



Decisioning Consultant

STUDENT GUIDE

© Copyright 2024
Pegasystems Inc., Cambridge, MA
All rights reserved.

This document describes products and services of Pegasystems Inc. It may contain trade secrets and proprietary information. The document and product are protected by copyright and distributed under licenses restricting their use, copying, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This document is current as of the date of publication only. Changes in the document may be made from time to time at the discretion of Pegasystems. This document remains the property of Pegasystems and must be returned to it upon request. This document does not imply any commitment to offer or deliver the products or services provided.

This document may include references to Pegasystems product features that have not been licensed by your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems service consultant.

PegaRULES, Process Commander, SmartBPM® and the Pegasystems logo are trademarks or registered trademarks of Pegasystems Inc. All other product names, logos and symbols may be registered trademarks of their respective owners.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors. This document or Help System could contain technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Pegasystems Inc. may make improvements and/or changes in the information described herein at any time.

This document is the property of:
Pegasystems Inc.
1 Rogers Street
Cambridge, MA 02142
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pegasystems.com

Mission: Decisioning Consultant

Product: Pega Customer Decision Hub™ '24.1

URL:

<https://academy.pegasystems.com/mission/decisioning-consultant/v7>

Date: 11 April 2024

Contents

One-to-one customer engagement.....	7
Next-Best-Action paradigm.....	8
1:1 Customer Engagement paradigm.....	13
Pega Customer Decision Hub overview.....	22
CDH Assistant overview.....	31
Next best action in an omnichannel environment.....	35
Next best action in a contact center.....	36
Next best action on digital.....	43
Arbitration fundamentals.....	53
Action arbitration.....	54
Action prioritization with AI.....	60
Optimizing your next-best-action strategy.....	66
Monitoring the health of the system using Impact Analyzer.....	67
Investigating the effect of business levers in action arbitration.....	74
Investigating the effect of applying engagement policies.....	78
Next-Best-Action on digital.....	82
Business use case: Cross-sell on the web.....	83
Next-Best-Action Designer.....	87
Defining and managing customer actions.....	94
Action hierarchy.....	95
Managing business structure.....	100
Creating actions.....	102
Renaming actions.....	106
Presenting a single offer on the web.....	109
Real-time containers.....	110
Creating and configuring a real-time container.....	114
Presenting a single offer on the web.....	116
Defining customer engagement policies.....	121

Customer engagement policies.....	122
Defining eligibility, applicability, and suitability Rules.....	127
Creating and testing personas.....	132
Testing Next-Best-Action with Persona and Channel Context.....	133
Testing the Engagement Policy configurations with Persona Test Cases.....	142
Avoiding overexposure of actions.....	146
Contact policies.....	147
Defining action suppression rules.....	149
Arbitrating between actions.....	153
Action arbitration.....	154
Prioritizing actions with business levers.....	160
Business use case: Cross-sell on the web extended.....	163
Business use case: Cross-sell on the web extended.....	164
Decision Strategies Overview.....	168
Creating decision strategies.....	169
Decision strategy execution.....	175
Creating engagement strategies using customer credit score.....	186
Customer credit score using a scorecard.....	187
Building a scorecard to calculate the creditscore.....	192
Using a scorecard in a decision strategy.....	200
Using predictive models.....	208
Predictive models drive predictions.....	209
Arbitrating across business issues.....	213
Using predictions in engagement strategies.....	215
Creating parameterized predictors.....	221
Defining and creating customer journeys.....	226
Pega Customer Journeys.....	227
Defining Customer Journeys.....	233
Essentials of always on outbound.....	242
Working of the always-on brain in outbound.....	243

Traditional marketing versus always-on outbound.....	249
Business use case: Always-on outbound customer engagement.....	254
Sending offer emails.....	257
Sending offer emails.....	258
Defining the starting population.....	267
Segmentation.....	268
Creating a segment.....	270
Defining an action for outbound.....	281
Email Treatments.....	282
Creating internal email treatments.....	284
Defining the action flow.....	303
Avoiding overexposure of actions on outbound.....	307
Avoid sending the same action for a time period.....	311
Limiting action volume on outbound.....	315
Volume constraints.....	316
Configuring a volume constraint.....	323
Sharing action details with third-party distributors.....	328
File and database templates.....	329
Creating an external treatment to share action details in a file or database.....	334
Business agility in 1:1 customer engagement overview.....	345
Agility in a customer engagement project.....	346
Change management process.....	348
Pega 1:1 Operations Manager overview.....	352
Building your business operations team.....	353
Lifecycle of a change request.....	357
Change request types.....	359
Launching a new offer on web.....	367
Using Automated insights when creating treatments in 1:1 Operations Manager.....	368
Submitting a request for a new action.....	371
Prioritizing change requests.....	374

Completing the action build tasks.....	376
Deploying changes to production.....	378
Creating treatments with GenAI.....	379
Update existing actions.....	385
Update existing actions.....	386
Updating existing actions in bulk.....	388
Implementing a business change using Revision Management.....	391
Modifying a decision strategy.....	392
Editing a group-level engagement policy.....	396
Implementing a fast release revision.....	398
Fast-release revisions.....	399
Implementing outbound communication in the BOE.....	402
One-time actions.....	403
Creating a One-time action.....	405

One-to-one customer engagement

Description

Familiarize yourself with the one-to-one customer engagement paradigm. Learn about the business problems Pega Customer Decision Hub™ can solve and discover how Pega's omni-channel AI delivers the right action during every customer interaction.

Learning objectives

- Explain the basics of the Next-Best-Action approach
- Describe the purpose of Next-Best-Action Designer and its user interface

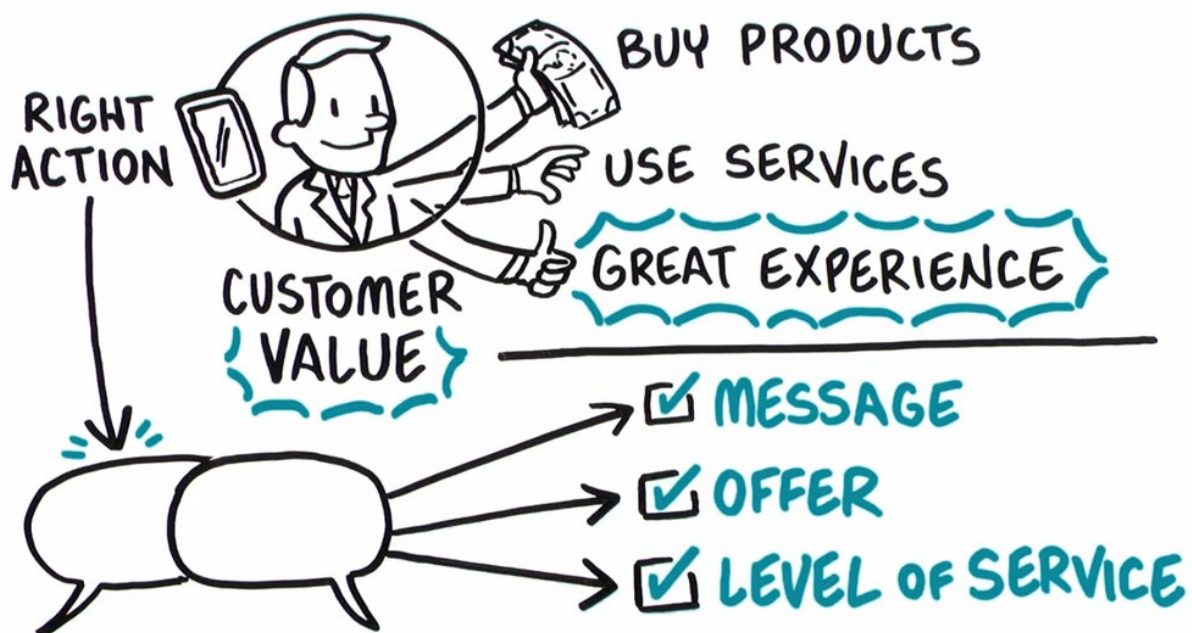
Next-Best-Action paradigm

Introduction

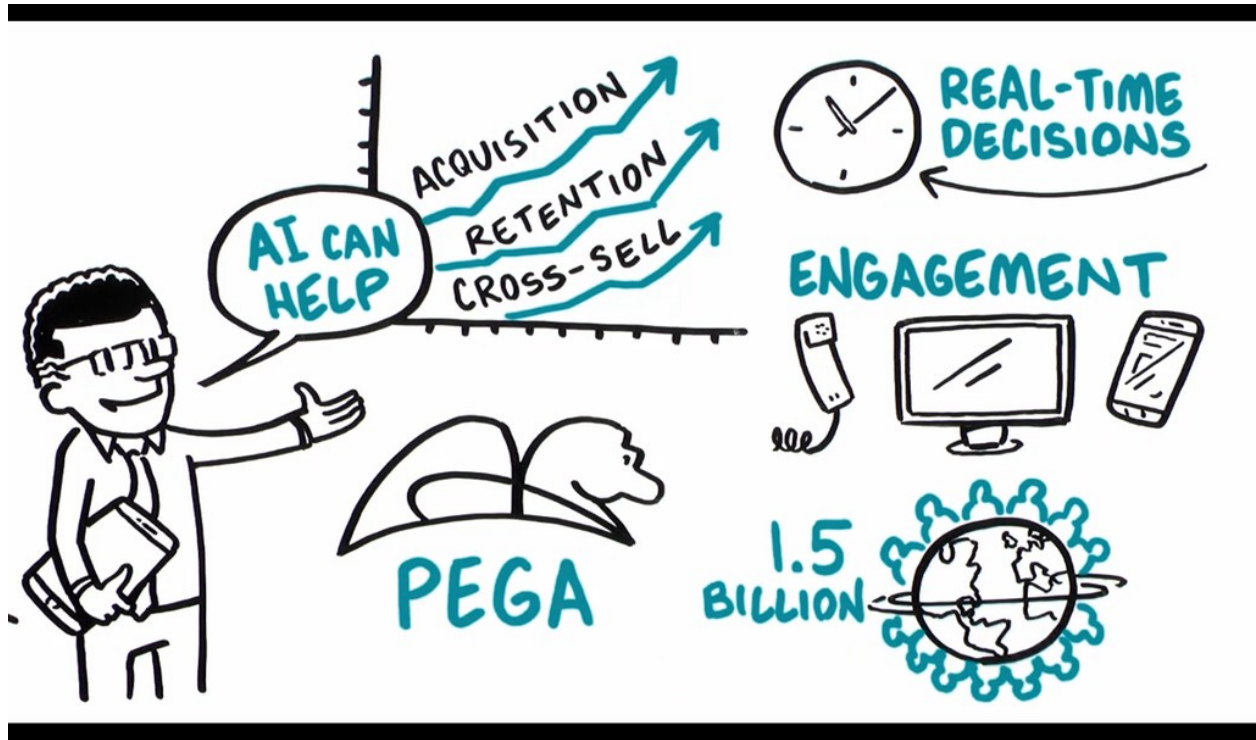
The value of big data and analytics is fully realized when every customer conversation delivers exactly the right message, the right offer, or the right level of service to provide a great experience while maximizing the customer's value to the organization. With Pega Next-Best-Action, business experts develop decision strategies that combine predictive and adaptive analytics with traditional business rules to maximize this value.

Transcript

This is your customer. You want him to buy your products, use your services and have a great experience. And your competitors want the same thing. To compete, you have to take the right action at every customer touch, ensuring that each conversation delivers exactly the right message, offer and level of service. You want to provide a great experience, while maximizing the customer's value to your organization.



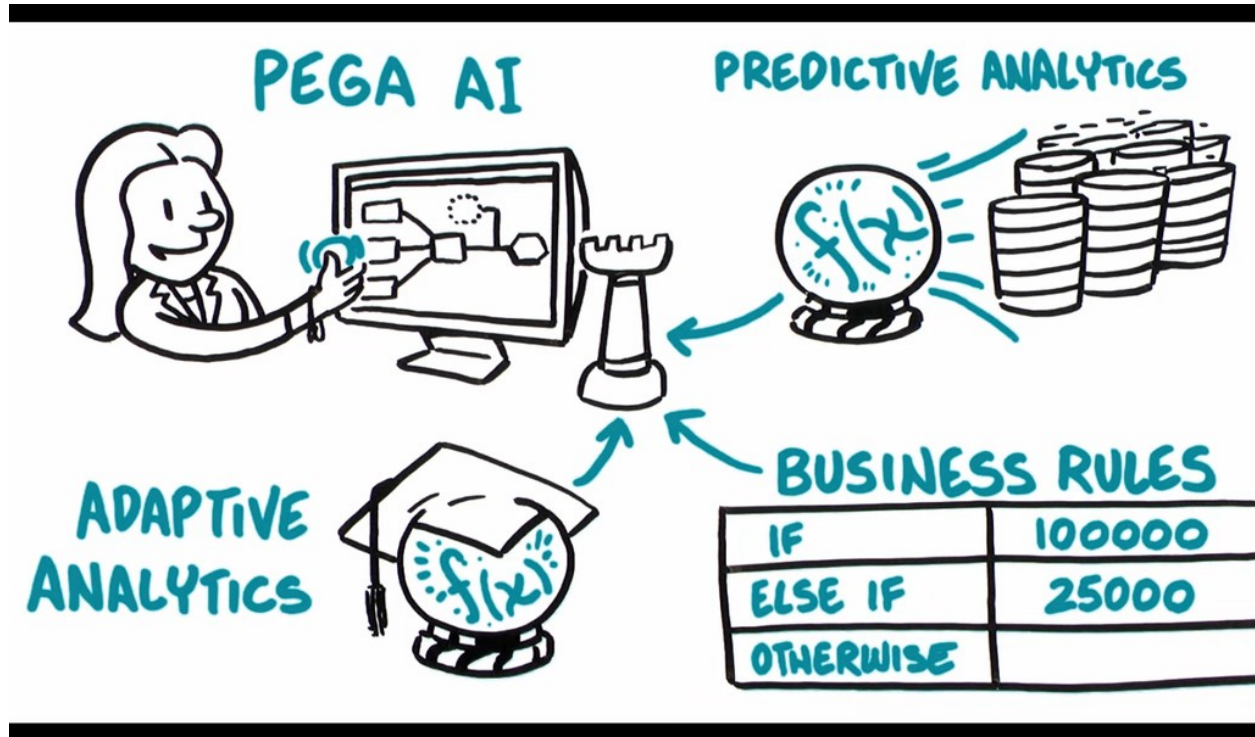
Artificial Intelligence, or AI, can help—if you can get past the hype. Pega has been using AI to create real business value for years, driving real-time decisions that deliver awesome engagement on any channel and improving experiences for over 1.5 billion customers across the globe.



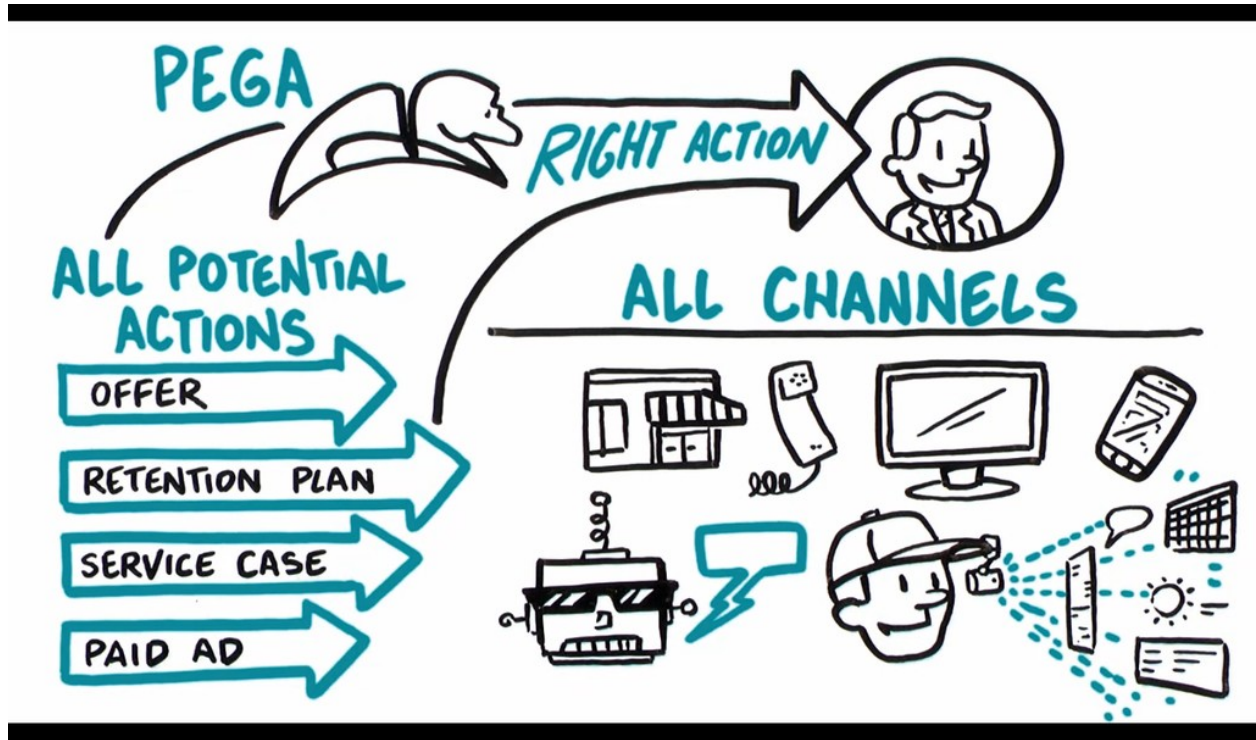
Pega's omni-channel AI delivers the right action at every customer touch by crunching millions of data points in real-time. Make an offer, initiate a retention plan, predict a problem before it happens. Every decision generates the next-best-action for your customer, and your business.



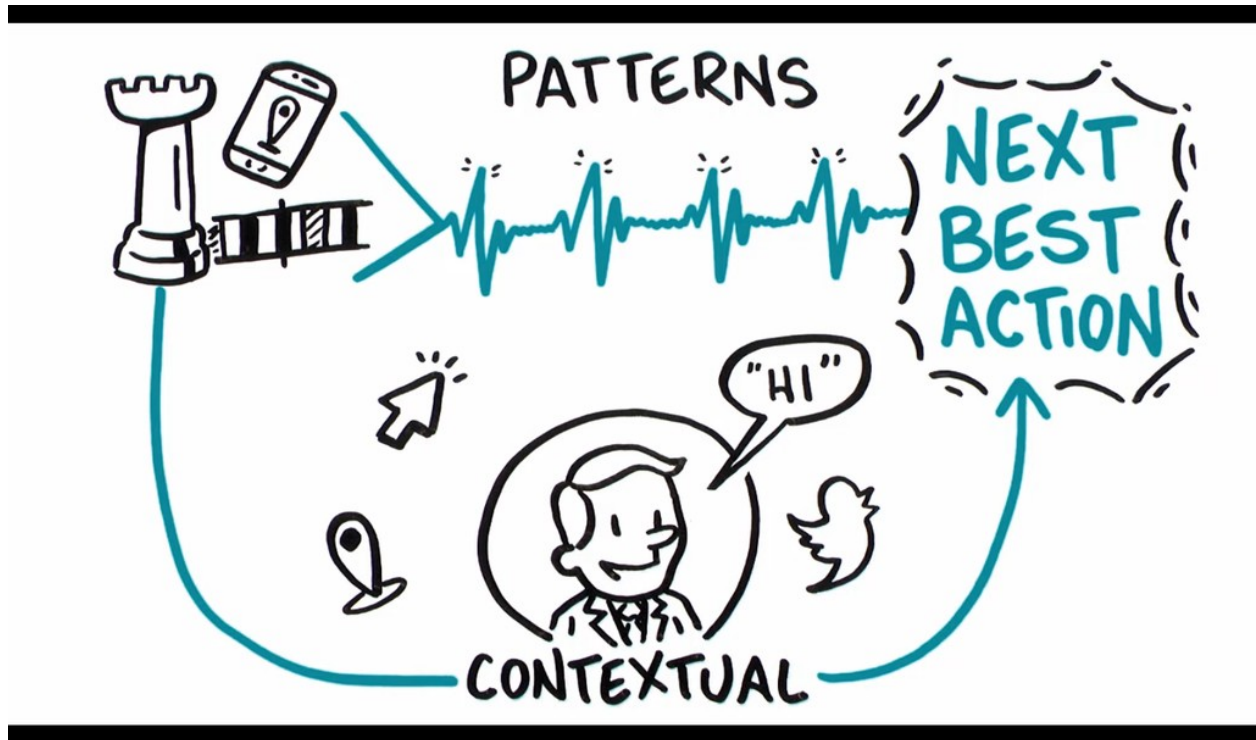
Pega's AI is built for business people, not scientists or developers. They design visual decision strategies that combine predictive analytics, algorithms developed through mining large sets of data, adaptive analytics, machine-learning algorithms that improve with each interaction, and traditional business rules that allow users to prioritize and arbitrate between decisions.



Pega uses the strategy to look across all the potential actions you may take with a customer, make an offer, initiate a retention plan, open a service case, place an ad, and ensure exactly the right action is taken at every interaction and it works across all channels to provide a consistent experience in a store, on the phone, on the web, mobile, with the chat bot, or just some crazy tech that hasn't even been invented yet.



And Pega connects to streams like mobile locations or network events to detect patterns and drive the Next Best Action proactively. And strategies are completely contextual. Any change in the customer's context — a click, a reply, a location change, a Tweet — will trigger the Next Best Action. So, you can really listen to your customers and act accordingly.



Pega's real-time, omni-channel AI puts the power in your hands, so you can optimize every customer interaction for experience, and value.



1:1 Customer Engagement paradigm

The optimal outcome of every customer interaction is to provide a great experience while maximizing the value of the customer to the company. To achieve this outcome, you must perform the right action in the right channel at the right moment for each customer. In Pega Customer Decision Hub™, this feature is known as One-to-one Customer Engagement.

Transcript

The optimal outcome of every customer interaction is to provide a great experience while maximizing the value of the customer to the company.

In a traditional approach marketers search for potential customers from a database based on target demographics, geographies, or financial means to make a purchase. Then, marketers target this customer segment across all channels, and all these customers receive offers for a specific product.

The problem with this approach is that only a low percentage of customers respond or make a purchase. Marketers might hit their short-term goal, but in the long term, this approach affects the relationship with customers.

As a result, the traditional marketing approach fails because of a lack of relevance, context, timing, and empathy.

Why this fails

Lack of **RELEVANCE**

Lack of **CONTEXT**

Lack of **TIMING**

Lack of **EMPATHY**

Customers are more empowered than ever before. As a result, they have very high expectations for the experiences they receive from their service providers. Their experiences must make sense within the context of their lives and this means they must be meaningful, consistent, and personalized across every channel with which they interact.

In business, the optimal outcome of every customer interaction is to provide a great experience while maximizing the customer's value to the company. To achieve this, you must perform the right action in the right channel at the right moment for each customer.



Traditional mass-marketing techniques that use segments, batches, and campaigns do not move the needle anymore; they are antiquated and unsustainable. The market has recentered around real-time technologies and one-to-one customer interactions.



The Pega approach to these interactions is through One-to-one Customer Engagement.

Through One-to-one Customer Engagement, companies can transition their marketing away from a traditional one-to-many campaign-driven approach. A one-to-one approach allows companies to have consistent, contextual, and relevant conversations with individual customers across any channel or touch point.



The key to achieving One-to-one Customer Engagement is one centralized brain.

In other words, one piece of intelligence acts as a single decision authority across your application ecosystem.

Each channel or system profits from this single source of customer intelligence and can use it to gain insights or perform relevant actions.



In Pega Customer Decision Hub, this centralized brain is the core feature that uses AI to enable One-to-one Customer Engagement.

In Pega Infinity, Pega Customer Decision Hub forms the core of the customer engagement platform, which sits at the center of existing systems and channels in an enterprise.

The “brain” collects data from every customer engagement across the enterprise to create predictions and decisions about every interaction in every channel.

Pega Customer Decision Hub can directly integrate with third-party content management platforms such as Adobe Experience Manager and use the content that you develop in these platforms for personalized customer engagement.

Continuous learning and decision-making are the foundation of a One-to-one Customer Engagement solution.



Customer Decision Hub combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that it uses to make intelligent decisions. These decisions are known as the next best actions.



Every next best action weighs customer needs against business needs to optimize decisions based on priorities set by the business manager.

In the milliseconds before interacting with a customer, Customer Decision Hub combines thousands of business rules and predictive and adaptive models that determine customer needs and interests to ensure the next best action is relevant by keeping it personal, timely, contextual, and empathetic.

Customer Decision Hub identifies the best moments to make a sale, provide a service, make a retention offer, inform about an offer, or do nothing at all (for example, if no offer is relevant enough to warrant the customer's attention). The system distributes next-best-action decisions in real time to your channels, such as the web, mobile, and contact center. Pega Customer Decision Hub can also distribute next best actions to real-time paid channels such as Google, YouTube, Facebook, LinkedIn, and Instagram. Pega Customer Decision Hub also integrates with non-real-time outbound channels such as data management platforms (DMPs) and email.

After the system distributes the next best actions and the "brain" receives customer responses, the process begins again. Customer Decision Hub distributes new next best actions in milliseconds. It captures every customer interaction in every channel to ensure consistency and an optimized customer experience across channels.



For example, consider a customer, Miranda. With the centralized brain in place, instead of looking only at sales offers, the system begins with a list of all potential actions, such as service, retention, nurture, or a hardship message, everything that you can do for this customer now.

Although AI drives the next best actions, AI does not run rampant; it experiments on every person with every topic. Customer Decision Hub still allows marketers to maintain control and establish the criteria that AI must meet to consider beginning one of these conversations.

For example, you establish eligibility rules that state the business cannot sell a card to someone under the age of 18.

Then there are applicability rules to define if the action is appropriate at a given time. If Miranda already owns a competing or more valuable product, the business does not offer this product to her even though she is eligible.

Suitability rules determine whether an action is in the best interest of a customer. For example, Miranda might be eligible for a card, and her current card does not offer as high of a cash-back offer. But because Miranda cannot make her monthly payments and might end up in collections, the business does not offer the card to her even though it can.

So, if a customer fails to meet the conditions that you establish for an action, the AI does not consider that action.

Next-Best-Action Decisions



Should we arbitrate?

Potential Actions for Miranda	Eligible?	Applicable?	Suitable?	Arbitrate?
● Sales: Rewards Card	✓	✓	✗	No
● Sales: Home Equity Loan	✓	✓	✓	Yes
● Sales: Mortgage Loan	✓	✓	✓	Yes
● Sales: Premium Checking	✓	✓	✓	Yes
● Service: Update Account Information	✓	✓	✓	Yes
● Service: Fraud Alert	✓	✗	—	No
● Service: Travel Notification	✓	✓	✓	Yes
● Retention: Waive Annual Fee	✓	✓	✓	Yes
● Retention: Double Rewards Points	✗	—	—	No
● Retention: \$100 Travel Credit	✓	✓	✓	Yes
● Nurture: Personal Finance eLearning	✓	✓	✓	Yes
● Nurture: Chat with Investment Team	✓	✓	✓	Yes
● Onboarding: Download Mobile App	✓	✓	✓	Yes
● Onboarding: Enroll in Autopay	✓	✓	✓	Yes

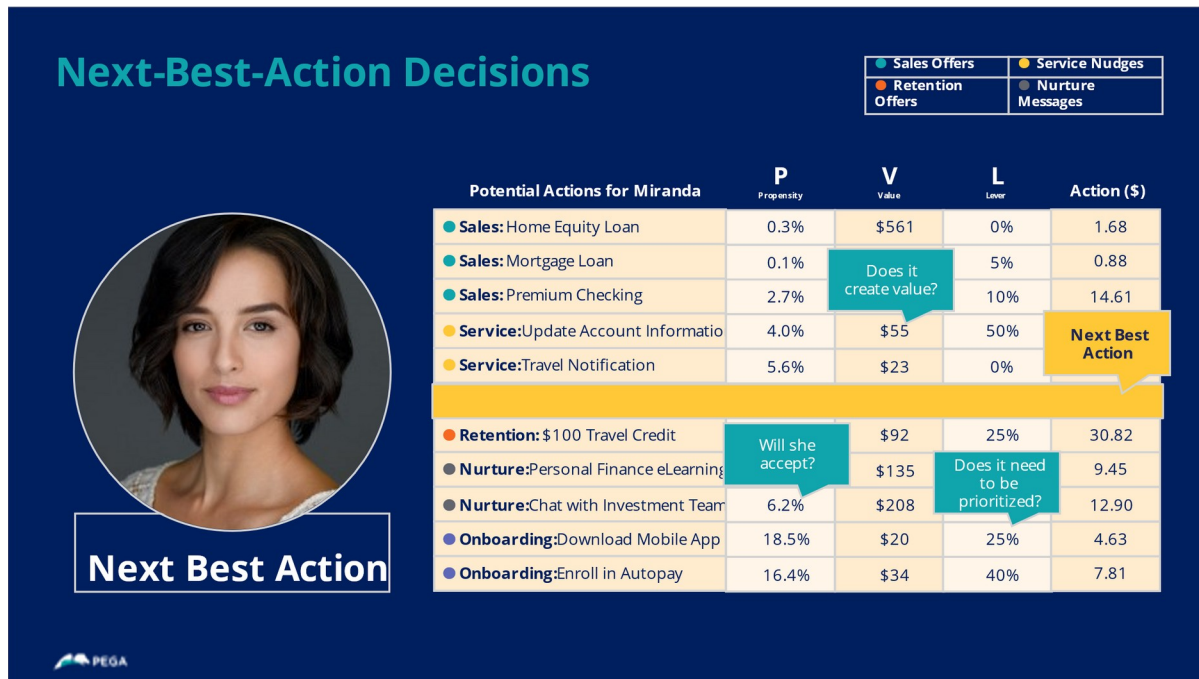
● Sales Offers	● Service Nudges
● Retention Offers	● Nurture Messages



Once the system narrows down the list, AI takes over. AI first determines how likely each action is desirable to the customer. Next, it determines the value each option generates for the business. What is the impact if the customer reads the content or accepts an offer? Does it give you more revenue or reduce your costs?

You can add levers to make adjustments based on the current business situation. For example, the business can decide to nudge an offer if it needs to meet its financial goals or ramp down an offer if it runs low on inventory.

For Miranda, the action that the system selects as the next best action is the one with the highest P*V*L value.



In summary, Customer Decision Hub is the always-on brain that acts as a single, centralized decision authority.

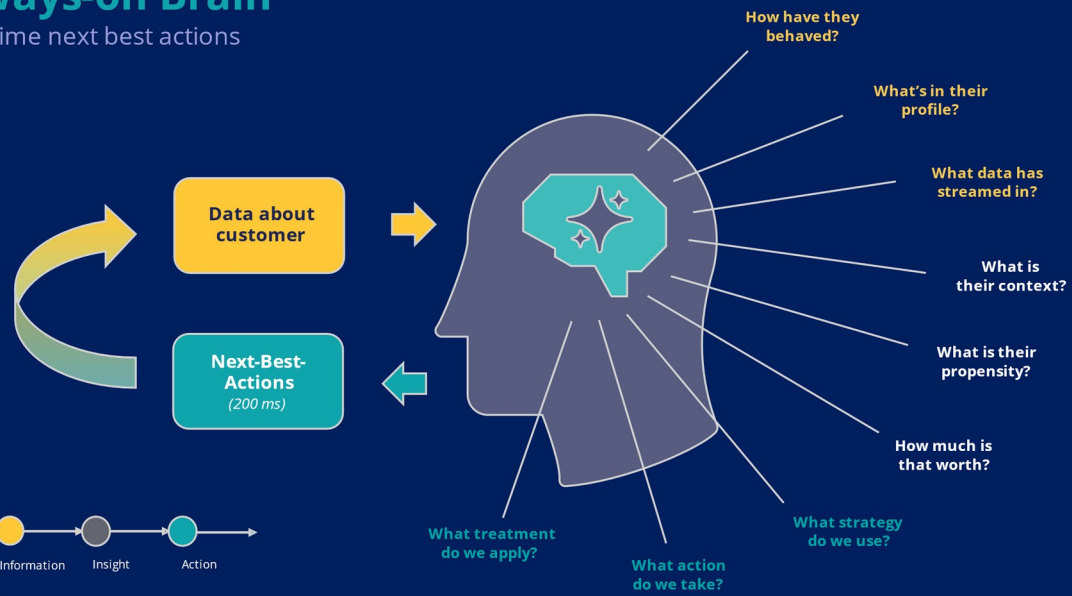
It uses data about the customer, including past interactions, as input.

It uses advanced AI techniques to make predictions.

Decision strategies (which combine traditional business rules with predictive, adaptive, and text analytics) deliver consistent and personalized next best actions across all channels.

Always-on Brain

Real-time next best actions



You have reached the end of this video.

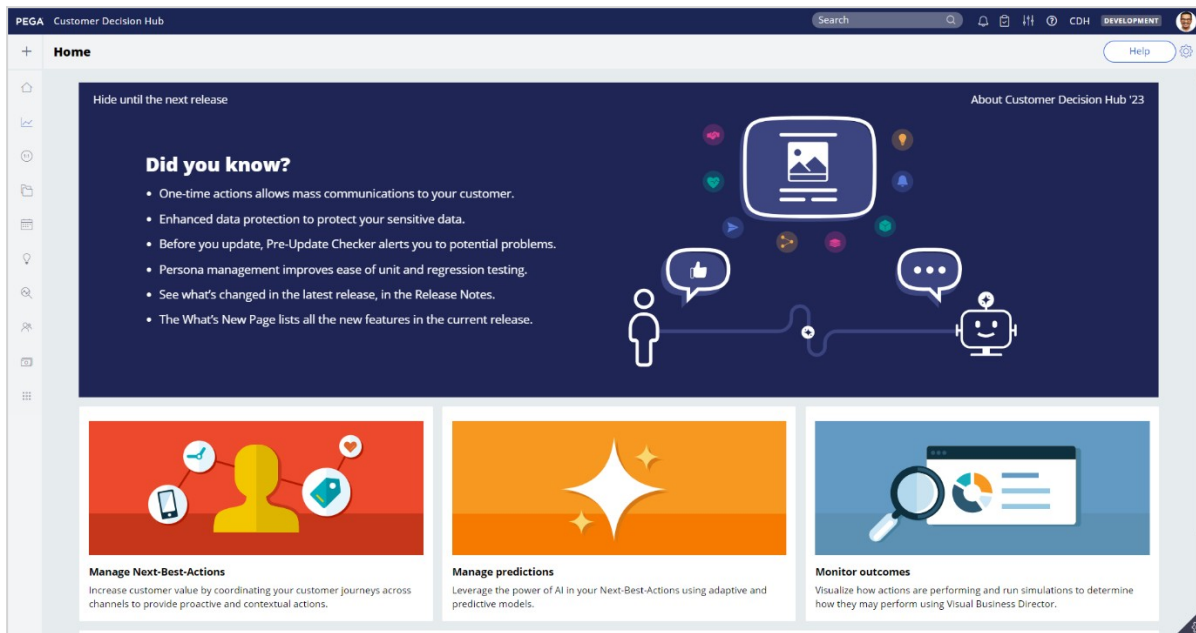
Pega Customer Decision Hub overview

Pega Customer Decision Hub™ optimizes customer lifetime value by providing an "always-on brain" for your business. With Customer Decision Hub, you can unify your data, analytics, and channels into a single, connected experience so that you can predict customer needs in real-time. You then deliver hyper-personalized next best actions that drive long-term loyalty across channels and at scale.

Transcript

Most marketing stacks use a combination of channel-based decisioning solutions, that do not work out-of-the-box together and cannot keep pace with today's customers. The design of these marketing stacks intends to push products rather than solve problems. Building real relationships with today's customers means engaging them one-to-one, anticipating their needs, and delivering relevant experiences across channels.

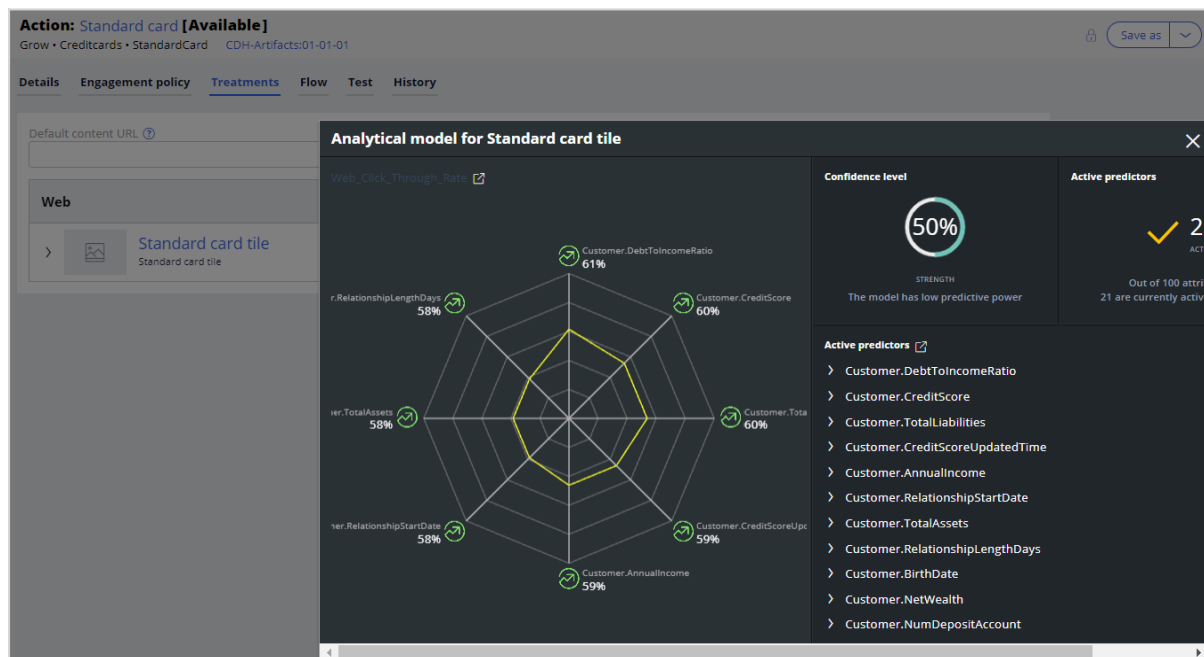
Pega Customer Decision Hub is the central decision engine of your marketing stack, constantly analyzing the context of the customer and recommending the most relevant, next best action to take, all in real time. Here is how it works.



You first identify the outcomes that you want to achieve, such as increasing sales, reducing churn, or preventing service issues. These outcomes are called business issues. A business issue is the purpose behind the actions that you offer to customers. The business issues contain business groups to organize actions into categories.

Issues / Groups	Description	Action naming
Acquire	Drive conversion rate and generate sustainable growth	
☰ Credit cards	Actions to improve customer's card acquisition journey experience	
☰ Auto loans	Actions to improve customer's auto account acquisition journey experience	
☰ Deposit accounts	Actions to improve customer's deposit account acquisition journey experience	
☰ Mortgages	Actions to improve customer's Home loan acquisition journey experience	
Grow	Develop deeper and more profitable relationship with existing clients	
☰ Credit cards	Credit card products for existing customers	
☰ Deposit accounts	Deposit account products for existing customers	
☰ Mortgages	Actions which help to build deeper relationships based on customers need for home loans	
Nurture	Nurture customer relationship to drive more engagement	
☰ Webinars	Includes webinars for developing customer relationships	
☰ Credit cards	Maximize the benefits card customers receive	
☰ Articles	Fuel you content marketing using knowledge articles	
☰ Mobile banking	Highlight the convenience & security of mobile banking	
Retain	Improve customer attrition	

Next, you create actions, such as upsell or cross-sell offers, retention bundles, and nurture messages that Customer Decision Hub uses to engage your customers. You create treatments, variations of these actions, to tailor that engagement to multiple channels. Each action and treatment automatically receive a predictive model, so our AI can determine exactly what each customer needs.



With Customer Decision Hub, the next-best-action customer journey takes a transformational approach by using real-time AI to prioritize across all potential journeys and actions to deliver more relevant customer experiences.

Customer Decision Hub helps marketers leverage thousands of pieces of historical data and identify the actions and journeys that lead to positive engagement. Because Customer Decision Hub takes advantage of real-time machine learning, all the predictions are up to date, which ensures that customers receive personalized offers at each stage.

You define a journey once across all business objectives, channels, and lines of business. Then, you add the different stages and assign the corresponding actions to each one. As customers move through their journeys, the AI learns continuously and automatically to identify which actions work best at each stage and pivots between selling, serving, and retaining in real time as the needs of customers change.

Then, you define specific channel and action limits and limit overexposure to a specific action or group of actions. With outbound and inbound action limits, you restrict the number of actions that the system presents to customers in a given period of time.

Constraints		CDH-Artifacts: 01-01-01	Edit	Actions
		Edited 6 days ago by Decisioning Architect		
Outbound channel limits				
> All outbound			2 limits defined	
> Outbound Retail			1 limit defined	
> Email			1 limit defined	
> Direct Mail			1 limit defined	
> SMS			1 limit defined	
> Push			1 limit defined	
> Outbound Call Center			1 limit defined	
Outbound action limits				
> All issues / All groups (All categories)			1 limit defined	
Inbound action limits				
> Call center			1 limit defined	
> Retail			1 limit defined	
> Web			1 limit defined	
Contact policy library				
> CP-IMPR-G-7D: Track Impressions for the action over the past 7 days				
> CP-ACPT-G-7D: Track Accepts for the action over the past 7 days				

Next, you set rules of engagement so that Customer Decision Hub knows what customers are eligible for, when certain actions make sense, and what to prioritize in different situations. Engagement policies are a set of business rules and practices that the organization uses to determine which customers qualify for which next best actions. These policies allow you to specify the conditions under which an action or group of actions a customer is eligible for.

Name	Specialized policies
Standard card	E
Rewards card	E
Rewards Plus card	S
Premier Rewards card	S

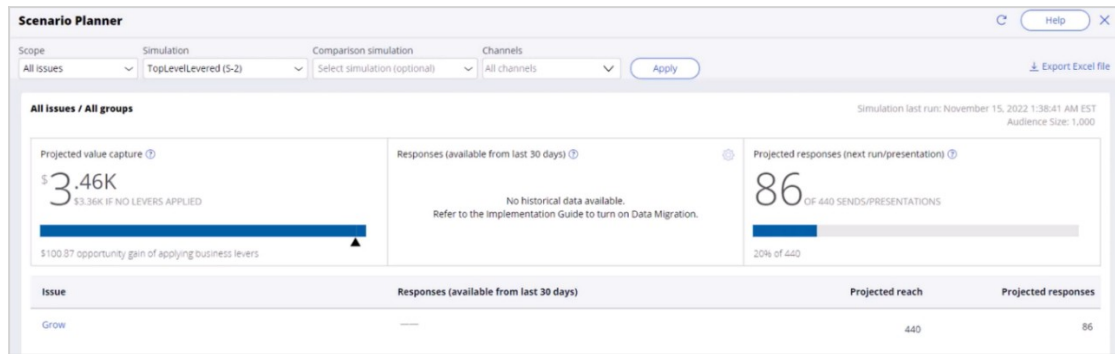
You balance customer relevance with business priorities by using arbitration. Arbitration is how Customer Decision Hub prioritizes the remaining list of eligible, applicable, and suitable actions from each group. On the **Arbitration** tab of Next-Best-Action Designer, you can define the AI controls that the system uses to rank the next best action for each customer.

Customer relevance × **Business priority** = **Final priority**

Propensity × Context weighting × Business value × Business levers = Explore arbitration results

- Propensity**: Ensure only relevant messages get communicated to customers and determine the source for propensities. 0 thresholds defined.
- Context weighting**: Upweight actions based on real-time contextual data passed by channel applications. 10 contexts defined.
- Business value**: Extend the way business value is calculated through the CalculateBusinessValue extension point. 0 thresholds defined.
- Business levers**: Upweight actions based on the business structure, and toggle all Individual action levers ON/OFF. 0 levers defined.
- Priority**: Ensure that only the most important messages get delivered to customers. 0 thresholds defined.

Finally, you activate your next best actions in the intended delivery channels, which creates a "brain" that powers your entire engagement program.



Customer categories

Analysis of the current customer engagement. Click on a customer category to find out more details.

Category:

>	329 customers (32.9%) No actions after suitability
>	212 customers (21.2%) Only irrelevant actions after eligibility
>	207 customers (20.7%) No actions after eligibility
>	75 customers (7.5%) No actions after applicability
>	32 customers (3.2%) Only irrelevant actions after suitability
>	3 customers (0.3%) Only irrelevant actions after arbitration

Pega Customer Decision Hub is designed to address the needs of your customers. To fulfill this objective, the application is specifically configured to align with business requirements. You can access a summary of your application's configuration through the **Application Overview**. This landing page provides crucial details about enabled modules, environments, Rulesets, and metrics, and offers insights into the health of your next best action, suggesting areas for further review.

The information from the **Application Overview** helps you understand how the application is currently configured and how it performs against the current configuration. For example, it showcases the number of active issues, groups, and actions within the

system, as well as provides information on treatments and the performance of actions. Additionally, it highlights areas that can benefit from improvement.

Each time a customer engages with an offer, the channel calls Customer Decision Hub for a new next best action. Because Customer Decision Hub updates itself every time you receive a new piece of data, you constantly trigger new offers and messages as you learn more about each customer.

So, whether customers browse the web, read an email or mobile message, talk with a CSR, or see a digital ad, that experience is all about them every time. Pega Customer Decision Hub helps you embrace empathy at scale across your entire enterprise, helps customers get what they need, and helps you generate hundreds of millions in incremental value every year.

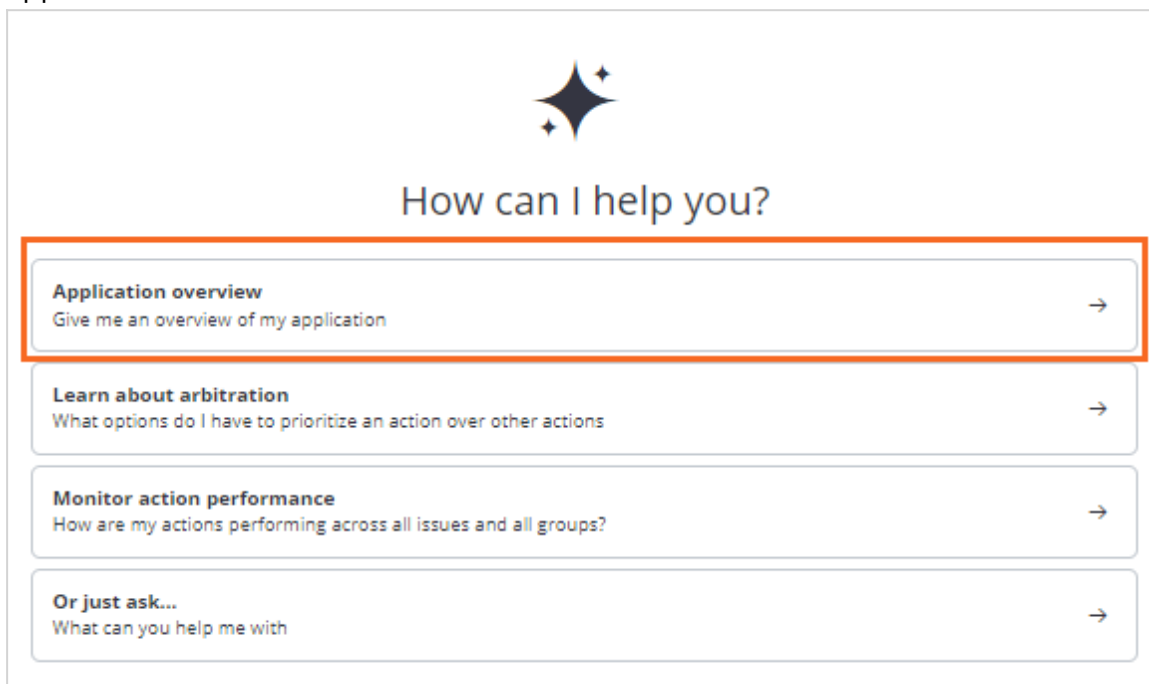
CDH Assistant overview

CDH Assistant is an AI-powered conversational feature that is seamlessly integrated with the Pega Customer Decision Hub™ and Pega 1:1 Operations Manager infrastructure. Its main purpose is to enhance your decision-making process, boost productivity, and unleash the full potential of your business by simplifying the search process and delivering personalized results.

This is Pega Customer Decision Hub. The Pega Infinity™ '24.1 software suite introduces a new feature to simplify knowledge searching in Pega Customer Decision Hub – CDH Assistant.

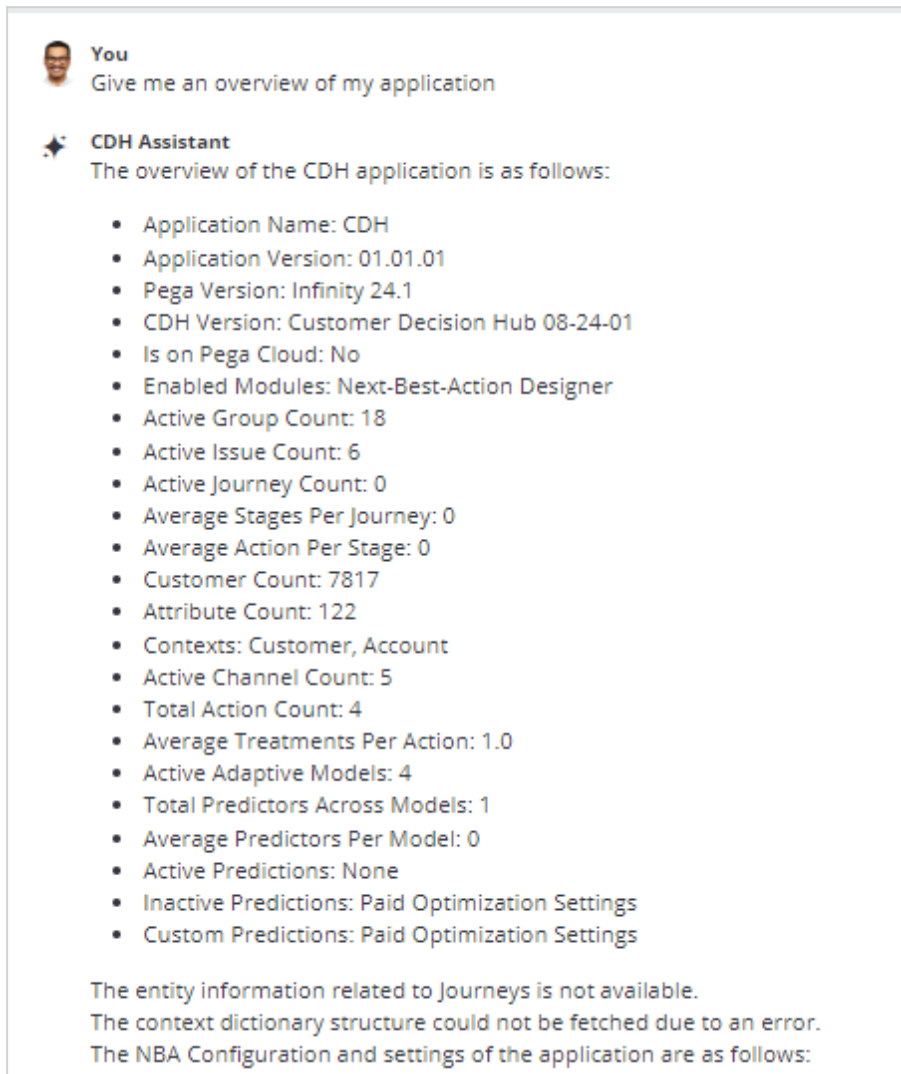
To access CDH Assistant, click the CDH Assistant tab, and you will be directed to the Assistant’s landing page. From there, you have two options to start your conversation. You can either choose a topic provided on the landing page, or enter your own questions to obtain the information that you need for your particular use case.

Click the **Application overview** tile to learn more about the Customer Decision Hub application.



By utilizing CDH Assistant, you gain a comprehensive overview of your application within Customer Decision Hub. This includes detailed insights into Containers, Channels, Context, Rulesets, and Actions. Additionally, you can access information about configurations and performance metrics, such as accepts, sends, clicks, and impressions for your actions,

issues, and groups.



The screenshot shows a chat interface. At the top, a user icon is followed by the text "You Give me an overview of my application". Below this, a star icon is followed by "CDH Assistant" and the text "The overview of the CDH application is as follows:". A bulleted list of application metrics follows, including application name, version, Pega version, CDH version, cloud status, enabled modules, and various counts for groups, issues, journeys, stages, actions, customers, attributes, channels, and treatments. At the bottom, there are three lines of error messages: "The entity information related to Journeys is not available.", "The context dictionary structure could not be fetched due to an error.", and "The NBA Configuration and settings of the application are as follows:".

You
Give me an overview of my application

CDH Assistant
The overview of the CDH application is as follows:

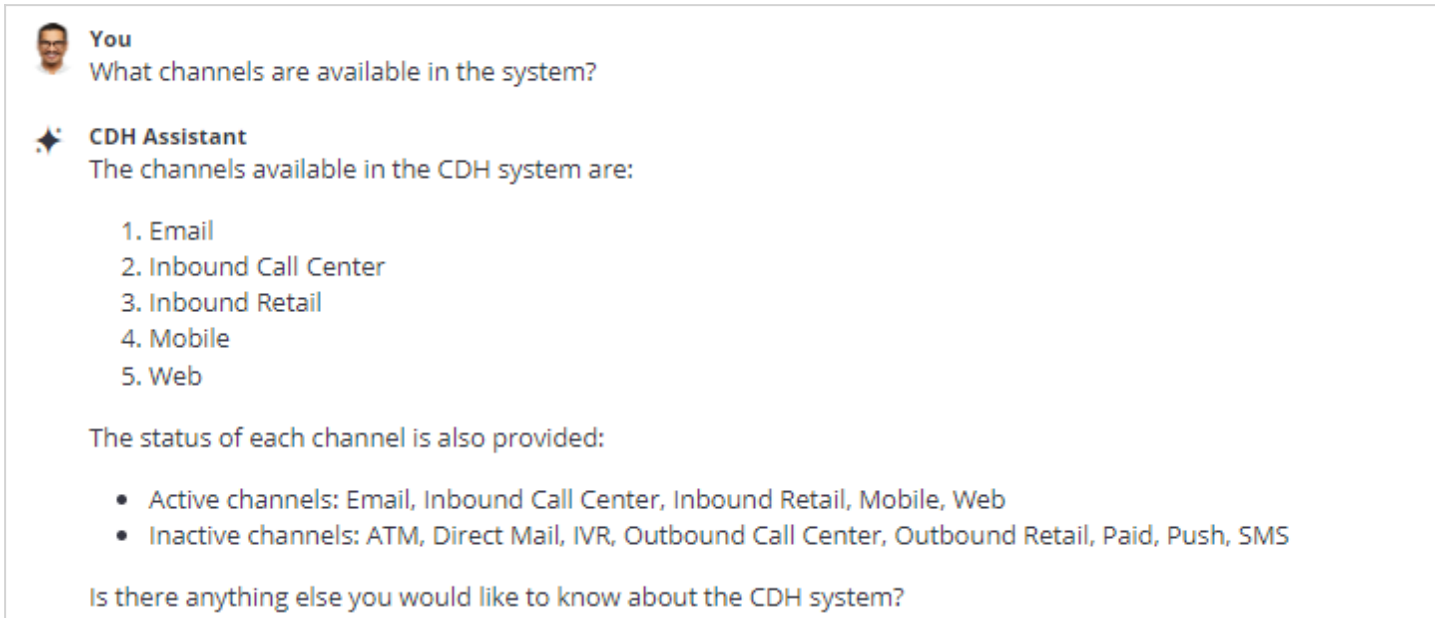
- Application Name: CDH
- Application Version: 01.01.01
- Pega Version: Infinity 24.1
- CDH Version: Customer Decision Hub 08-24-01
- Is on Pega Cloud: No
- Enabled Modules: Next-Best-Action Designer
- Active Group Count: 18
- Active Issue Count: 6
- Active Journey Count: 0
- Average Stages Per Journey: 0
- Average Action Per Stage: 0
- Customer Count: 7817
- Attribute Count: 122
- Contexts: Customer, Account
- Active Channel Count: 5
- Total Action Count: 4
- Average Treatments Per Action: 1.0
- Active Adaptive Models: 4
- Total Predictors Across Models: 1
- Average Predictors Per Model: 0
- Active Predictions: None
- Inactive Predictions: Paid Optimization Settings
- Custom Predictions: Paid Optimization Settings

The entity information related to Journeys is not available.
The context dictionary structure could not be fetched due to an error.
The NBA Configuration and settings of the application are as follows:

Furthermore, CDH Assistant not only provides application-related information but also enables you to inquire about other Pega products. You can seek guidance on installation, implementation, configuration, and troubleshooting.

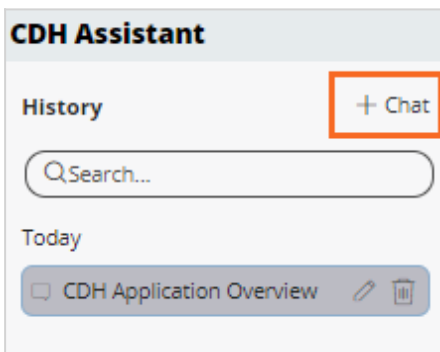
The Assistant will search the extensive Pega documentation set and provide you with the most relevant response, tailored to your needs, together with a link to the source of the knowledge.

You can continue this conversation by asking additional questions, for example:

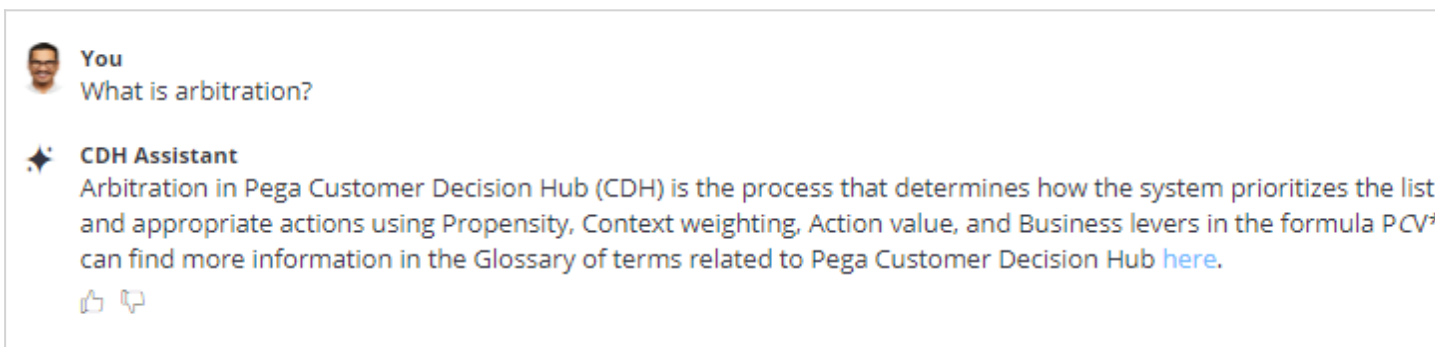


A screenshot of a chat interface. At the top, a user icon is followed by the text "You" and the question "What channels are available in the system?". Below this, a star icon is followed by "CDH Assistant" and the response "The channels available in the CDH system are:". A numbered list follows: 1. Email, 2. Inbound Call Center, 3. Inbound Retail, 4. Mobile, 5. Web. Below the list, it says "The status of each channel is also provided:" followed by two bullet points: "Active channels: Email, Inbound Call Center, Inbound Retail, Mobile, Web" and "Inactive channels: ATM, Direct Mail, IVR, Outbound Call Center, Outbound Retail, Paid, Push, SMS". At the bottom, it asks "Is there anything else you would like to know about the CDH system?"

To keep your conversations in order, always start a new chat for a specific concept, feature, or product. This is a good practice because the context of the overall conversation can affect the Assistant's answers.



A screenshot of the CDH Assistant chat interface. At the top, it says "CDH Assistant". Below that, there is a "History" section with a "+ Chat" button highlighted in an orange box. Underneath is a search bar with the placeholder text "Search...". Below the search bar, there is a "Today" section with a chat bubble titled "CDH Application Overview" and edit/delete icons.



A screenshot of a chat interface. At the top, a user icon is followed by the text "You" and the question "What is arbitration?". Below this, a star icon is followed by "CDH Assistant" and the response "Arbitration in Pega Customer Decision Hub (CDH) is the process that determines how the system prioritizes the list of offers and appropriate actions using Propensity, Context weighting, Action value, and Business levers in the formula PCV*...". It then says "You can find more information in the Glossary of terms related to Pega Customer Decision Hub [here](#)." At the bottom, there are thumbs up and thumbs down icons.

You can also rename your conversations so that the names reflect their main objective, and this can help you search past conversations to more easily find answers.

Lastly, all your past conversations are conveniently stored in the History section, ensuring that you can refer back to them whenever necessary.

CDH Assistant not only answers questions about the product but also allows you to interact with the documentation by providing links to the relevant sources of knowledge.

You have reached the end of this video. You have learned:

- What is CDH Assistant.
- How to use CDH Assistant.

Next best action in an omnichannel environment

Description

Pega's omni-channel AI delivers the right action at every customer touch by crunching millions of data points in real-time.

Learn how you can use Next-Best-Action to ensure a customer service representative takes relevant actions at every step during a customer interaction. Learn how easy it is to configure the brain by using the Next-Best-Action Designer to select the right offer for the right customer on a digital channel.

Learning Objectives

- Explain the benefits of Next-Best-Action in a contact-center
- Explain how adaptive models are used to predict customer behavior
- Describe the cross-sell on the web use case
- Explain the purpose of key configurations in the Next-Best-Action Designer

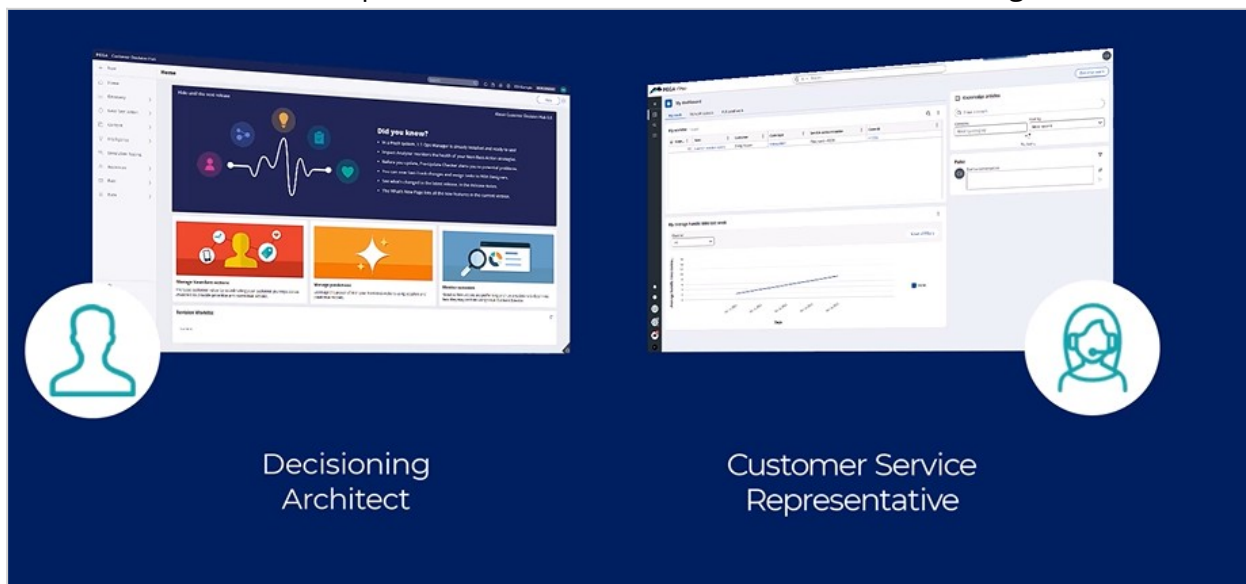
Next best action in a contact center

Pega Customer Decision Hub™ is an "always on" centralized decisioning "brain" that calculates a one-to-one business case for every Next-Best-Action recommendation. To create the business case, Customer Decision Hub combines the customer profile with previous interaction results, the current call context, and business rules, and then applies predictive analytics. Next-best-action recommendations occur across multiple channels, including the contact center.

Transcript

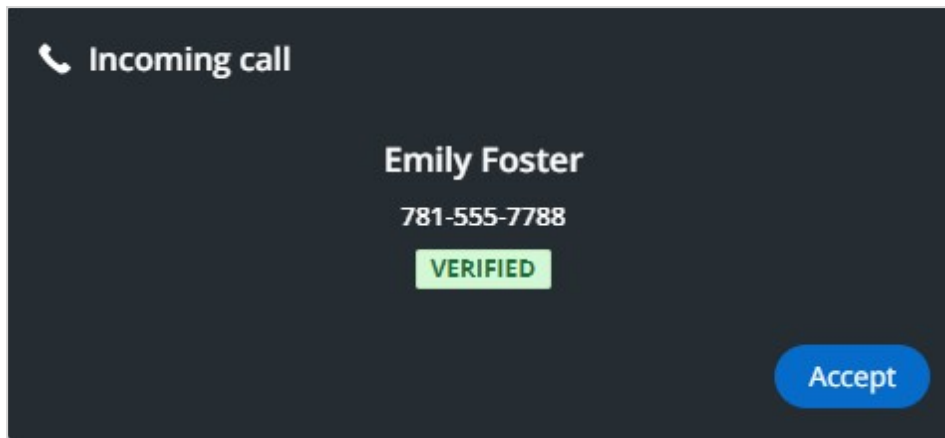
Next-best-action recommendations help to ensure that a Customer Service Representative (CSR) takes relevant actions at every step during a customer interaction.

In this scenario, U+ Bank is a retail bank that uses **Pega Customer Decision Hub (CDH)** and **Pega Customer Service** in its contact center. All the configurations implemented by the decisioning architect in Pega Customer Decision Hub are visible to the CSR in Pega Customer Service, and help them achieve the best results when interacting with customers.



Let's consider an example of a customer interaction:

A call comes into the U+ Bank service center from Emily Foster, a U+ customer. The system immediately routes the call to a CSR.



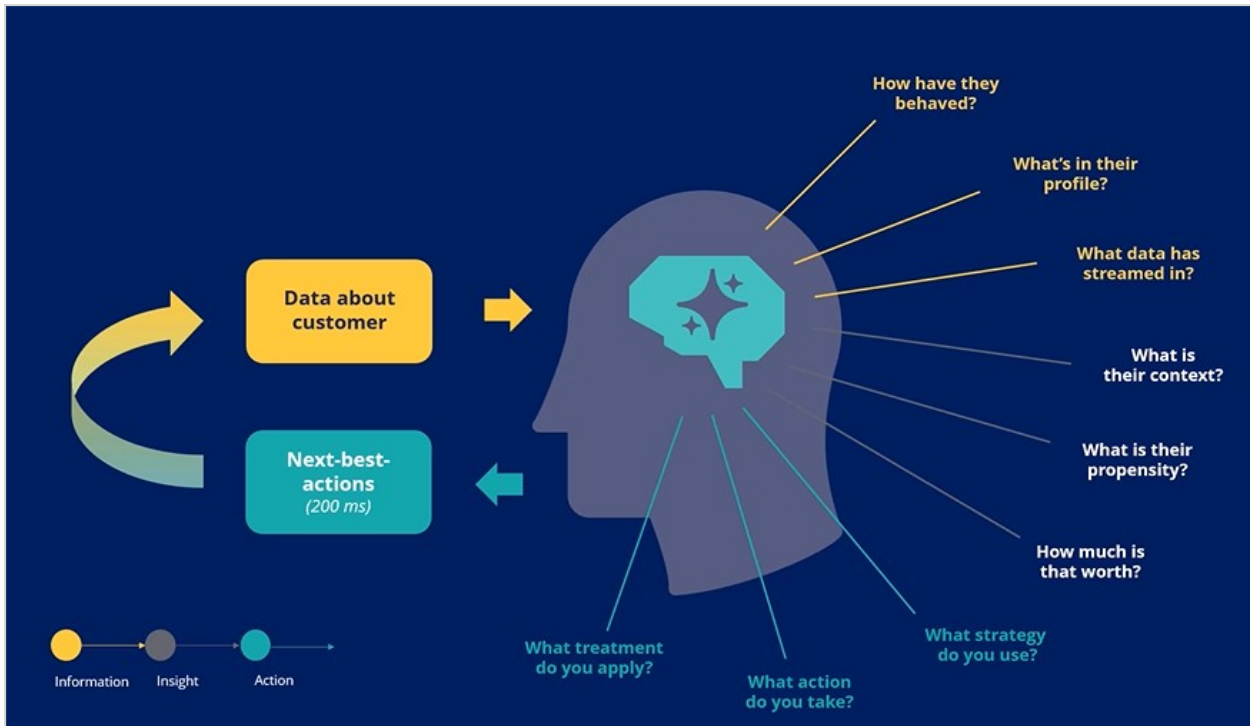
The call details from the interactive voice response (IVR) system indicate that Emily wants to discuss recent credit card transactions.

Once the CSR accepts the call, all relevant details about Emily are displayed on the main application window. The next-best-action recommendations then guide the CSR to take the next step with Emily.

In the lower-left corner of the screen, you can see the next best action that Customer Decision Hub recommends to the CSR.

Customer Decision Hub is an "always on" centralized decision management "brain" that calculates a one-to-one business case for every next-best-action recommendation. Customer Decision Hub combines the customer profile with previous interaction results, the current call context, and business rules to create the business case, and then applies predictive analytics.

Customer Decision Hub reevaluates the next best action and delivers a new recommendation when any new information becomes available. For example, when the customer responds to the recommended action.



In this case, the recommended action is to start a service task to handle Emily's transaction dispute. So, the CSR carries out the task. However, the CSR is always in control and can select other service actions as appropriate during the conversation.

Dispute transactions

Capture dispute details S-2003 PENDING-INTAKE

Card * Dispute reason *

*****7340 - Emily Foster (Active) ▼

Duplicate charge ▼

Cancel

Once the CSR completes the task, the system refreshes the next best action to show the next recommended action, which is to present a credit card offer.

Customer Decision Hub analyzed Emily's credit score, which indicates that she is a customer with a high credit score.

So, the highest recommendation is for a Standard credit card, a top offer, which is relevant for customers with high credit scores.

The CSR can view more details about the recommended offer to discuss its benefits with Emily further.

Get 1% cash back on every purchase

Why this is relevant

Customer with high credit score

∨ **Benefits**

\$200 cash rewards after you spend \$500 on purchases in the first 3 months from account opening. Earn unlimited 1% cash back on every purchase. No annual fee.

∨ **Pricing**

10% interest for 18 months when you transfer other card balances.

After learning about the benefits of the offer, Emily is not convinced that it is a good offer for her. Therefore, the CSR presents the other available offers recommended by Customer Decision Hub.

The CSR then presents the second top offer that is in the suggestions section. Now, Emily is interested in the offer, and she decides to accept it.

Other suggestions

Rewards card

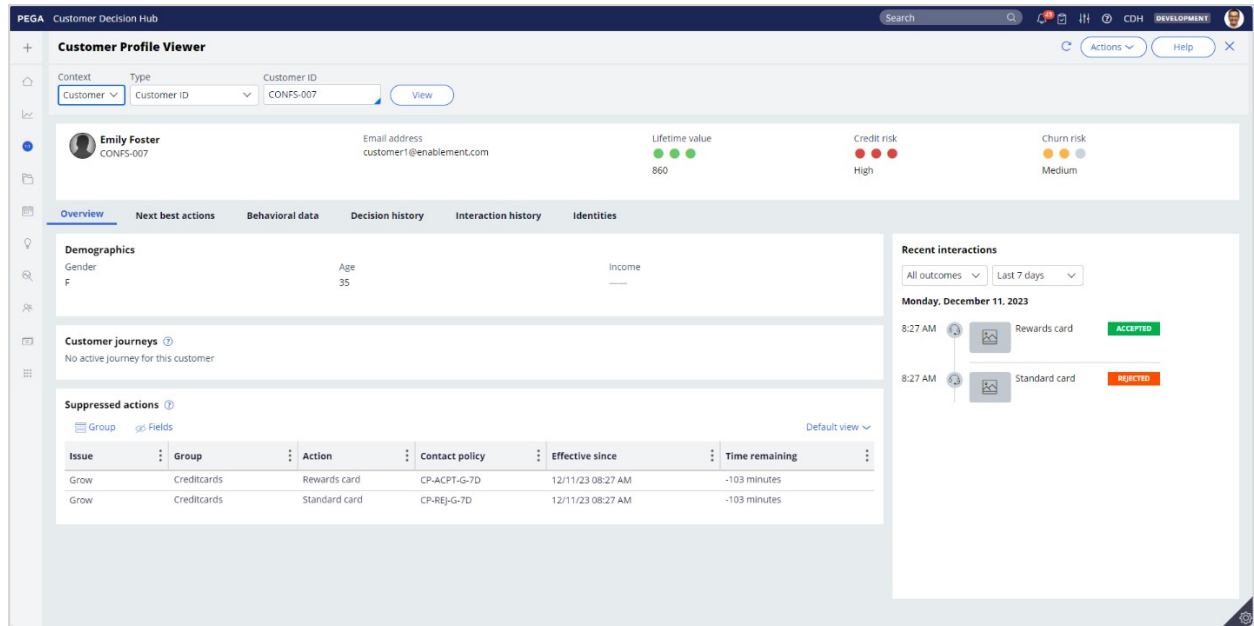
Get 1% cash back on every purchase

[More](#)

When the CSR accepts the offer, the offer fulfillment is complete, and the customer response is recorded in the Interaction History.

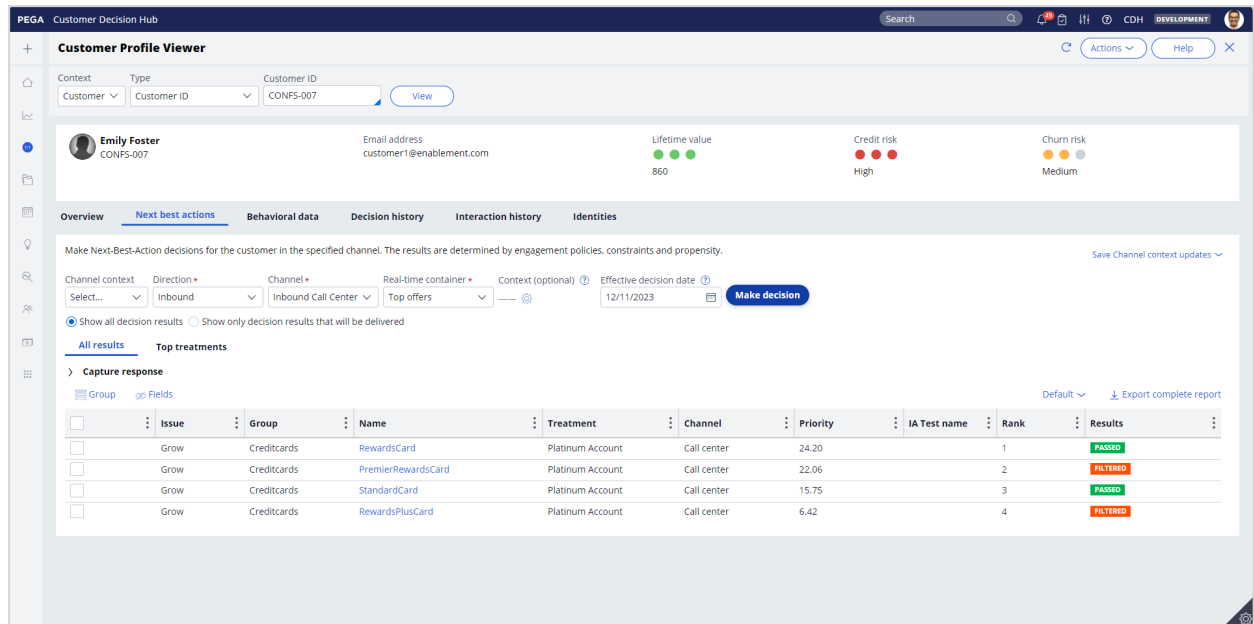
As a decisioning architect, you can use Pega Customer Decision Hub to input or configure data necessary to make the next best recommendations.

The recent interaction between the CSR and the customer, Emily Foster, was captured and is now visible on the overview tab of Customer Profile Viewer.



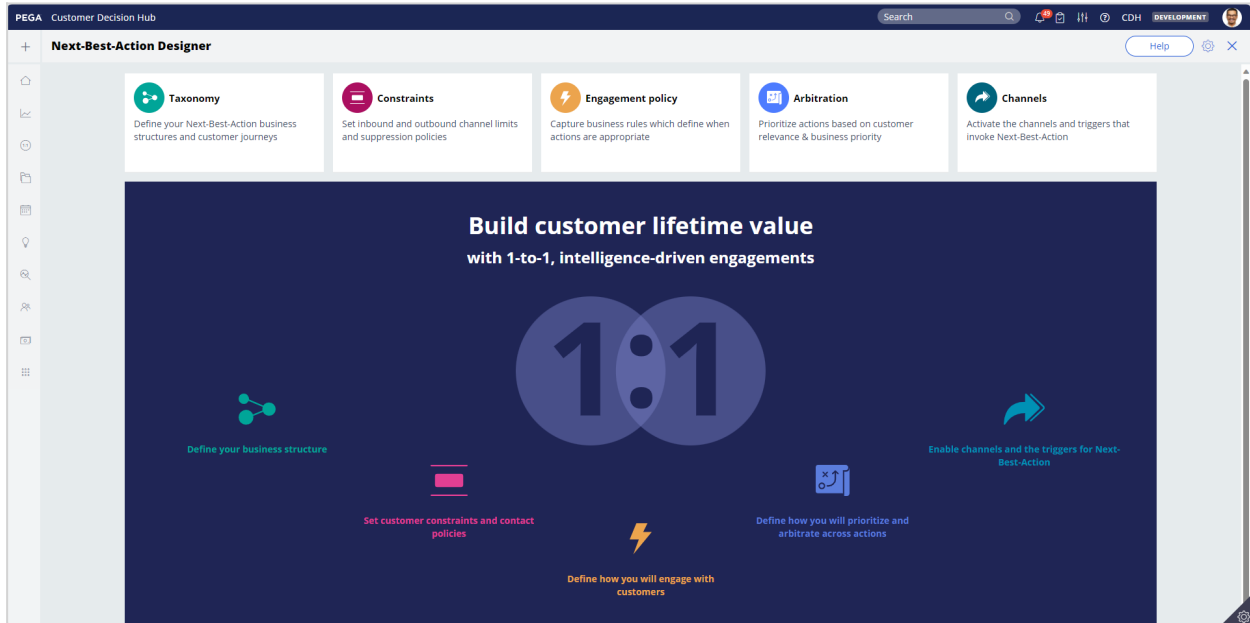
Customer Profile viewer allows you to explore customer profile details such as demographics, active customer journeys, suppressed actions, recent interactions behavioral data, decision history, and interaction history.

The Next best actions tab in Customer Profile Viewer shows what actions can be presented to the customer. You can provide some input parameters like direction, channel type, which real-time container to test, and what the next-best-action results would be.



Apart from the tailored offers, you can also examine the arbitration factors: prioritization, weighing, and propensity for the final result.

Next-Best-Action Designer guides you through the creation of a next-best-action strategy for your business. Using its intuitive interface, proven best practices, and sophisticated underlying decisioning technology, you can automatically deliver personalized customer experiences across inbound, outbound, and paid channels.



On the engagement policy tab of NBA designer, you can observe previously defined sets of actions belonging to specific issues and groups (in this case cross sell, credit cards).

You, as the decisioning architect, can also define group-level engagement policies and action-level engagement policies. All the applied eligibility, applicability, and suitability conditions are visible on the Customer offers tab.

Customer actions Policies: **E** **A** **C** Actions: 4

E Eligibility ⓘ
(Customer: Age is greater than or equal to 18)

A Applicability ⓘ
(Customer: Count of credit card accounts is equal to 0)

S Suitability ⓘ
No group criteria defined

C Contact policy ⓘ

- > 7-day group clicks: Track Clicks for all actions in the group over the past 7 days
- > 7-day action impressions: Track Impressions for the action over the past 7 days

Actions ⓘ

4 Actions (4 with specialized policies)

Name	Specialized policies
Standard card	E
Rewards card	E
Rewards Plus card	S
Premier Rewards card	S

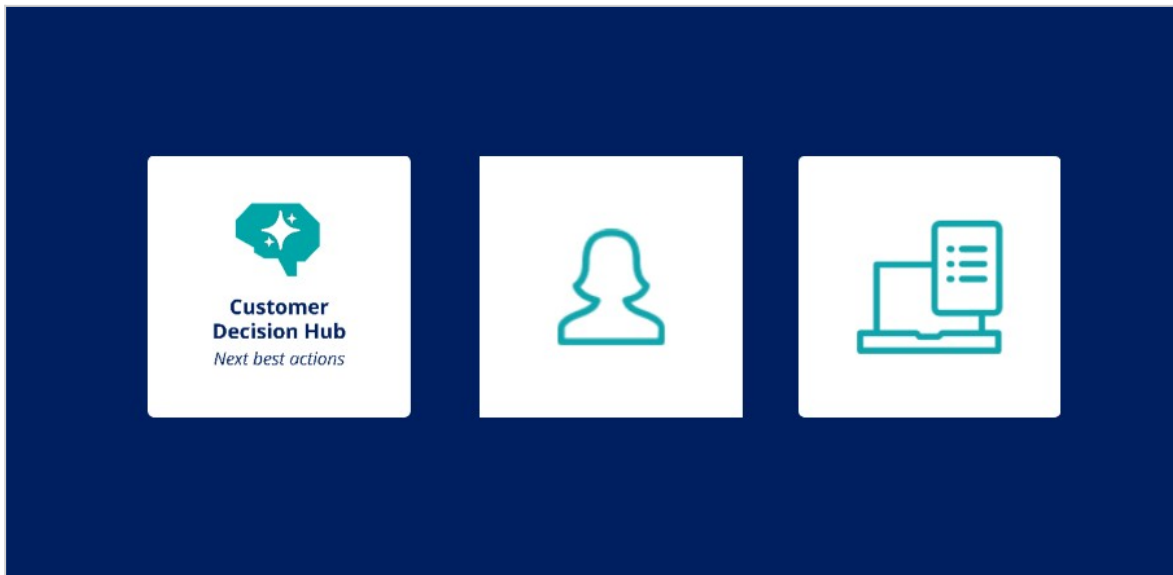
For an action, you can also define channel-specific treatments, which means that the action can be delivered to a customer by using a particular channel. The treatment contains all the data necessary for the CSR to offer the action to a customer.

To summarize, the Customer Sales Representative offered a customer a Next-Best-Action that was selected by the “always on” centralized decision management “brain” from a set of actions defined and configured by the Decisioning Architect.

Next best action on digital

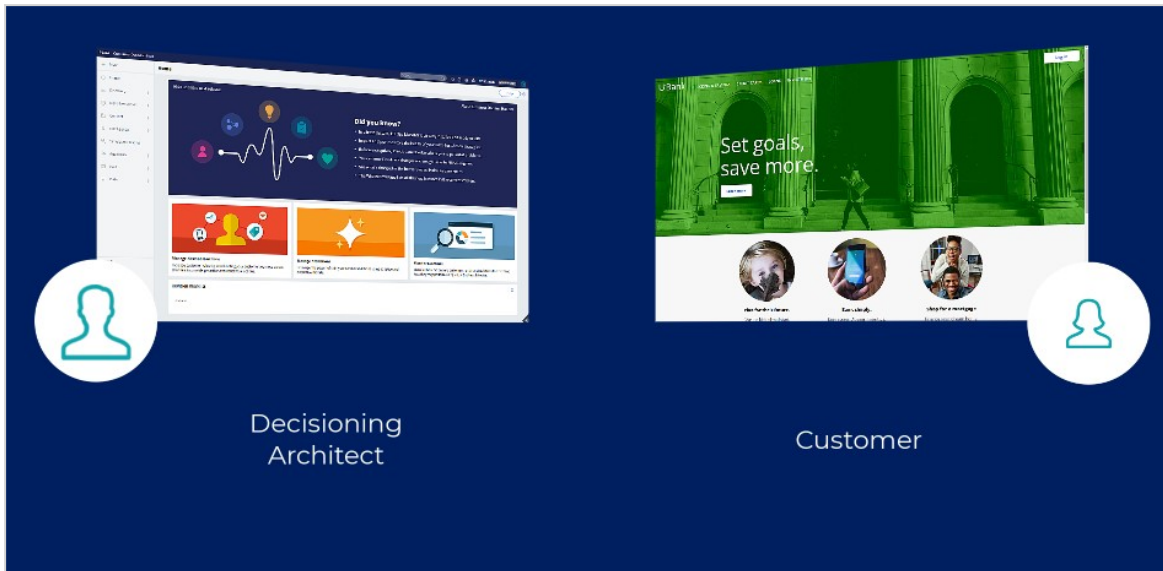
Transcript

Pega Customer Decision Hub can deliver the next-best-action recommendations through various inbound and outbound channels. Outbound messages involve proactively reaching out to customers, while inbound channels demand action from a customer. One such inbound channel is the web. For example, when a customer visits the website, they see the intended offers.



Consider the following web channel scenario, which is a typical cross-selling use case.

The next-best-action recommendations help to ensure that the customers of U+ Bank can see the tailor-made offer when visiting the website of the bank. The centralized decision management "brain" of Customer Decision Hub selects the next best action that is displayed for a customer based on configurations implemented by the Decisioning Architect.



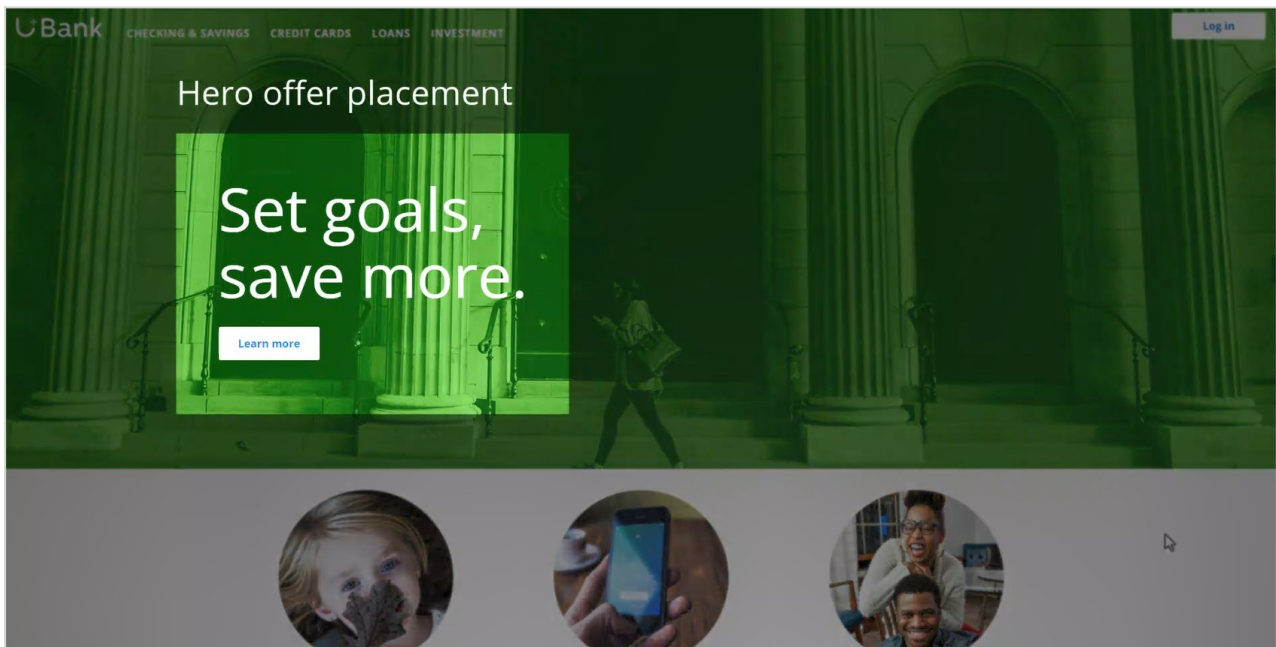
U+ is a retail bank that wants to use its website as a marketing channel to improve One-to-one Customer Engagement, drive sales, and deliver next best actions in real time.

The bank has decided to use Pega Customer Decision Hub™ to recommend more relevant banner ads to its customers when they visit the website.

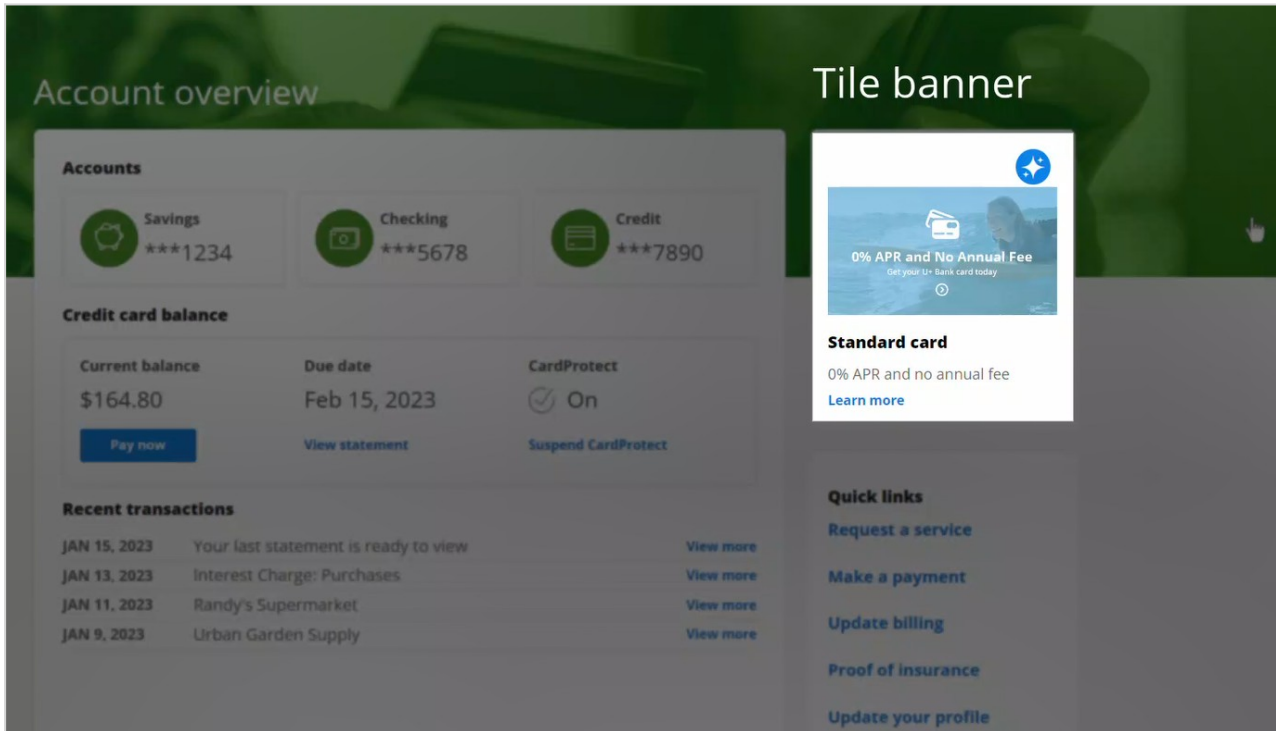
When visiting the U+ Bank website, Troy, a customer, can see banner ads on various pages.

For example, on the home page, U+ displays a hero banner at the top of the page, which is typically a larger image with larger typeface.

Under that banner, there is space to display several tile banners, which are typically smaller.



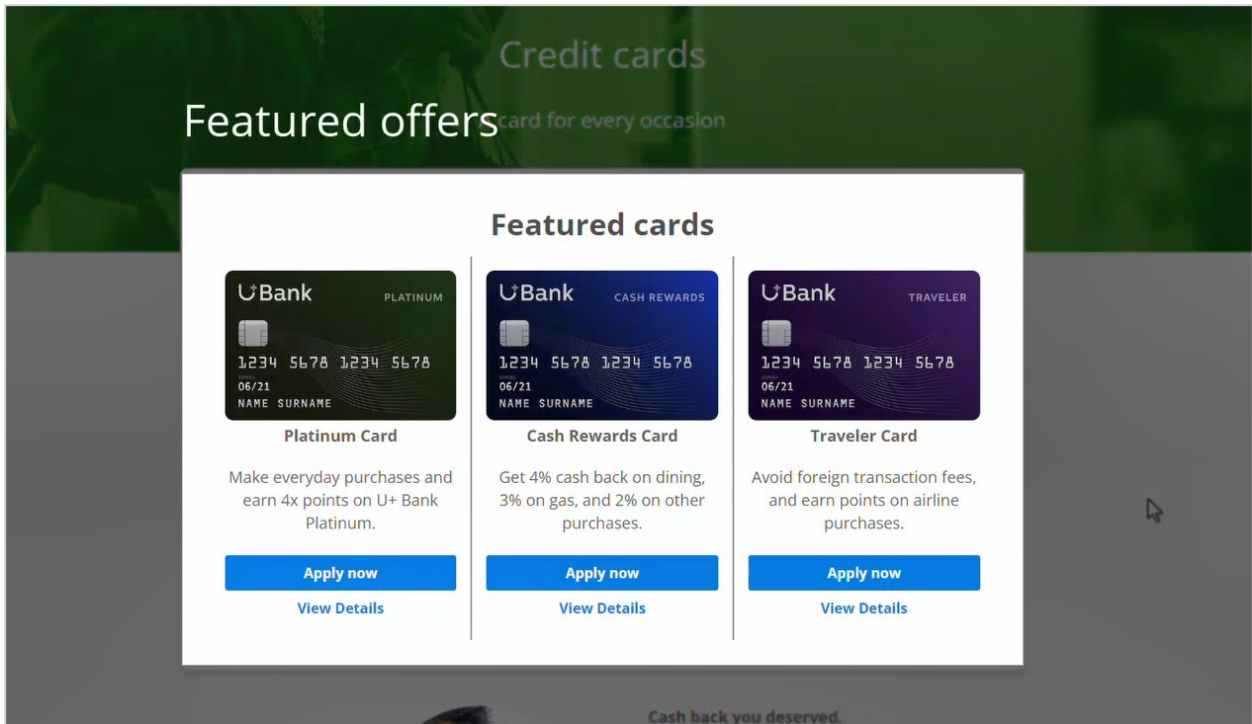
When Troy logs in to his personal portal, he also sees a tile banner on the **Account overview** page.



The main goal of U+ at this stage is to increase customer web engagement. When Troy clicks the **Learn more** link, the action shows his interest.

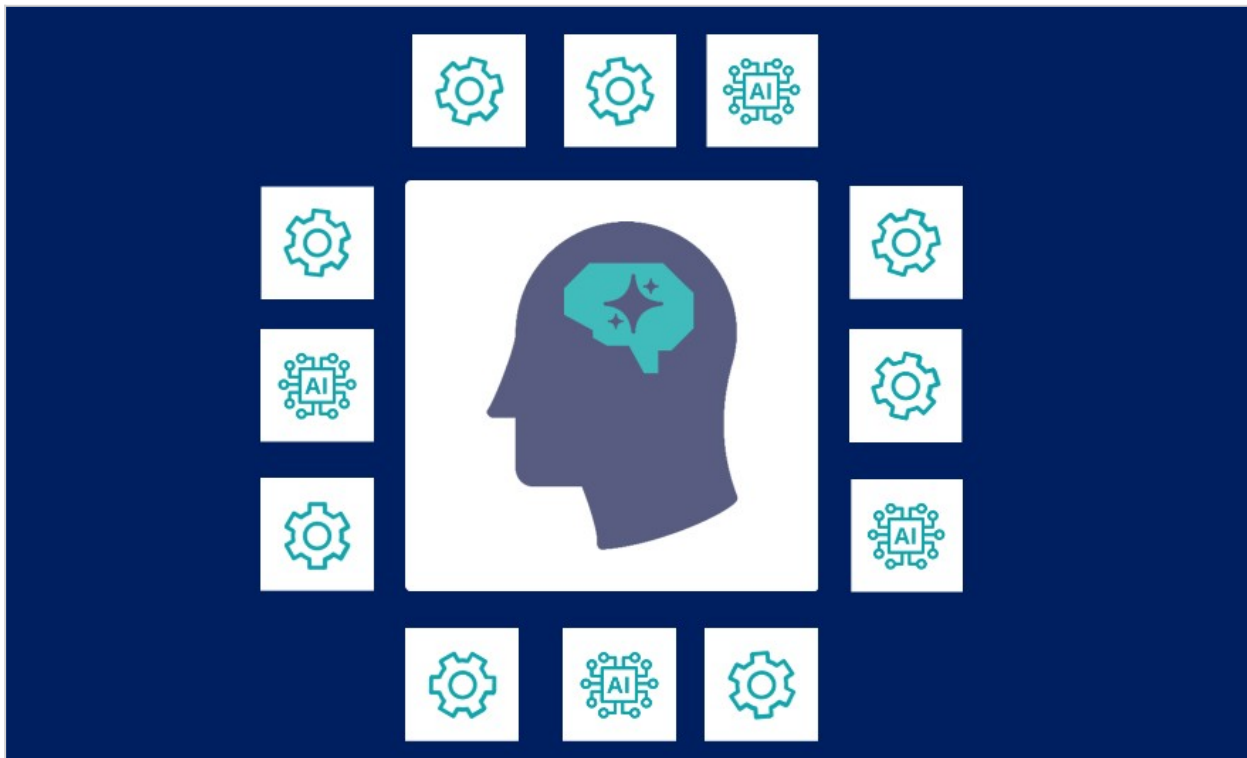
This interaction is recorded as a click-through and helps measure the web engagement of the customer.

After clicking **Learn more**, Troy can see featured credit card offers for which he can apply.

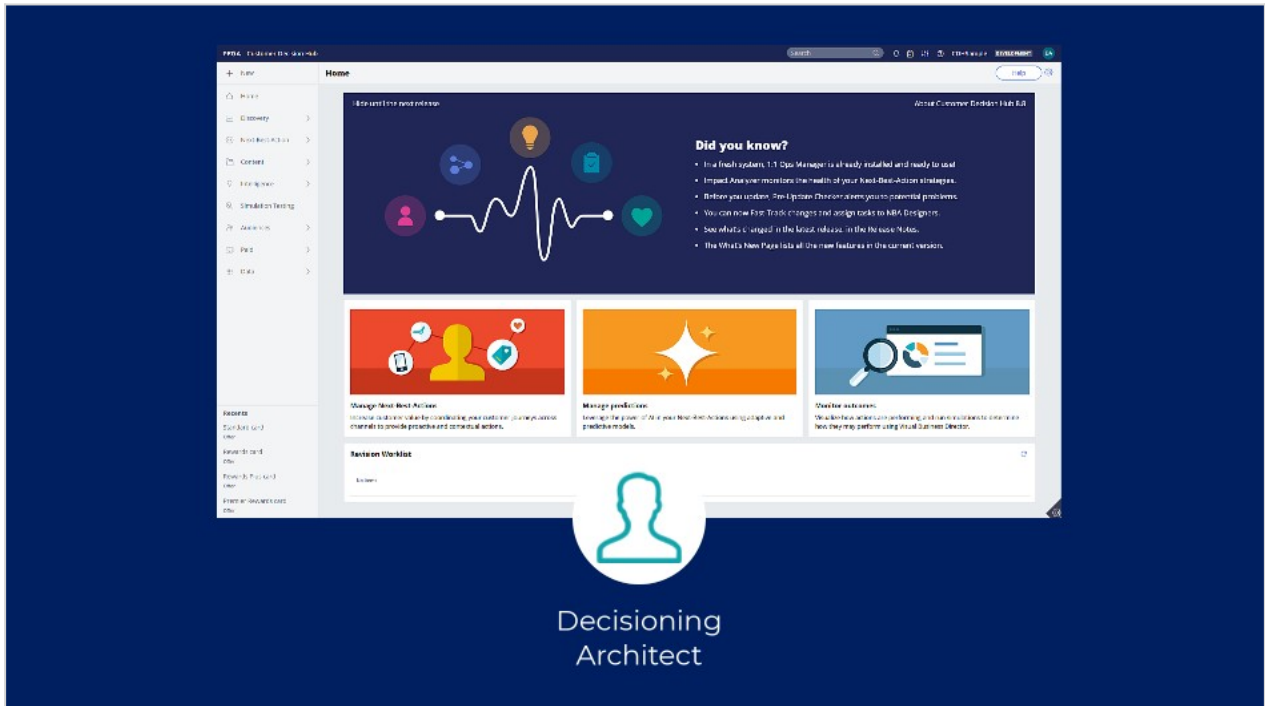


Behind the scenes of the next best actions, the complex decision engine is working to rank and select the best offer to display for each customer who visits the website.

A combination of artificial intelligence (AI) and other business rules determine the options.



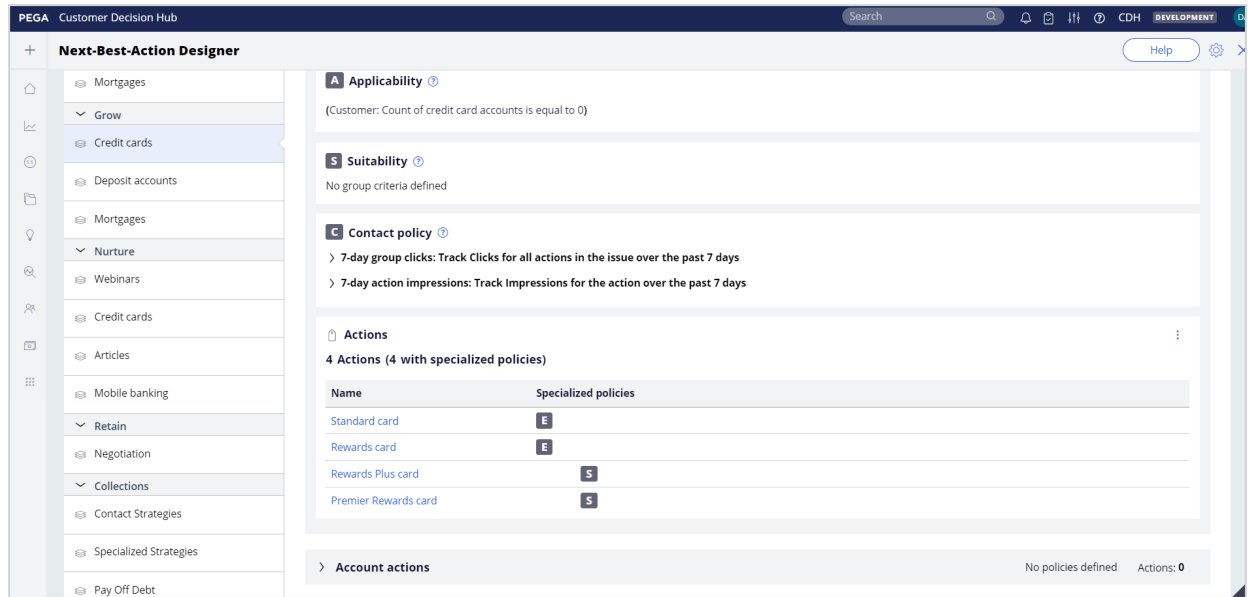
As a Decisioning Architect, you can define the business rules and other settings in Pega Customer Decision Hub by using **Next-Best-Action Designer**.



Next-Best-Action Designer allows you to configure how you want the always-on brain to select the best offer for a customer.

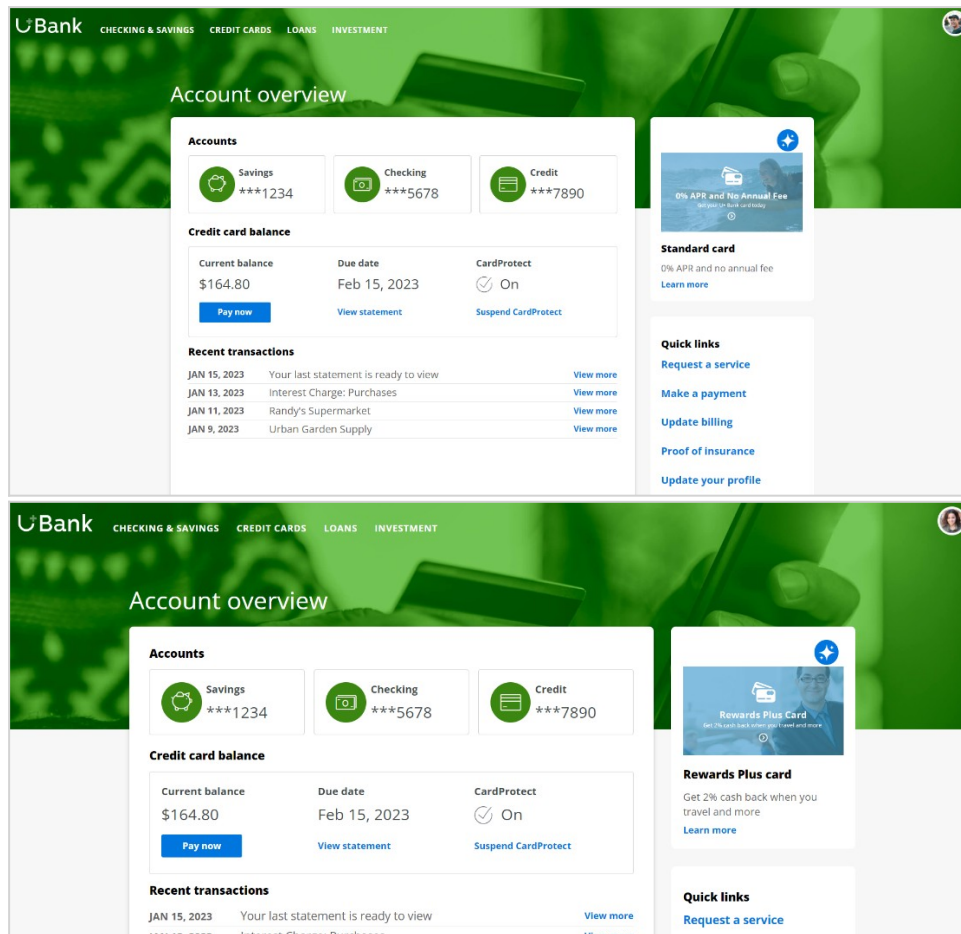
First, to ensure that the offer is displayed on the website, you need to verify that the business structure is in place and that the actions and treatments are defined. You can check that on the **Engagement policy** tab of Next-Best-Action Designer.

In the **Grow** issue and **Credit cards** group, four actions are defined.



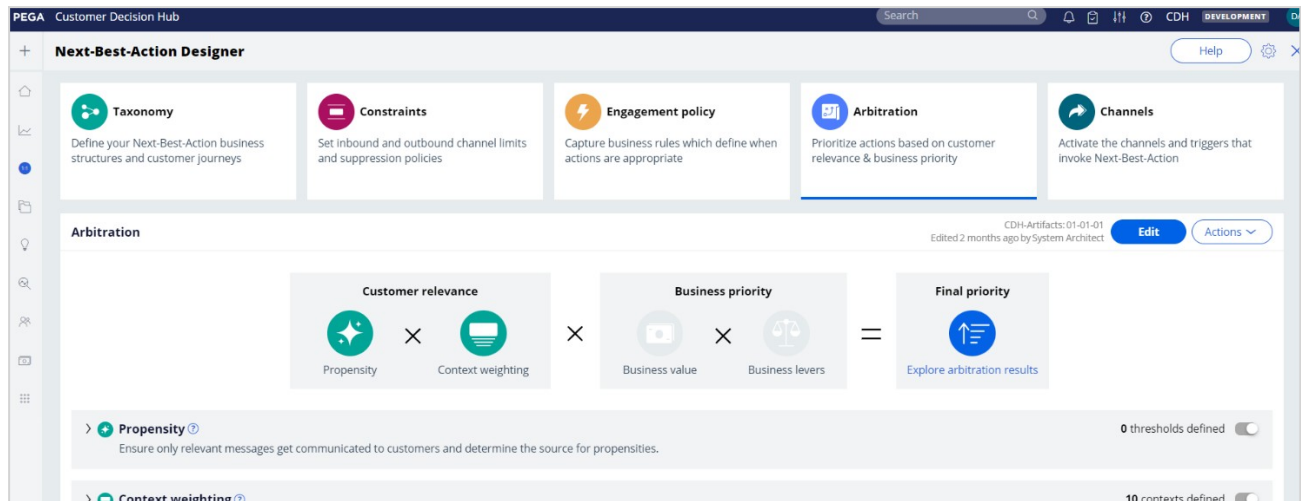
Each action has a set of conditions such as eligibility, applicability, and suitability specified. These conditions are engagement policies, and they qualify an offer or a group of offers for a customer. As a result, customers see only the offers which the organization believes they should receive.

For example, when Troy logs in, he sees the Standard Card, but this offer is not applicable for Barbara, so it is never displayed; instead, she sees the Rewards Plus Card.



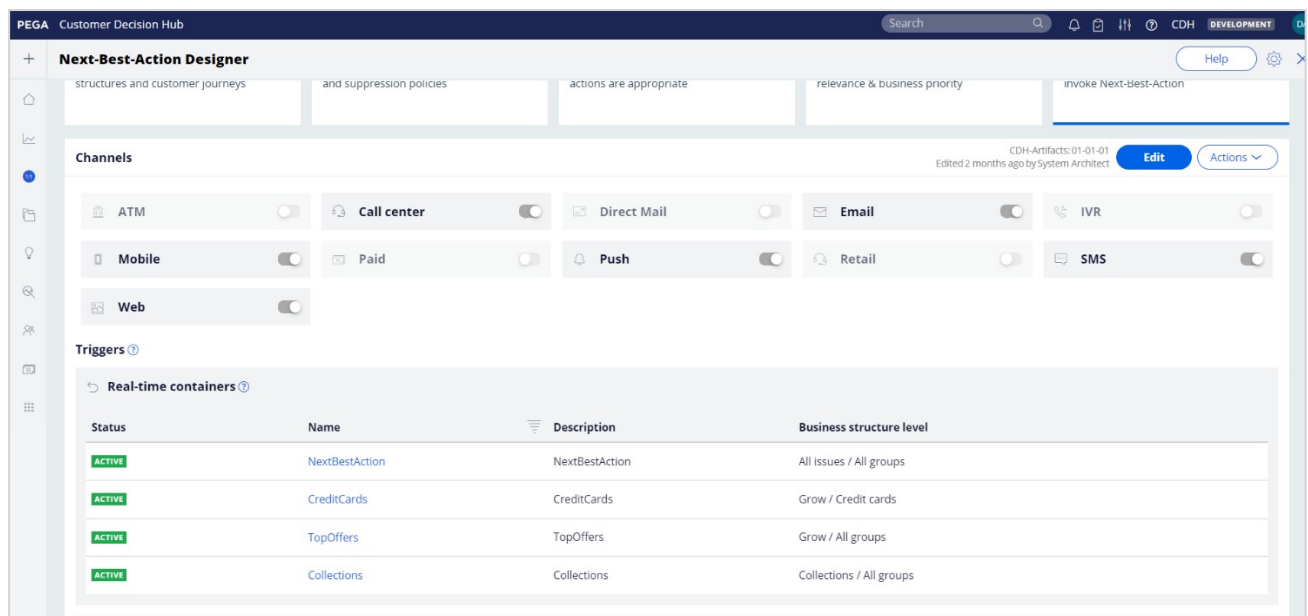
After the engagement policies have "whittled down" the total possible offers to a few, Customer Decision Hub uses arbitration to choose the top offer based on what is relevant for the customer right now.

Arbitration aims at balancing customer relevance with business priorities. Specifically, propensity, context weighting, business value, and business levers have numerical values. The system then uses a simple formula to arrive at a prioritization value, which determines the selection of the top offer.

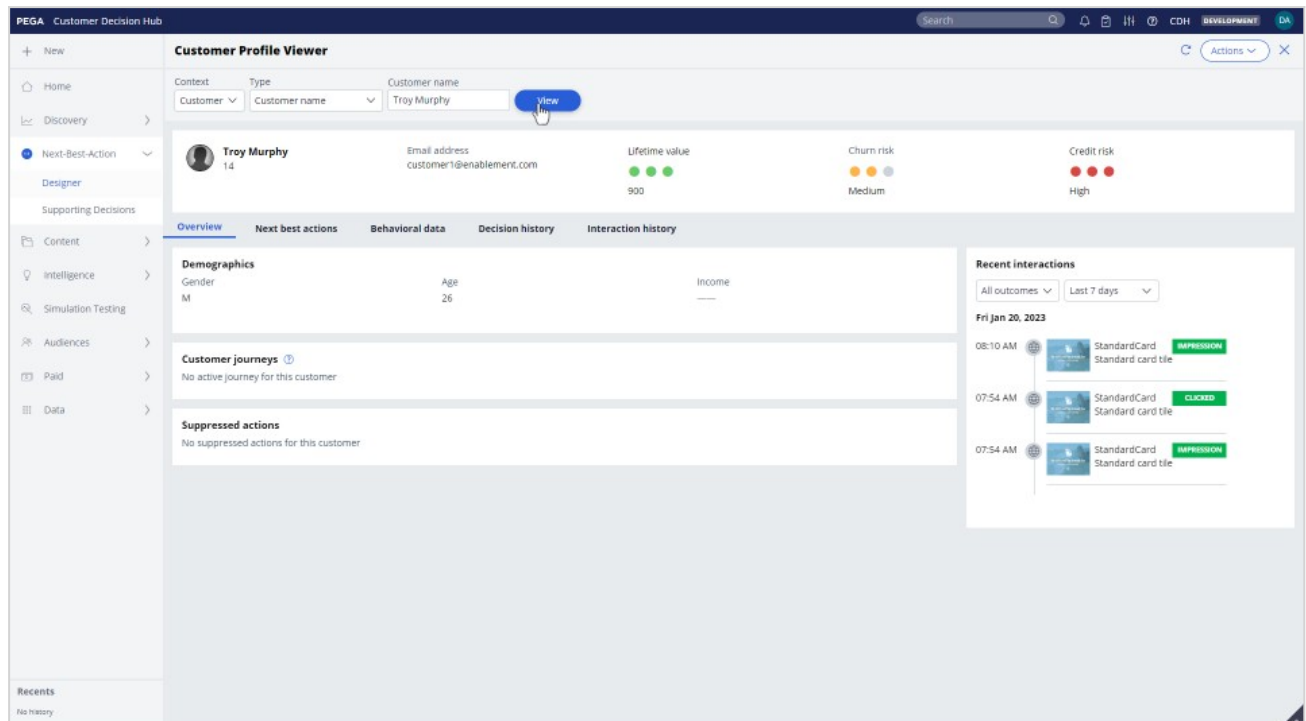


Additional configurations are required, such as enabling a web channel and adding a real-time container, to present the offer on the website. Both settings are configurable on the **Channels** tab of Next-Best-Action Designer.

A real-time container is a service that manages communication between Customer Decision Hub and external channels. It fetches the data configured in Customer Decision Hub and makes it possible to display it on a website.



The recent Troy interaction that Customer Decision Hub captured is now visible in Customer Profile Viewer. You can observe whether the interaction was an impression or a click.



On the **Next best actions** tab of the Customer Profile Viewer, you can see the next-best-action results based on the provided input parameters (for example, inbound direction and a web channel).

In the action results table, you can observe that two out of four credit cards were filtered out, which means that they will not be presented to Troy.

You have access to exclusive Pega GenAI™ capabilities if you are on Pega Cloud. After you view the next-best-action decisions for a customer, you can use the AI Insight feature, which uses Pega GenAI™ technology, to better explain the results that are displayed in the table.

Enabling Pega GenAI, results in changes to the UI of Customer Profile Viewer. The fields marked with the Polaris icon give you access to the AI Insight feature.

AI Insight



The Grow - Creditcards Issue Group has a total of 4 candidate Offers. Out of these, 2 actions passed and 2 actions were filtered out. The most occurring filter reason is "Excluded by suitability", which filtered out 2 actions.

Group Fields Default Export complete report

	Issue	Group	Name	Treatment	Channel	Priority	IA Test name	Rank	Results
	Grow	Creditcards	RewardsPlusCard	Rewards Plus card tile	Web	26.36		1	FILTERED
	Grow	Creditcards	PremierRewardsCard	Premier Rewards card tile	Web	17.27		2	FILTERED
	Grow	Creditcards	RewardsCard	Rewards card tile	Web	4.06		3	PASSED
	Grow	Creditcards	StandardCard	Standard card tile	Web	3.50		4	PASSED

After processing the results, the AI Insight provides the summary for the tested configuration. In this example, you receive a clear explanation that there are four candidate offers for Troy. However, only 2 actions passed, and 2 actions were filtered out due to the Suitability conditions.

By clicking on the Polaris icon for each row in the Results column, you can access an explanation of the action results in the Action results explanation window, specific to the action.

Name	Treatment	Channel	Priority	IA Test name	Rank	Results
RewardsPlusCard	Rewards Plus card tile	Web	26.36		1	FILTERED 
PremierRewardsCard	Premier Rewards card tile	Web	17.27		2	FILTERED 
RewardsCard	Rewards card tile	Web	4.06		3	PASSED
StandardCard	Standard card tile	Web	3.50		4	PASSED

In this example, you will learn that the RewardsPlusCard is filtered out due to the suitability condition.

Action results explanation GENERATED USING AI ×

Grow / Creditcards / RewardsPlusCard / Rewards Plus card tile FILTERED

The offer you mentioned, **RewardsPlusCard**, is currently being filtered out due to the reason **"Excluded by suitability"**. This means that the offer does not meet the required criteria or properties to be delivered to the customers.

To make this offer pass and be delivered to the customers, you need to address the suitability issue. The specific reason for the exclusion is not provided in the information you provided, so it would be helpful to investigate further to determine the exact criteria that are not being met.

Once you have identified the specific criteria, you can take the following probable approach to make the offer pass:

1. Review the properties of the offer and compare them to the desired criteria.
2. Identify any gaps or discrepancies between the offer properties and the required criteria.
3. Make necessary adjustments to the offer properties to align them with the desired criteria.
4. Ensure that the adjusted offer properties meet the suitability requirements.
5. Test the modified offer to confirm that it now passes the suitability criteria.

By following this approach, you can increase the chances of the offer being delivered to the customers based on the desired properties.

Learn more about:
[Excluded by suitability](#)

You can explore the reasons behind this decision by utilizing the "Filtering by feature" option. The table displaying the decision results has been enhanced with multiple categories that might serve as the underlying cause for excluding certain actions.

You can for example, see how the engagement policies are used or why a customer does not receive an action.

To summarize, when visiting the bank's website, the customer can see the next best action that the "always on" centralized decision management "brain" of Customer Decision Hub selected from a set of actions defined and configured by the Decisioning Architect.

You have reached the end of this video.

Arbitration fundamentals

Description

Customers might still qualify for more than one action after you apply engagement policy rules. Learn how every next best action weighs customer needs against business objectives to optimize decisions based on the priorities that the business sets.

Learning Objectives

- Describe what action arbitration is and how it works.
- Explain how the system considers customer needs and business objectives during arbitration.
- Prioritize actions based on AI.

Action arbitration

Pega Customer Decision Hub™ (CDH) combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that CDH uses to make intelligent decisions. Arbitration aims to balance customer relevance and business priorities by weighing numerical values for the following factors: propensity, context weighting, business value, and business levers. Learn to create a simple formula for arriving at a prioritization value, which is used to select the top actions.

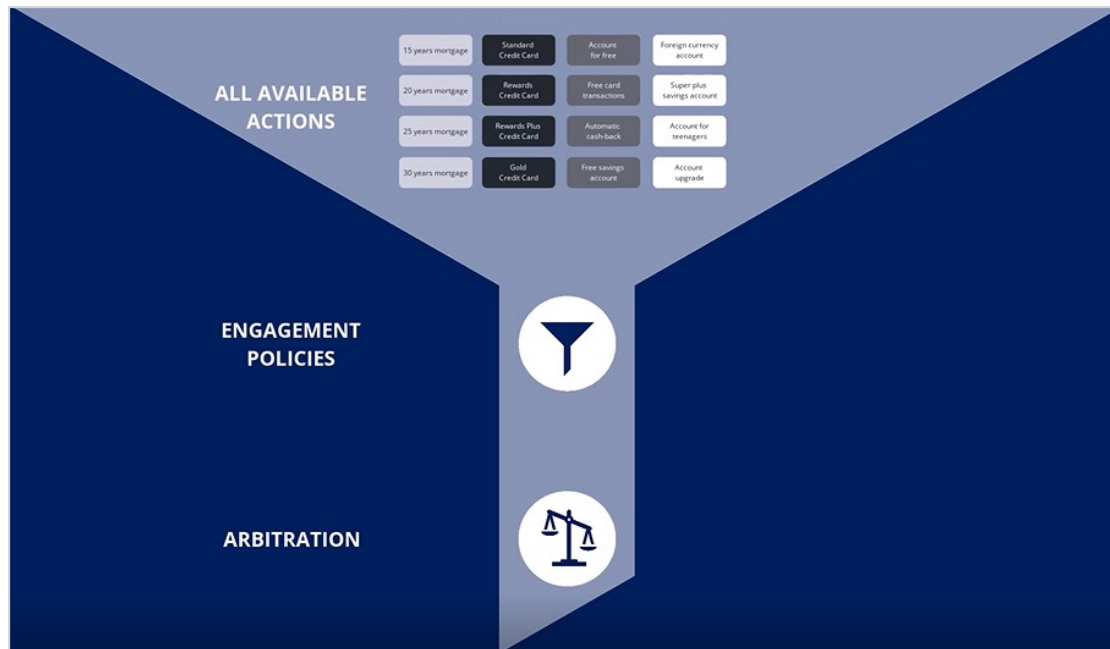
Transcript

This video explains the concept of action arbitration in Pega Customer Decision Hub.

Pega Customer Decision Hub combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that it uses to make intelligent decisions. These decisions are known as Next-Best-Actions. Every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business manager.



U+, a retail bank, wants to avoid offering random actions to its customers, so it uses Pega CDH to rank and select the next best actions for every customer. All the actions from the Bank's offer move through several steps that whittle them down until the Next Best Action is selected.



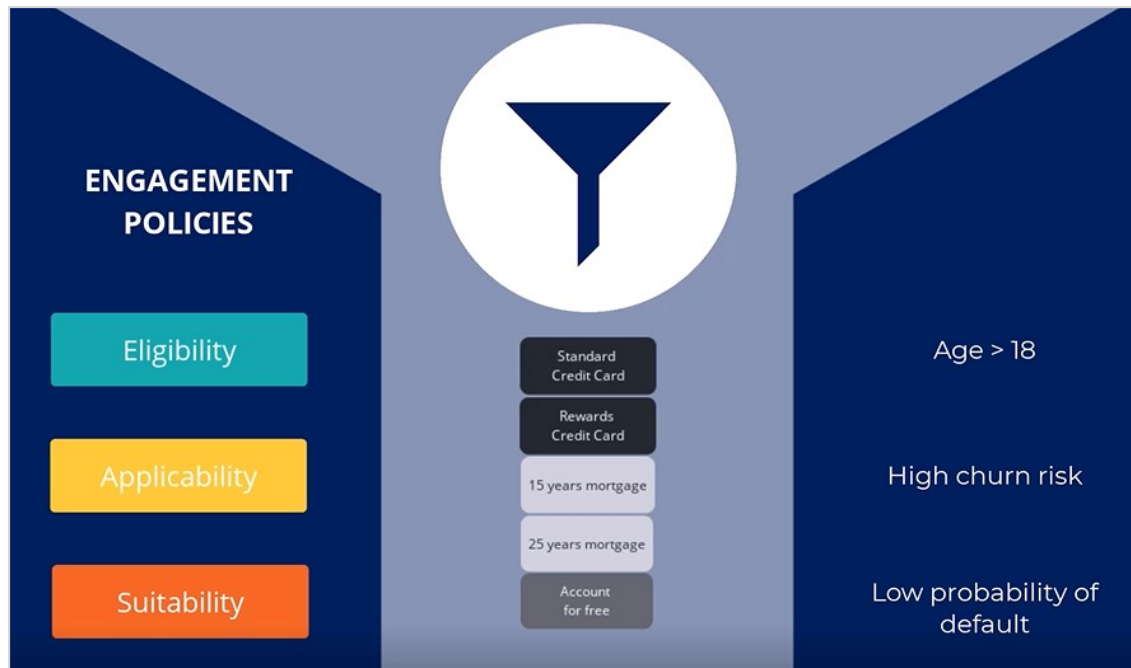
The process starts with the catalog full of actions that the organization can offer to its customers. These actions belong to various issues and groups.

The first step is to apply the Engagement Policies: Eligibility, Applicability, and Suitability conditions.

Eligibility rules are used to determine whether customers are eligible for an action. Users must meet the condition for CDH to consider the action eligible. An example of an eligibility condition is the customer's age. A customer can be offered a credit card only if they are 18 years old or older.

Applicability rules are used to limit what to offer, based on a customer's current situation. For example, the bank wants to show a retention offer, instead of a credit card offer, to customers who are likely to churn in the near future.

Suitability rules are used to define whether an offer is appropriate for a customer. For example, if a customer is in debt and has a lot to pay off, the bank does not want to offer them a 30-year mortgage as it would not be empathetic.



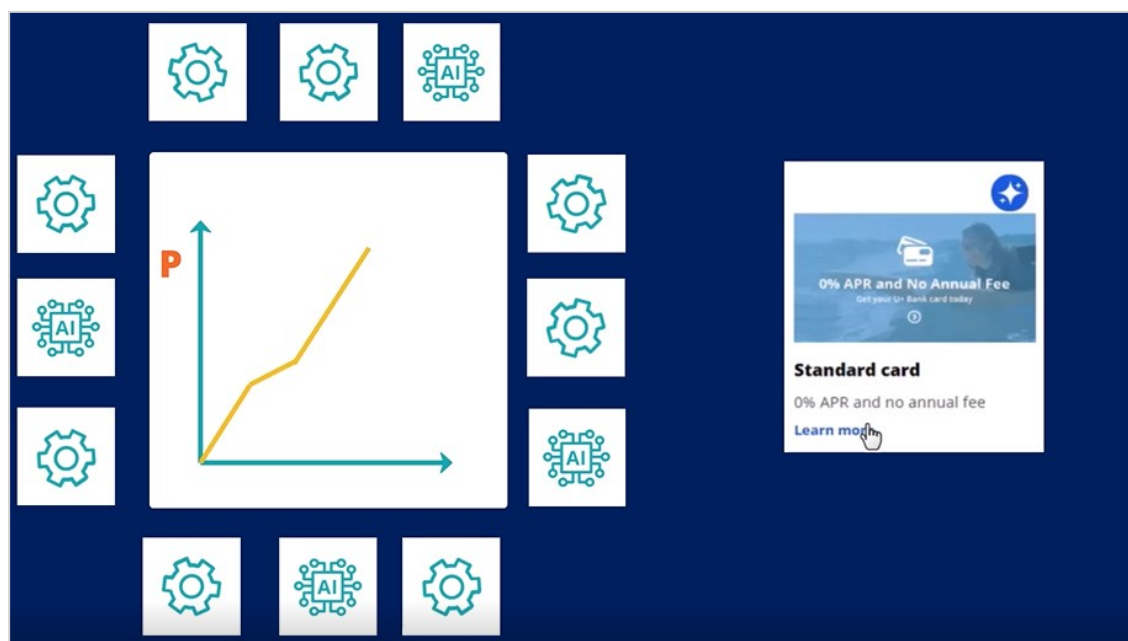
The initial catalog of actions has been narrowed down by engagement policies. These engagement policies are a combination of business rules, for example, certain age requirements for a loan or some product compatibility rules, and predictive models which help predict customer behavior as customer churn or probability of default. These predictive models are created by data scientists in Pega, or other tools familiar to data scientists like Python, R, and H2O.AI.

Before being presented to customers, the remaining actions go through a final stage known as the arbitration stage. This stage focuses on selecting and prioritizing the best actions that are most relevant for the customer at the present time. Arbitration balances customer relevance with business priorities using four components: **Propensity, Context Weighting, Business Value**, and **Business Levers**, each represented by numerical values.



Propensity is the predicted likelihood of positive behavior, such as the likelihood of a customer accepting an offer.

The propensity value is calculated by AI and it is the foundation of the arbitration process. The higher the likelihood of a customer accepting an offer, the higher the Propensity value for that offer. For Example, when a customer is eligible to receive a Standard Credit Card and clicks the Standard Credit card offer on the Bank's website the propensity of this action increases.



The AI is driven by adaptive models, self-learning mathematical models that use machine learning to make calculations. Adaptive models are a sub-group of predictive models that continuously adapt to new data as it becomes available. This means that the models are updated incrementally in real-time based on incoming data.

The Customer Decision Hub is configured to calculate the propensity for each treatment. For the web treatment, **Web_Click_Through_Rate** is the out-of-the-box adaptive model. It predicts the likelihood of a customer clicking on a banner, which is considered a positive behavior.

Context Weighting allows Pega Customer Decision Hub to consider the situational context for each action. For example, if a customer contacts the bank to close their account, the highest-priority action is to ensure that the customer is retained, even if they are eligible for other offers.

Business Value enables you to assign a financial value to an action and prioritize high-value actions over low-value ones. This value is typically normalized across Issues and Groups. For example, a 25-year mortgage is more profitable than a 15-year mortgage. So, in a situation where a customer is eligible for both plans, the 25-year mortgage will be ranked higher because of its higher business value.

Business Levers allow the business to assert some level of control over the prioritization of actions defined within the system. Levers are used to manually nudge Customer Decision Hub toward Next-Best-Actions based on external factors. For example, the recommended Next-Best-Action might be to offer a credit card to a customer when they visit the home page. However, to meet a business goal, the Mortgage Line of Business favors a mortgage offer, even if that offer is ranked a little lower on the list of possible actions.

Pega AI considers all four components to select the top offer for a customer. A simple formula $P * C * V * L$ is used to arrive at a prioritization value, which is used to select the top actions. The action with the highest priority will be presented to a customer.



In the current example, there are 5 actions that made it until the arbitration phase.

Assume the AI generates a propensity value for each of the actions which is a value between 0 and 1.

Each action has a context weight set, which is a percentage value between 0% and 100%, business value in dollars, and business lever value in a percentage between 0% and 100%.

Using the priority formula PCVL, with all of these values taken into account, we end up with a computed priority. Finally, the action with the highest priority is selected. In this case, it is a Standard Credit Card.

	PROPENSITY	CONTEXT	BUSINESS VALUE	BUSINESS LEVERS	PRIORITY
Standard Credit Card	0.45	5%	\$186	25%	5.23
Rewards Credit Card	0.3	5%	\$238	25%	4.46
15 years mortgage	0.25	2%	\$665	10%	3.6
25 years mortgage	0.2	2%	\$965	10%	4,25
Account for free	0.4	10%	\$56	5%	0.58

In summary, arbitration is used to select and prioritize the best actions that are most relevant for the customer at the present time. Arbitration balances customer relevance with business priorities using the formula $P * C * V * L$. The top actions are selected based on the result of this multiplication – the prioritization value.

Action prioritization with AI

Explore how AI-based arbitration works and how AI predicts customer behavior. Arbitration aims to balance customer relevance with business priorities. Pega Customer Decision Hub™ uses a formula to arrive at a prioritization value and select the top actions. The formula uses the propensity value that AI calculates. Propensity is the predicted likelihood of positive behavior, such as the likelihood of a customer accepting an offer.

Transcript

This demo shows you how AI-based arbitration works and explains how AI predicts customer behavior.

U+ Bank, a retail bank, uses Pega Customer Decision Hub to display marketing offers to customers on its website. The bank wants to display more relevant offers to customers based on their behavior. There is an initial set of actions that belong to **Grow** Issue and **Credit Cards** Group created in Customer Decision Hub.

The screenshot displays the 'Next-Best-Action Designer' interface. On the left, a navigation menu includes 'New', 'Home', 'Discovery', 'Next-Best-Action', 'Designer', 'Supporting Decisions', 'Content', 'Intelligence', 'Simulation Testing', 'Audiences', 'Paid', and 'Data'. The main area shows a list of actions under the 'Grow' issue and 'Credit cards' group. The 'Actions' section lists 4 actions with specialized policies:

Name	Specialized policies
Standard card	E
Rewards card	E
Rewards Plus card	S
Premier Rewards card	S

Below the table, there is a section for 'Account actions' with 'No policies defined' and 'Actions: 0'.

After filtering by Engagement Policies, customer Troy is eligible for two credit card offers – Standard Card and Rewards Card. When he logs into the bank’s website, he sees the **Standard Card** offer. When you click the **Polaris** icon next to the card offer, you can examine the arbitration details, such as propensity value and priority value.

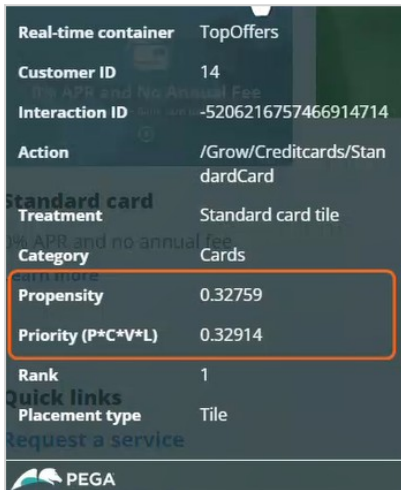
The screenshot shows a bank's account overview page with a Pega overlay. The bank's navigation bar includes 'CHECKING & SAVINGS', 'CREDIT CARDS', 'LOANS', and 'INVESTMENT'. The main heading is 'Account overview'. The 'Accounts' section lists Savings (***1234), Checking (***5678), and Credit (***7890). The 'Credit card balance' section shows a current balance of \$164.80, a due date of Mar 15, 2023, and CardProtect status 'On'. The 'Recent transactions' section lists several transactions from February 2023. The Pega overlay displays offer details for a 'Standard card' with a propensity of 0.32759 and a priority (P*C*V*L) of 0.32914. The overlay also shows a 'Real-time container' with 'TopOffers' and a 'PEGA' logo at the bottom.

The offer presented to the customer is the one with the highest priority that the Pega Customer Decision Hub selects based on the arbitration settings. You define the Arbitration settings in the Next-Best-Action Designer of Customer Decision Hub. The purpose of arbitration aims is to balance customer relevance with business priorities. The system uses four components of arbitration, **Propensity (P)**, **Context weighting (C)**, **Business value (V)**, and **Business levers (L)**, represented by numerical values, to achieve this balance. With the $P \times C \times V \times L$ formula, Customer Decision Hub arrives at a prioritization value, which the system uses to select the top offers.

The screenshot shows the 'Next-Best-Action Designer' interface. The top navigation bar includes 'Taxonomy', 'Constraints', 'Engagement policy', 'Arbitration', and 'Channels'. The 'Arbitration' section is highlighted, showing a formula for calculating 'Final priority': $\text{Propensity} \times \text{Context weighting} \times \text{Business value} \times \text{Business levers} = \text{Final priority}$. Below the formula, there are four sections: 'Propensity' (0 thresholds defined), 'Context weighting' (10 contexts defined), 'Business value' (0 thresholds defined), and 'Business levers' (0 levers defined). The interface also includes a 'Help' button and a 'CDH-Architect 01/01/21' timestamp.

Typically, the propensity for every action starts at 0.5 or 50 percent, the same as the flip of a coin. This value is the default because the AI has no past customer behavior on which to base its predictions. After every interaction, the propensity increases or decreases accordingly.

The **Standard Card** that is displayed for the customer Troy shows the propensity and priority equal 0.32, and it means that the system captured several interactions before.



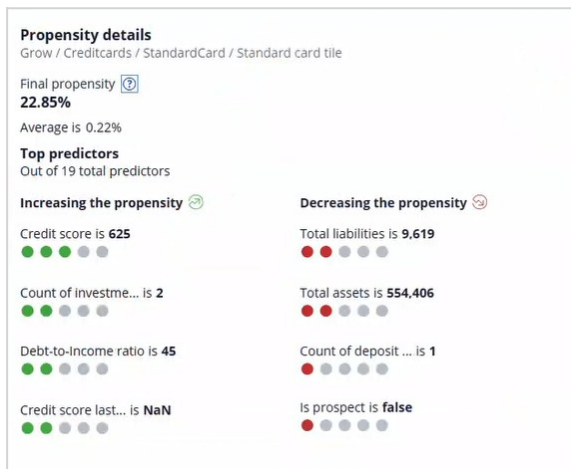
Consider the following interaction as an example: Troy logs in multiple times and sees the same Standard Card offer. On the first three visits, Troy ignores the offer. When he visits the website the fourth time, he clicks the offer to learn more.

In Customer Profile Viewer, you can examine Troy's interactions, the decision history, and the next best actions. First, load the decision history for the current use case to view the interactions recorded. In the table, the propensity changes after the system records each interaction. You can examine that the propensity decreases because Troy ignored the offer three times. Then, after he clicked **Learn more**, the propensity increased.

Decision time	Name	Treatment	Group	Issue	Direction	Channel	Journey	Journey stage	Final propensity	Context weight	Value	Total lever weight	Priority	Rank
2/22/23 9:57 ...	StandardC...	Standard card tile	Creditcards	Grow	Inbound	Web			0.3400	1.00	35.00	1.00	0.34	1
2/22/23 9:57 ...	StandardC...	Standard card tile	Creditcards	Grow	Inbound	Web			0.3158	1.00	35.00	1.00	0.32	1
2/22/23 9:56 ...	StandardC...	Standard card tile	Creditcards	Grow	Inbound	Web			0.3342	1.00	35.00	1.00	0.33	1
2/22/23 9:56 ...	StandardC...	Standard card tile	Creditcards	Grow	Inbound	Web			0.3442	1.00	35.00	1.00	0.34	1

The configuration of the AI model behind these offers treats impressions that do not result in a click as a negative outcome. As a result, the propensity, and therefore the priority of that offer, decreases. The propensity and priority of the not-clicked offer keep decreasing until the model records a click (positive outcome). However, if Troy clicks the offer, the propensity and priority increase. Next, view the next-best-action recommendations to learn more about the action for which Troy qualifies. For the current use case, the direction is **Inbound**, and the channel is the **Web**. TopOffers is the real-time container service that manages communication between the Customer Decision Hub and the website of the bank.

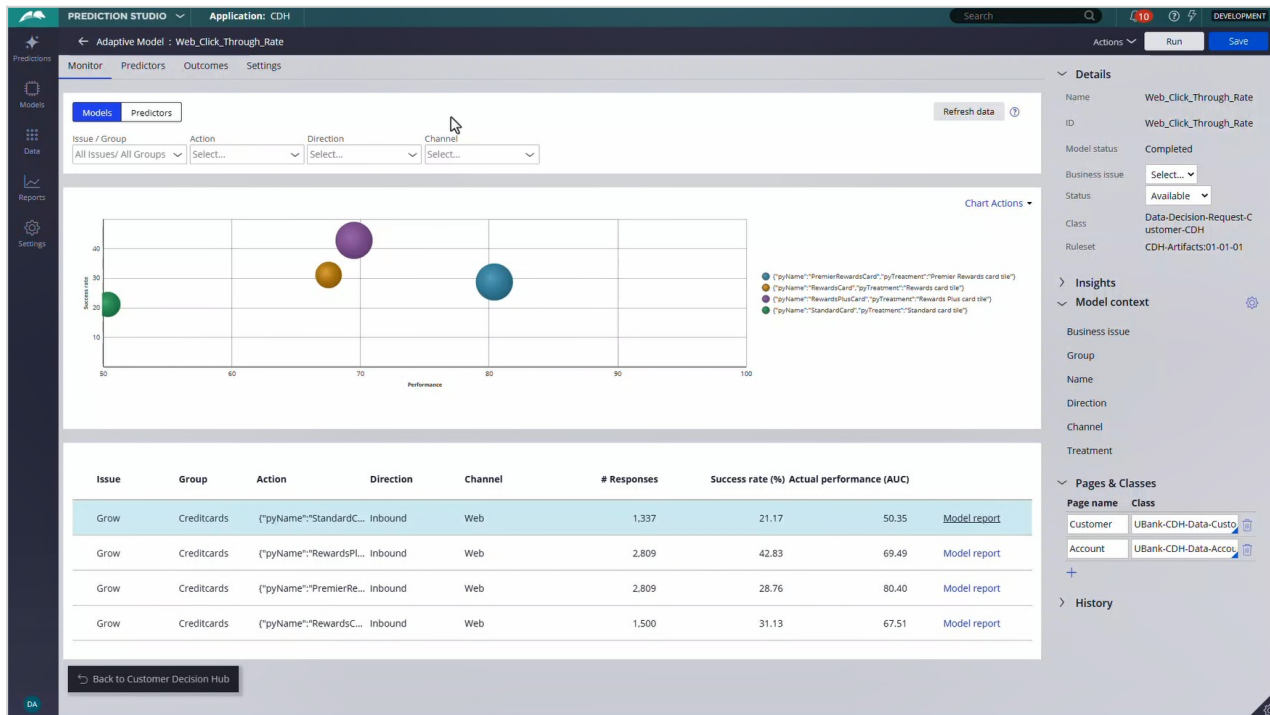
Based on the engagement policy rules, Troy is eligible for two credit card offers: the **Standard card** and the **Rewards card**. In Customer Profile Viewer, you can examine the arbitration factors using one of the following options: **Explain prioritization, Explain weighting, or Explain propensity**. In the **Explain propensity** view, you can check the influencing factors of a specific action for which Troy qualifies. The factors are the best-performing predictors that contribute positively to the propensity of the offer and the predictors that contribute negatively to the propensity of the offer.



Customer Decision Hub calculates the propensity for each treatment. To understand how this works, study the **Analytical model** behind a treatment. This pop-up window provides a summary of the AI behind the treatment. In Customer Decision Hub, the AI that determines the propensity for positive behavior towards an action or treatment is an adaptive model.

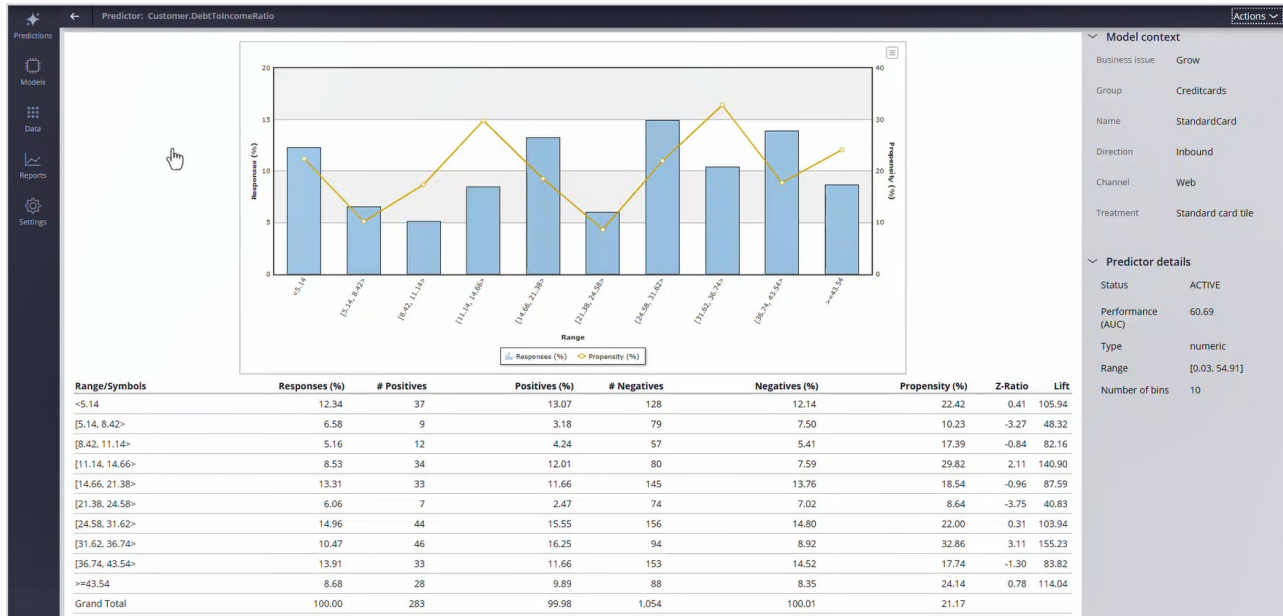


From here, you can navigate to the adaptive model in Prediction Studio, a workspace used to manage AI models. On the landing page of the Prediction Studio, every action is represented as a separate bubble on the bubble chart. Under the chart, separate **Model reports** for every action are available.



An adaptive model is a self-learning predictive model that uses machine learning to calculate propensity scores. It automatically determines the factors that help in predicting customer behavior. These predictors can include a customer's demographic details, product and service usage, past interactions with the bank, and even contextual information such as the current channel of interaction. In the **Predictor** report, you can examine the

performance of individual predictors. For example, you can see how the system automatically groups the values of a numeric predictor into bins and how the propensity to accept varies across the bins.



The behavior of one customer influences the propensity calculation for other customers with a similar profile.

- You have reached the end of this video. You have learned:
- How customer behavior influences the Propensity value.
- Where in CDH you can update the Arbitration settings.
- How customer behavior and next best actions are displayed in Customer Profile Viewer.
- What are the adaptive models, and how to manage them.

Optimizing your next-best-action strategy

Description

Learn how to check the health of Pega Customer Decision Hub™ and improve the effectiveness of your next best actions. Impact Analyzer, Scenario Planner, and Value Finder are optimization tools that provide insight into the performance of your next best actions, identify opportunities for improvement, and help you adjust settings to create higher customer engagement and value capture.

Learning objectives

- Use Impact Analyzer to monitor the performance of your next best actions.
- Use Scenario Planner to simulate “what-if” scenarios to forecast results.
- Use Value Finder to identify underserved customers and take action to improve their experience
- Explain the connection between Impact Analyzer and Scenario Planner.
- Explain the connection between Impact Analyzer and Value Finder.

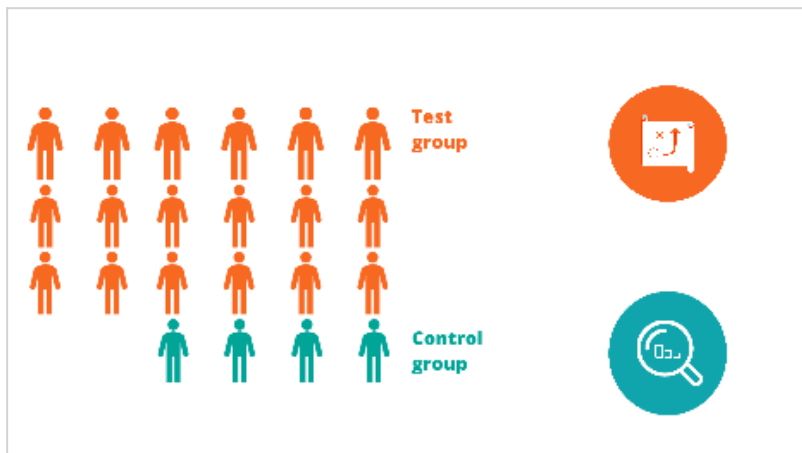
Monitoring the health of the system using Impact Analyzer

Impact Analyzer is a proactive monitoring tool that shows the overall health of your actions and highlights opportunities for improvement. The tool provides insight into the performance of your next best actions through the results of experiments that tell you what actions and configurations are working as expected, and where you can still improve.

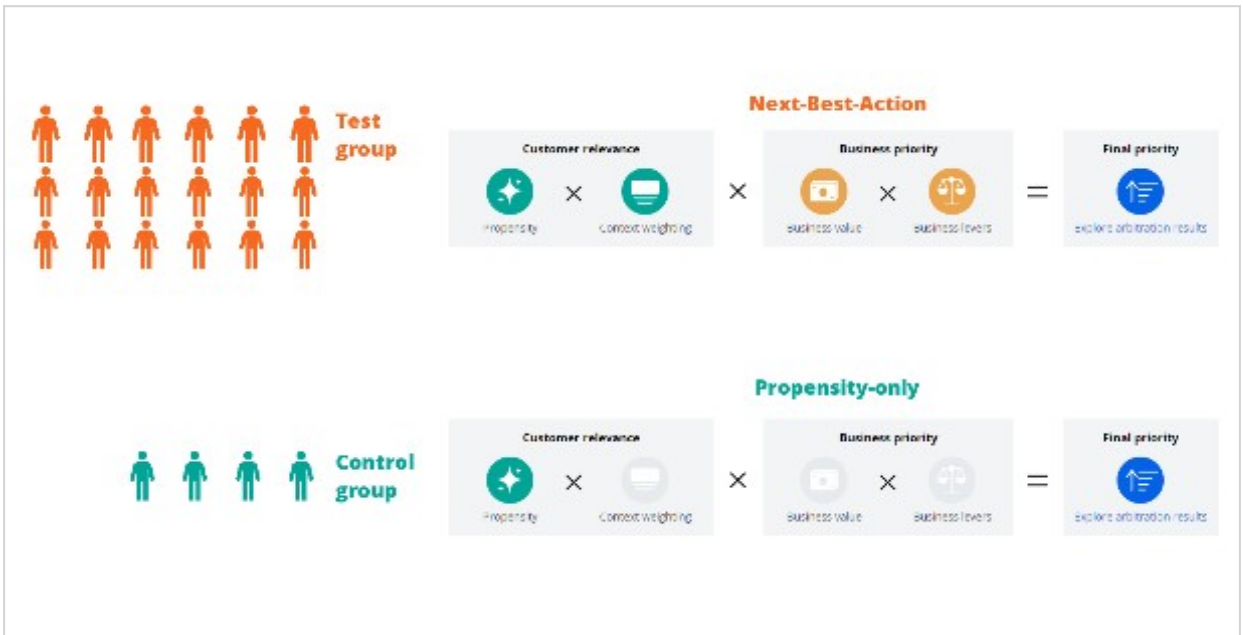
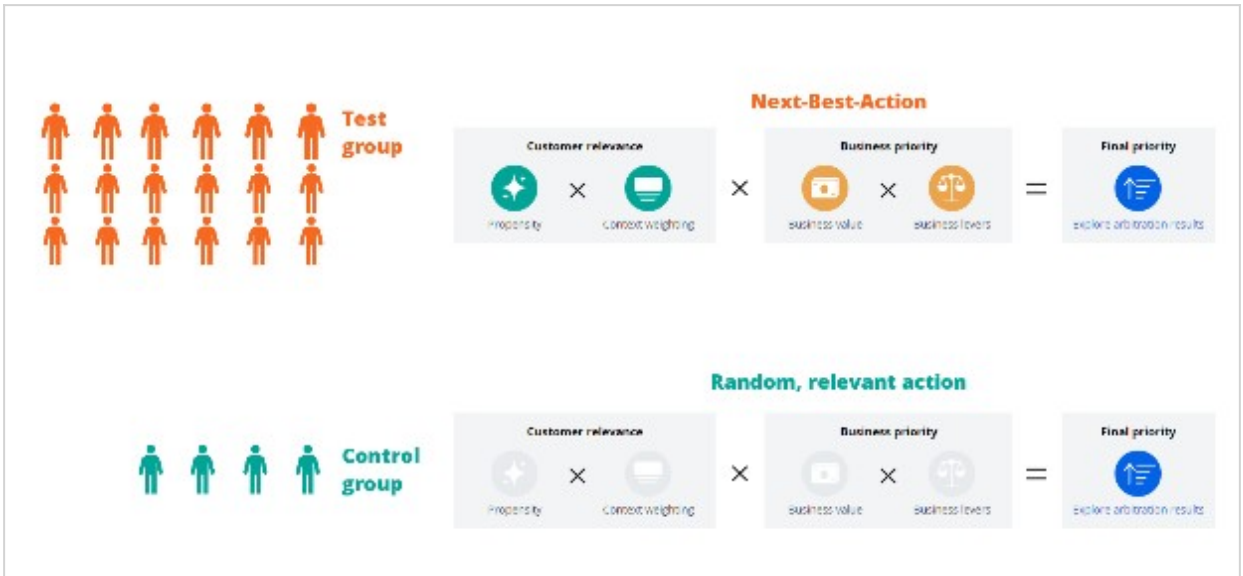
Transcript

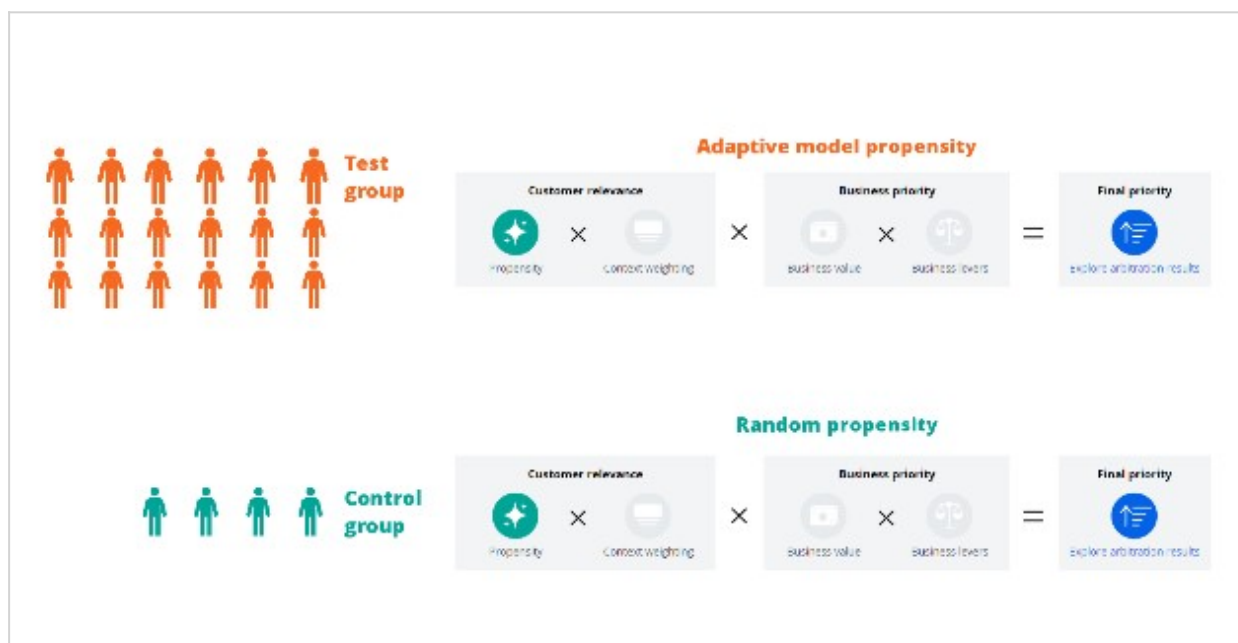
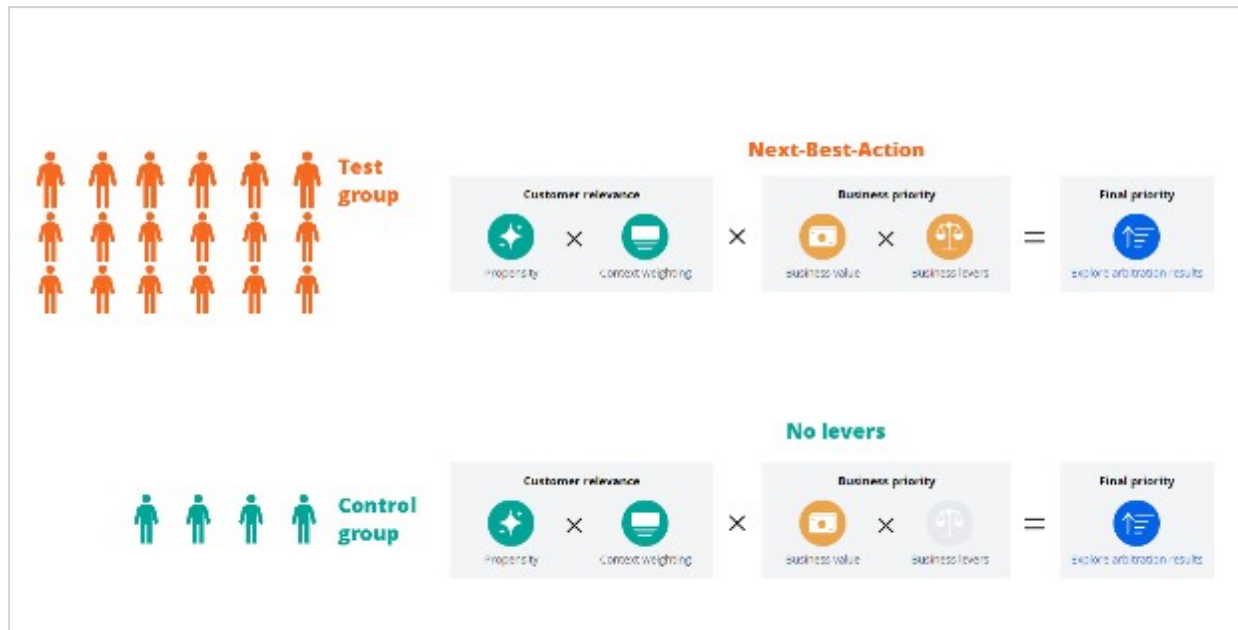
Impact Analyzer is a feature of Pega Customer Decision Hub™ that uses experimentation to track the effectiveness and overall health of next best actions. By utilizing this tool, you can gain insights into the effects of arbitration. Impact Analyzer conducts experiments with alternative prioritization and engagement policy filters on a small control group of customers. This enables you to assess whether or not the next best action is producing a positive effect. In addition to monitoring for basic lift, these control group experiments can also reveal opportunities to optimize and align the next best actions with your business goals.

An experiment is a test with two sets of participants, the test group and the control group, in which the test group receives the originally arbitrated action and the control group receives an alternate action based on the type of experiment.



You can use Impact Analyzer to carry out a variety of tests. For example, you can assess the effectiveness of the next best action against a random relevant action or actions, chosen based on propensity alone. Additionally, you can measure the performance of the next best action when arbitrating with no levers. You can also evaluate the performance of the adaptive model propensity against a random propensity.

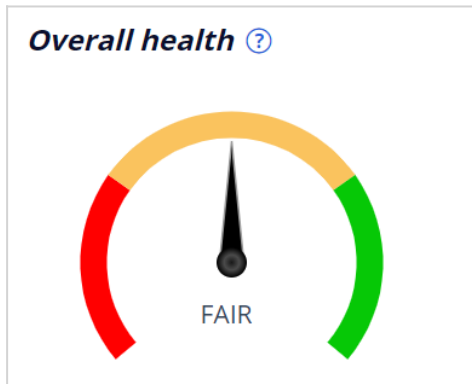




Impact analyzer continuously monitors the health of the next best actions in the experiment, and presents the results in the form of two key performance indicators (KPIs): value lift, and engagement lift. Value lift measures how much a business can make from suggesting certain next best actions to customers, compared to a random relevant action. Engagement lift measures how much more likely customers are to click or accept a suggested next best action, compared to the random relevant action.

When a next best action is delivering lift, the overall health indicator highlights in green. The middle position of the indicator shows that there is room for improvement to better align

actions to the business. The red highlight [the section on the left] means that something is wrong and needs your attention. The indicator is colored gray when the general health is not available due to insufficient data in one or more active tests.



Impact Analyzer is available under **Discovery** tab in the left menu of Customer Decision Hub.

One of the experiments that you can run is called *How is Next-Best-Action performing against a random relevant action?*

How is Next-Best-Action performing against a random relevant action? ? ACTIVE :

All issues

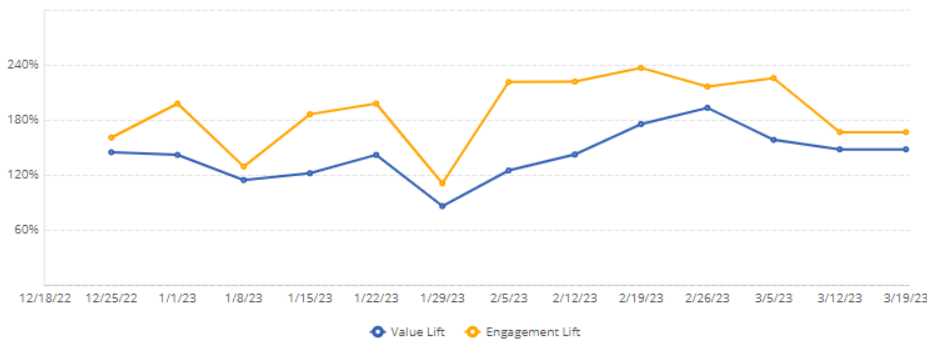
HIGH CONFIDENCE

VALUE LIFT	ENGAGEMENT LIFT
128%	171%
+/- 13%	+/- 4%
Lift details	

Well done. Next-Best-Action is delivering value and engagement lift.
 Nothing specific to do right now. Continue to monitor.
[Go to Next-Best-Action Designer](#)

View less

Weekly value and engagement lifts (*week starting*)

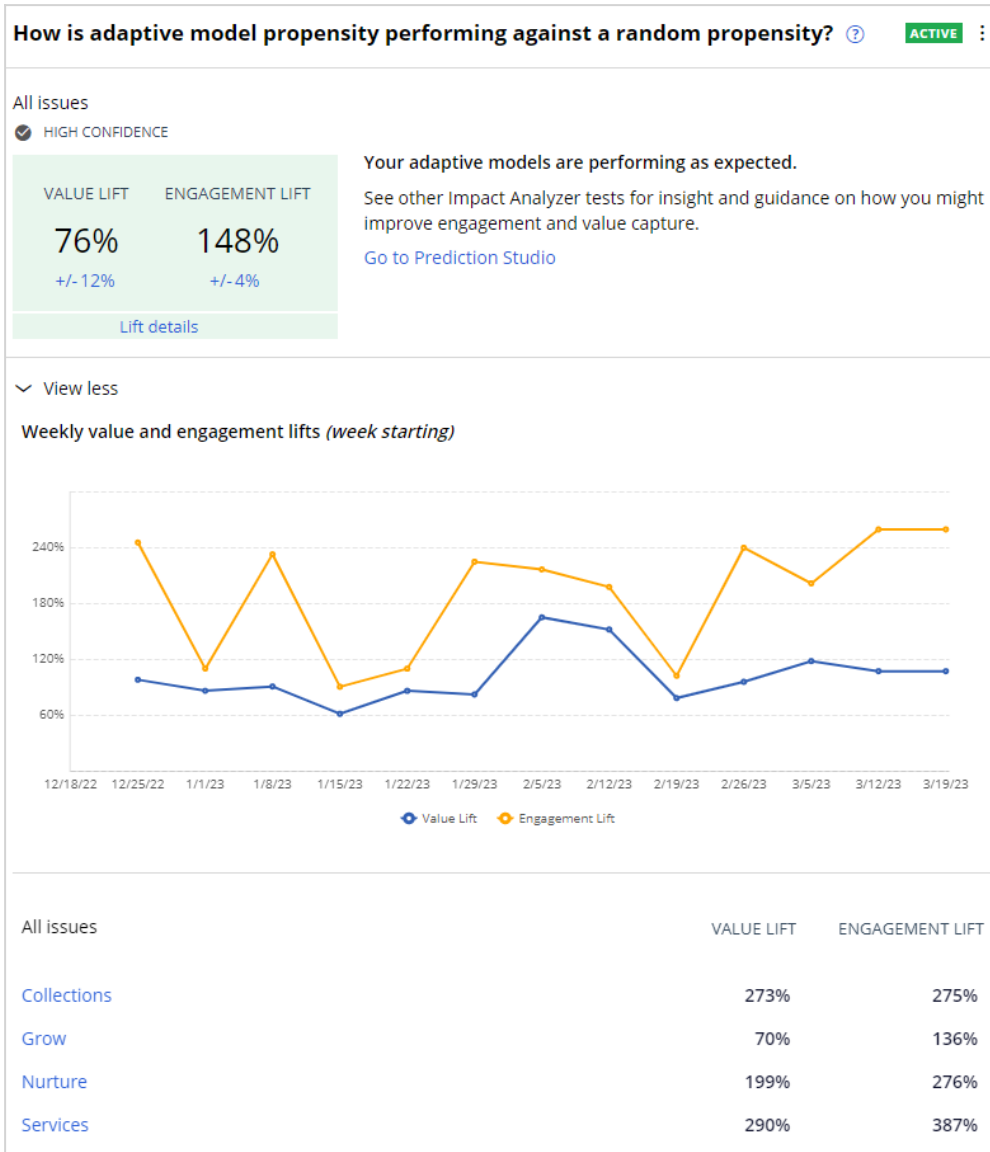


All issues	VALUE LIFT	ENGAGEMENT LIFT
Collections	251%	254%
Grow	121%	156%
Nurture	171%	215%
Services	335%	335%

This test serves as a baseline check to ensure that the next best action is beating the random relevant action. At the very least, we expect the value lift - and ideally, both the value and engagement lift - to be positive, at the highest level across all issues and groups aggregated. If both value and engagement lifts are negative, there is an issue. This could mean that there is a problem with the underlying adaptive models, or that your engagement policies are too targeted. Another reason might be that your levers are excessively biased, and the next best action lacks the aptitude to optimize engagement or to capture value.

The test widget presents a value lift of 128 percent and an engagement lift of 171 percent. Both values are positive, and the test is highlighted in green.

Another experiment is called *How is adaptive model propensity performing against a random propensity?*



This experiment compares only Pega AI-driven model propensity with no context weight, no business value, and no business levers, against only randomly generated propensity. The experiment is another baseline check to ensure that the adaptive models that a next best action uses are delivering lift over random propensity.

To monitor for basic lift, both presented experiments should be on at all times.

Impact Analyzer offers a range of benefits to its users. One of the most significant benefits is the ability to collect data in real-time, allowing for real-time monitoring and testing of KPIs. Impact Analyzer also enables users to look for opportunities that they may have otherwise missed.

The primary user of Impact Analyzer is the Next Best Actions operations team, who are responsible for ensuring that the system runs optimally. However, the tool brings value to various lines of business, sales, marketing, and Data Scientists.

You have reached the end of this video. You have learned:

- How Impact Analyzer continuously monitors the performance and health of the system.
- How Impact Analyzer conducts experiments by implementing alternative prioritization and engagement policy filters on a small control group of customers, to assess the effectiveness of next best actions.
- How the experiments reveal opportunities to optimize and align next best actions with business goals.

Investigating the effect of business levers in action arbitration

Impact Analyzer is a tool that uses experimentation to monitor the health and effectiveness of next best actions. Through various experiments, Impact Analyzer checks if the next best actions are delivering lift, and identifies opportunities to better optimize and align next-best-action optimization with your business goals. To further analyze such cases, you can use Scenario Planner, which allows you to easily simulate 'what-if' scenarios to accurately forecast results, optimize strategies to hit specific goals, and explore the potential trade-offs of each option. Both Impact Analyzer and Scenario Planner are optimization tools that help you adjust Pega Customer Decision Hub™ settings so that the actions bring higher customer engagement and value capture.

Transcript

Business levers are one of the arbitration factors that allow the business to assert some level of control over the prioritization of actions defined within the system. One of the experiments in Impact Analyzer is the business levers experiment, which highlights opportunities that are missed because of the use of business levers. Using Scenario Planner, you can simulate different business levers settings, and assess which one has the right balance between missed opportunities and alignment with your business goals.

Business Levers are used to manually nudge Customer Decision Hub toward next best actions based on external factors. For example, the recommended next best action might be to offer a credit card to a customer when they visit the home page. But to meet a business goal, the Mortgage Line of Business favors a mortgage offer, even if that offer is ranked a little lower on the list of possible actions.

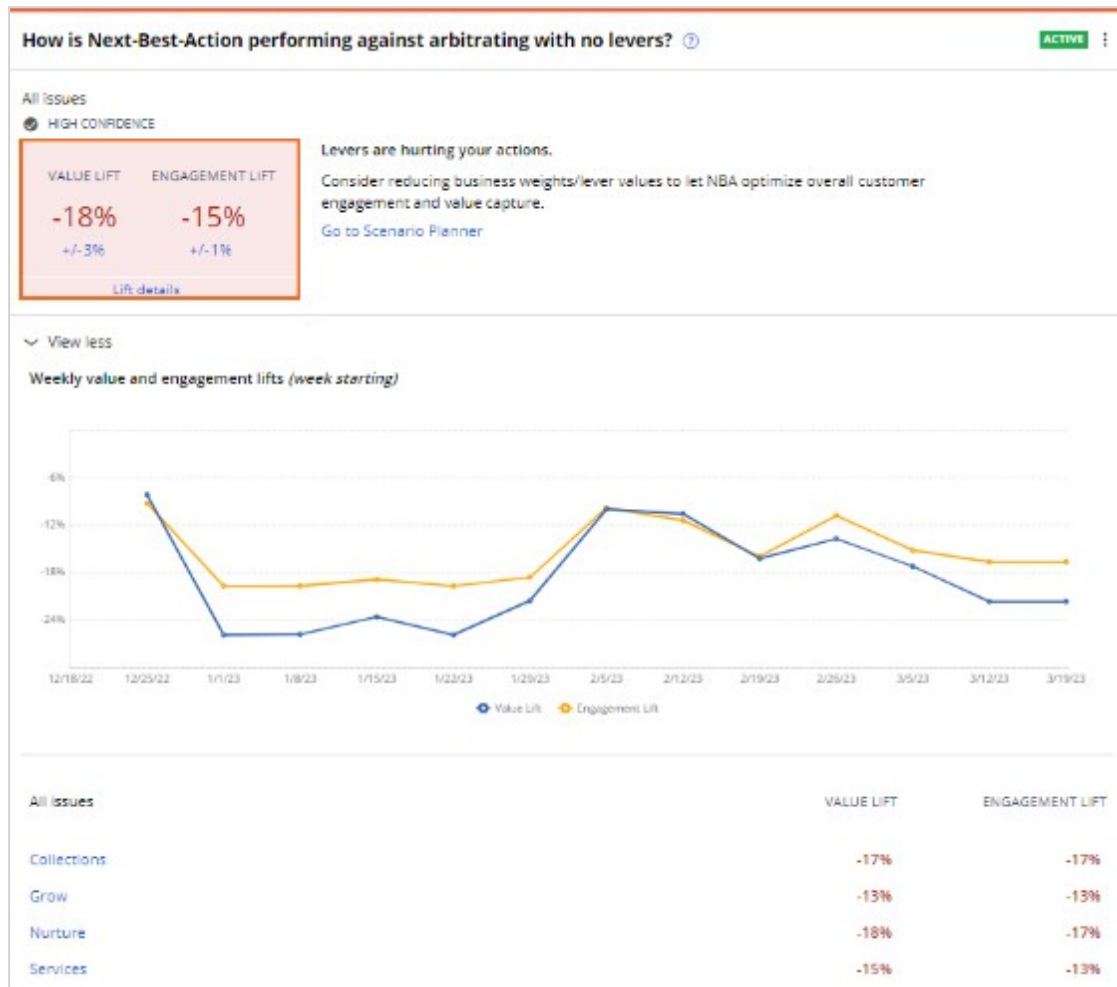
One of the experiments you can perform using Impact Analyzer is "How is next best action performing against arbitrating with no business levers?".

The business levers experiment highlights opportunities that are missed because of the use of business levers. This experiment evaluates the effectiveness of the full PCVL priority formula, which includes Propensity, Context, Business Value, and Business Levers, compared to arbitration with no business levers.



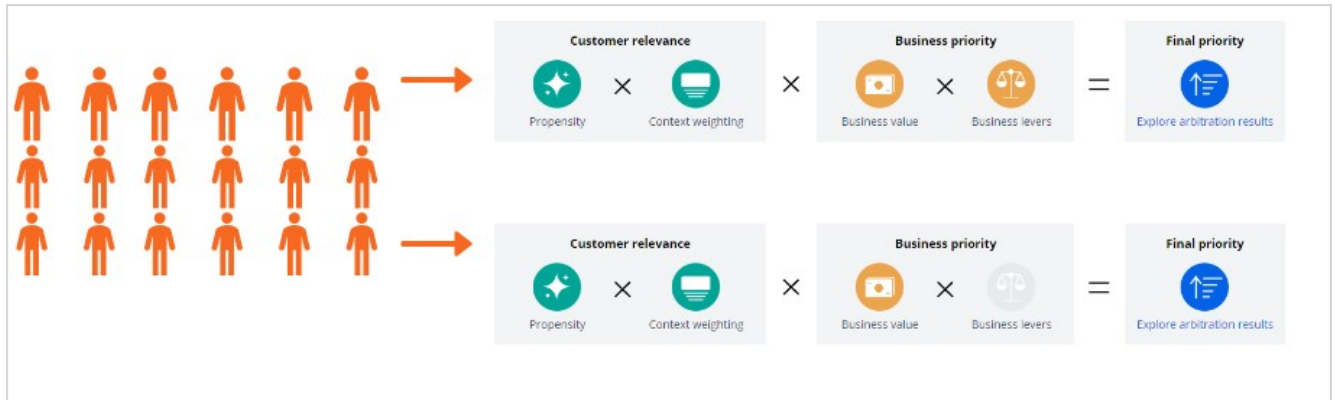
We expect that the next best actions should not outperform the control group with no levers in terms of value capture during this experiment. This is because the action that wins the impression using the full PCVL formula may not have the highest expected value capture. When next best actions arbitrate without any business levers, actions with the highest expected value capture will be presented. However, when business levers are applied, actions with lower expected value capture may be boosted and win the impression. Therefore, for this experiment, the value lift KPI is expected to be negative. This experiment can identify missed opportunities due to the use of business levers. When engagement or value lift is negative in this test, reducing lever values at the issue, group, and action levels can improve performance by giving next best actions the space to optimize. To ensure that the next best action is optimizing lift, the test widget provides recommendations to follow.

In the following example, the experiment indicates an -18 percent value lift and -15 percent engagement lift, with an overall red status, which is in line with expectations.

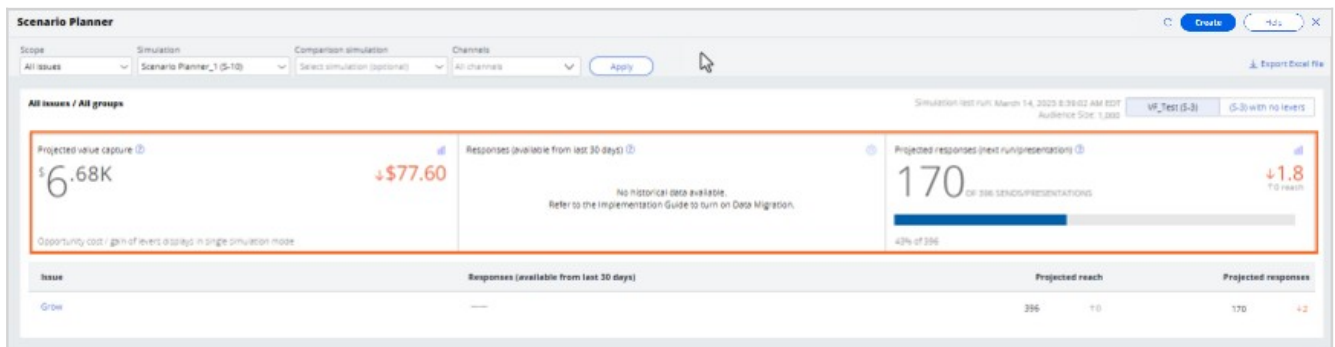


To examine the impact of changing lever values, use Pega Scenario Planner. Scenario Planner allows you to easily simulate 'what-if' scenarios to accurately forecast results, optimize strategies to hit specific goals, and explore the potential trade-offs of each option. Scenario Planner also estimates the value that the subsequent next-best-action run creates using its current configuration. For example, you can observe whether prioritizing a specific action by increasing its business weight will likely result in an opportunity gain or an opportunity cost.

Consider running a comparison simulation using a Scenario Planner to examine the difference between applying the full PCVL formula and the PCV formula without levers. Customer Decision Hub has various subsets of customer data available to run comparison simulations. The subsets of customers represent a sample of real customer data. For a Scenario Planner simulation, you use the same customer data sample for both simulations – one sample with full PCVL, and the comparison sample with the PCV formula.



The Scenario Planner landing page shows a number of useful key performance indicators (KPIs) including projected reach and projected acceptances, projected value capture, as well as historical acceptances, sends, and impressions.



In this example, the value capture is shown as \$6.68k, which represents the estimated value that will be generated by the upcoming next best action using the currently selected configurations. The delta is displayed in red and indicates that actions arbitrated without business levers result in a higher value. Based on the results obtained from the Scenario Planner, and taking into account your business goals, you can determine whether or not to update the business levers.

If you choose to update the business levers settings in an attempt to increase the value lift and engagement lift, you can use Scenario Planner to compare the current simulation with the adjusted simulation. This time, Scenario Planner will compare two simulations with full PCVL formulas, to determine whether the newly adjusted settings are aligned with the business objectives and bring value.

You have reached the end of this video. You have learned:

- How Impact Analyzer identifies the value and engagement loss caused by applied business levers.

- How Scenario Planner enables users to simulate different scenarios and explore potential trade-offs, to optimize their business strategies.
- How updating your business levers influences the Scenario Planner results.

Investigating the effect of applying engagement policies

Impact Analyzer is a tool that uses experimentation to monitor the efficiency and health of next best actions. Through various experiments, it checks whether the next best actions deliver lift and identifies opportunities to better optimize and align next-best-action optimization with your business goals. With the help of Impact Analyzer, you can identify the value and engagement loss caused by excessively restrictive engagement policies. To further analyze such cases, you can use Pega Value Finder to identify underserved customers and take action to improve their experience.

Transcript

Engagement policies are a set of business rules and practices used by the organization to determine which customers are eligible for which next best actions. There is an experiment in Impact Analyzer, which measures the value and engagement loss caused by possibly too strict engagement policies. Pega Customer Decision Hub™ provides an additional optimization tool, Value Finder, which enables users to further explore this issue. The Value Finder helps users identify underserved customers and optimize their next best actions to deliver higher value.

You can use the engagement policies to specify the conditions under which an action or group of actions a customer is eligible. There are three engagement policy conditions: **Eligibility, Applicability, and Suitability**.

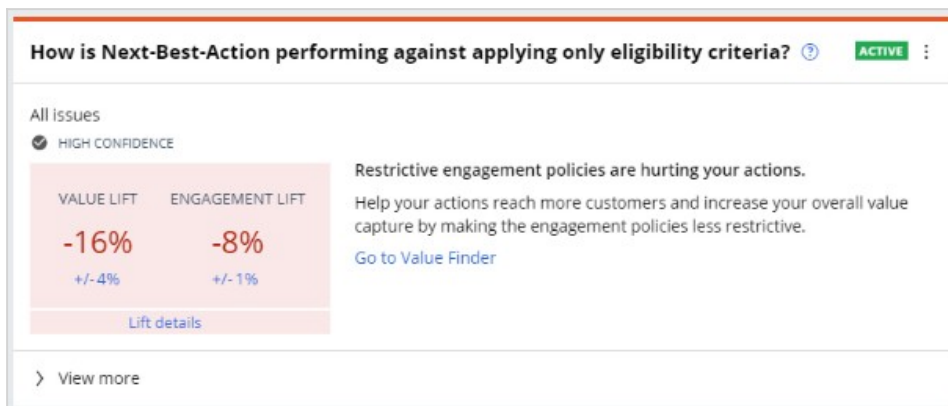
One of the experiments you can perform with Impact Analyzer is "How is Next-Best-Action performing against applying only eligibility criteria?". This experiment compares the next best action configured with full engagement policies and arbitration against the action configured with only Eligibility (without the Applicability and Suitability rules) followed by arbitration. The engagement policies need to be appropriately categorized for the "How is Next-Best-Action performing against applying only eligibility criteria?" experiment to work. Specifically, all the 'required' engagement policies need to be categorized as Eligibility criteria, while discretionary policies need to be categorized as Applicability and Suitability criteria.

You can use this experiment to identify potential opportunity gains by making your engagement policies less restrictive. Sometimes, when engagement policies are too restrictive, actions with a high potential for engagement and value capture are filtered out and do not have a chance to compete for the impression. By eliminating non-essential engagement policies, you can put these offers back in place and increase customer engagement and value capture. In this experiment, the test group receives the fully-

configured next best action, while the control group receives an action with only the eligibility criteria applied.



In this example, Impact Analyzer indicates a -16 percent of value lift and -8 percent of engagement lift, and the overall status of the experiment is negative. The widget displays the recommendation that you can make your engagement policies less restrictive to increase the overall value capture.

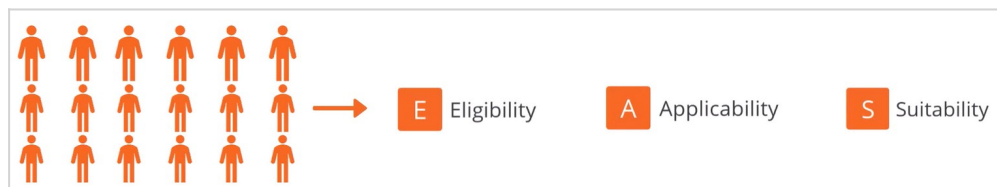


Pega Customer Decision Hub provides opportunities to engage empathetically by identifying "under-served" customers. By using the simulation tool Pega Value Finder, you can identify areas for improvement and monitor scenarios where customers receive no or only low-propensity actions. Value Finder also suggests actions you can take to increase the value capture, such as adjusting engagement policies or creating new actions and treatments.

Consider running a Value Finder simulation to indicate missed opportunities and underserved customers.



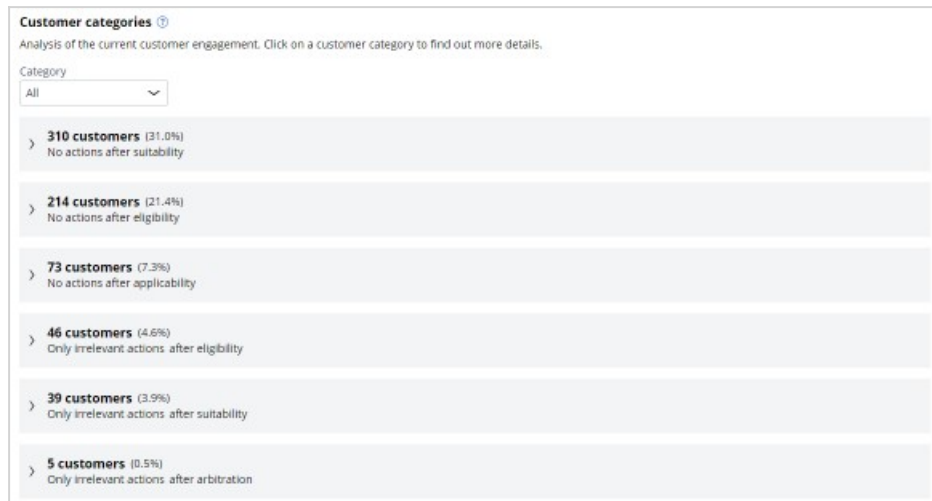
Customer Decision Hub has various subsets of customer data available to run simulations, and the subsets of customers represent a sample of real customer data. For a Value Finder simulation, you use a subset of the customer data and apply all three engagement policy conditions.



Once the simulation run is complete, the Value Finder landing page displays a pie chart with numbers of customers in different categories:

- **Receiving no actions** represents the number of customers who received no actions.
- **Receiving only irrelevant actions** represents the number of customers who received low-propensity actions.
- **Receiving at least 1 relevant action** represents the customers who received high-propensity actions.
- **Receiving only new actions** represents the customers who received only new actions.

Value Finder also displays customer categories tabs, with the number of customers filtered out because of the given engagement policy. In this case, the first customer category shows the number of customers who receive irrelevant actions because of eligibility conditions. In this case, it is 370 customers. Analysis of the first customer category suggests that the eligibility conditions might be too strict and exclude many customers.



When you click a particular category, you can run a simulation to see which conditions influence the actions negatively. Considering the business objectives and current conditions that you want to maintain, you can decide whether to loosen your eligibility rules or not.

If you decide to update the engagement policies to increase lift and engagement values, you can perform an additional Value Finder simulation to determine the number of underserved customers that remain after you apply the new settings.

You have reached the end of this video. You have learned:

- How Impact Analyzer identifies the value and engagement loss caused by engagement policies.
- How Value Finder identifies underserved customers and provides valuable insights.
- How Impact Analyzer and Value Finder help businesses improve the optimization of next best actions and align them more effectively with their overall business goals.

Next-Best-Action on digital

Description

Learn about a typical cross-selling use case on the web channel and its four phases of implementation to improve 1-to-1 customer engagement, drive sales, and deliver Next-Best-Actions in real time.

Learning objectives

- Describe a typical cross-selling use case on the web channel
- List the four phases of use case implementation

Business use case: Cross-sell on the web

Introduction

Pega Customer Decision Hub's Next-Best-Action Designer lets you configure how you want the always-on brain to select the best offer for a customer. The best offer is the result of a series of decisions that are executed in a hierarchical fashion by the brain. Cross-selling on the web channel will help you improve 1-to-1 customer engagement, drive sales, and deliver Next-Best-Actions in real-time.

Transcript

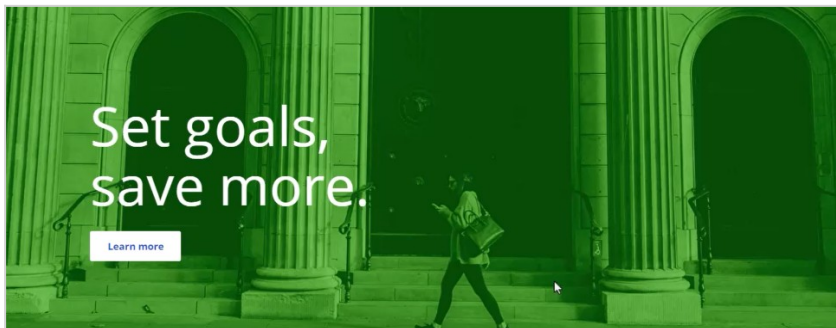
This video describes a typical cross-selling use case on the web channel.

U+ is a retail bank. The bank would like to leverage its website as a marketing channel to improve 1-to-1 customer engagement, drive sales, and deliver Next-Best-Actions in real-time.

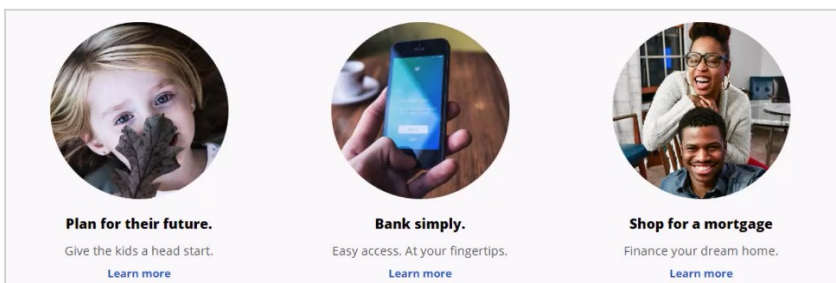
The bank has decided to use the Pega Customer Decision Hub™ to recommend more relevant banner ads to its customers when they visit their personal portal.

Banner ads are shown on various pages throughout the website.

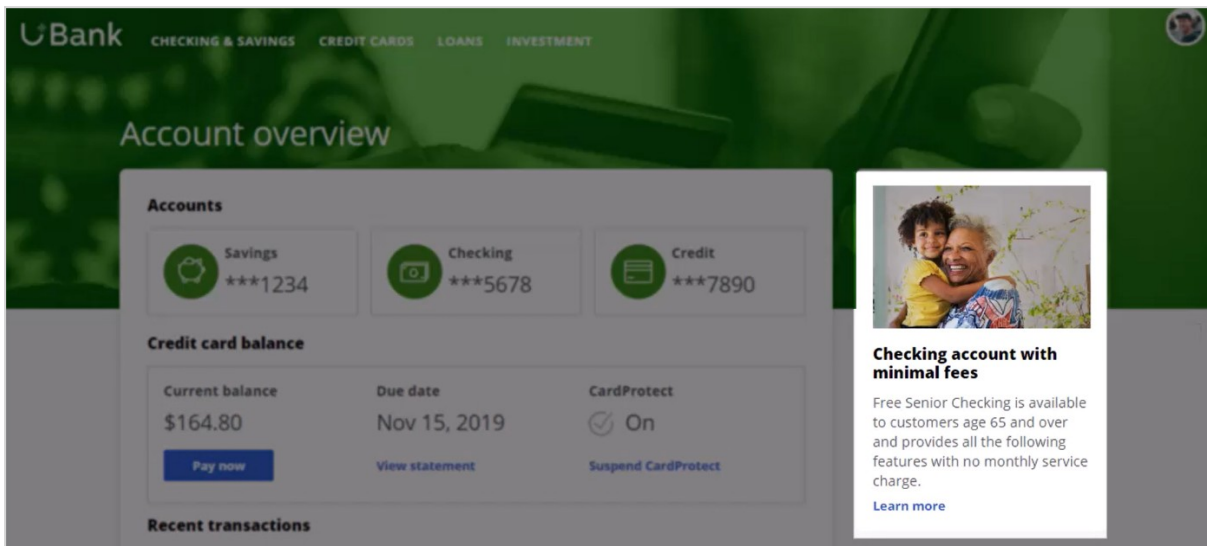
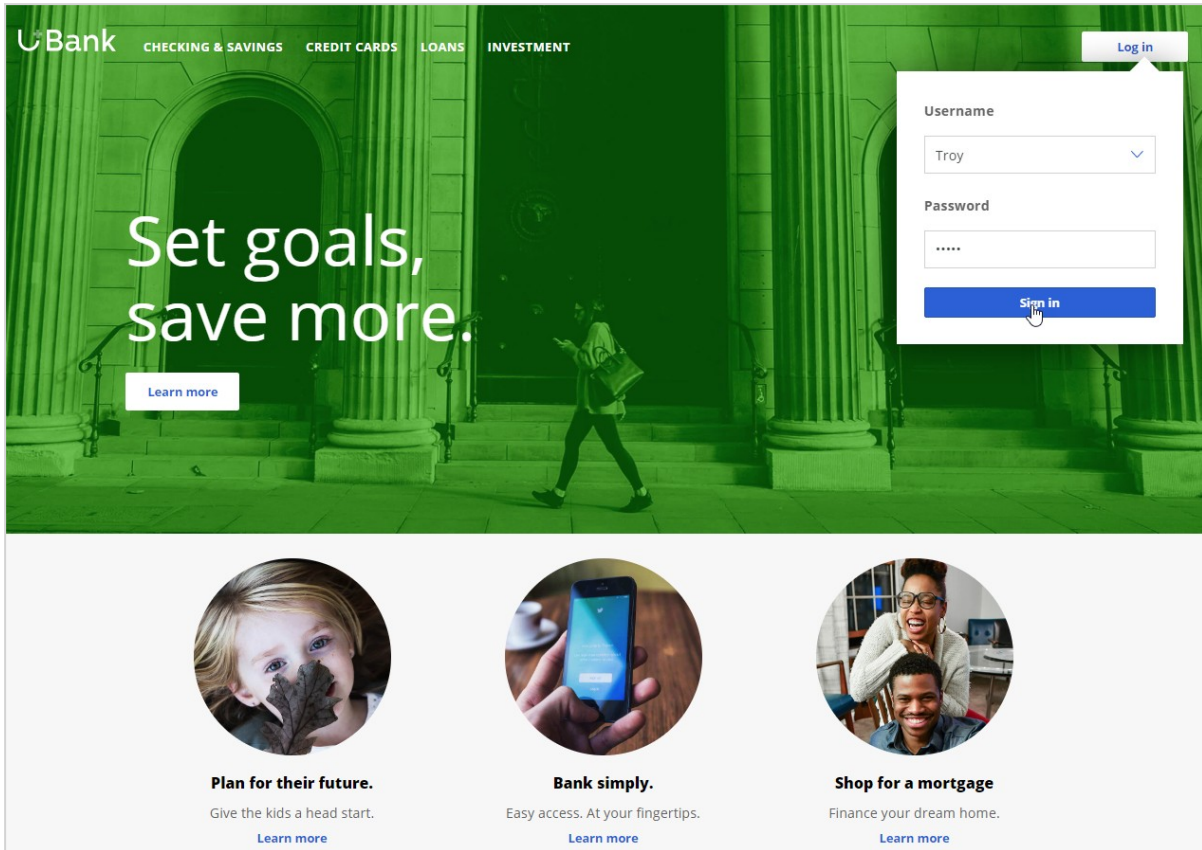
For example, on the home page, U+ can display a Hero banner at the top of the page, which is typically a larger image with bigger typeface.



Below that, there is space to display several Tile banners, which are typically smaller.



When customers log in to their personal portal, they also see a Tile banner on the 'Account overview' page.



The main intent of U+ at this stage is to increase their web engagement. This can be measured by click-through rate. A Click-through is recorded when the customer clicks on the 'Learn more' link.

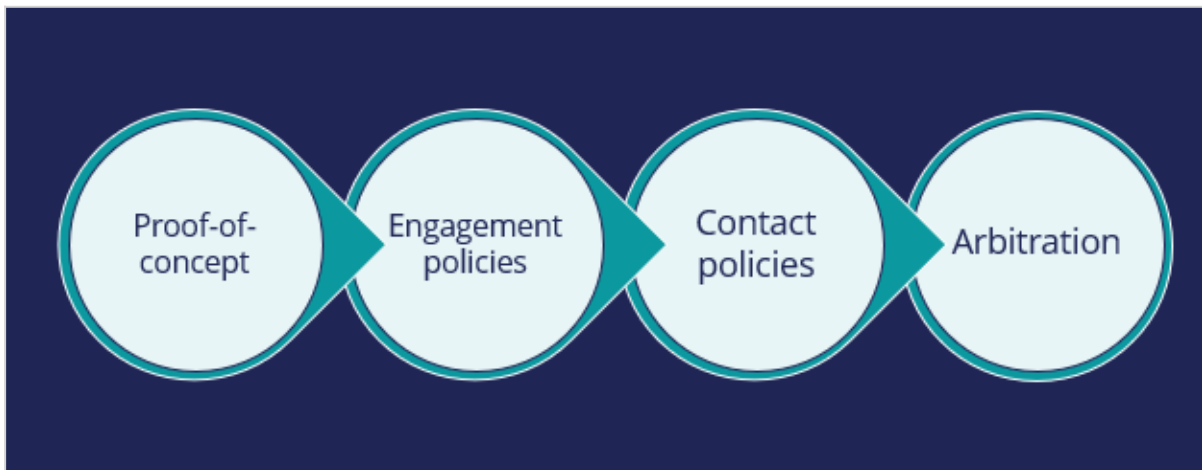
The bank would like to use these banners on the 'Account overview' page, to display offers that are more relevant and likely to receive a positive response.

The offers will be selected by a combination of artificial intelligence (AI) and other business rules. The AI and business rules are defined in the Pega Customer Decision Hub.

The Pega Customer Decision Hub is the always-on customer brain that acts as a single, centralized decision authority. The always-on customer brain selects the right offer to be displayed to each customer who visits the bank's website.

Next-Best-Action Designer lets you configure how you want the always-on brain to select the best offer for a customer. The best offer is the result of a series of decisions that are executed in a hierarchical fashion by the brain.

The bank plans to implement the requirement in multiple phases.



The first phase is a proof-of-concept phase. In this phase, the goal is to display a credit card offer on the U+ website. This requires getting the basic environment up and running, setting up the business structure, defining an Action and a Treatment, and enabling channels and triggers for Next-Best-Action.

As a result of this phase, a credit card offer will be displayed on the 'Account overview' page to all customers who visit the U+ web site. For example, if customer Troy logs in to his account, the 'Cash back' offer is displayed. If another user logs in, they will see the same offer. However, in practice, more offers should be displayed. Also, not all offers may be available to a customer for various reasons.

The next phase is to add customer engagement policies. Engagement policies are the set of conditions such as eligibility, applicability, and suitability that qualify an offer, or a group of offers for a customer. As a result of engagement rules, customers will see only those offers that the organization believes they should be exposed to. For example, Troy logs in, he sees

the Rewards Card, but for Barbara this is not applicable, so it will never appear; instead, she sees the Rewards Plus Card.

Too many contact attempts over a short period of time can have a negative impact on a customer's attitude toward further offers by your company. Therefore, in the next phase, U+ implements some contact policies using suppression rules, which allow an offer to be put on hold after a specific number of outcomes. For example, if Troy ignores an ad a few times, then the ad will no longer be shown to him over a period of time. Instead, his 'Account overview' page will show a different ad.

Basically, from a set of all available offers, the choice is narrowed down by engagement policies. Then the selection is further narrowed down by suppression rules.

After the engagement policies and suppression rules have "whittled down" the total possible offers to a few, Arbitration is used to choose the top offer based on what is relevant for the customer right NOW.

Arbitration is the last phase of cross-sell in the U+ web use case.

Arbitration aims at balancing customer relevance with business priorities. Specifically, Propensity, Context Weighting, Business Value, and Business Levers are given numerical values. A simple formula is then used to arrive at a prioritization value, which is used to select the top offer. For example, Troy qualifies for three credit card offers. When he logs in, he sees the top offer for him, the Standard Card. This offer is the Top 1 because the priority value is the highest among all other offers.

This video has concluded. It showed you a typical cross-selling use case on the web channel and its four phases of implementation.

Next-Best-Action Designer

Introduction

Next-Best-Action Designer guides you through the creation of a Next-Best-Action strategy for your business. Its intuitive interface, proven best practices, and sophisticated underlying decisioning technology enable you to automatically deliver personalized customer experiences across inbound, outbound, and paid channels. Next-Best-Action Designer is organized according to the high-level sequence of steps needed to configure the next-best-action strategy for your organization.

Transcript

Next-Best-Action Designer guides you through the creation of a next-best-action strategy for your business. Its intuitive interface, proven best practices, and sophisticated underlying decisioning technology enable you to automatically deliver personalized customer experiences across inbound, outbound, and paid channels.

Next-Best-Action Designer user interface allows you to easily define, manage, and monitor next best actions.

The tabs across the top of the user interface represent the steps to complete to define next best actions.

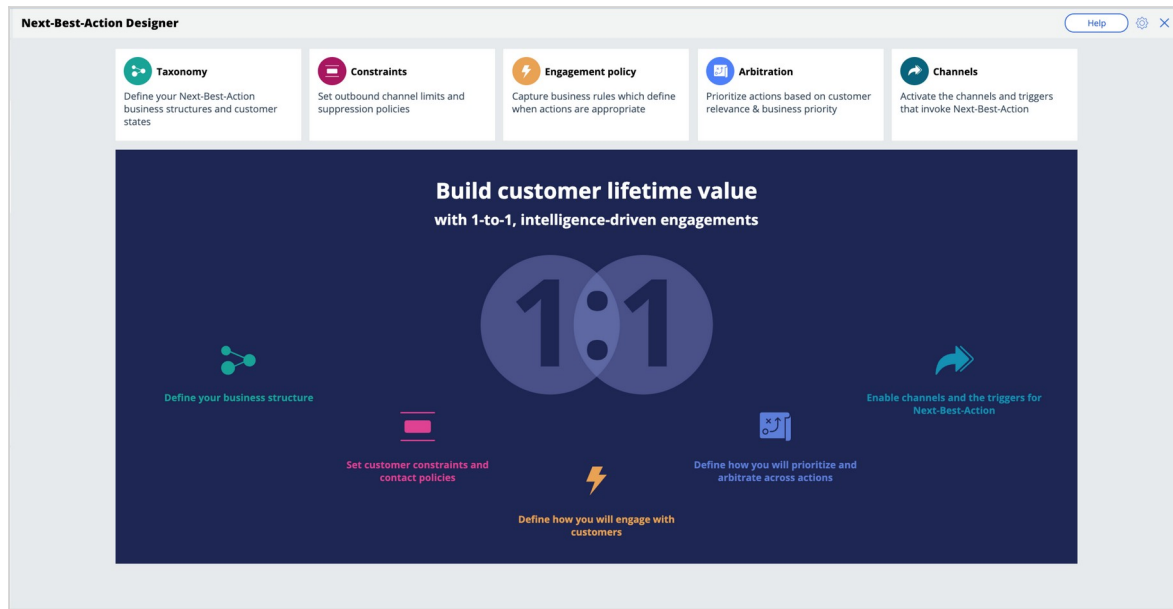
Use the **Taxonomy** component to define the business structure for your organization.

Use the **Constraints** component to implement channel limits and constraints.

Use the **Engagement policy** component to define the rules that control which actions are offered to which customers.

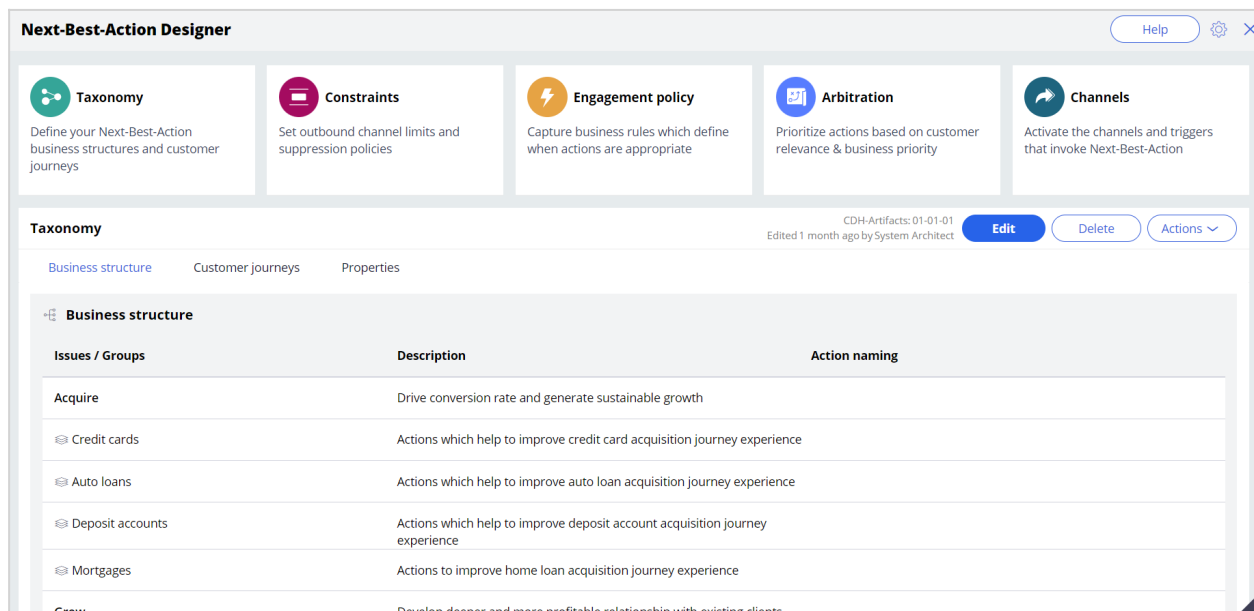
Use the **Arbitration** component to configure the prioritization of actions.

Use the **Channels** component to configure when and where the next best actions initiate.



The system uses these definitions to create an underlying Next-Best-Action Strategy framework. This framework leverages best practices to generate next-best-actions decision strategies at the enterprise level. These decision strategies are a combination of the business rules and AI models that form the core of the Pega Customer Decision Hub, which determines the personalized set of next best actions for each customer.

Use the **Taxonomy** tab to define the hierarchy of business issues and groups to which an action belongs and create and manage customer journeys.



A business issue is the purpose behind the actions you offer to customers. For example, actions with the purpose of acquiring new customers belong to the business Issue of **Acquire**. Actions with the purpose of retaining existing customers are grouped under the business issue of Retain.

Business groups organize customer actions into categories. For example, as part of the Acquire business Issue, you can create groups for products like credit cards, mortgages, or auto loans, with the intention of offering these to potential customers.

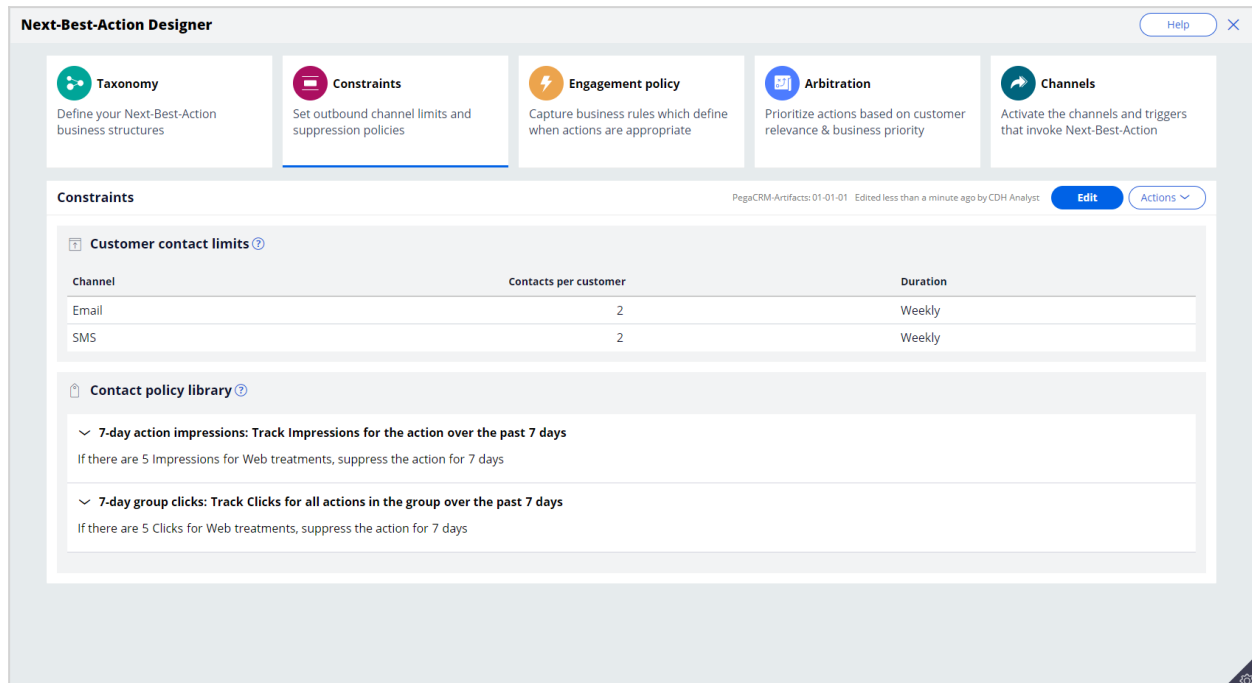
Customer journeys are the sum of the phases that customers go through when interacting with an organization. You can create customer journeys in Next-Best-Action Designer, and then map customer actions to different journey stages. Each stage represents an experience that a customer goes through while interacting with an organization.

On the **Properties** tab, you can define, manage, and view all the strategy result properties and action properties that are available in the application, including both default system properties and custom properties that you define for the outbound treatments or real-time containers.

The screenshot displays the Next-Best-Action Designer interface. At the top, there are five main sections: Taxonomy, Constraints, Engagement policy, Arbitration, and Channels. The Taxonomy section is selected, and the Properties tab is active. The Properties table lists several properties with their names, descriptions, types, usages, and whether they are included in strategy results.

Name	Description	Type	Usage	Include in strategy results	Expressions
PrintingCost	Printing Cost	Decimal			
LabelPriority	LabelPriority	Decimal	Dynamic		
AverageCost	AverageCost	Decimal	Static	Inbound,Outbound	
CreditScore	CreditScore	Integer	Dynamic		

Use **Constraints** to specify outbound contact limits and limit overexposure to a specific action or group of actions.



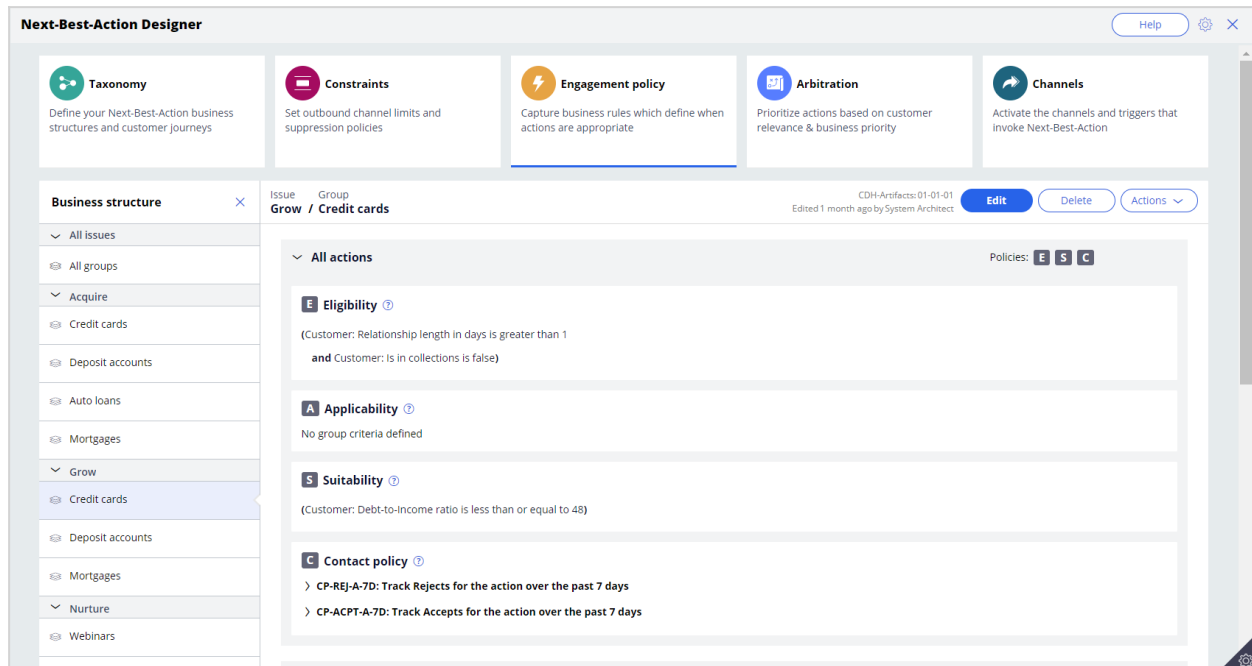
Outbound channel limits allow you to limit the number of outbound communications that a customer can receive over a given period of time on a specific channel. For example, you can decide that you do not want your customers to receive more than two emails per week or two SMS messages per week.

With action limits (outbound action limits and inbound action limits), you limit the number of actions that the customer sees during an interaction on an inbound or outbound channel.

On the **Constraints** tab of Next-Best-Action Designer, you can define more extensive suppression rules by creating Contact Policy rules in the library. Contact Policy rules are reusable across all Business Issues and Groups.

In the Contact Policy library, you define suppression rules that automatically put an action on hold after a specific number of outcomes are recorded for some or all channels. For example, an action can be suppressed for a customer for seven days after the customer has seen an ad for that action five times. Suppressing or pausing an action prevents over-exposure by limiting the number of times a customer is exposed to the same action.

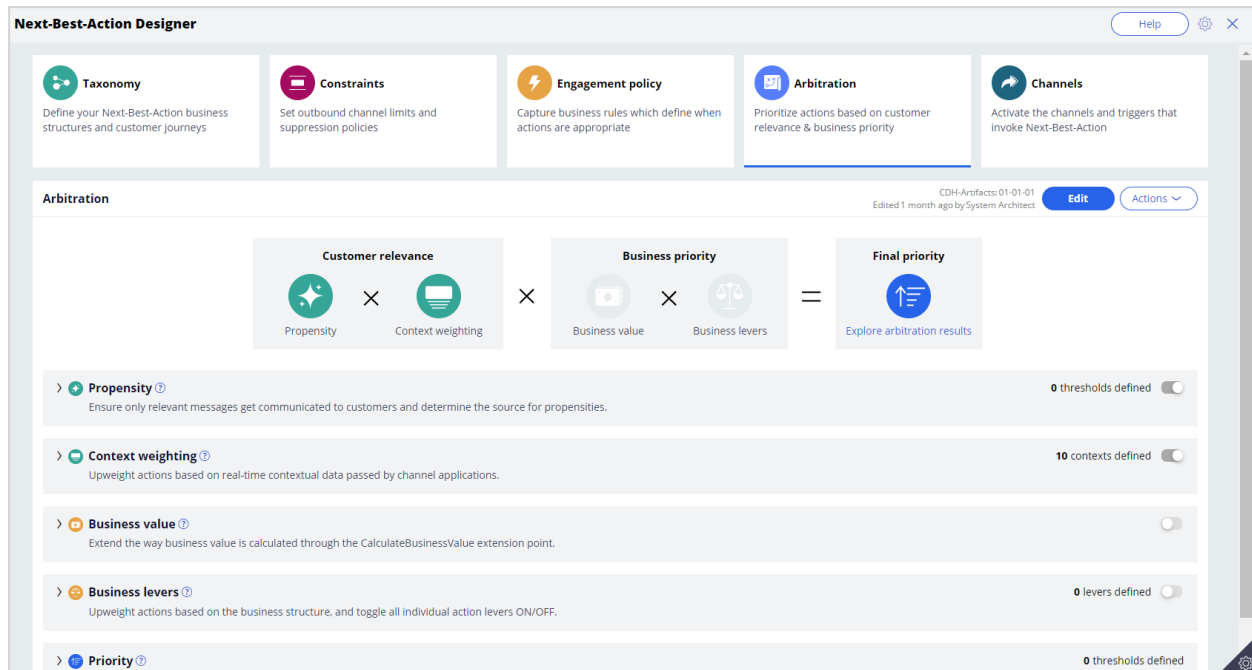
Use **Engagement policies** to define when specific actions or groups of actions are appropriate for customers.



There are four types of engagement policies:

- **Eligibility** determines whether or not a customer qualifies for an action or group of actions. For example, an action may only be available for customers over a specific age or living in a specific geographic location.
- **Applicability** determines if an action or group of actions is relevant for a customer at a particular point in time. For example, a discount on a specific credit card may not be relevant for a customer who already owns a card.
- **Suitability** determines if an action or group of actions is appropriate for a customer for ethical or empathetic reasons. For example, a new loan offer may not be appropriate for a customer whose credit score is low, even though it might be profitable for the bank.
- **Contact Policies** determine when an action or group of actions should be suppressed and for how long. For example, you can suppress an action after a specific number of promotional messages have been sent to the customer. To activate *Contact Policy* rules created in the library on the **Constraints** tab, add them to the **Engagement Policy** tab.

Arbitration determines how Customer Decision Hub prioritizes the list of eligible and appropriate actions that come out of each group.



The factors weighed in arbitration are **Propensity**, **Context weighting**, **Business value**, and **Business levers**, each represented by numerical values.

- **Propensity** is the likelihood of a customer responding positively to an action. The propensity is calculated by AI. For example, clicking an offer banner or accepting an offer in the contact center are considered positive behaviors.

Real-time contextual data is an important part of making highly relevant recommendations. **Context weighting** allows you to assign a weighting to a specific context value for all actions within an Issue or Group. For example, if a customer contacts the bank to change their address, the weight of the Service context increases, and the highest priority action is to ensure that the relevant service is delivered to the customer.

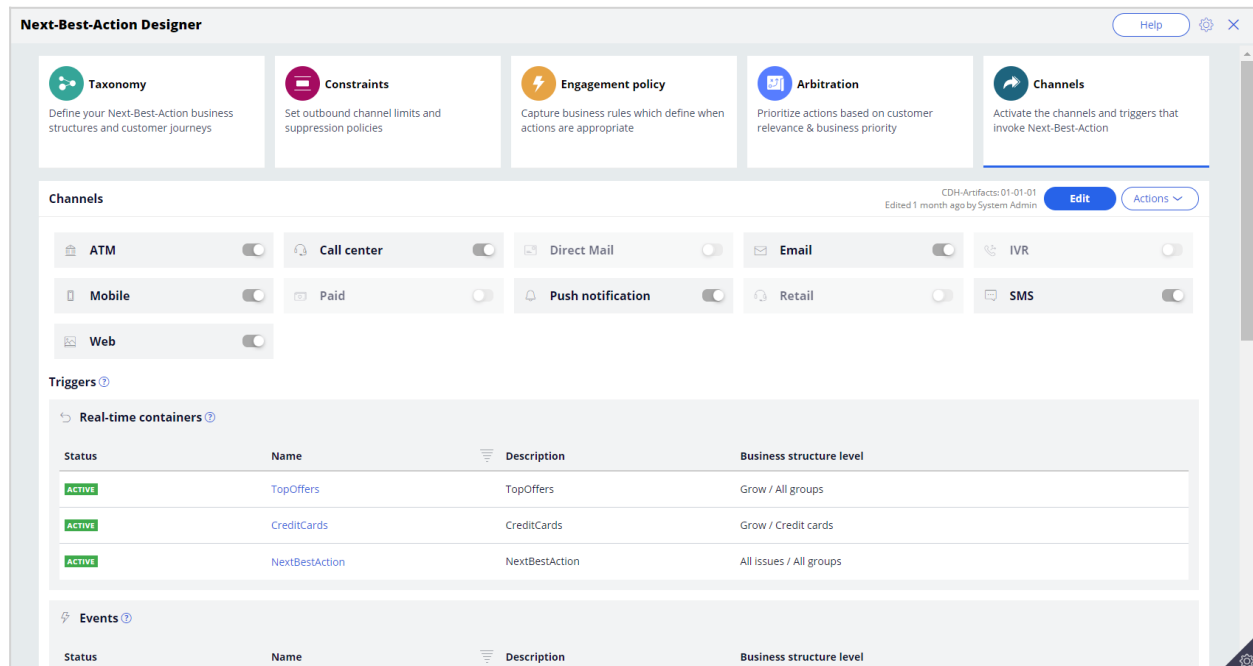
- **Business value** enables you to assign a financial value to an action and prioritize high-value actions over low-value ones. For example, promoting an unlimited data plan might be more profitable for the company than a limited data plan. Action values are typically normalized across Issues and Groups.
- **Business levers** enable you to accommodate ad hoc business priorities by specifying a weight for an action or group of actions and/or its associated business issue.

A simple formula determines a prioritization value, which is used to select the top actions.

The final priority indicates the top action selected from the list of eligible and appropriate actions for the customer. You can explore the arbitration matrix in Customer Profile Viewer to understand what type of actions are displayed to the customer.

You can use Customer Profile Viewer to examine specific profile and behavioral data, including previous next-best-action decisions and interaction history.

Next-Best-Action Designer enables the delivery of next best actions through inbound, outbound, and paid channels.



You can turn these channels on or off. If you turn off a channel, the next best actions are not delivered to that channel.

An external real-time channel is any channel that displays the actions that Customer Decision Hub selects for a customer. These channels can include a website, a call-center, or a mobile application. A real-time container is a placeholder for content in an external real-time channel.

A trigger is a mechanism whereby an external channel invokes the execution of a Next-Best-Action decisioning process for specific Issues and Groups. The result returns to the invoking channel. For example, when a real-time container called *TopOffers* is configured, the website invokes this real-time container before loading the account landing page.

As you have seen in this video, Next-Best-Action Designer is organized according to the high-level sequence of steps needed to configure the Next-Best-Action strategy for your organization. These steps involve:

- Defining the business structure for your organization.
- Implementing the channel limits and constraints.
- Defining the contact policies that control for which actions a customer qualifies.
- Configuring the prioritization of actions.
- Configuring when and where to initiate next best actions.

You have reached the end of this video.

Defining and managing customer actions

Description

In Pega Customer Decision Hub™, learn how to use Next-Best-Action Designer to build the business issue/group hierarchies for your organization. Create an action under a specific business issue/group hierarchy and learn to rename an action to better reflect its business context.

Learning objectives

- Explain an actions hierarchy
- Configure a business structure with issues and groups
- Create actions
- Rename actions

Action hierarchy

Customer action introduction

In Pega Customer Decision Hub™, next-best-action customer recommendations can take many forms, such as a banner advertisement, a retention offer, or a service message. Learn how to recognize different types of actions in your organization and organize similar actions into action groups.

Customer actions in your organization

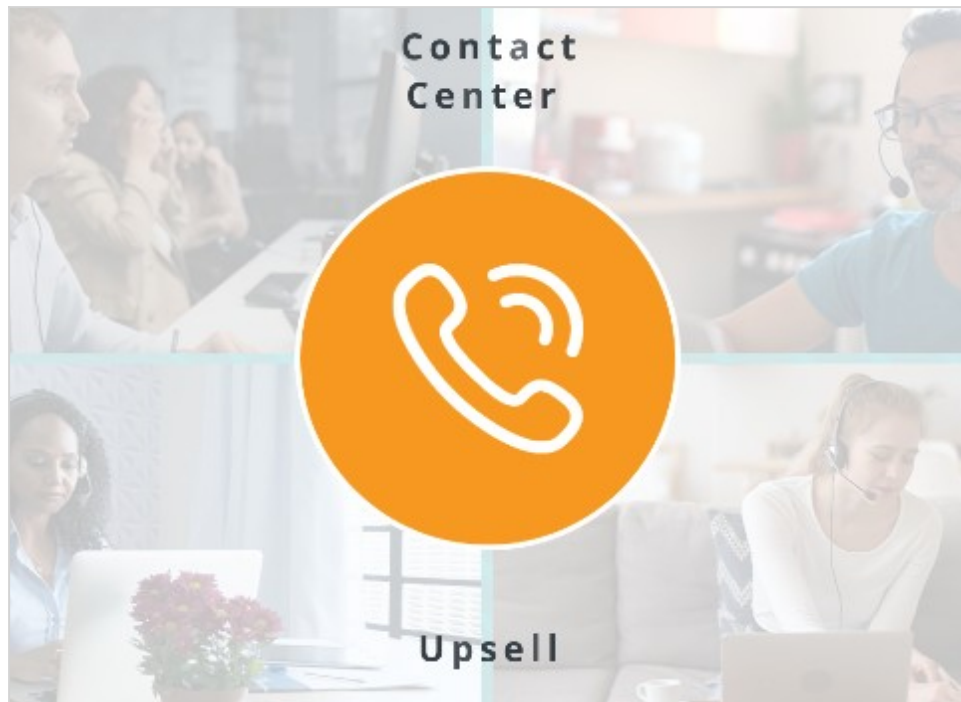
In Pega Customer Decision Hub, customer actions are the specific actions that the system or application can take based on customer interactions and decisions. These actions can include sending personalized messages, offers, recommendations, or any other form of engagement with the customer. The actions are determined by the decisioning capabilities of Customer Decision Hub, which utilize analytics, business rules, and customer data to make intelligent decisions and drive personalized customer experiences.

Consider the following scenarios in various channels: contact center, mobile, and web.

Scenario 1: Contact center: Upsell

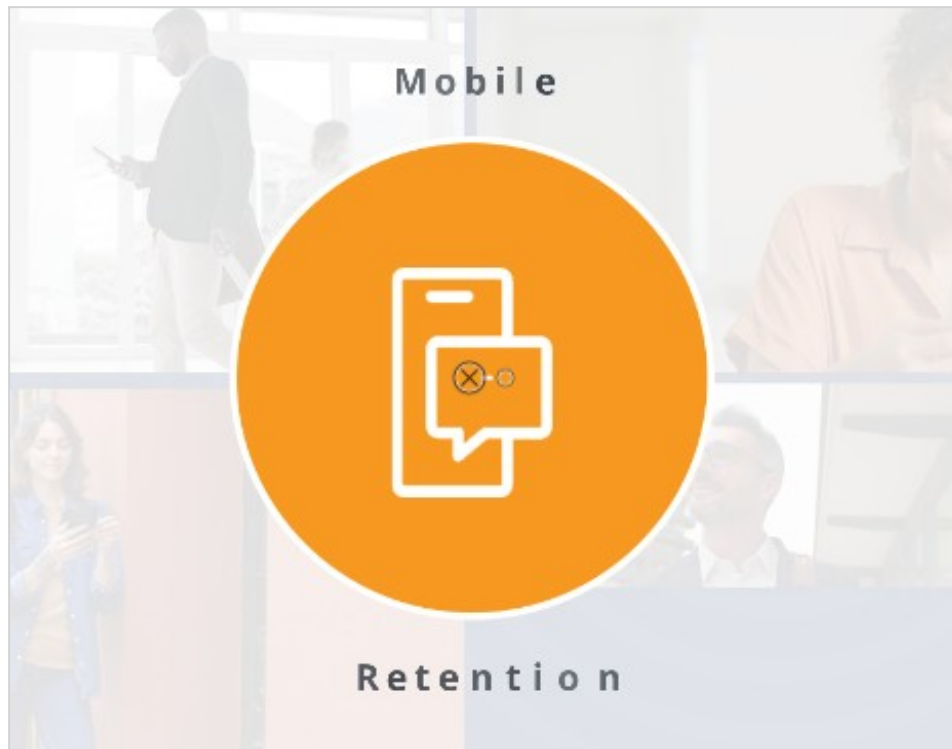
A customer, Peter, has a subscription with a Telco company. A Customer Service Representative (CSR) calls Peter to encourage him to purchase an upgraded version of the subscription.

This scenario is an upsell offer made in a contact center and is a valid customer action.



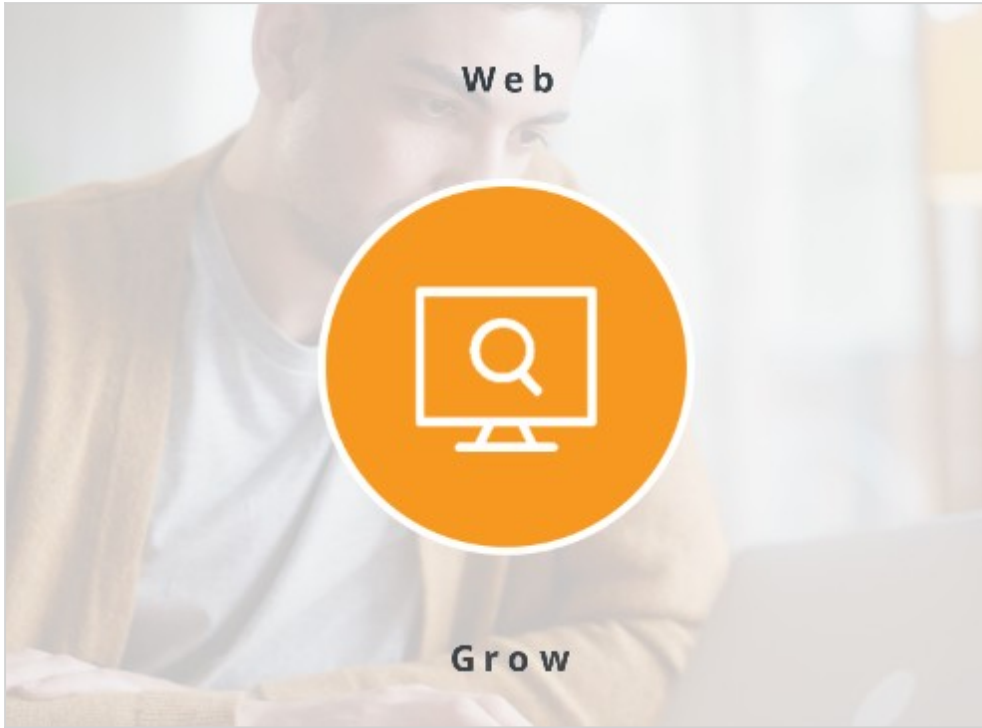
Scenario 2: SMS: Retention

A customer, Troy, doesn't use his credit card that often anymore and has decided to cancel the contract with U+ Bank. The Bank sends him an SMS message with the discount retention offer. This scenario is a retention offer made on the mobile channel to retain a customer and is a valid customer action.



Scenario 3: Inbound-web: cross-sell

A customer, Arnold, searches for a new phone subscription on a Telco company website. He finds an advertisement for a monthly subscription with unlimited mobile internet and calls. This scenario is a banner advertisement on a website and is a valid customer action.



Customer action properties

Every customer action has properties that define its characteristics. For example, a banner advertisement on a website usually consists of several elements, which help customers quickly understand the offer.

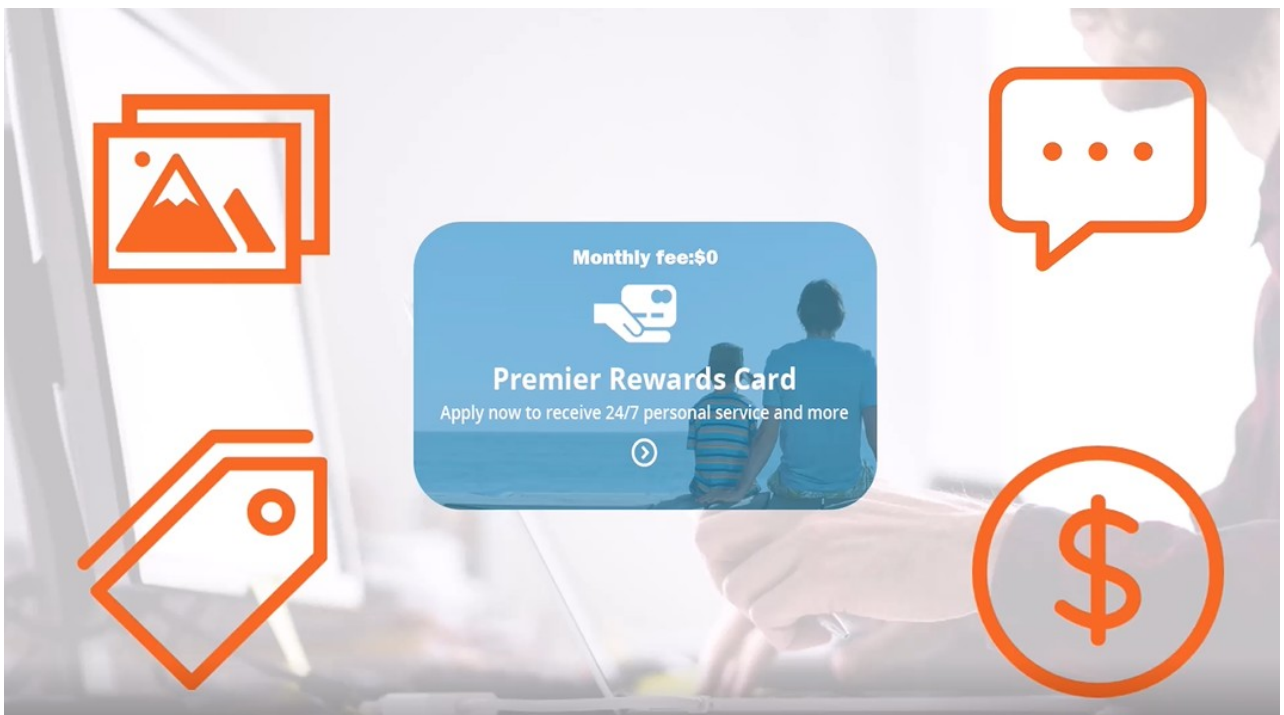


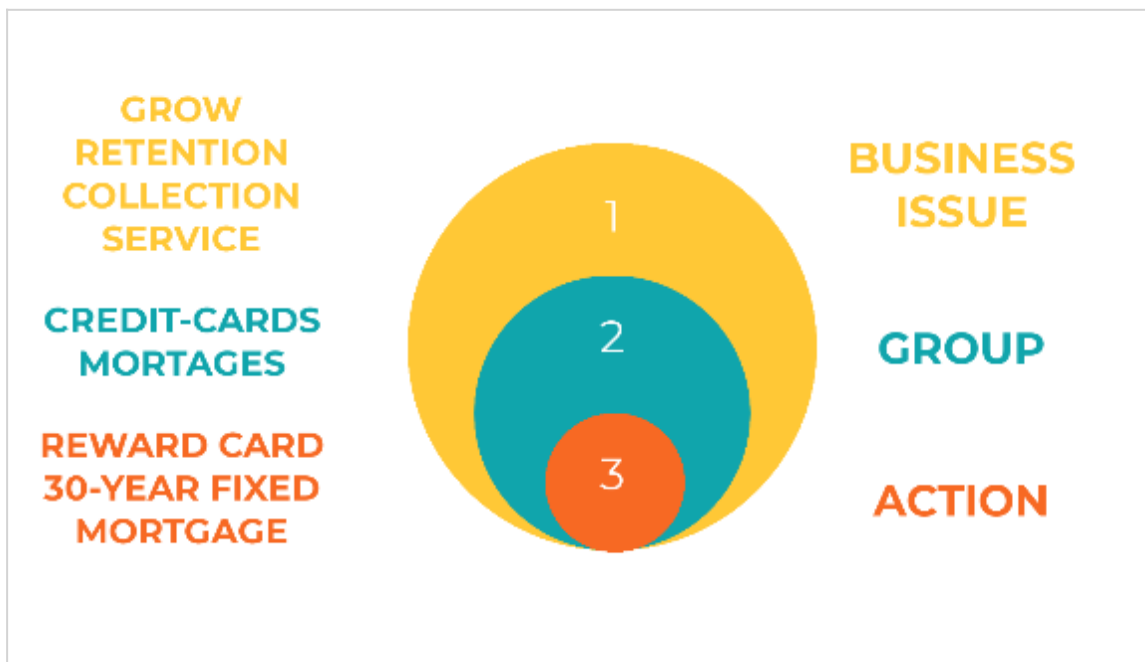
Image represents the visualization that is used for the banner. **Price** is a numeric property that represents the price of the customer action. **Benefit** represents the value proposition to the customer. **Short title** is a property that represents a short description of the customer action.

Business hierarchy

In Pega Customer Decision Hub, customer actions apply to various business issues and are organized into a three-level hierarchy. The business issue is the purpose of the actions that you offer to your customers. Each action that you prepare for your customers is associated with an issue and a group. Because of the association, you must always define the hierarchy before creating actions.

In Pega Customer Decision Hub, a **Business Issue** represents the business area for which a customer action is applicable (for example, **Grow, Retention, Collections, and Service**), a **Group** organizes actions into logical categories (for example, **Credit Cards or Mortgages**), and an offer, which is referred to as an **Action** (for example, a **Reward card or 30-year fixed rate**).

Each action that you prepare for your customers is associated with an issue and a group.



You have reached the end of this video. What did it show you?

- What are the customer actions in Customer Decision Hub.
- What are the customer action properties.
- Business hierarchy in Customer Decision Hub.

Managing business structure

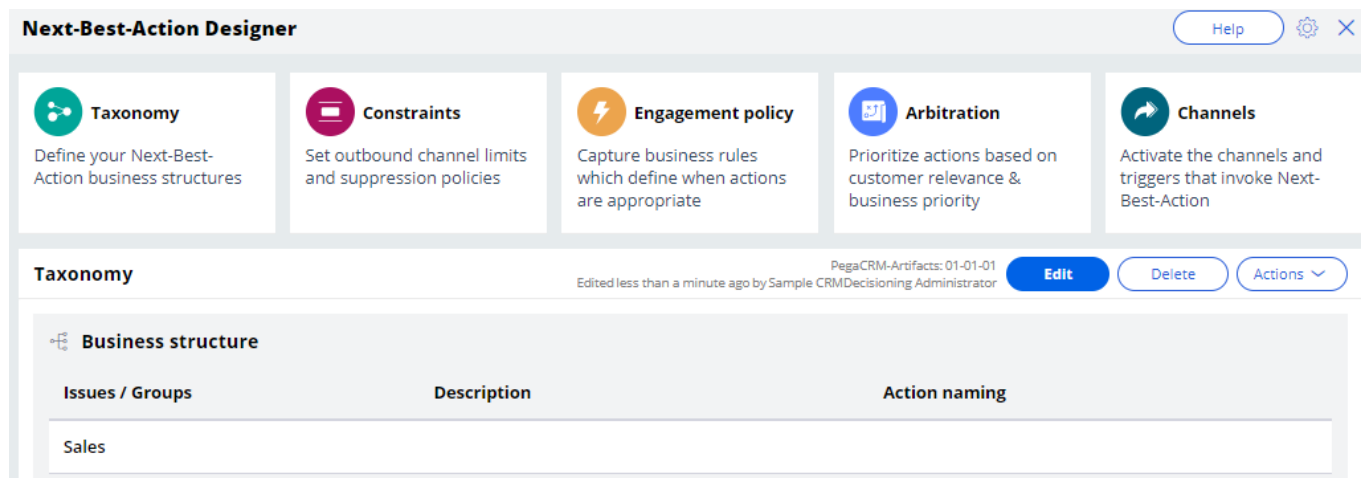
Introduction

In Pega Customer Decision Hub™, Next-Best-Action Designer is used to build the business issue/group hierarchies for your organization. Together, business issues and groups form the organizing structure for your customer interactions. Each Next-Best-Action that is presented to a customer is associated with a business issue and group.

Transcript

This demo will show you how to set up your business structure in Next-Best-Action Designer.

This is the Pega Customer Decision Hub™ portal. To manage the business structure, navigate to Next-Best-Action Designer. Customer actions are organized using a hierarchical business structure called Issues and Groups. Under the Taxonomy tab, you define the Issues and Groups that will play a role in the Next-Best-Action decisioning process.



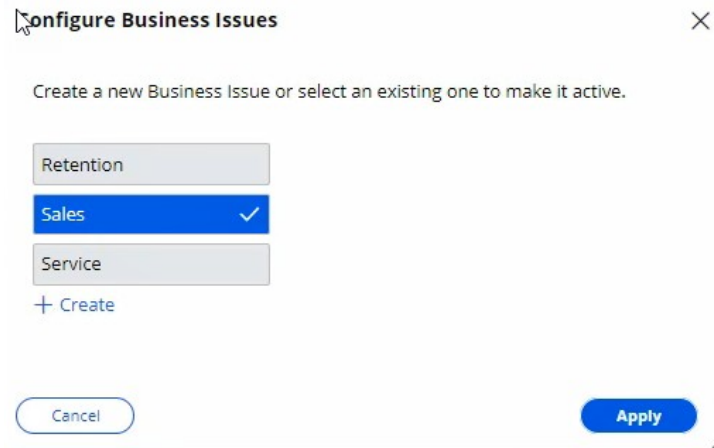
The screenshot shows the 'Next-Best-Action Designer' interface. At the top, there are five main tabs: Taxonomy, Constraints, Engagement policy, Arbitration, and Channels. The 'Taxonomy' tab is selected and active. Below the tabs, there is a section titled 'Taxonomy' with a sub-section 'Business structure'. This section contains a table with the following data:

Issues / Groups	Description	Action naming
Sales		

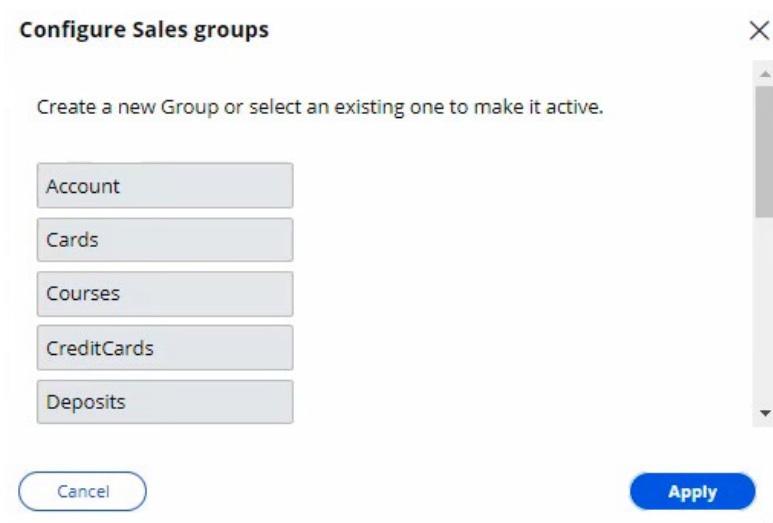
Additional UI elements include a 'Help' button, a settings gear icon, and a close 'X' icon in the top right corner. Below the 'Taxonomy' header, there is a metadata line: 'PegaCRM-Artifacts: 01-01-01 Edited less than a minute ago by Sample CRMDecisioning Administrator'. Action buttons for 'Edit', 'Delete', and 'Actions' are also visible.

In this case, you will set up U+ bank's business structure based on their various areas of business focus. Currently, there is only one Issue in the business structure.

You can add, remove, or create new Issues here by editing and configuring the hierarchy. Note that, this can be edited only with an additional privilege. Here is a list of Issues that were created before. You can also create new Issues.



Now, add an existing Issue. Similarly, you can add, remove, or create business Groups. To add more groups, configure the groups.



To complete the configuration, save the changes.

This demo has concluded. What did it show you?

- How to add an Issue to the hierarchy.
- How to add Groups to an Issue.

Creating actions

Introduction

An action holds various details about a particular offering, such as Start Date, End Date, and Expected Revenue. Each action is backed by a decisioning proposition. The system automatically manages this relationship by creating, deploying, and deleting the proposition instances as needed. Since an action is closely tied to a proposition, it must always be created in the context of an issue and group.

Transcript

This demo will show you how to create an Action. It will also explain how to create Actions in bulk.

U+ Bank, a retail bank, recently introduced several new credit card offers that they would like to display to their customers on a website. As a decisioning consultant, you have been tasked with creating these offers in the **Sales** Issue and **CreditCards** Group as per the pre-defined business hierarchy.

This is the Pega Customer Decision Hub™ portal. Actions are created and managed on the Actions landing page.

You can either create a single Action or Actions in bulk.

To create a single Action, enter a short description of the new Action. Select the appropriate **Issue** and **Group**. Now, open the Action.

The attributes of an Action distinguish it from other Actions. Attributes are used by the Customer Decision Hub to select the right Action for a customer. Fill in the attributes relevant to this Action. For example, provide a **Description** that explains its purpose and **Benefits** that describe how this Action will benefit the customer.

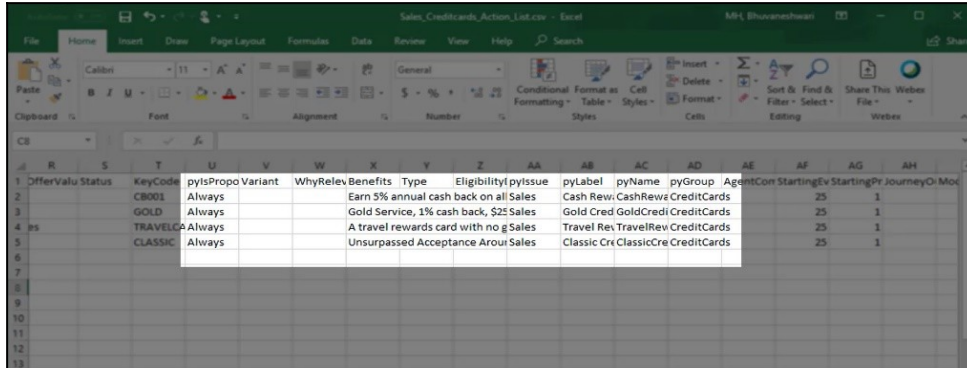
Bear in mind that some of the values you enter may be customer-facing information. For example, U+ wants to display the content of the Benefits attribute when this Action is presented on its website. *To complete the configuration, save the Action.*

To view Actions within a certain Group, you can filter by a specific **Issue** and **Group**. This is the Action you just created. Thus far you have created a single Action. However, sometimes you may want to create multiple Actions at once. You can do that by uploading a list of

Actions from a Comma Separated Value (CSV) file. To create such a CSV file you need to download a CSV template and fill it in.

Open the downloaded CSV template. You can use this template to add/update Actions. Add new Actions with corresponding attributes to the file.

Properties that start with py, such as pyName, pyIssue, pyGroup, pyLabel and pyIsPropositionActive are mandatory internal properties. It is important that you input these values in the correct format into the CSV file.



OfferValu	Status	KeyCode	pyIsPropo	Variant	WhyRelev	Benefits	Type	Eligibility	pyIssue	pyLabel	pyName	pyGroup	AgentCon	StartingEv	StartingPr	JourneyO	Moc
		CB001	Always		Earn 5% annual cash back on all Sales				Cash Rev: CashRewe	CreditCards				25	1		
		GOLD	Always		Gold Service, 1% cash back, \$25 Sales				Gold Cred	GoldCredi	CreditCards			25	1		
		TRAVEL	Always		A travel rewards card with no g Sales				Travel Res	TravelRes	CreditCards			25	1		
		CLASSIC	Always		Unsurpassed Acceptance Arout Sales				Classic Crx	ClassicCre	CreditCards			25	1		

The property pyIsPropositionActive corresponds to the Availability property, which is visible on the Details tab of an Action. Possible values for this property are 'Always', 'Never' and 'Date'.

Choose 'Always' to ensure that the Action is selected during the Next-Best-Action decisioning process by the Customer Decision Hub.

Set it to 'Never' if you never want the Action to be selected, for example when you want to retire the Action

If you set the value to 'Date', you should enter a date range in the 'StartDate' and 'EndDate' fields. The Customer Decision Hub will select this Action only during that time period.

pyIssue corresponds to the Issue the Action was created for.

pyLabel corresponds to a short description of the Action.

pyName corresponds to the Action name.

pyGroup is the Group to which the Action belongs.

Save and close the CSV file.

Now, import Actions from the file. Select the CSV file that contains the list of Actions you want to import. If you want to delete existing Actions from this Issue/Group that have been created in the system but not in the CSV file, you can select this option.

Create actions from file [X]

Issue / Group
Sales / CreditCards ▾

Select File Sales_Creditcards_Action_List.csv

Acceptable file type: CSV [Download CSV Template](#)

Action flow template

None

Reference flow from another action

Copy flow from another action

Delete existing actions that are not in the file

Import

For now, you are only interested in adding new Actions.

Notice that the window displays a summary of the result of the import.

Create actions from file [X]

File Name
Sales_Creditcards_Action_List.csv

Issue / Group
Sales / CreditCards

Summary

Actions added or updated	Actions deleted	Errors
4	0	0

<< Back Finish

For example, when a duplicate Action is added, the Errors count is incremented, and you can download the error file to learn the exact issue. Complete the import. Notice that the new Actions are now listed on the **Actions** page.

This demo has concluded. What did it show you?

- How to create an Action.

- How to set Action attributes.
- The most important properties of Actions in a CSV file.
- How to create multiple Actions using a CSV file.

Renaming actions

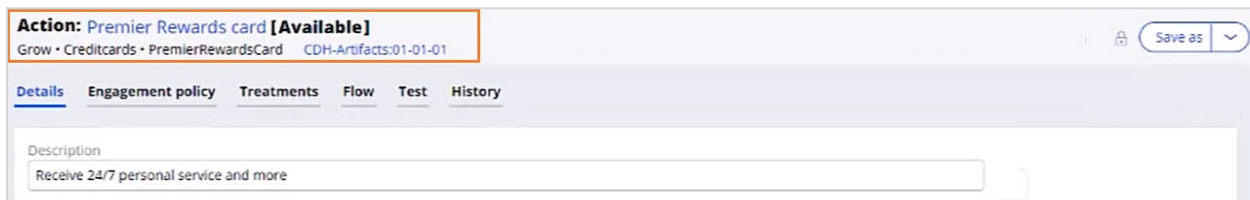
Introduction

By default, Pega Customer Decision Hub uses the term 'actions' to refer to messages delivered during a customer interaction. If needed, you can change this to a more appropriate term, such as Promotions or Nudges. Actions can be renamed with different terms based on the business issue and group to which the product belongs.

Transcript

This demo will show you how to change the default naming of Actions to something that better reflects their business context.

This is the Pega Customer Decision Hub™ portal. On the **Actions** landing page, you can view all the Actions that have been created. To view Actions within a certain Group, you can filter by a specific **Issue** and **Group**. In this case, filter Actions in the **Grow** Issue and **CreditCards** Group. Open an Action to view the name. Notice that the Grow/CreditCards Group uses the default terminology '**Action**'.

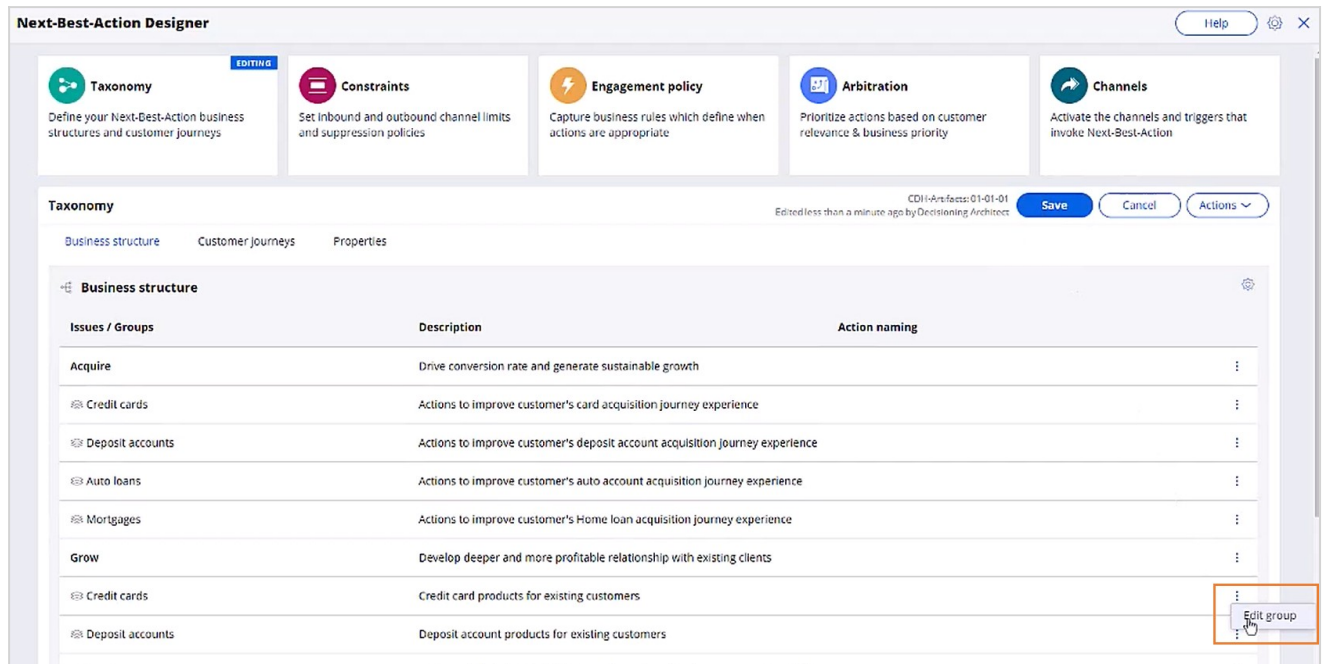


U+ bank wants to rename Actions under the **Grow** Issue and **CreditCards** Group to better suit their business purpose. The Actions under the **CreditCards** Group are always promotional credit card offers, therefore the bank wants to change the naming convention from 'Action' to 'Offer'.

You can change the default naming convention of Actions in Next-Best-Action Designer.

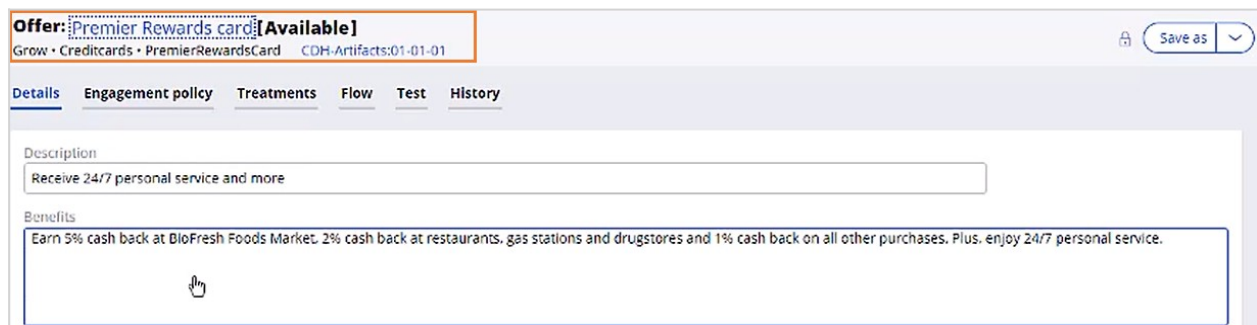
To manage the business structure, navigate to the Taxonomy tab of Next-Best-Action Designer. The 'Action naming' convention is set in a business Group's configuration settings. This can be set for each group in the business hierarchy.

Edit the hierarchy to modify the Group attributes.



In this case, change the Action naming convention to Offer.

To complete the configuration, save the changes. Verify that the Action is renamed. Open any Action under the **Grow** Issue and **CreditCards** Group to view the effect of renaming.



From now on, Actions under Grow/CreditCards will be referred to as Offers.

This demo has concluded. What did it show you?

- How to change the default naming convention of Actions, to a name that better reflects their business context.

Presenting a single offer on the web

Description

Learn about real-time containers and how they are used to display actions selected by Pega Customer Decision Hub in real-time channels. Gain experience creating and configuring real-time containers by learning how to configure Next-Best-Action Designer to select a single offer that will be displayed on a website.

Learning objectives

- Describe the concept of real-time containers
- Explain how to create and configure a real-time container
- Explain how to configure Next-Best-Action designer to display an offer on the web

Real-time containers

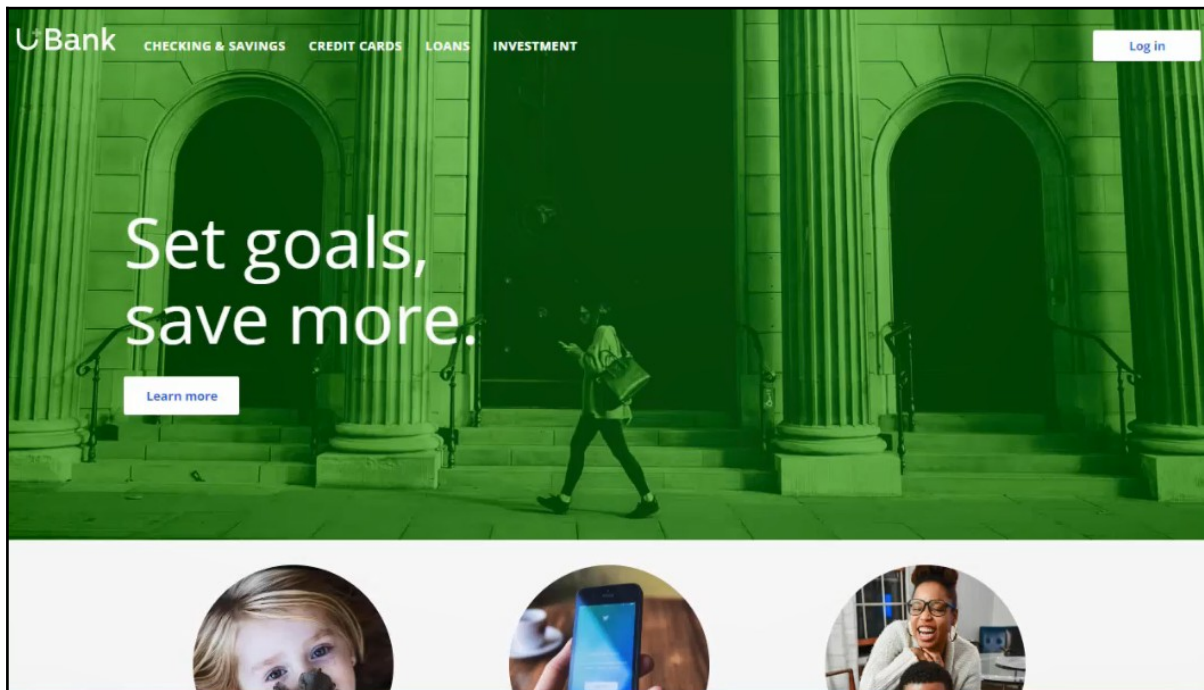
Introduction

A real-time container is a service that manages communication between Pega Customer Decision Hub and external channels. An external real-time channel is any channel that presents actions selected by the Customer Decision Hub to a user or customer. For example, a website, a call-center application, or a mobile application.

Transcript

This video explains the concept of real-time containers, which manage communication between the Pega Customer Decision Hub™ and external channels.

This is the website of a retail bank called U+ Bank.



The bank plans to promote new offers on the account page, which is displayed when customers log in.

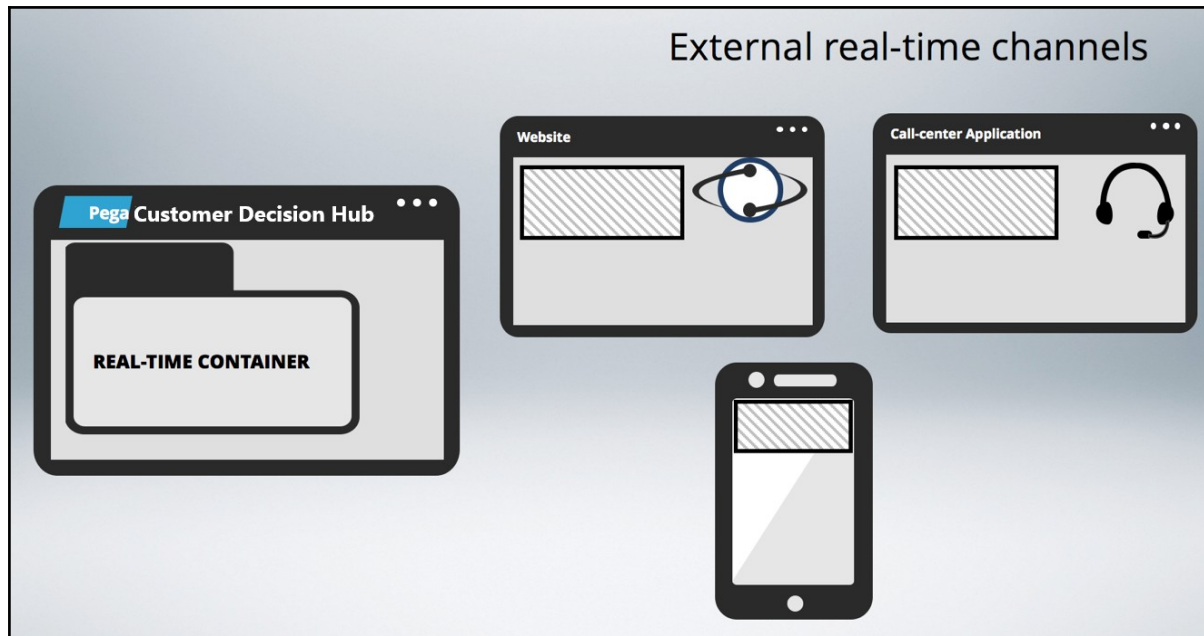
The screenshot displays a user's account overview on the UBank website. The interface is clean and organized, with a prominent green header. The 'Accounts' section provides a quick overview of the user's financial products. The 'Credit card balance' section offers detailed information about the current balance, due date, and CardProtect status, along with actionable buttons like 'Pay now' and 'View statement'. The 'Recent transactions' section provides a clear list of recent activity with dates and descriptions. The sidebar on the right is personalized, featuring a family photo and a targeted offer for 'Checking account with minimal fees', which is relevant for older customers. The 'Quick links' section provides easy access to common banking tasks.

The marketing department wants to leverage the Customer Decision Hub's Next-Best-Action capability to display the right offer for each customer.

The Customer Decision Hub's real-time container functionality is used to implement this requirement.

A real-time container is a placeholder for content in an external real-time channel.

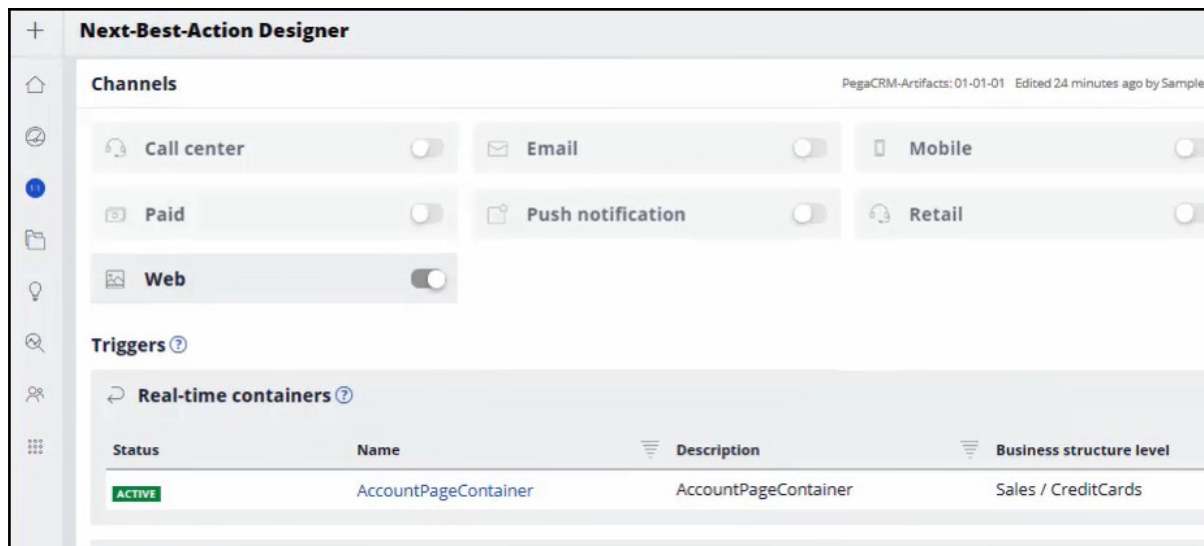
An external real-time channel is any channel that presents actions selected by the Customer Decision Hub to a user or customer. For example, a website, a call-center application, or a mobile application.



Here's how the website invokes the real-time container to present credit card offers on the account page.

In the Customer Decision Hub's Next-Best-Action Designer, a real-time container called "Account Page Container" is configured. The website invokes this real-time container before loading the account page.

The Customer Decision Hub then evaluates the actions from the associated Issue/Group, which in this case is Sales/CreditCards and returns the resulting offer details back to the website.



The website then loads the account page with the content returned by the Customer Decision Hub, such as the offer image, description and other relevant attributes.

UBank CHECKING & SAVINGS CREDIT CARDS LOANS INVESTMENT

Account overview

Accounts

- Savings ***1234
- Checking ***5678
- Credit ***7890

Credit card balance

Current balance \$164.80 Pay now	Due date Nov 15, 2019 View statement	CardProtect On Suspend CardProtect
--	--	--

Recent transactions

OCT 15, 2019	Your last statement is ready to view	View more
OCT 13, 2019	Interest Charge: Purchases	View more

Cash back
Earn 10% Cash Back
Use your U+ bank card today to get 10% cash back
[Learn more](#)

Quick links

- [Request a service](#)
- [Make a payment](#)
- [Update billing](#)

Meanwhile, the Customer Decision Hub records these customer interactions in the Interaction History. An Impression is recorded to indicate that the action was shown to the customer, a Click-through is recorded when the customer clicks on the action. Marketers use these metrics, i.e. Impressions and Click-throughs, to measure the level of customer engagement, and therefore, the success of the marketing effort.

Creating and configuring a real-time container

Introduction

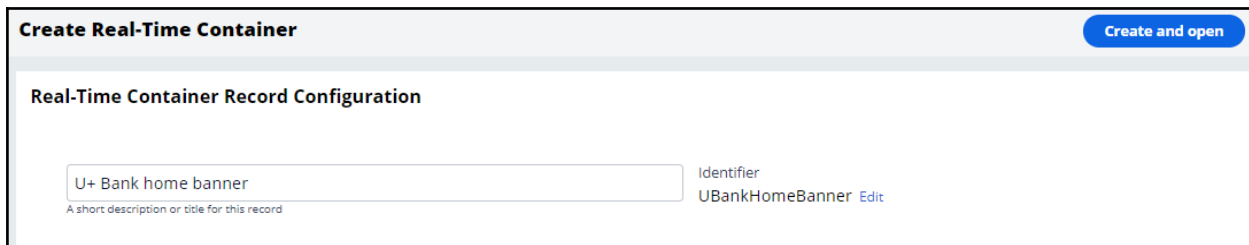
Learn how to create and configure a real-time container which allows you to manage communication between Pega Customer Decision Hub and external channels.

Transcript

This demo will show you how to create and configure a real-time container.

To create real-time containers, navigate to the Real-Time Artifacts landing page.

Enter a short description of the new real-time container.



Create Real-Time Container Create and open

Real-Time Container Record Configuration

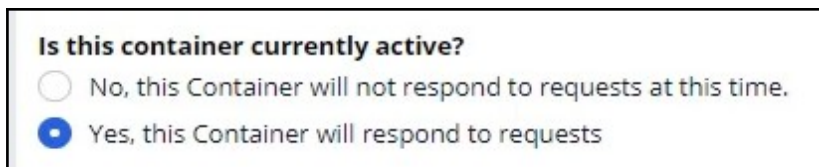
U+ Bank home banner A short description or title for this record Identifier UBankHomeBanner Edit

An external channel uses the real-time container's identifier when it makes a request to the Pega Customer Decision Hub.

You can change the identifier to a value that the external channel in your environment expects.

For example, the U+ Bank website is pre-configured to call a real-time container with the identifier Ubank_home_banner. Now, open the real-time container.

Here you can turn the real-time container's ability to respond to requests from external channels on or off.



Is this container currently active?

No, this Container will not respond to requests at this time.

Yes, this Container will respond to requests

Keeping the default value will allow the Customer Decision Hub to serve requests from the website.

'Impression capture' allows the external channel to control how the Customer Decision Hub records impressions.

Impression capture

Captured on retrieval

Captured by channel

'Captured on retrieval' means that the Customer Decision Hub records an impression in the Interaction History immediately after sending the action details to the external channel.

'Captured by channel' means that the Customer Decision Hub does not immediately record an impression. Instead, the external channel can explicitly request that the Customer Decision Hub records an impression at a later point in time.

In this case, the bank wants the impressions not followed by a click to be captured, so keep the default value.

To record a click in the Interaction History without initiating an action flow for the customer, keep the default value.

Click through behavior

Capture click through and initiate offer flow

Capture click through only

This option is suitable for actions that do not have a defined flow, for example, on the Web channel when a customer clicking on an ad does not trigger any follow-up steps.

Here you can view the list of Next-Best-Action strategies that you have associated with this real-time container.

Next-Best-Actions associated with this container

Associated NBA	Enabled	Start date	End date	Status
No Next-Best-Actions are associated with this container. Configure through Next-Best-Action Designer.				

Save the configuration. Navigate to the Containers tab to see the newly created real-time container.

This demo has concluded. What did it show you?

- How to create a real-time container.
- How to configure the real-time container settings.

Presenting a single offer on the web

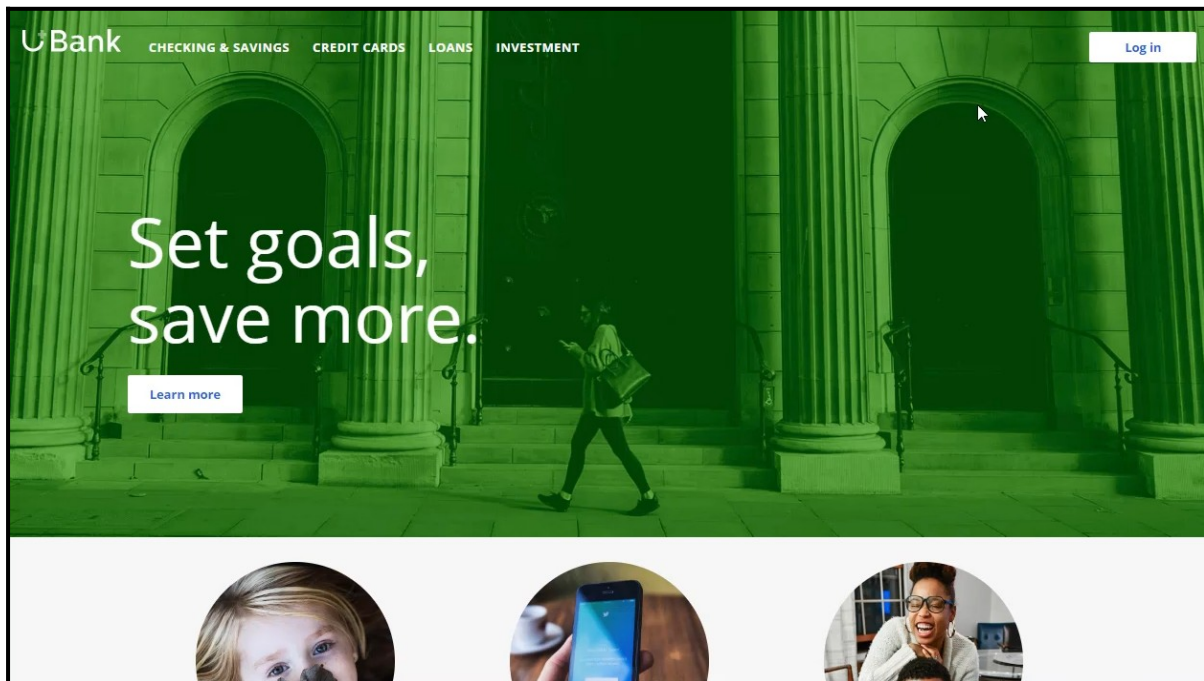
Introduction

Pega Customer Decision Hub is the always-on customer brain used to select the right offer to be presented to each customer in any real-time channel. Learn how to configure Next-Best-Action Designer to select a single offer that will be displayed on a website.

Transcript

This demo will show you how to configure Next-Best-Action Designer to select a single offer that will be displayed on a website.

U+, a retail bank, would like to use the Pega Customer Decision Hub™ to display a single offer on its website.



U+ bank wants to make offers related to credit cards and display the same 'Cash back' offer to every customer who logs in to the web site.

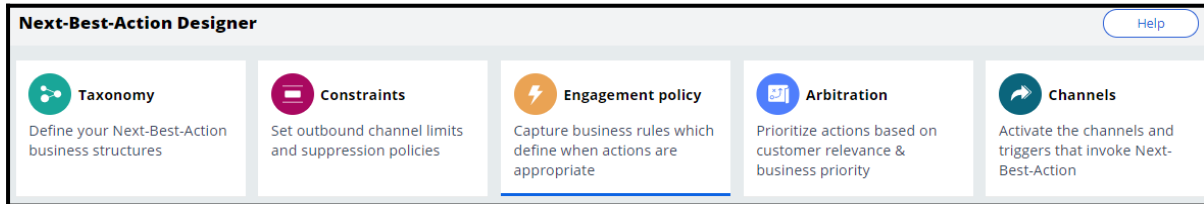
For example, if customer Troy logs in to his Accounts page, the 'Cash back' offer is displayed.

If Troy clicks on the 'Learn more' button, it takes him to the Offers page.

This page shows the offer details.

To implement this business requirement, navigate to Next-Best-Action designer.

Here you can configure business rules to define when specific actions or groups of actions are appropriate for customers.



Notice that the NBA hierarchy currently has three Business issues with Groups under them. In this case, U+ wants to promote credit card offers. So, open the CreditCards Group.

You can configure the actions here.

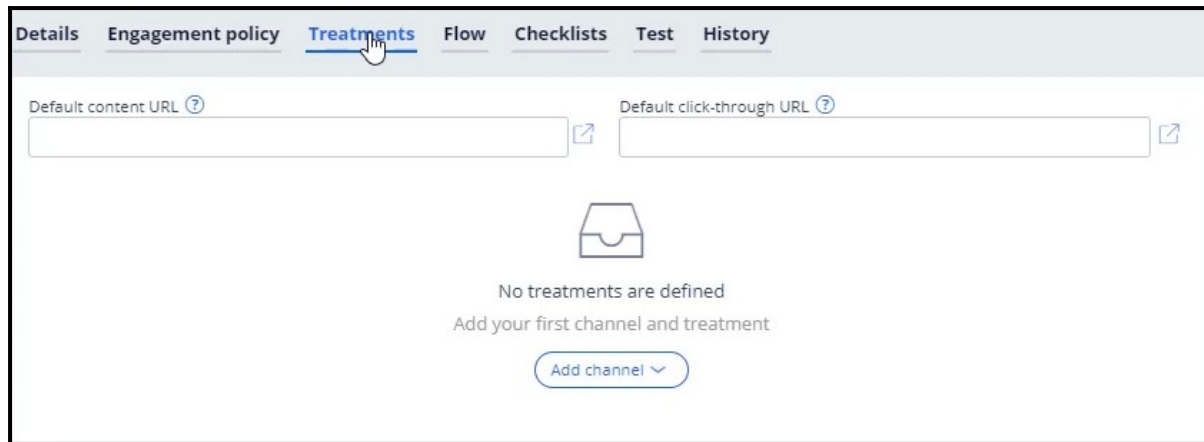


A consultant has already created a few actions under Sales/CreditCards. However, in this scenario, the bank wants to show only the 'Cash back' card.

Now, open on the 'Cash back' offer.

To display an offer, you need to add a treatment that is specific to the channel.

In this scenario, U+ bank wants to present the offer on their website, so select the right treatment type.



A web treatment represents the message that you want to communicate visually to your customer on the web channel. You can either use an existing treatment or create one here.

Now, fill in the required fields for the web treatment.

The screenshot shows a 'Create web treatment' form. At the top right, there is a blue 'Create' button and a 'Help' button with a close icon. The form has two columns. The left column contains: 'Name' (required, indicated by a red star) with an input field; 'Active' section with three radio buttons: 'Always' (selected), 'Never', and 'Within a defined time period'; 'Content URL' with an input field and an external link icon; 'Click-through URL' with an input field and an external link icon; and 'Dimensions' with an input field. The right column contains: 'Description' with an input field; 'Format' with a dropdown menu showing 'Select...'; 'Placement type' with a dropdown menu showing 'Select...'; and 'Language' with a dropdown menu showing 'Select...'. There is a horizontal separator line between the 'Active' section and the URL/Format section.

Here you provide a link to an image that will display the action for this treatment.

Once you provide the link, a preview of the content is displayed.

This is the URL that you want the customer to go to when they click on the action.

Now, select the intended location and style of the treatment design. For example, the treatment can appear as a large central banner, a rotating strip of images, or a footer on the website.

In this scenario, select Tile to display the 'Cash back' offer on the top right of the Account page.

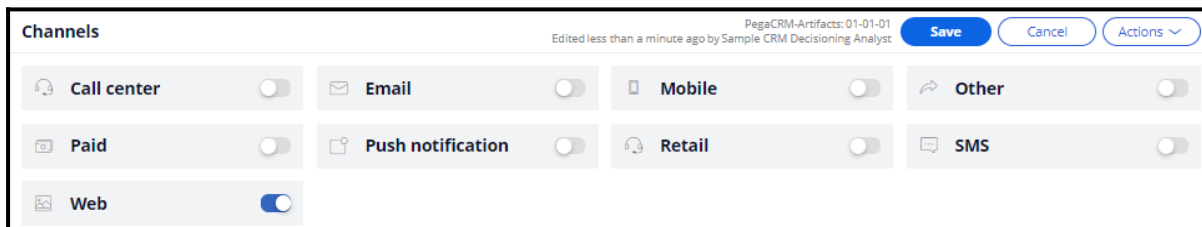
You can specify the language of the treatment if required. Specifying the treatment language allows Pega Customer Decision Hub to consider the customer's preferred language when selecting the treatment.

Add the web treatment you just created.

In this phase, U+ does not have any further eligibility or prioritization requirements for this action. Save the changes.

Here, you can enable the channels and triggers that will invoke Next-Best-Actions.

As U+ wants to display the offer on the web, enable the web channel.



Now, configure the real-time container that the U+ website will use to display the offer banner on the account page.

The real-time container manages communication between the Pega Customer Decision Hub and external channels such as the web and call center. A decisioning consultant has already configured a real-time container for you, so select it to be added to your configuration.

Once the real-time container is added, configure it to select the results from an appropriate Business issue and Group. In this case, U+ bank wants to display the 'Cash back' offer, which is under Sales/CreditCards.

Status	Name	Description	Business structure level
ACTIVE	SalesCreditCards	SalesCreditCards	Sales / CreditCards

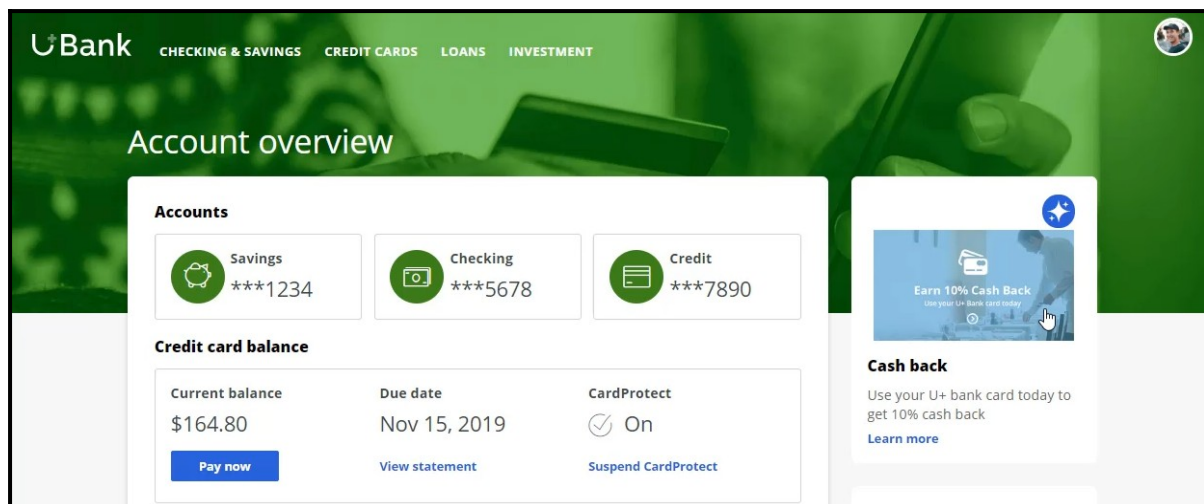
With that, all the necessary configuration for this scenario is complete. Save the changes for the configuration to take effect.

The account page on the U+ Bank website has been configured to use the real-time container with the name 'SalesCreditCards' and with Placement type 'Tile'.

The web treatment is shown as a tile on the top right of the page.

When Troy clicks on the 'Learn more' button, it takes him to the URL that was configured as the click-through URL for the treatment.

If another user logs in, they will see the same offer.



This demo has concluded. What did it show you?

- How to configure Next-Best-Action designer to display a web treatment on the U+ bank website.

Defining customer engagement policies

Description

Engagement policies are the set of conditions such as eligibility, applicability, and suitability that qualify an offer or a group of offers for a customer. As a result of engagement rules, customers will see only those offers that the organization believes they should be exposed to.

Learning objectives

- Define engagement policy conditions in Next-Best-Action Designer at the action group level
- Define engagement policy conditions at the action level
- Differentiate between eligibility, applicability and suitability conditions

Customer engagement policies

Introduction

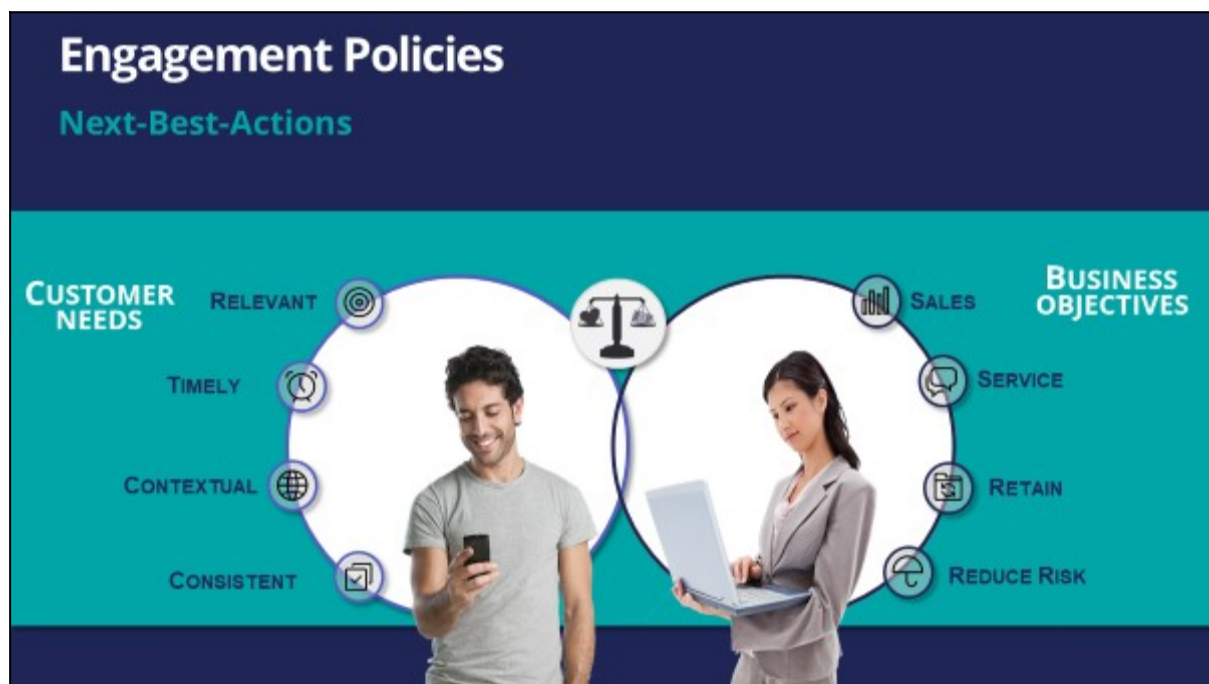
Engagement policies are a set of business rules and practices used by the organization to determine which customers are eligible for which Next-Best-Actions. These policies allow you to specify the conditions under which an action or group of actions are eligible for a customer.

To learn more about the three types of engagement policy conditions, see [Understanding engagement policies](#).

Transcript

This video explains the concept of customer engagement policies.

The Pega Customer Decision Hub™ combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that it uses to make intelligent decisions. These decisions are known as the Next-Best-Action. Every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business manager.



Typically, the business defines a set of rules that make certain actions available to certain customers. This set of rules is called an engagement policy.

As part of an engagement policy, three types of conditions are defined – Eligibility, Applicability, and Suitability.

Consider the following examples: a retail bank is promoting a Gold Credit Card; a telco is offering a new iPhone upgrade with an unlimited data plan; and a communications and media company is promoting a new bundle of HD channels.

Let's see how to define engagement policy conditions that will ensure the bank's Next-Best-Action decisions support these promotions.

In Eligibility, you define strict rules for what is legal, and even possible, to offer customers. For example, to be eligible for the Gold Card offer, customers must be 18 years or older.

Similarly, for the iPhone upgrade offer, customers are eligible for a new contract only if their old contract ends in less than three months.

For the TV channels offer, customers must already own a TV subscription. This offer is not available for customers who have only mobile or landline subscriptions.

Engagement Policy

Offers	Eligibility	Applicability	Suitability
Gold Card	Age >= 18		
New iPhone unlimited data plan	ContractEndMonth<=3		
50 Full HD TV channels	IsCustomer=True and OwnsTVPackage=True		

In Applicability, you specify rules for limiting what to offer based on a customer's current situation, which is often defined by the products they currently have. These rules are not as rigid as those for Eligibility.

For example, a Gold Card is not applicable if the customer already has a higher value card, such as a Platinum Credit Card. If a customer already has a Platinum Card, they might be eligible for the Gold Card, but the Gold Card is not applicable to them. If they ask for it, they may get it, but the business would prefer not to present them with the Gold Card offer.

Similarly, with the iPhone upgrade offer, if a customer explicitly expressed in the last survey that they weren't interested in an iPhone, this action is not applicable to them. For the TV channels offer, the business does not want to advertise HD channels to a customer who has recently bought a set top box that is not capable of HD.

Engagement Policy

Offers	Eligibility	Applicability	Suitability
Gold Card	Age >= 18	OwnedCard != Platinum	
New iPhone unlimited data plan	ContractEndMonth<=3	CustomerPreference=iPhone	
50 Full HD TV channels	IsCustomer=True and OwnsTVPackage=True	SettopBox=FullHD	

In Suitability, you specify conditions that define an offer as appropriate for a customer. Suitability rules are in place to promote the concept of empathy. That is, to help an enterprise be empathetic toward their customers and refrain from making offers that may not be a good fit.

For example, as the Gold Card is a high value card, it is only suitable for a customer whose debt-to-income ratio is below a certain threshold. Although a customer might be eligible for it, and the offer might be applicable to them, it would be inappropriate to market it to them, as there is a risk of default.

Similarly, an unlimited data plan is not suitable to be offered to a customer with low Internet usage. In the last example, if the customer's favorite TV shows are not available in HD, then it's not empathetic to offer them the new HD channels package.

Engagement Policy

Offers	Eligibility	Applicability	Suitability
Gold Card	Age >= 18	OwnedCard != Platinum	Debt to Income ratio < 50%
New iPhone unlimited data plan	ContractEndMonth<=3	CustomerPreference=iPhone	MonthlyUsage > 5GB
50 Full HD TV channels	IsCustomer=True and OwnsTVPackage=True	SettopBox=FullHD	FavoriteChannelList contains HD content

U+, a retail bank, has configured its engagement policy to suit its own business objectives as well as the needs of its customers.

In this scenario, a marketer for U+ has designed 200 actions that could be presented to customers. To select the Next-Best-Actions from these, the Pega Customer Decision Hub first checks eligibility conditions and filters the actions. Then, the applicability conditions are run to filter them further. Next, the suitability conditions are checked to derive the final set of available actions.

These actions will go through one final stage before being presented to customers: the arbitration stage. Arbitration is used to prioritize and choose the Next-Best-Actions based on what is relevant for each customer right now. This is decided by considering factors such as AI-calculated propensity, the action value, and various business levers.

Selecting actions for a customer



Defining eligibility, applicability, and suitability Rules

You define engagement policies in Next-Best-Action Designer. Engagement policies specify the conditions under which an action or group of actions is available for a customer. Configure these policies in the following categories: eligibility, applicability, and suitability, which represent the true nature of the associated conditions.

Transcript

This demo shows you how to define engagement policy conditions such as eligibility, applicability, and suitability by using Next-Best-Action Designer.

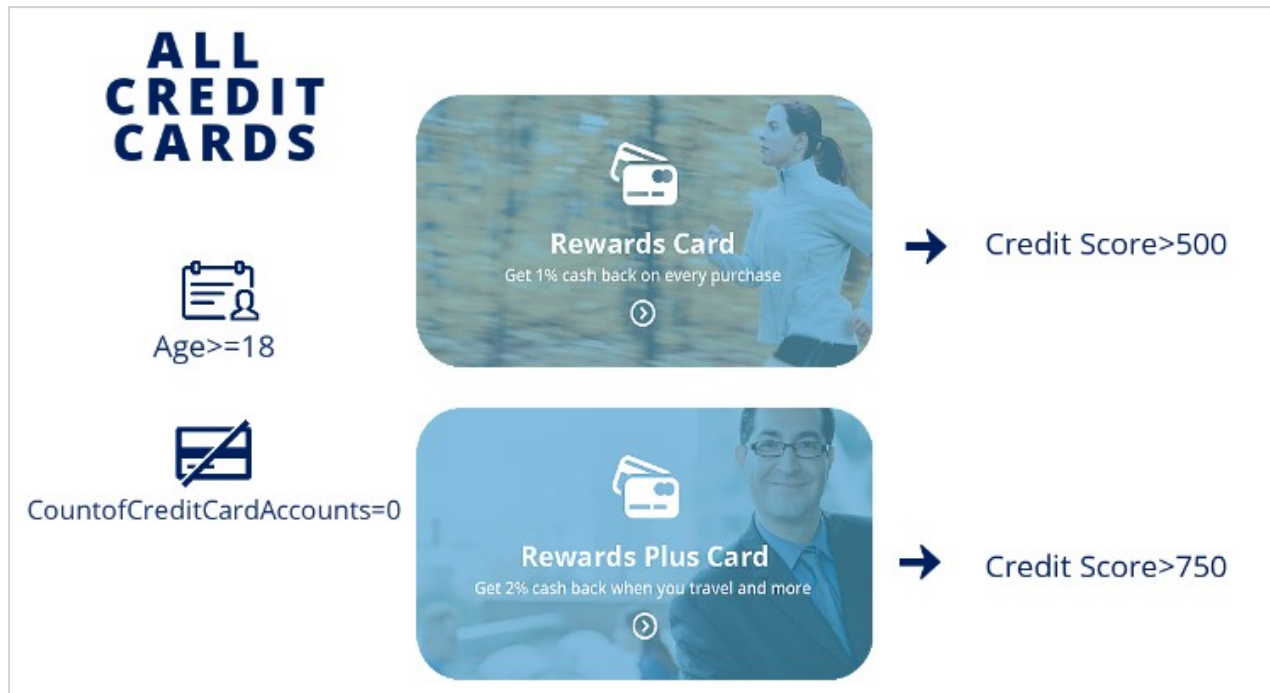
U+ Bank, a retail bank, has introduced two new credit cards and wants to offer them to customers based on certain criteria.

First, all credit cards are eligible only for existing U+ customers who are at least 18 years old.


In addition, the business wants to ensure that these two new credit cards are available for new customers who do not yet have a credit card.

Last, the business understands that not all credit cards are suitable for everyone.

Because of the credit limits of each card, the business wants to offer the Rewards Card to customers with a credit score higher than 500 and the Rewards Plus Card to customers with a credit score higher than 750.




In the Pega Customer Decision Hub™ portal, you define engagement policies in Next-Best-Action Designer. Engagement policies specify the conditions under which an action or group of actions is available for a customer.




Taxonomy

Define your Next-Best-Action business structures and customer journeys



Constraints

Set inbound and outbound channel limits and suppression policies



Engagement policy

Capture business rules which define actions are appropriate

Business structure ×

Expand All Collapse All

- > All issues
- > Acquire
- > Grow
- > Nurture
- > Retain
- > Collections
- > Service

Engagement policy


Engagement policies allow you to specify the conditions under which an action or group of actions is eligible for a customer. You can define four types of conditions as part of your engagement policy.

[Learn more](#)

Eligibility

The criteria which the customer must fulfill in order to qualify for the action or group of actions.


For example, an action may only be available for customers over a specific age or living in a specific geographic location.



Suitability

The criteria which must be true in order for the action or group of actions to be deemed in the best interest of the customer.

For example, a new credit card offer may not be appropriate for a customer who is likely to default on payments.



Now, you set up the engagement policies to reflect U+ Bank's requirements. You can define the engagement policies for a specific group within an issue and for individual actions. Edit to configure the engagement policies. Notice that the two actions are displayed in this group.

As a best practice, use one of the following categories to define your engagement policies: **Eligibility**, **Applicability**, and **Suitability**. These categories represent the true nature of the associated conditions.

First, define the group-level eligibility condition to ensure that only customers who are at least 18 years old are considered for the actions.

Then, define the group-level applicability condition so that only customers who currently do not have a credit card qualify for the actions.

If you wish to reuse a specific condition or a set of conditions in the future, there is an option to save them to a library. This feature helps to avoid recreating identical conditions repeatedly. Saved conditions are available on the **Intelligence>Engagement policies** tab, where you can modify and prepare them for reuse.

The **Save to library** option is unavailable if the conditions include either strategy Rules or When conditions with parameters. The option is also inactive for eligibility conditions, which include default criteria created with the **Use advanced logic** option.

The screenshot shows a configuration interface for 'Customer actions'. It is divided into two sections: 'Eligibility' and 'Applicability'. Each section contains a set of dropdown menus and a text input field to define a condition.

- Eligibility:** The first dropdown is 'Customer', the second is 'Age', the third is 'is greater than or equal to', and the text input field contains '18'.
- Applicability:** The first dropdown is 'Customer', the second is 'Count of credit card acc...', the third is 'is equal to', and the text input field contains '0'.

Lastly, U+ Bank wants different suitability conditions for individual cards depending on the limit. Because the conditions are specific to each credit card, you must define the suitability condition for each at the action level.

Open the **Rewards Card** action. This card comes with a certain credit limit. Therefore, U+ Bank believes that this card is only suitable for customers with a credit score higher than 500, even though customers may satisfy the eligibility and applicability conditions. Thus, define the suitability condition accordingly.

Suitability ?

Inherited from [Creditcards](#)

All actions

(Customer: Debt-to-Income ratio is less than or equal to 48)

and ⓘ

Customer



Credit score



is greater than



500

Next, open the **Rewards Plus Card** action. This card has an even higher credit limit than the Rewards card. Thus, this card is suitable for customers with a credit score higher than 750. To complete the configuration, save the changes.

Suitability ?

Inherited from [Creditcards](#)

All actions

(Customer: Debt-to-Income ratio is less than or equal to 48)

and ⓘ

Customer



Credit score

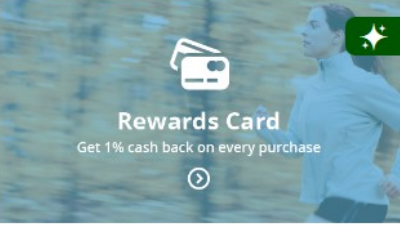


is greater than



750

Log in as Troy, a 25-year-old chef who became a customer of U+ Bank three months ago, and his credit score is 600. Notice that the Rewards Card offer is displayed because Troy satisfies all eligibility and applicability conditions. Troy receives this card only because his credit score is higher than 500 but lower than 750.

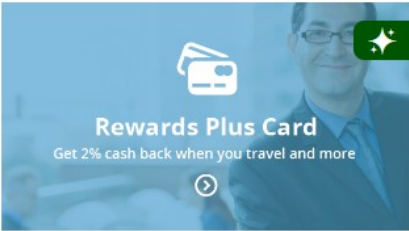


Rewards Card
Get 1% cash back on every purchase

[Learn more](#)

The image shows a promotional card for a 'Rewards Card'. It features a background image of a woman in a light blue shirt. Overlaid on the image is a white credit card icon, the text 'Rewards Card', and the offer 'Get 1% cash back on every purchase'. A small green star icon is in the top right corner, and a right-pointing arrow is at the bottom center.

Now, log in as Barbara, a 40-year-old engineer who became a customer of U+ Bank one month ago; her credit score is 800. Notice that the Rewards Plus Card offer is displayed because Barbara also satisfies all eligibility and applicability conditions. Barbara, on the other hand, receives offers for both cards because her credit score is higher than 750.



Rewards Plus Card
Get 2% cash back when you travel and more

[Learn more](#)

The image shows a promotional card for a 'Rewards Plus Card'. It features a background image of a man in a suit and glasses. Overlaid on the image is a white credit card icon, the text 'Rewards Plus Card', and the offer 'Get 2% cash back when you travel and more'. A small green star icon is in the top right corner, and a right-pointing arrow is at the bottom center.

You have reached the end of this video. You have learned:

- How to define group-level engagement policy conditions in Next-Best-Action Designer.
- How to define action-level engagement policy conditions.
- How to differentiate between eligibility, applicability, and suitability conditions.

Creating and testing personas

Description

Learn how to create personas and channel contexts in Pega Customer Decision Hub™, and how to use them for next-best-action testing. Explore the reasons behind action filtering through persona-based testing. Examine the results of the persona test cases and analyze how changes in the engagement policies affect the test results.

Learning objectives

- Create personas and channel context.
- Use personas for next-best-action testing.
- Create and analyze persona test cases.

Testing Next-Best-Action with Persona and Channel Context

In Pega Customer Decision Hub™ (CDH), after applying engagement policies and suppression rules to actions, it is necessary to test the settings to ensure that customers receive the expected and desired results. There are two different types of tests that the Decisioning Architect can perform:

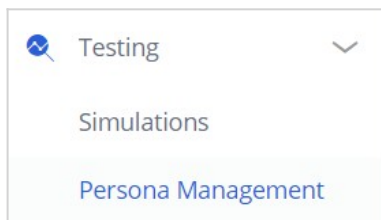
- Next-Best-Action testing in Customer Profile Viewer.
- Persona testing in Next-Best-Action Designer.

Transcript

This video shows you how to create Personas and Channel Contexts and how to use them for next best action testing.

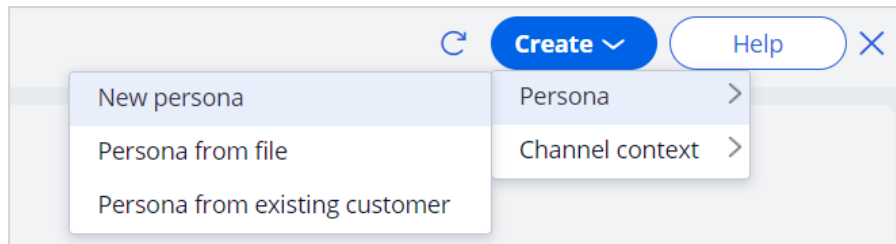
U+Bank, a retail bank, is cross-selling on the web. The bank wants to test the recent next best action configurations in Customer Profile Viewer. To generate the test results of the next best action strategy framework, the bank needs to create personas and channel context.

The dedicated space for creating, updating, and managing the Personas and Channel Contexts in Customer Decision Hub is the **Persona Management** landing page.



A persona is a profile of a fictitious person, representing a group of customers with a similar set of attributes. There are three types of personas you can create in CDH:


- **New persona**, where you input the customer properties manually,
- **Persona from file**, when creating Personas in bulk, using an external file,
- **Persona from the existing customer**.



To test the next best action configurations, U+Bank decided to use the Persona created based on existing customer Troy Murphy. Troy is a 26-year-old chef, who has been a customer of the bank for over three months, with a high credit risk. Troy is also considered financially vulnerable.

It is a good practice to use such customer information to name your Persona. The descriptive naming convention helps users to recognize the profile faster and to avoid repetitive Persona profiles. Following this rule, you can name the persona created based on Troy's profile: **Young High Risk Financially Vulnerable**.

Customer preview

 Troy Murphy 14	Email address customer1@enablement.com	Lifetime value 900	Credit risk High	Churn risk Medium
---	---	-----------------------	---------------------	----------------------

Persona name *

When you submit the new persona, the system auto-populates the actual customer information and organizes it in the customer attributes tree, which represents the Customer Insights cache. All the customer details are editable. You use this option to create

the persona based on the customer, and then adjust it to specific industry needs.

Entites ?		Customer + Add property	
Persona		Property	Value
>	Customer	Age ?	26 🗑️
>	Interaction history summaries	Annual income ?	13652.000 🗑️
		Date of birth ?	7/8/1995 📅 <input type="checkbox"/> Use dynamic value 🗑️
		Churn risk ?	Medium 🗑️
		CLV designation ?	Medium 🗑️
		Customer lifetime value ?	900 🗑️

In the Persona profile, you can also add a Persona description and tags.

Name	Young High Risk Financially Vulnerable
Description	26-year-old chef, relationship length over three months, with a high credit risk

Tags are the labels that you add to personas for the purpose of identification and easier searching. Tags can consider customer properties like **Credit risk**, **Age**, and **Relationship length in days** and so on, as well as engagement policy rules.

ID	YoungHighRiskFinanciallyVulnerable
Tags	new_customer × financially_vulnerable × low_debt × 🗑️
Last modified by	Decisioning Architect
Created date	Sep 21, 2023 3:12:23 AM

For example, if the applicability rule is Relationship length in days is ≤ 90 , and Troy's relationship with the bank is 88 days, we can tag him as a new_customer. The tag Financially_vulnerable means that Troy does not meet the suitability condition of Is financially vulnerable=false. Low_debt means that the Debt-to-income ratio for Troy is less than or equal to 48.

Personas include only raw information about the customer. Therefore, to run the tests in context, an additional factor that contains information about the **Direction**, **Channel**, and

Real-time container is necessary. This factor is called **Channel context** and it can also be created on the Persona Management Landing Page.

U+Bank is cross-selling on the web, therefore there is a need to perform a test for the direction **Inbound** and the channel **Web**. The real-time container for this channel context is **Top_Offers**.

When you create the channel context, the system automatically displays it on the Channel Context tab of the Persona Management landing page.

The screenshot shows a 'Create channel context' form with the following fields and values:

- Name ***: Web Inbound
- ID**: Webinbound
- Description**: Describe the use cases of this channel context
- Tags**: web, inbound
- Direction ***: Inbound
- Channel ***: Web
- Real-time container ***: Top offers (highlighted with a blue border)
- Page placements (in priority order) ?**: (empty)
- Context (optional) ?**: (empty)

Buttons: Cancel, Submit

Creating personas using Pega GenAI features

Pega introduces new features that use the power of Pega GenAI™ to simplify your daily tasks in Pega Customer Decision Hub.

With Pega GenAI, creating personas is effortless and efficient. When creating a new persona, you can describe the characteristics you want to include in natural language. It will allow you to create a diverse set of personas quickly that accurately represent your target audience.

Create new: Persona Save Actions ×

General

Name ID

Description Tags

Created by: Decisioning Architect | Last modified date: | Last modified by: | Created date: Sep 18, 2023 5:08:48 AM

Persona details

Entities ⓘ

- Persona
 - Customer ✦ ⋮
 - Interaction history summaries

References ⓘ + Add

No entities selected

Persona entities contain all the data that makes up a persona. On the left you will find a list of entities, some refined further into sub entities. When selecting the entity, its respective properties will appear here. Select an entity or sub entity, or add new entities or sub entities and populate them with properties to personalize this persona to your needs

In this case, you can generate customer details by typing your persona characteristics in the text box. Create a mature, low-debt, financially invulnerable customer, who lives in New York.

Generate customer details with AI ×

Additional description of persona

Cancel Preview prompt Generate

Based on the given details, the system generates the Customer data for the new persona. In this case, the age information provided is automatically set to 45, "low debt" corresponds to a low debt-to-income ratio, and "financially vulnerable" is translated to the "isFinanciallyVulnerable" flag being set to false.

Customer		+ Add property
Property	Value	
Age ?	45	
Annual income ?	75000.0	
Date of birth ?	5/20/1975 <input type="checkbox"/> Use dynamic value	
CLV designation ?	Low Value	
Credit score ?	700	
Credit score last updated time ?	1/15/2020 10:30 AM <input type="checkbox"/> Use dynamic value	
Customer ID ?	CUST123	
Debt-to-Income ratio ?	0.25	
Push notification opt-in ?	true	
Is in active duty ?	false	
Is in bankruptcy ?	false	
Credit score stale ?	false	
Is customer active ?	true	
Is deceased ?	false	
Is financially vulnerable ?	false	
Is in active complaint ?	false	
Is in arrears ?	false	

The remaining properties are randomly filled and can be reviewed and updated as necessary.

Testing next best actions with newly created persona and channel context

U+Bank wants to test the next best actions using the newly created Persona and Channel context. The objective is to identify whether any actions have been filtered out for the Young High Risk Financially Vulnerable Persona and determine the reasons why.

You can proceed with next best action testing in the Customer Profile Viewer. Search for Persona Young High Risk Financially Vulnerable profile, and on the next best actions tab, make the decision for the Web Inbound Channel context.

Make Next-Best-Action decisions for the customer in the specified channel. The results are determined by engagement policies, constraints and propensity. Save Channel context updates

Channel context: Web Inbound | Direction: Inbound | Channel: Web | Real-time container: Top offers | Page placements (in priority order): | Context (optional): | Effective decision date: 9/26/2023 Make decision

Show all decision results Show only decision results that will be delivered

All results | Top treatments

> Capture response

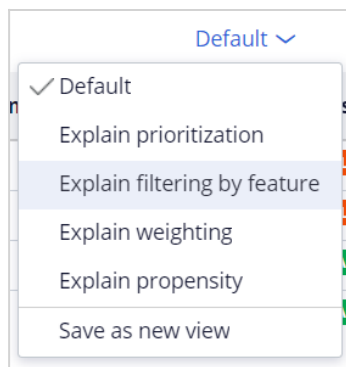
+ AI Insight

The Grow Issue Group in the Creditcards Group has a total of 4 candidate offers. Out of these, 50% (2) of the actions passed and 50% (2) were filtered out. The most occurring filter reason is "Excluded by suitability".

Group	Fields	Default	Export complete report					
Issue	Group	Name	Treatment	Channel	Priority	IA Test name	Rank	Results
Grow	Creditcards	RewardsPlusCard	Rewards Plus card tile	Web	16.14		1	FILTERED
Grow	Creditcards	PremierRewardsCard	Premier Rewards card tile	Web	8.72		2	FILTERED
Grow	Creditcards	RewardsCard	Rewards card tile	Web	3.58		3	PASSED
Grow	Creditcards	StandardCard	Standard card tile	Web	3.23		4	PASSED

In the action results table, you can observe that two out of four credit cards were filtered out, which means that they will not be presented to the customer.

You can explore the reasons behind this decision by utilizing the "Filtering by feature" option. The action results table has been enhanced with multiple categories that might serve as the underlying cause for excluding certain actions.



For instance, actions may be excluded based on eligibility, applicability, suitability, contact policy suppression, or if no treatment has been defined. When the labels in the table columns display "false," it indicates that the action is not excluded and can be presented to the customer.

In the case of Persona Young, High Risk, Financially Vulnerable, two actions have been filtered out due to the Suitability condition. The label in the Excluded by suitability column of the action results table indicates 'true'. Next to the label is an information icon that provides an explanation of the decision.

The Rewards Plus card and the Premier Rewards card were excluded because of the suitability condition: Is financially vulnerable=false. Young High Risk Financially Vulnerable Persona is financially vulnerable, he does not meet this condition and therefore, the two actions cannot be presented to him.

Suitability explanation ✕

There are multiple levels of Engagement policies that effect whether or not an action will be considered as part of Arbitration or filtered out. The results for each of these levels is provided below with an indicator if the customer passed these policies

Standard view **Advanced view**

- NBA_Grow_Creditcards_S_All** PASSED
 - Inherited from Creditcards PASSED
 - (Customer: Debt-to-Income ratio is less than or equal to 48)
 - Action Suitability
 - No conditions defined
- NBA_Grow_Creditcards_S_Customer** FAILED
 - Inherited from Creditcards PASSED
 - Always true
 - Action Suitability FAILED
 - (Customer: Is financially vulnerable is false)**

Additionally, in CPV, another GenAI feature: the AI Insight is available. It provides the summary of the tested configuration. In this example, you receive a clear explanation that there are four candidate offers for the Persona. However, only 2 actions passed, and 2 actions were filtered out due to the Suitability conditions.

> Capture response

AI Insight

Based on the provided data, the performance of the "Grow" issue group for the "Creditcards" group is as follows: - Total candidate offers: 4 - Passed actions: 2 - Filtered actions: 2 - Number of filtered actions per criteria: 2 actions were excluded by suitability. This means that 50% of the actions passed and 50% were filtered out due to suitability issues.

Group Fields

<input type="checkbox"/>	Issue	Group	Name	Treatment	Channel	Excluded by eligibility	Excluded by applicability	Excluded by suitability	Excluded by treatment eligibility	Excluded by Journey stage	Suppressed by contact policy	No active channel	Action Inactive	No treatments defined	No active treatments	Excluded by model maturity
<input type="checkbox"/>	Grow	Creditcards	RewardsPlusCard	Rewards Plus card tile	Web	false	false	true	false	false	false	false	false	false	false	false
<input type="checkbox"/>	Grow	Creditcards	PremierRewardsCard	Premier Rewards card tile	Web	false	false	true	false	false	false	false	false	false	false	false
<input type="checkbox"/>	Grow	Creditcards	RewardsCard	Rewards card tile	Web	false	false	false	false	false	false	false	false	false	false	false
<input type="checkbox"/>	Grow	Creditcards	StandardCard	Standard card tile	Web	false	false	false	false	false	false	false	false	false	false	false

By clicking on the Polaris icon for each row in the Results column, you can access an explanation of the action results in the Action results explanation window, specific to the action.

In this example, you will learn that the RewardsPlusCard is filtered out due to the suitability condition.

Action results explanation ★ GENERATED USING AI

Grow / Creditcards / PremierRewardsCard / Premier Rewards card tile FILTERED

Filtered Offer:

- **Issue:** Grow
- **Group:** Creditcards
- **Offer:** PremierRewardsCard
- **Treatment:** Premier Rewards card tile
- **Channel:** Web
- **Model evidence:** 3132
- **Model positives:** 1032
- **Model propensity:** 0.2956989247311828
- **Final propensity:** 0.2775043986616834
- **Filtered:** true

Exclusion Reason:

- **Reason:** Excluded by suitability

Explanation:

The offer has been filtered out due to suitability concerns. The specific reason for exclusion is not provided in the given information. To make this offer pass, you need to address the suitability issues. Here are some probable approaches:

1. **Review Targeting Criteria:** Double-check the targeting criteria for the offer to ensure it aligns with the intended audience. Make sure the offer is suitable for customers who fall under the "Grow" issue, "Creditcards" group, and are interested in the "PremierRewardsCard" offer.
2. **Refine Treatment:** Evaluate the "Premier Rewards card tile" treatment and ensure it effectively communicates the benefits and features of the Premier Rewards Card. Consider optimizing the treatment to better resonate with the target audience.
3. **Analyze Model Results:** Review the model evidence, positives, propensity scores, and final propensity. Identify any potential issues or discrepancies that might have contributed to the exclusion. Adjust the model parameters or retrain the model if necessary.
4. **Address Suitability Concerns:** Investigate the specific suitability concerns that led to the exclusion. It could be related to compliance, legal, or ethical considerations. Take appropriate actions to resolve these concerns and ensure the offer meets all necessary requirements.

Learn more about:

[Excluded by suitability](#)

You have reached the end of this video. What did it show you?

- What is Persona and Channel context.
- How to create a Persona based on a customer profile.
- How to create a Persona using the Pega GenAI feature
- How to create a channel context.

- How to test the next best actions using Personas.
- How AI insights are generated using Pega GenAI.

Testing the Engagement Policy configurations with Persona Test Cases

In Customer Decision Hub, after applying engagement policies and suppression rules to actions, you need to perform tests to ensure that customers receive the expected and desired results.

- Next-Best-Action testing in Customer Profile Viewer.
- Persona testing in Next-Best-Action Designer.

Transcript

This video shows you how to conduct a persona test to evaluate the next-best-action strategy results.

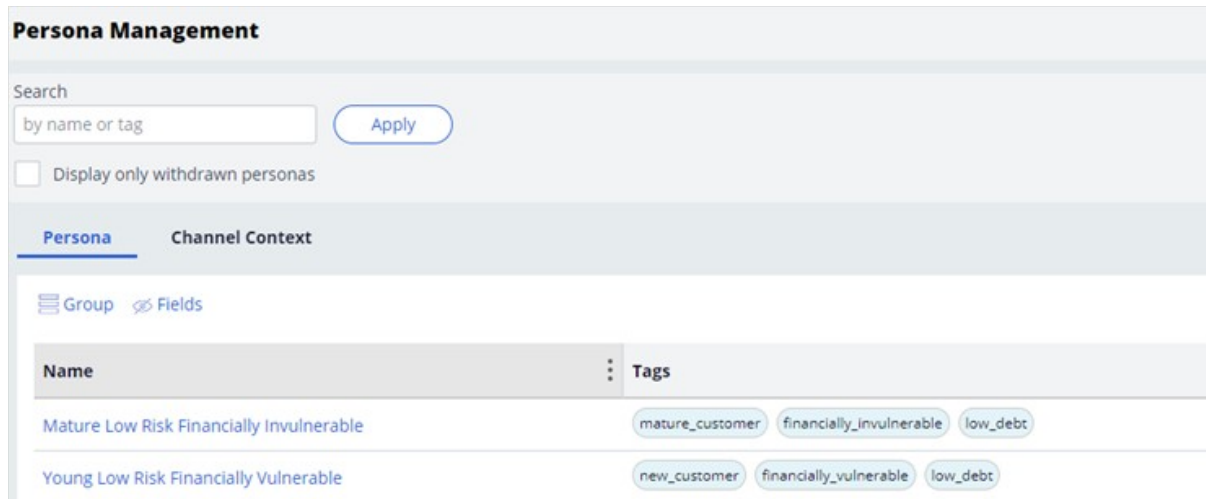
U+ Bank, wants to verify that customers receive the correct offers from the engagement strategy and that the offers are in line with the business requirements of the bank.

Persona-based tests use customer personas with specific characteristics to evaluate next-best-action engagement strategy results.

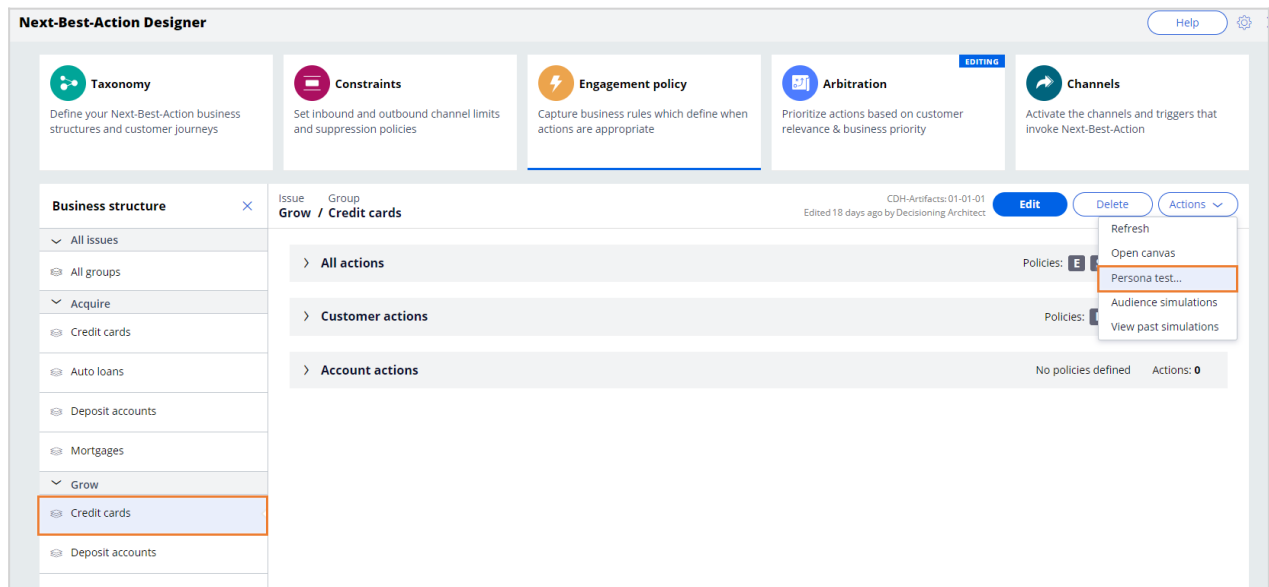
To test the configurations, the bank decides to use a **Mature Low Risk Financially Invulnerable** and a **Young Low Risk Financially Vulnerable** personas which are created based on customers Barbara and Troy profiles.

You can see the attributes of each persona in the Persona Management landing page:

- **Mature Low Risk Financially Invulnerable** is a 32-year-old engineer who became a customer of U+ bank a year ago, and her annual income is 14452.
- **Young Low Risk Financially Vulnerable** persona is a 26-year-old chef who became a customer of U+ Bank 3 months ago, and his annual income is 13652.



With persona-based tests, you verify that the next-best-action strategy gives the expected results. On the **Engagement Policy** tab of Next-Best-Action Designer, you can create test cases for a specific group or all groups. Running test cases for all groups implies that you run the next-best-action strategy for all issues and groups in your business structure. As the bank wants to target the credit cards group, create a persona test case at the group level.



To configure a persona test case, select the action and treatment that you expect the test persona to receive according to your next-best-action strategy. In this case, **Young Low Risk Financially Vulnerable** Persona is supposed to receive the Standard card and Rewards card actions.

In the **Persona (Data transform)** field, select the persona against which you want to test the strategy. Select **Young Low Risk Financially Vulnerable**.

In the Channel context field, select *WebInbound channel context*, which defines the Direction, channel and the Real-time container for the test.

In the Next-Best-Action scope section, select whether the test checks only the engagement policy configuration or includes additional elements such as constraints and arbitration.

Selecting **Engagement policies only** validates that your policy conditions provide the desired results. The selection ensures that the test looks at the eligibility, applicability, and suitability.

Selecting **Engagement policies and arbitration** validates the effectiveness of your policies when you arbitrate across all actions. The test considers arbitration, adaptive analytics, treatment and channel processing, and constraints. In this case, select **Engagement policies only**.

Mature Low Risk Financially Invulnerable Persona is eligible for the Rewards Plus card and the Premier Rewards card. Create a test case for this persona. Run the tests to confirm that they pass.

The bank now wants to offer the Standard card to customers who have an annual income greater than 14000. See how this engagement policy change affects these test results. To define an engagement policy, click **Standard card**. Define a suitability condition so that only customers who have a minimum annual income of 14000 qualify for the actions.

Suitability ?

Inherited from Creditcards

All actions
(Customer: Debt-to-Income ratio is less than or equal to 48)

and ?

Group ANDs

Customer Annual income is greater than 14000

Run both tests to see how this engagement policy configuration affects the test cases.

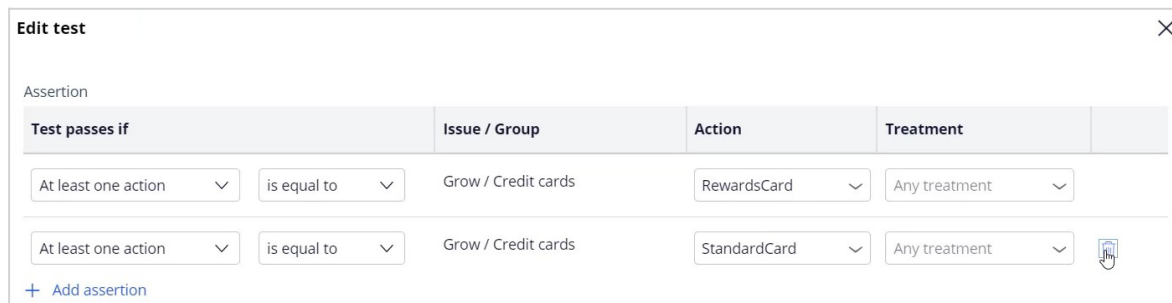
Observe that **Young Low Risk Financially Vulnerable** persona fails the test. The failure means that customers with a profile similar to Troy does not satisfy all the existing engagement policies. Analyze the results to see why the test did not give the expected outcome.

Based on the report, **Young Low Risk Financially Vulnerable** persona is not eligible for the Standard card.


Young Low Risk Financially Vulnerable persona's annual income is 13652, which is less than the defined applicability condition. **Mature Low Risk Financially Invulnerable** persona's annual income is 14452, which is higher than the defined applicability condition.

Although the **Young Low Risk Financially Vulnerable** persona fails the test, the **Mature Low Risk Financially Invulnerable** persona passes it.

Now, edit the test for the **Young Low Risk Financially Vulnerable** persona to delete the Standard card assertion, and then run the test.



The screenshot shows a dialog box titled "Edit test" with a close button (X) in the top right corner. Below the title is the word "Assertion". There is a table with four columns: "Test passes if", "Issue / Group", "Action", and "Treatment".

Test passes if		Issue / Group	Action	Treatment	
At least one action	is equal to	Grow / Credit cards	RewardsCard	Any treatment	
At least one action	is equal to	Grow / Credit cards	StandardCard	Any treatment	

At the bottom left of the dialog box, there is a link: "+ Add assertion".

You can design the personas according to the business requirements and test them every time an engagement policy changes.

You have reached the end of this video. What did it show you?

- How to create persona test cases.
- How engagement policy changes affect the persona test results.
- How to edit persona test cases.

Avoiding overexposure of actions

Description

Learn about contact policies and how they are used in Pega Customer Decision Hub. Gain experience defining contact policy rules to suppress actions after a specific number of outcomes.

Learning objectives

- Describe contact policy requirements
- Explain how to define contact policy rules to suppress a single action or group of actions for a limited time period

Contact policies

Understanding contact policy requirements

Too many contact attempts over a short period of time can have a negative impact on a customer's attitude toward further actions by your company. To maximize the lifetime value of every customer relationship, organizations must prevent outreach fatigue by optimizing the number of actions taken.

In the Pega Customer Decision Hub, contact policies allow you to suppress actions after a specific number of outcomes.

Suppressing or pausing an action prevents oversaturation by limiting the number of times a customer is exposed to the same action.

Defining contact policies

Contact policies determine when and for how long an Action or group of Actions should be shown to a customer. Contact policies track responses to Actions over a specific period of time, allowing you to implement rules such as the following:

- Do not show an ad to a customer for two weeks if the customer ignores the ad five times in a one-week timeframe.

Note: If outcomes are tracked for an individual action, then the action is not shown once the suppression criteria are met.

- Do not show a group of ads for six months if a customer clicks on any ad in the group 3 times over a period of 30 days.

Note: If outcomes are tracked for all actions in the group, then all of these actions are not shown once the suppression criteria are met.

An Interaction History Summary rule is used to determine the number of impressions and clicks generated by a customer over a period of time. The default time periods are 7 and 30 days. There might be business requirements to track a customer's response to an offer over different time periods, for example, 14 days.

You can add more tracking periods by creating a new Interaction History Summary rule for the required time period and then updating the part of the Next-Best-Action strategy that references it.

Defining action suppression rules

Introduction

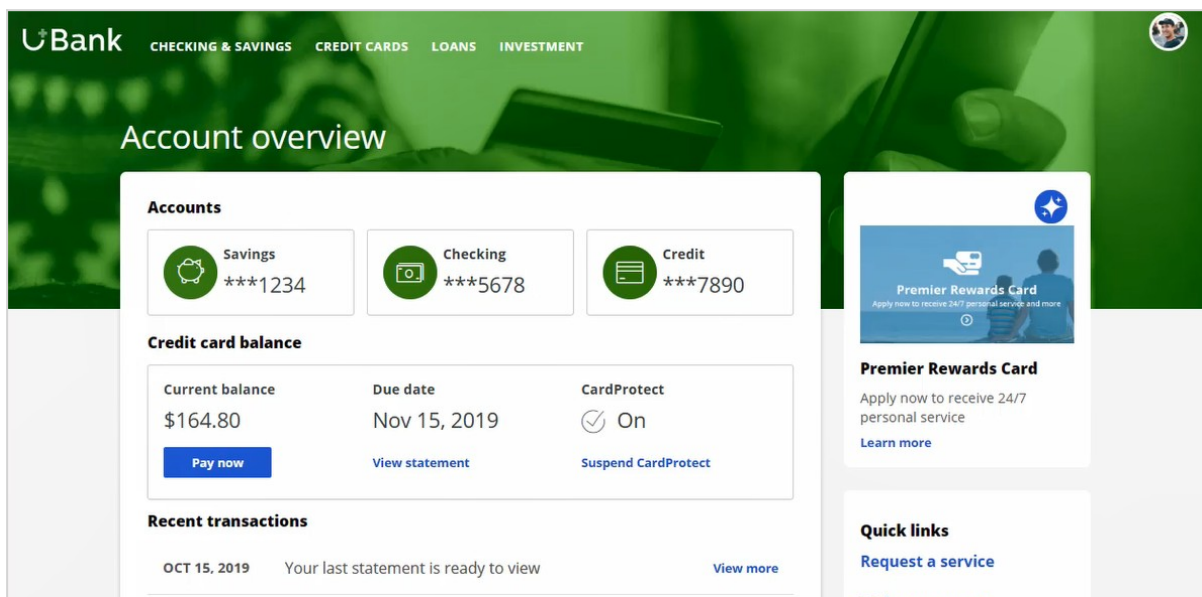
Suppression rules determine when and for how long an action or group of actions should not be shown to a customer. These suppression rules put an action on hold after a specific number of outcomes are recorded for some or all channels.

Transcript

This demo will show you how to suppress a single action or group of actions for a limited time period.

U+, a retail bank, currently displays various credit card offers to each customer who logs in to the website.

For example, every time Troy logs in to his accounts page, a credit card offer is shown. Sometimes the same offer is shown multiple times.



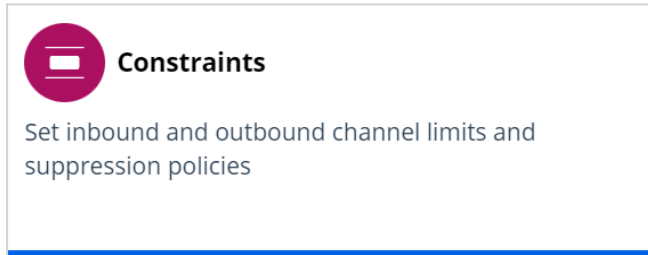
For a limited time period, the bank wants to automatically suppress offers that are shown or clicked too often.

In this scenario, the bank has two requirements. First, do not show a credit card for ten days if the card was shown three times in the last seven days.

Second, do not show any credit cards for ten days if a user has clicked on a credit card five times in the last seven days.

Contact policies are used to implement these business requirements. You create contact policies in Next-Best-Action Designer.

On the Constraints tab, you can define the suppression rules by creating contact policy rules.



For the first requirement, showing a credit card a maximum of three times, configure a contact policy to track Impressions at the Action level.

Provide a name for the contact policy.

Then select the type of outcome that will be tracked by the contact policy, in this case Impressions.

You can specify whether the responses are tracked for one specific action, or for all actions in the group. Track the first requirement at the Action level, since you want to show one specific card a maximum of three times.

You can select the time period over which the responses should be tracked. In this case, responses should be tracked over a period of seven days.

A screenshot of a dialog box titled "Add contact policy" with a close button (X) in the top right corner. The dialog contains the following fields: a "Name" field with a red asterisk and the text "Three Offer Impressions"; a "Scope" section with "Track" followed by a dropdown menu set to "Impressions", the word "for", another dropdown menu set to "the action", the text "within the past", a third dropdown menu set to "7", and the word "days". At the bottom, there are two buttons: "Cancel" on the left and "Submit" on the right.

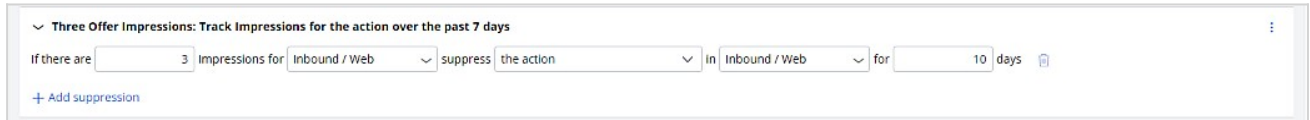
The newly configured contact policy has been added.

The first business requirement is to suppress the action for ten days if there are three impressions for the web treatment, so fill in the details accordingly.

Enter the number of responses required to fulfill the suppression criteria.

Select the channel for which the responses are tracked. Note that if you want to track impressions across multiple channels, you can select **Any**.

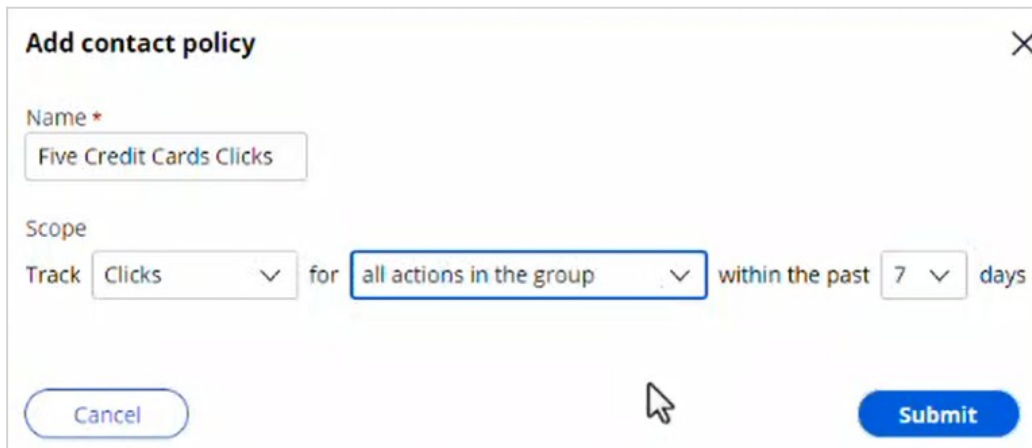
Enter the number of days for which an action should be paused after the suppression criteria are met.



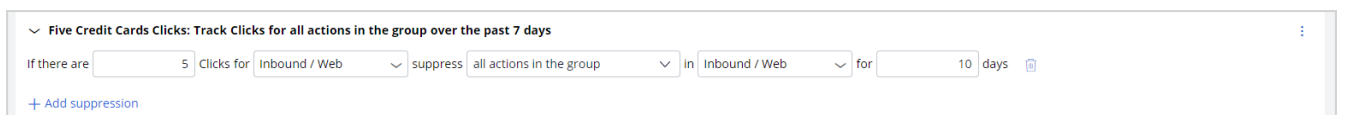
The next business requirement is to suppress the entire group of actions if there are five clicks for web treatments.

The first contact policy is configured to track Impressions, so add another contact policy.

For this requirement you will be tracking Clicks for all actions in the group. This because you want to hide all credit cards if there are five clicks on any one credit card.



Once the contact policy is created, fill in the suppression rule details. If there are five clicks on web treatments, suppress the group of actions for ten days.



Save the changes.

The contact policy rules are reusable as policy rules across all business issues and groups.

As the bank wants to suppress credit card offers, open the CreditCards group.

Add the contact policy rules you just created.

With that, all the necessary configuration for this scenario is complete. Save the changes.

When customer Troy logs in to his accounts page, the Premier Rewards Card offer is displayed.

After showing him this offer three times, it is automatically suppressed, and a different credit card offer is shown.

When customer Barbara logs in to her accounts page, a credit card offer is displayed.

After clicking on any credit card offer five times, the credit card offers are not shown again.

This demo has concluded. What did it show you?

- How to define contact policy rules to suppress a single action or group of actions.

Arbitrating between actions

Description

After applying engagement policy rules, a customer may still qualify for more than one action. Learn how every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business.

Learning Objectives

- Describe what action arbitration is and how it works
- Explain how customer needs and business objectives are considered during arbitration
- Prioritize actions based on AI
- Prioritize actions based on AI & business levers

Action arbitration

Pega Customer Decision Hub™ (CDH) combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that CDH uses to make intelligent decisions. Arbitration aims to balance customer relevance and business priorities by weighing numerical values for the following factors: propensity, context weighting, business value, and business levers. Learn to create a simple formula for arriving at a prioritization value, which is used to select the top actions.

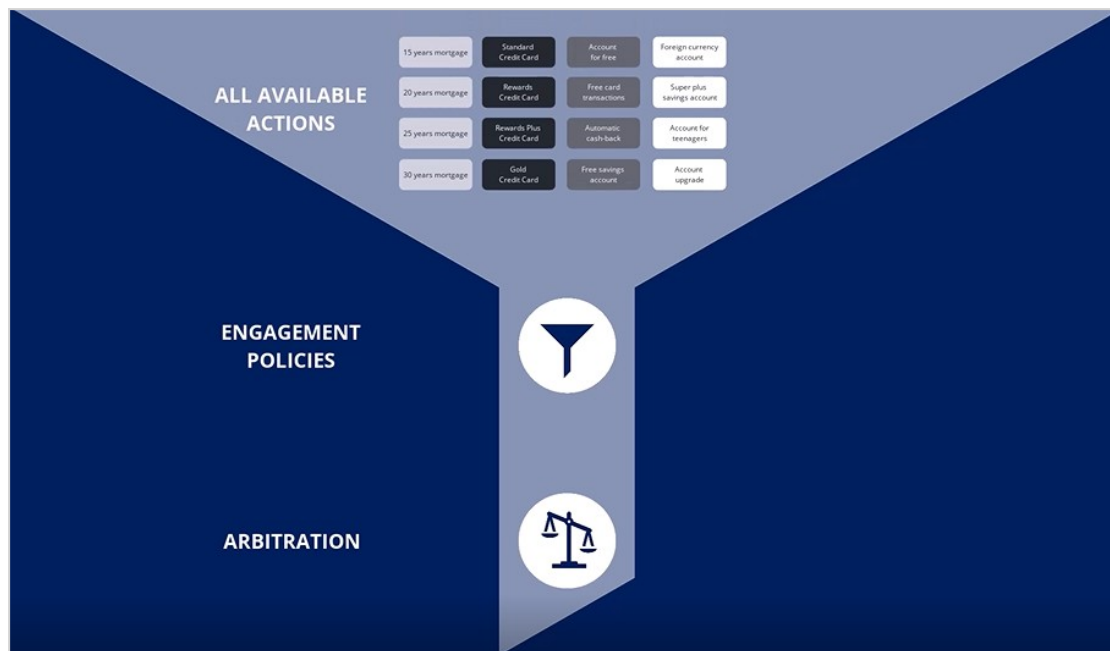
Transcript

This video explains the concept of action arbitration in Pega Customer Decision Hub.

Pega Customer Decision Hub combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that it uses to make intelligent decisions. These decisions are known as Next-Best-Actions. Every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business manager.



U+, a retail bank, wants to avoid offering random actions to its customers, so it uses Pega CDH to rank and select the next best actions for every customer. All the actions from the Bank's offer move through several steps that whittle them down until the Next Best Action is selected.



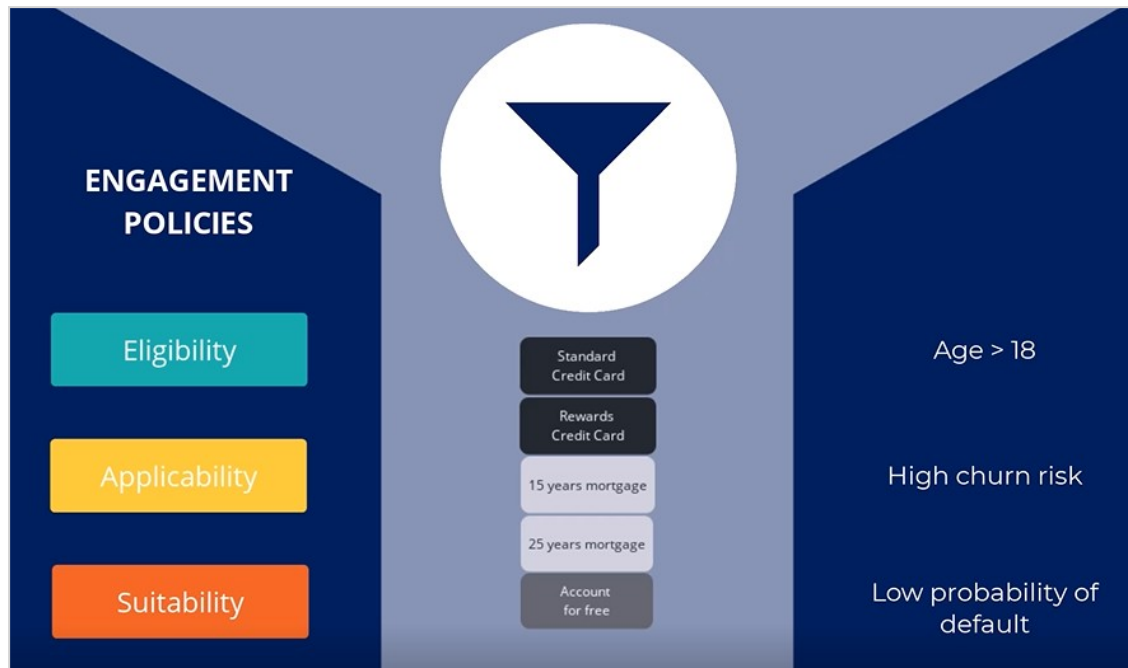
The process starts with the catalog full of actions that the organization can offer to its customers. These actions belong to various issues and groups.

The first step is to apply the Engagement Policies: Eligibility, Applicability, and Suitability conditions.

Eligibility rules are used to determine whether customers are eligible for an action. Users must meet the condition for CDH to consider the action eligible. An example of an eligibility condition is the customer's age. A customer can be offered a credit card only if they are 18 years old or older.

Applicability rules are used to limit what to offer, based on a customer's current situation. For example, the bank wants to show a retention offer, instead of a credit card offer, to customers who are likely to churn in the near future.

Suitability rules are used to define whether an offer is appropriate for a customer. For example, if a customer is in debt and has a lot to pay off, the bank does not want to offer them a 30-year mortgage as it would not be empathetic.



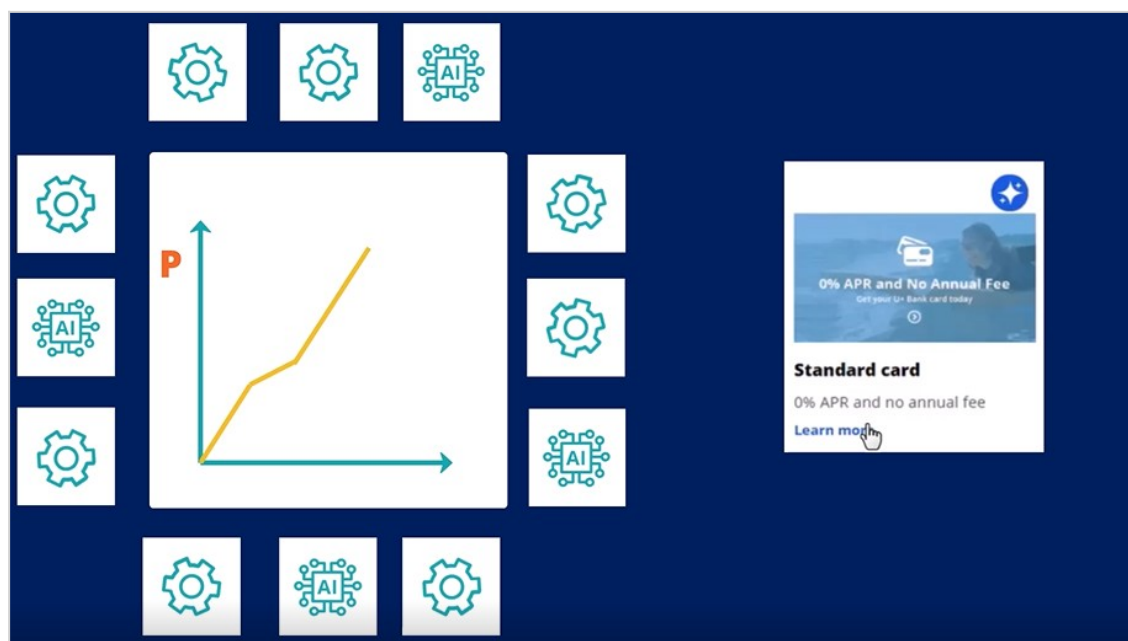
The initial catalog of actions has been narrowed down by engagement policies. These engagement policies are a combination of business rules, for example, certain age requirements for a loan or some product compatibility rules, and predictive models which help predict customer behavior as customer churn or probability of default. These predictive models are created by data scientists in Pega, or other tools familiar to data scientists like Python, R, and H2O.AI.

Before being presented to customers, the remaining actions go through a final stage known as the arbitration stage. This stage focuses on selecting and prioritizing the best actions that are most relevant for the customer at the present time. Arbitration balances customer relevance with business priorities using four components: **Propensity, Context Weighting, Business Value**, and **Business Levers**, each represented by numerical values.



Propensity is the predicted likelihood of positive behavior, such as the likelihood of a customer accepting an offer.

The propensity value is calculated by AI and it is the foundation of the arbitration process. The higher the likelihood of a customer accepting an offer, the higher the Propensity value for that offer. For Example, when a customer is eligible to receive a Standard Credit Card and clicks the Standard Credit card offer on the Bank's website the propensity of this action increases.



The AI is driven by adaptive models, self-learning mathematical models that use machine learning to make calculations. Adaptive models are a sub-group of predictive models that continuously adapt to new data as it becomes available. This means that the models are updated incrementally in real-time based on incoming data.

The Customer Decision Hub is configured to calculate the propensity for each treatment. For the web treatment, **Web_Click_Through_Rate** is the out-of-the-box adaptive model. It predicts the likelihood of a customer clicking on a banner, which is considered a positive behavior.

Context Weighting allows Pega Customer Decision Hub to consider the situational context for each action. For example, if a customer contacts the bank to close their account, the highest-priority action is to ensure that the customer is retained, even if they are eligible for other offers.

Business Value enables you to assign a financial value to an action and prioritize high-value actions over low-value ones. This value is typically normalized across Issues and Groups. For example, a 25-year mortgage is more profitable than a 15-year mortgage. So, in a situation where a customer is eligible for both plans, the 25-year mortgage will be ranked higher because of its higher business value.

Business Levers allow the business to assert some level of control over the prioritization of actions defined within the system. Levers are used to manually nudge Customer Decision Hub toward Next-Best-Actions based on external factors. For example, the recommended Next-Best-Action might be to offer a credit card to a customer when they visit the home page. However, to meet a business goal, the Mortgage Line of Business favors a mortgage offer, even if that offer is ranked a little lower on the list of possible actions.

Pega AI considers all four components to select the top offer for a customer. A simple formula $P * C * V * L$ is used to arrive at a prioritization value, which is used to select the top actions. The action with the highest priority will be presented to a customer.



In the current example, there are 5 actions that made it until the arbitration phase.

Assume the AI generates a propensity value for each of the actions which is a value between 0 and 1.

Each action has a context weight set, which is a percentage value between 0% and 100%, business value in dollars, and business lever value in a percentage between 0% and 100%.

Using the priority formula PCVL, with all of these values taken into account, we end up with a computed priority. Finally, the action with the highest priority is selected. In this case, it is a Standard Credit Card.

	PROPENSITY	CONTEXT	BUSINESS VALUE	BUSINESS LEVERS	PRIORITY
Standard Credit Card	0.45	5%	\$186	25%	5.23
Rewards Credit Card	0.3	5%	\$238	25%	4.46
15 years mortgage	0.25	2%	\$665	10%	3.6
25 years mortgage	0.2	2%	\$965	10%	4,25
Account for free	0.4	10%	\$56	5%	0.58

In summary, arbitration is used to select and prioritize the best actions that are most relevant for the customer at the present time. Arbitration balances customer relevance with business priorities using the formula $P * C * V * L$. The top actions are selected based on the result of this multiplication – the prioritization value.

Prioritizing actions with business levers

Introduction

Often, due to an internal ad-hoc priority, the business would like to boost the chance of certain actions being selected. To achieve this, they would like to present more relevant offers to customers based not only on their behavior but also on business priorities. Learn how to include business requirements in an action prioritization calculation to boost the chance of an action being selected.

Transcript

This video will show you how to include business requirements in an action prioritization calculation to boost the chances of an action being selected.

U+, a retail bank, noticed that one of the offers, the Rewards card offer, was not presented frequently enough due to its low propensity because customers ignored it during the initial launch.

For example, Troy, a customer, qualifies for two credit card offers – the Standard Card and the Rewards Card. When he logs in to the bank's website, he sees the top offer for him, Standard Card.

The screenshot displays a bank's account overview interface. At the top, there is a green header with the text "Account overview". Below this, the page is divided into several sections:

- Accounts:** This section contains three cards representing different account types:
 - Savings:** Represented by a green circular icon with a dollar sign, showing the account number ***1234.
 - Checking:** Represented by a green circular icon with a checkmark, showing the account number ***5678.
 - Credit:** Represented by a green circular icon with a card symbol, showing the account number ***7890.
- Credit card balance:** This section provides details for the credit card:
 - Current balance:** \$164.80, with a "Pay now" button below it.
 - Due date:** Dec 15, 2019, with a "View statement" link below it.
 - CardProtect:** Status is "On" (indicated by a checkmark icon), with a "Suspend CardProtect" link below it.
- Credit Card Offer:** A promotional banner on the right side of the page features a blue background with a white credit card icon and the text "0% APR and No Annual Fee" and "Get your U+ Bank card today". Below this banner, the text "Standard card" is displayed, followed by "0% APR and no annual fee" and a "Learn more" link.

Now, due to an internal ad-hoc priority, the bank wants to boost the chances of the Rewards Card being selected as the top offer. That is, the bank would like to present more relevant offers to customers based on not only their behavior but also on business priorities.

To implement this requirement, you must first enable the Action Weighting, a Business Lever, in Next-Best-Action Designer's Arbitration equation. This ensures that an action's business weight is used in the priority value calculation.

Business levers 0 levers defined

Upweight actions based on the business structure, and toggle all individual action levers ON/OFF.

Action weighting

Apply all of the action weighting, which is set on individual actions, to the prioritization formula.

In this case, the bank wants to boost the Rewards Card. So, open the Rewards Card offer. Edit the offer to set a business weight, a value in percentage, that is required to boost the offer. In this case, U+ wants to increase the changes of this action selected by 40%.

Starting propensity

Starting evidence

Business weighting ?

+

-

40

Business value

Save the offer for the changes to take effect.

Now, when Troy logs into the website, he will see that Rewards card is the top offer.

Account overview

Accounts

Savings
***1234

Checking
***5678

Credit
***7890

Credit card balance

Current balance	Due date	CardProtect
\$164.80	Dec 15, 2019	<input checked="" type="checkbox"/> On
Pay now	View statement	Suspend CardProtect

Rewards Card
Get 1% cash back on every purchase

Rewards card

Get 1% cash back on every purchase

[Learn more](#)

In Customer Profile Viewer, you can examine the detailed results of the total lever weight after your updates, for the customer Troy.

☐	Issue	Group	Name	Treatment	Channel	Journey	Stage	Final propensity	Total context weight	Value	Total lever weight	Priority	Rank	Results
☐	Grow	Creditcards	RewardsCard	Rewards card tile	Web			0.1548 ▬▬▬	1.00	1.0	1.4 ▬▬▬	0.22	1	PASSED
☐	Grow	Creditcards	StandardCard	Standard card tile	Web			0.1823 ▬▬▬	1.00	1.0	1.0 ▬▬▬	0.18	2	PASSED

First, view the next-best-action decision to learn more about the details of the actions for which Troy qualifies. For the current use case, the direction is **Inbound**, and the channel is the **Web**.

`TopOffers` is the real-time container service that manages communication between Customer Decision Hub and the website of the bank.

When you request a decision for Troy, the Customer Profile Viewer shows you the offers for which Troy is eligible.

In the Explain prioritization view, you can see how the final Priority is computed, based on 4 factors: Propensity, Context, Business Value, Business levers. In this specific example, you can see how the Total lever weight influences the ranking. Even though the Final Propensity is higher for the Standard Card, the Rewards Card is ranked 1. This is caused by the Total Lever Weight factor, which is now 1.4 for the Rewards Card because the lever for this card was increased by 40%.

This demo has concluded. What did it show you?

- How to include business requirements in action prioritization calculation.
- How to boost an action using business weight.

Business use case: Cross-sell on the web extended

Description

Next-Best-Action Designer guides you through the creation of a core Next-Best-Action strategy for your business. Learn how to customize the core strategy by creating decision strategies from scratch that extend Next-Best-Action Designer capabilities.

Learning objectives

- Explain when to consider creating decision strategies from scratch

Business use case: Cross-sell on the web extended

Introduction

This video describes several use cases where decision strategies are used to extend Next-Best-Action Designer capabilities.

Transcript

U+ is a retail bank. The bank is leveraging its website as a marketing channel to improve 1-to-1 customer engagement, drive sales, and deliver Next-Best-Actions in real-time.

The bank is using the Pega Customer Decision Hub™ to recommend more relevant banner ads to its customers when they visit their personal portal. In the start-up phase, U+ successfully implemented all their use cases using Next-Best-Action Designer.

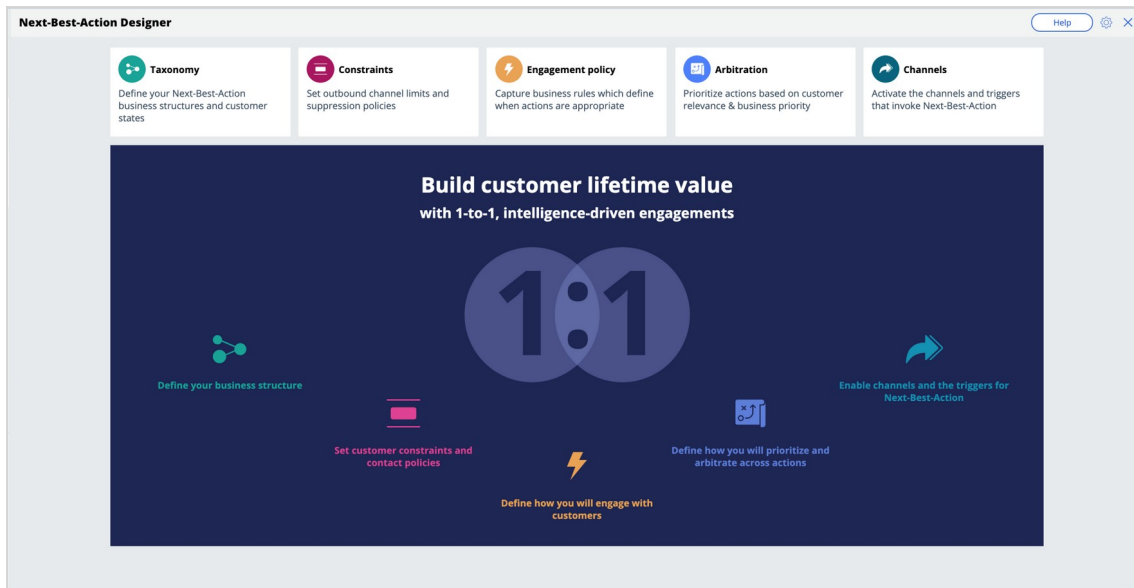
Next-Best-Action Designer guides you through the creation of a Next-Best-Action strategy for your business.

With the Next-Best-Action Designer user interface you define high-level business rules and AI controls, which the system uses to configure the underlying Next-Best-Action strategy framework.

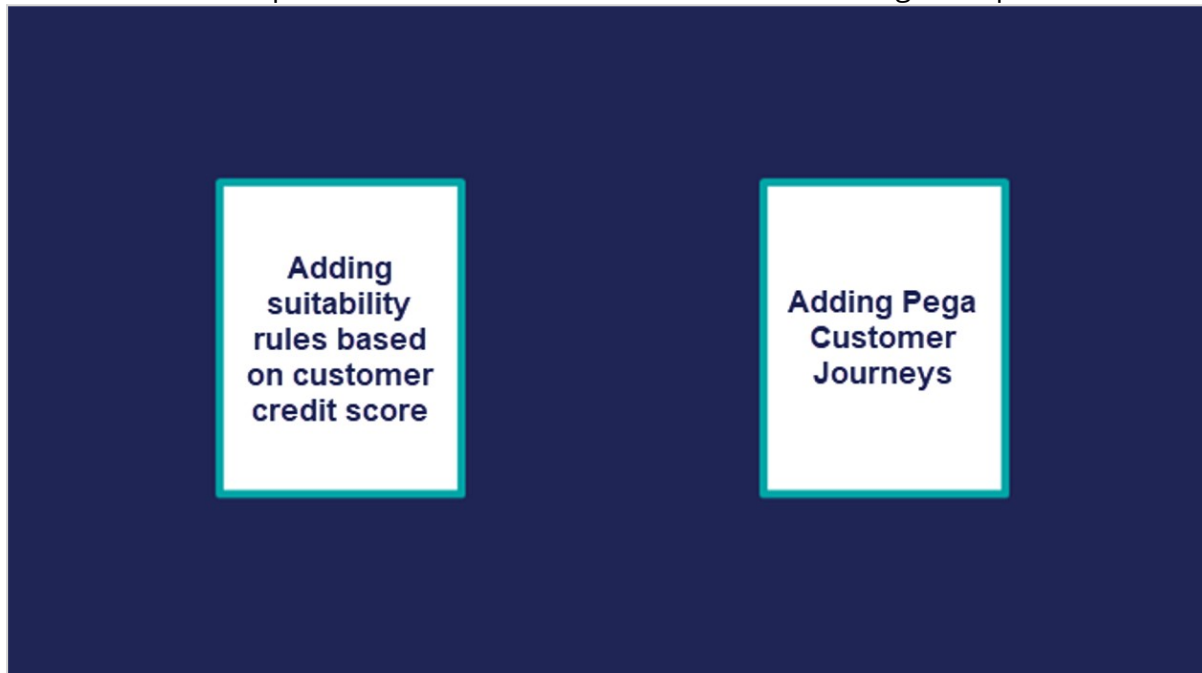
This framework leverages best practices to automatically generate Next-Best-Action decision strategies at the enterprise level.

These decision strategies are a combination of the business rules and AI models that form the core of the Pega Customer Decision Hub, which determines the personalized set of

Next-Best-Actions for each customer.



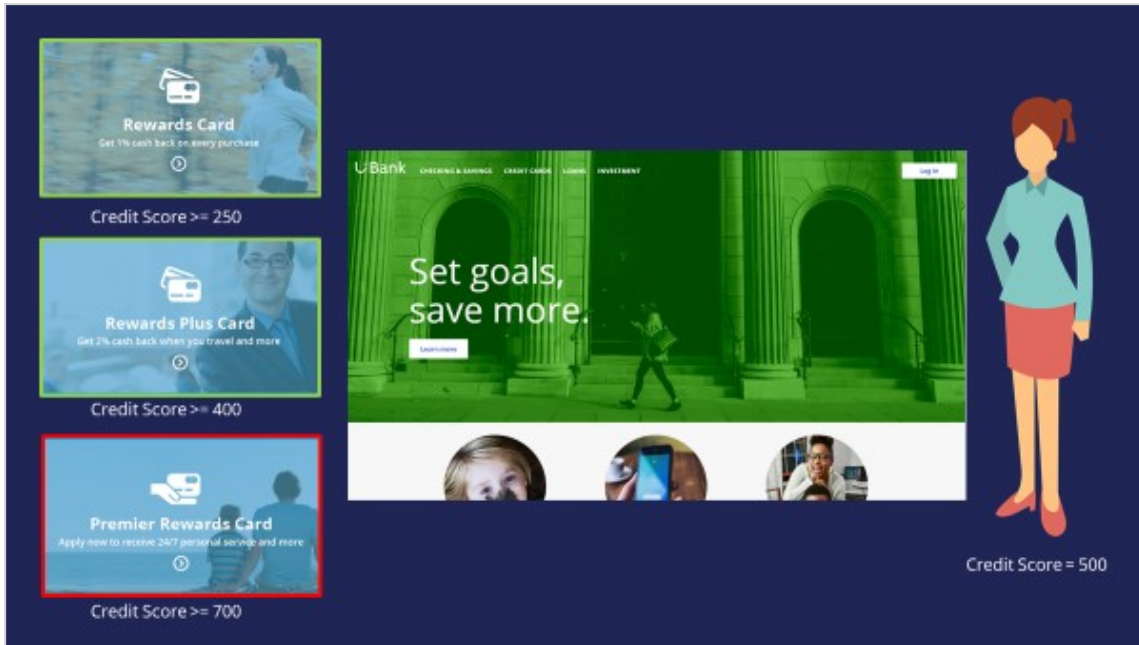
U+ Bank wants to implement additional use cases to meet new business requirements. These use cases require U+ bank to extend Next-Best-Action Designer capabilities.



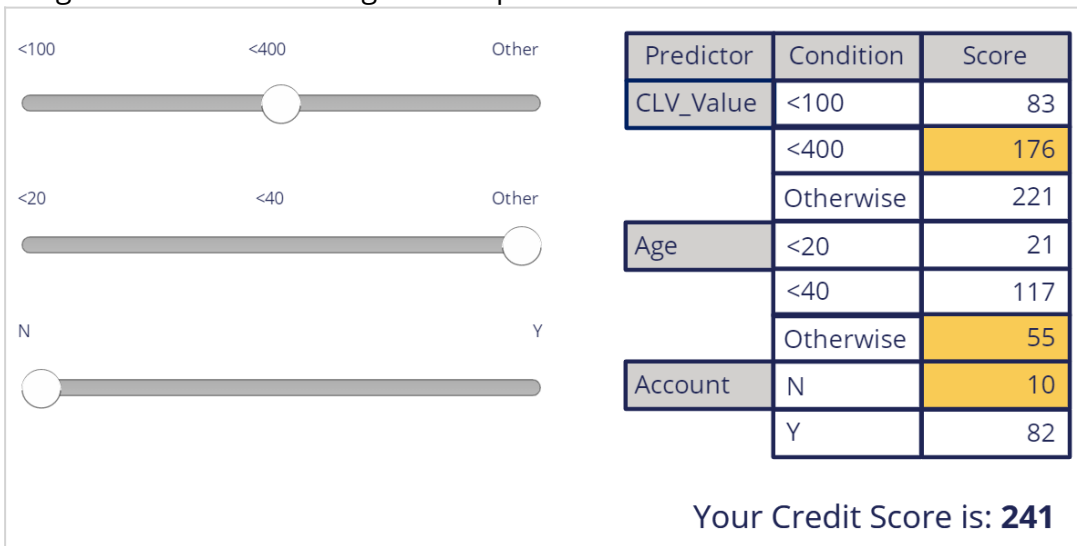
The first use case is to create suitability rules based on a customer's credit score. The bank wants to determine whether or not a customer is suitable for an offer based on their credit score. In this scenario, the credit score is not available, it needs to be computed in real-time based on customer profile information.

For example, when customer Barbara logs in, U+ bank only wants to present her with the Rewards and Rewards Plus offers, not the Premier Rewards offer. Barbara's credit score is

500. This makes her unsuitable for Premier Rewards, which is only suitable for customers with a credit score over 700.



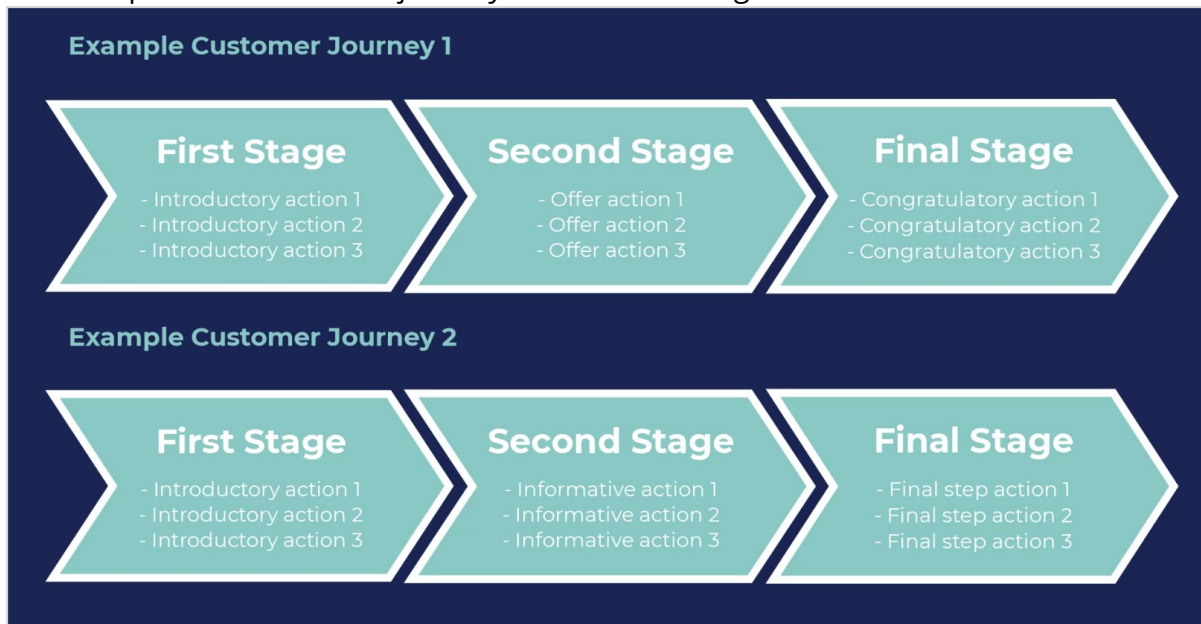
To determine suitability, the bank wants to calculate a customer's credit score using a scorecard. A scorecard is used to assign importance to pieces of data for use in a calculation. A scorecard uses a subset of customer property values divided into ranges and assigns scores to each range to compute a final score.



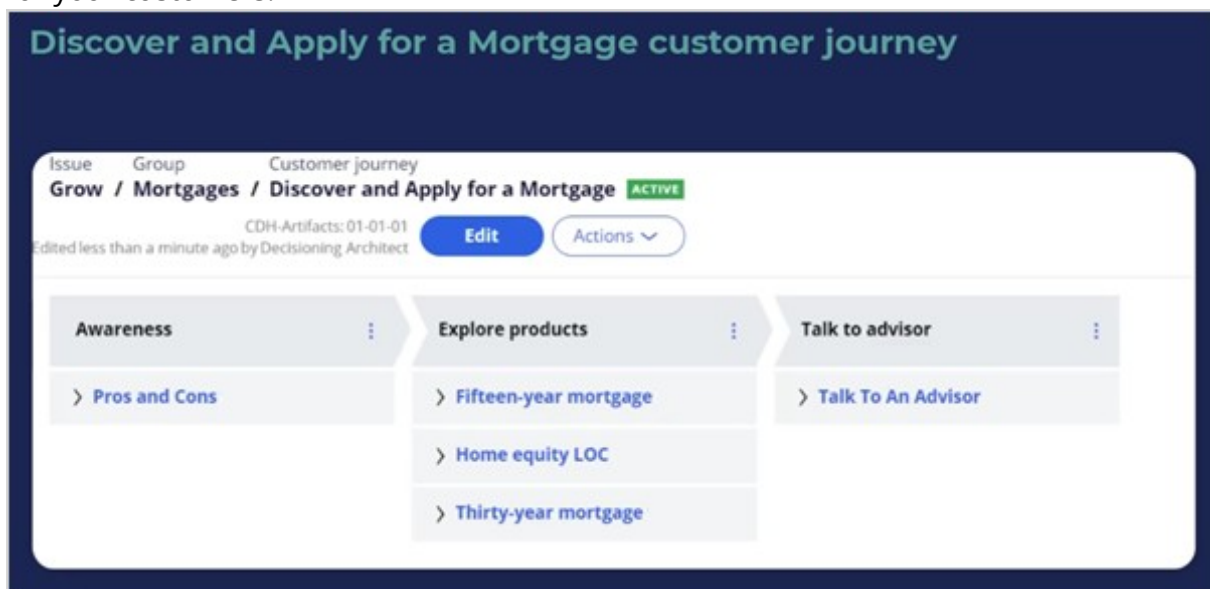
To implement this, you use a special Suitability condition in Next-Best-Action Designer. The Suitability condition uses a decision strategy that references a Scorecard rule. The Scorecard rule is used to determine the customer's credit score.

In the next use case, the bank wants to regulate and document the full customer experience. To implement this business requirement, you use Pega Customer Journeys. You

can set up various customer journeys divided into stages that contain a number of actions.



The customers can advance the journey by moving to the next stage when they meet the required entry criteria. Customer journeys provide a good mental model for managing content and influencing next-best-action decisions to present suitable actions seamlessly for your customers.



In summary, this video has shown you the various use cases U+ Bank would like to implement by extending Next-Best-Action Designer capabilities in this phase of the project.

Decision Strategies Overview

Description

Strategy canvas provides a guided and intuitive UI to bootstrap your application development with proven best practices that help you generate the strategies. These strategies can be customized using designated extension points or by building decision strategies from scratch, depending on the business requirement.

Learning objectives

- Describe how decision strategies are used
- Explain the decision strategy canvas and its building blocks
- Create decision strategies from scratch
- Explain what's going on inside each component when a decision strategy is executed

Creating decision strategies

Introduction

You can optimize decision points in a business process or in a case life cycle with the help of decision strategies. Decision strategies allow you to perform data-driven, real-time arbitration to select the best action for a given context. Learn the type of decision components and how they are used to create decision strategies. Gain hands-on experience designing and executing your own Next-Best-Action decision strategy.

Transcript

This demo will show you how to create a new decision strategy.

It will also describe three important decision components and the types of properties available for use in expressions during strategy building.

In this demo you will build a Next-Best-Label strategy. The Next-Best-Label strategy is a sample strategy, used to illustrate the mechanics of a decision strategy.

Start by creating a new strategy from scratch.

Decision strategies output actions, utilizing the so-called Strategy-Results class.

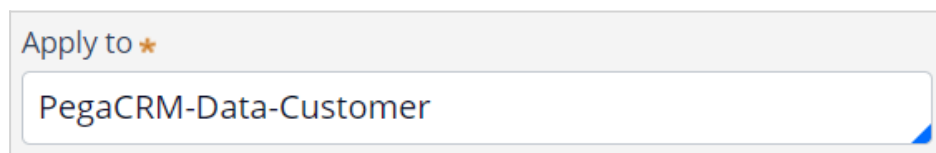
The Strategy-Results class limits the output of the strategy to the actions contained in the Business issue and Group.

The strategy you build will select a Label action from a set of predefined actions. The Label action selected will be the one with the lowest printing cost.

Notice that the complete definition of the Next-Best-Label strategy needs to include a reference to the PegaCRM-Data-Customer class.

This is the 'Apply to' class and it indicates the context of the strategy.

It ensures that from within the strategy, you have access to customer-related properties such as Age, Income, Address, Name, etc.

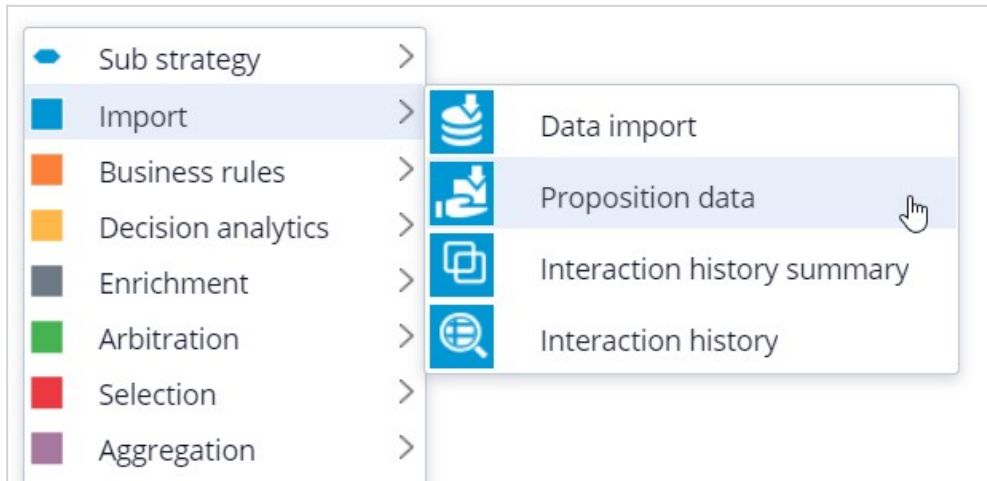


You can now start building the strategy. Right-click on the canvas to get the Context menu, which shows all component categories.

The first component to add is an Import component.

By expanding the Import category, you can see the Import component types available.

In this case you need a Proposition Data component to define the actions that will be considered by the strategy.



Now you need to configure the component. First, right-click to open the Proposition Data properties panel.

Notice that the Business issue and Group are grayed out.

This cannot be changed because the Enablement Business issue and Labels Group have already been selected for this decision strategy.

By default, the strategy will import all actions within that Group, unless you select a specific action.

For this component, you only want to import the Green Label, so let's select that.

Selecting the action from the drop-down menu automatically gives the component the appropriate name.

The description, which will appear under the component on the canvas, will also be generated automatically.

If you want to create your own description, you can do so by clicking the 'Use custom' radio button.

Now you want to import a second action into the strategy. You can use the Copy and Paste buttons to quickly add more Proposition Data components to the canvas.

You can use Alignment Snapping and Grid Snapping for easy placement of the components.

By turning these off, you can place a component anywhere on the canvas, but it makes it more difficult to align the shapes.



Now you need to add the next component in the strategy, which is an Enrichment component called Set Property.

You can add this component to the canvas by selecting it from the component menu.

Next, connect it to the Proposition Data components.

Ultimately, the result of this strategy should be the Label action with the lowest printing cost.

This printing cost is the sum of a base printing cost, which is specific to each label, and a variable cost, which depends on the number of letters.

The Set Property component is where you will calculate the printing cost for each of the actions.

The information in the 'Source components' tab is populated automatically by the Proposition Data components connected to this component.

Notice that the Black Label action is in the first row.

On the Target tab you can add properties for which values need to be calculated.

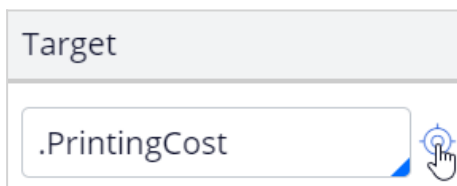
Click 'Add Item' to create the equation that will calculate the printing cost for each of the components.

Begin by setting the Target property to 'dot' PrintingCost.

In Pega, all inputs begin with a dot. This is called the dot-operator and it means that you are going to use a strategy property.

The PrintingCost property is a new strategy property that does not yet exist.

To create the new PrintingCost strategy property, click on the icon next to the Target field.



By default, the property type is Text. In Pega, there are various types supported. In this case, the PrintingCost is a numeric value, so change its type to Decimal.

Next, you need to make PrintingCost equal to the calculation you create. To create the calculation, click on the icon next to the Source field.

Using the Expression builder, you can create all sorts of complex calculations, but in this use case, the computation is very basic.

PrintingCost should equal $\text{BaseCost} + 5 * \text{LetterCount}$.

To access the BaseCost you type a dot. Notice that when you type the dot, a list of available and relevant strategy properties appears.

This not only makes it easy to quickly find the property names you're looking for; it also avoids spelling mistakes.

In a decision strategy, you have two categories of properties available to use in Expressions.

The first category contains the strategy properties, which can be one of two types.

An Action property is defined in the Action form. Examples are the BaseCost and LetterCount properties you are using here.

These properties have a value defined in the Action form and are available in the decision strategy via the Proposition Data component.

The property values can be overridden in the decision strategy but will often be used as read only.

The second type of strategy property is a calculation like the one you just created, PrintingCost. Such calculations are often created and set in the decision strategy.

These types of properties are either used as transient properties, for temporary calculations, or for additional information you want the strategy to output.

The second category contains properties from the strategy context, also called customer properties.

Suppose you want to use a customer property in your Expression, such as Age or Income.

In that case, you would have to type the prefix 'Customer dot', instead of just dot.

This is the list of available properties from the strategy context, also known as Customer properties.

For now, you calculate the printing cost for each action that does not use customer properties.

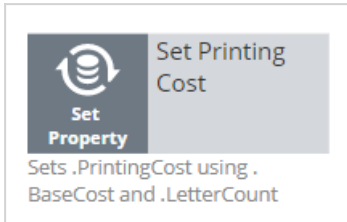
Finalize the Expression.



Even though you used the dot-operator to build your Expression, it's best practice to validate it, so click Test.

If the Expression isn't valid, you will receive an error message on screen.

On the canvas, you can see the automatically generated description for the component: Sets PrintingCost using BaseCost and LetterCount.



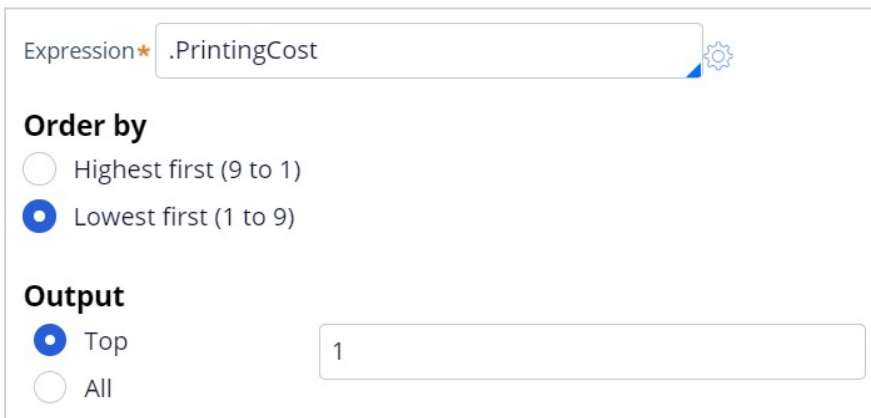
Now you want to ensure that the actions will be prioritized based on the lowest printing cost. So, you need to add the Prioritize component from the Arbitration category.

The prioritization can either be based on an existing property, or it can be based on an equation. Let's select an existing property using the dot construct.

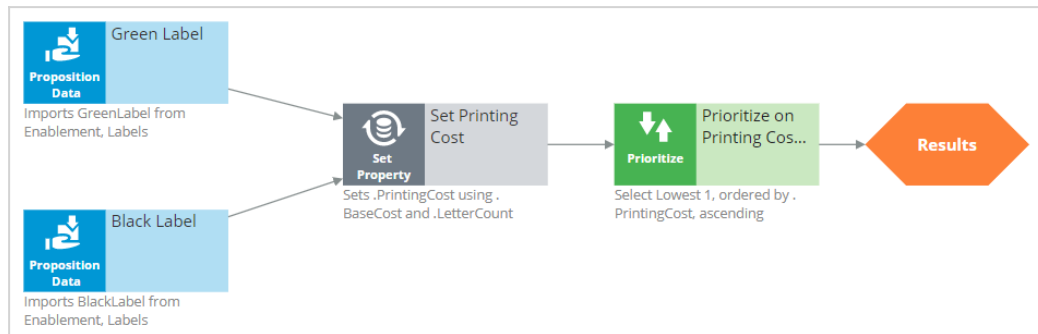
Here you can select the order in which the top actions are presented. Since you are interested in the lowest printing costs, configure it accordingly.

You can also select the number of actions that will be returned by the strategy.

If you want to output only one label, select Top 1 here.



Now you can connect the components and save the strategy.



To test the strategy, first check it out. Then, expand the right-hand side test panel and click 'Save & Run' to examine the results.

You can view results for any of the components by selecting that component.

If more than one action is present, each one is presented as a Page.

For the Set Property component, the Results contain a page for the Black Label and one for the Green Label.

For the Black Label the PrintingCost is 70.

For the Green Label the PrintingCost is 60.

On the canvas, you can show values for strategy properties such as Printing Cost.

For this exercise, you execute this strategy against a Data Transform called UseCase1.

If you open UseCase1, you can see the customer data the strategy uses when you run it.

To test the strategy on a different use case, you can create a Data Transform with different properties.

You can also select a Data Set that points to an actual live database table.

This demo has concluded. What did it show you?

- How to create a decision strategy from scratch.
- How to configure Proposition Data, Set Property and Prioritize decision components.
- How to build expressions in strategies.
- The two categories of properties available for expressions.
- How to test a decision strategy using a use case stored in a data transform.

Decision strategy execution

Introduction

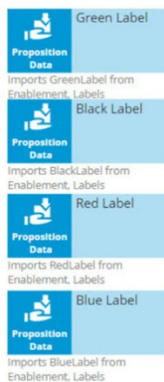
Using Pega Decision Management, you do not need to be an expert in programming, math or data science to design and execute sophisticated decision strategies that engage your customers throughout the customer journey. With its highly intuitive graphical canvas, Pega Decision Management enables you to easily embed Pega or third-party predictive models into your decision strategies. The result is customer-centric interactions that improve the customer experience while increasing customer value, retention and response rates.

Transcript

This demo explains what's going on inside each component when a Decision Strategy is executed.

For example, what happens 'under the covers' when a Filter component is executed, and how does it interact with the components around it?

In the interest of keeping it simple, this example is limited to four actions. In reality, decision strategies will involve many more actions than that.



Here are our 4 actions: 'Green Label', 'Black Label', 'Red Label' and 'Blue Label'; they are represented by a Data Import or, more specifically, a Proposition Data component.

In this example, the Proposition Data components import three data properties for each action: Name, BaseCost and LetterCount.

Green Label
Proposition Data
Imports GreenLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

Black Label
Proposition Data
Imports BlackLabel from Enablement, Labels

Red Label
Proposition Data
Imports RedLabel from Enablement, Labels

Blue Label
Proposition Data
Imports BlueLabel from Enablement, Labels

The first action's Name is Green Label, its BaseCost is 10, and its LetterCount is 10.

Likewise, the other actions have a Name, BaseCost and LetterCount.

Green Label
Proposition Data
Imports GreenLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

Black Label
Proposition Data
Imports BlackLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Black Label	20	10

Red Label
Proposition Data
Imports RedLabel from Enablement, Labels

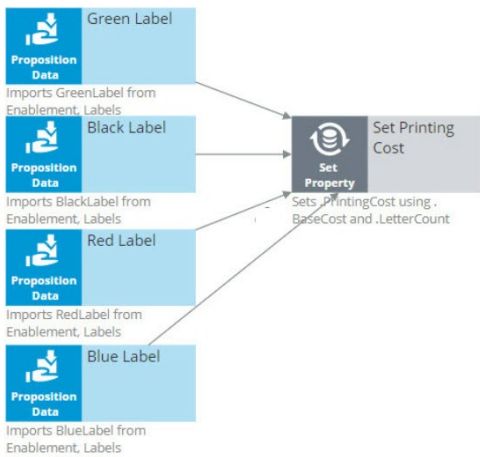
Rank	Name	BaseCost	LetterCount
1	Red Label	30	8

Blue Label
Proposition Data
Imports BlueLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Blue Label	40	9

One property is automatically populated for you; this is the Rank. We will come back to this later, but notice that, as separate components, each action has a Rank of 1.

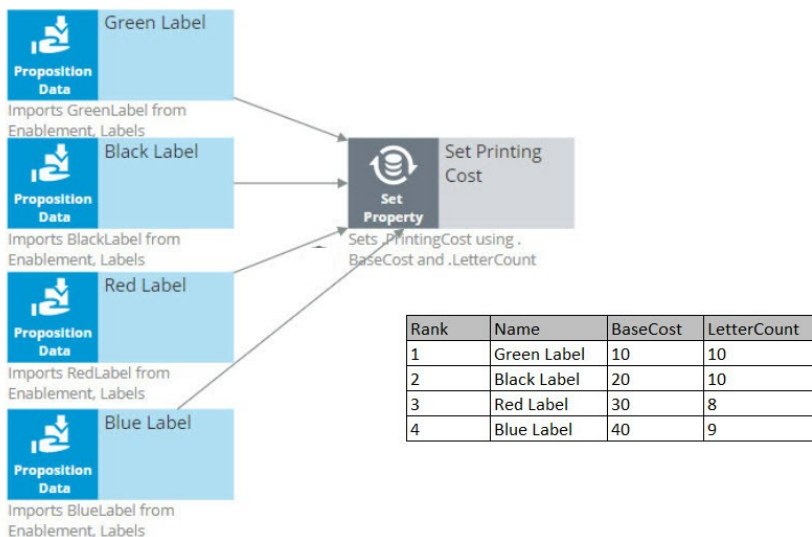
On the strategy canvas, components are connected by drawing arrows from component to component. So, what do these arrows mean exactly?



Well, when you draw an arrow, what happens is that, at runtime, all information in the component you're drawing the arrow from is available as a data source to the component you're drawing the arrow to.

So now, the Name, BaseCost and LetterCount for all of the actions are available in a single Set Property component.

The only data element that changes is the row number, or as we call it in the strategies, the Rank. In each decision component, the Rank value is automatically computed.



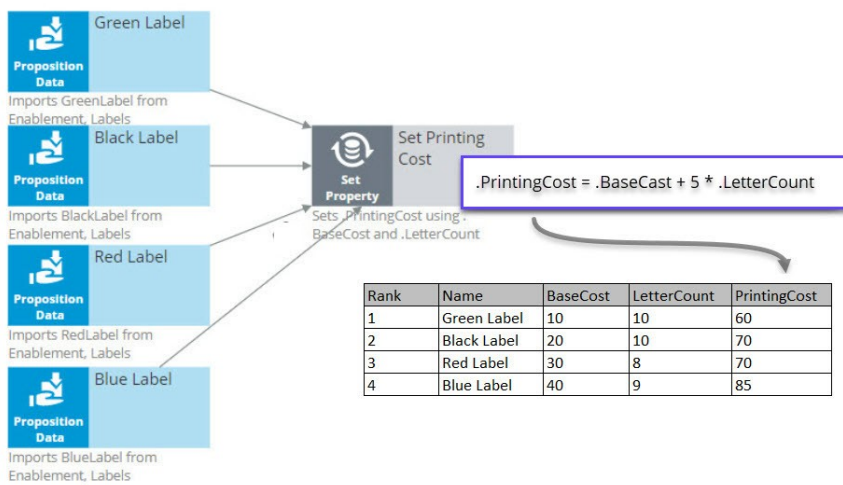
In the Set Property component, the Rank is determined by the order in which the actions are received by the component.

As a result, in this instance, the Green Label action has a Rank of 1, Black has a Rank of 2, Red has a Rank of 3, and Blue has a Rank of 4.

Ultimately, you want to select the best Label action. That is the Label with the lowest printing cost.

The printing cost of a Label is the sum of the BaseCost and a variable cost based on the LetterCount.

You configure the Set Property component to compute the printing cost of each Label action.



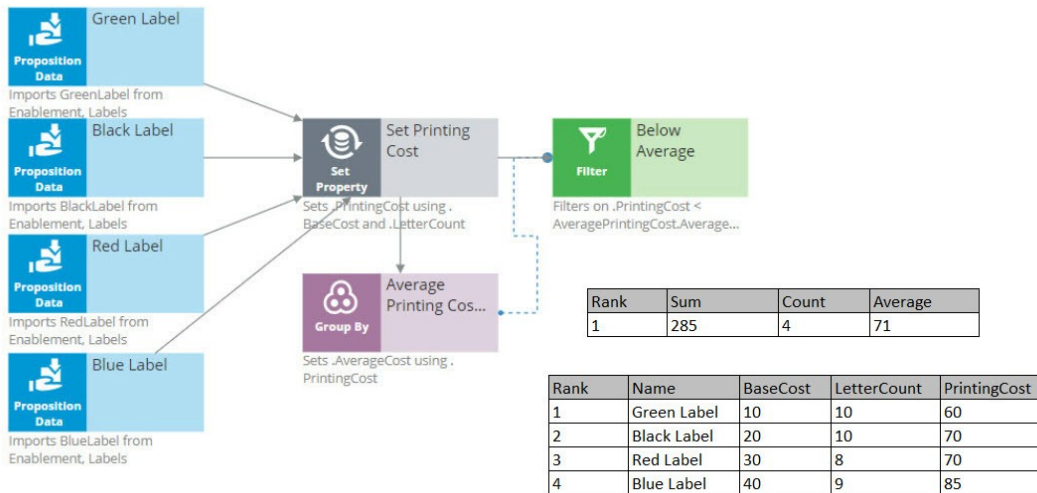
Because we are combining the data in our four Proposition Data components into one Set Property component, we only need to add one PrintingCost property to the new component, and it automatically computes the printing cost for all four actions.

For the Green Label action, PrintingCost equals a BaseCost of 10 plus 5 times the LetterCount of 10 which equals 60.

Similarly, the PrintingCost for the Black and Red Label actions is 70, and for the Blue Label action is 85.

Now, let's say the business rule is to select only Label actions with a printing cost lower than the average printing cost of all labels. For this requirement we use a 'Group by//Filter' component combination.

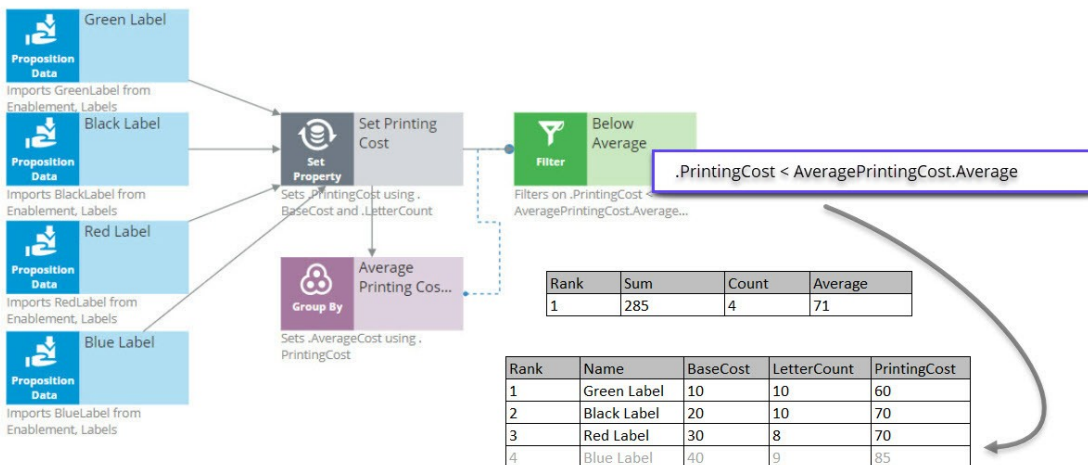
A 'Group by' component offers essential aggregation capabilities, like Sum and Count, that are used in many decision strategies. We will use it to calculate the average printing cost.



Again, we have our set of actions, each with their own specific PrintingCost value. The 'Group by' component combines all actions into one row. How does that work?

Well, it sums the PrintingCost values for all the actions, it counts the actions, and it calculates the average printing cost by dividing the summed printing cost by the count.

In this example, the sum of the PrintingCost values is 285, and the count of the actions is 4, so the average printing cost is 71.



Now that you have calculated the average printing price using a 'Group by' component, configure the Filter component to filter out actions that have a printing cost equal to or higher than this average.

So far in this strategy, we've seen only the solid line arrows, which copy information from one component to another. But now we also see a dotted line arrow.

This tells us that a component refers to information in another component.

Here, the Filter component is referencing the average printing cost that exists inside the Aggregation component. This is an important capability to understand.

The Filter component filters out actions when the printing cost for that action is equal to or above the average printing cost and propagates the other actions.

First, via the solid arrow, the filter looks at the actions sourced from the Set Property component.

Then, it applies the filter condition, which references the average printing cost in the 'Group by' component via the dotted arrow.

The Filter Condition in the Filter component is the Expression: 'dot PrintingCost is smaller than AveragePrintingCost dot Average'.

By using this ComponentName dot Property construct, any decision component can be referenced by any other component by name.

Important to note that the Filter component lets actions through when the condition Expression evaluates to **true** and filters out actions when the condition Expression is not met.

When you refer to a component, you always refer to the first element in the component, the one with Rank 1.

In this case, you are referring to the one and only row in the 'Group by' component, which naturally has Rank 1.

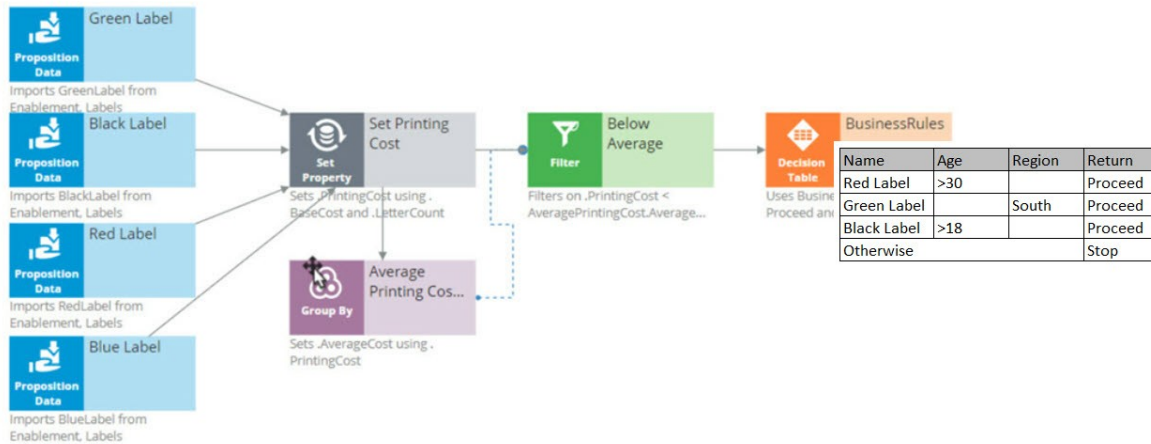
The Rank 1 average equals 71 in the 'Group by' component. This means that the filter will allow Label actions through that have a printing cost lower than 71.

By this standard, the printing cost of the Blue Label action is too high, so it is filtered out. The printing cost of the other Label actions are below 71, so they survive.

The result is that the table contains three surviving actions: Green Label with Rank 1, Black Label with Rank 2, and Red Label with Rank 3.

The next component is a Decision Table. A Decision Table in Pega is an artifact that can be used to implement business requirements in table format.

In a Decision Table, the business rules are represented by a set of conditions and a set of Return values.



The Decision Table receives information about the remaining actions via the solid arrow from the Filter component.

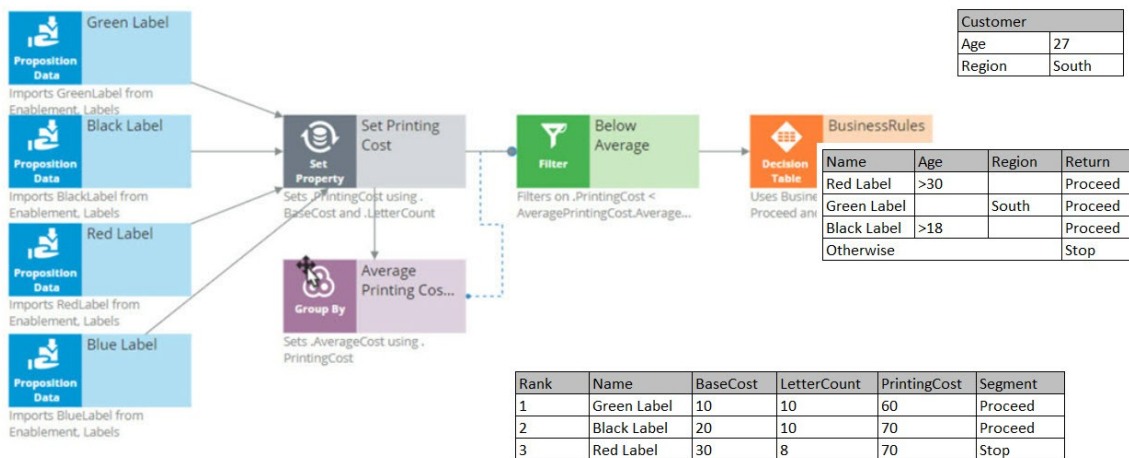
The business criteria say that the Red Label action can be offered if the customer's age is over 30 and they are from any region. If these criteria are met, the Return value is 'Proceed'.

The Decision Table also says that the Green Label action can be offered to anyone in the Southern region. So, if the Region value is South, the Return value for Green is 'Proceed'.

The Black Label action can be offered to anyone over the age of 18.

But in all other cases, or, Otherwise, no Label action meets the criteria, and the Return value is 'Stop'.

As an example, consider a customer with Age 27 and Region South.



Now, the Decision Table applies the business criteria for each action against the customer information and returns a value. The value returned by a Decision Table is also called a Segment.

The Decision Table checks the Green Label action with Rank 1 first, and in this case, it can proceed because the customer's Region is South.

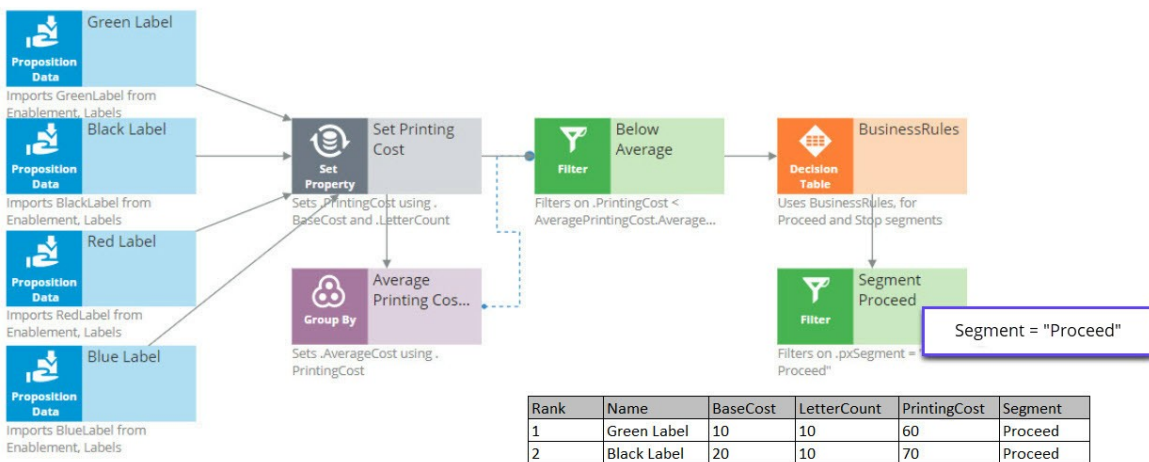
Next, it looks at the Black action and sees that the criteria for Black is that the customer's age is greater than 18. This customer is 27.

Black doesn't care about the Region, so the Segment value for the Black action is 'Proceed'.

Finally, it looks at the Red action, and the Age criteria don't match up, so the Segment value for Red is 'Stop'.

The result of the component is that you get a new segmentation column that flags which of the actions comply with the business rules.

You're now going to filter out the actions that do not match the business rules. This happens in the 'Segment Proceed' Filter component.



Again, via the solid arrow, the strategy copies the data over from the Decision Table component into the Filter component.

Now each action has a Rank, Name, BaseCost, LetterCount, PrintingCost and Segment. The filter condition is applied to this data.

The filter condition says: allow this action through if the Segment value equals 'Proceed'.

What this Filter component now does is go through the list of actions to find the actions with value 'Proceed' in their Segment property.

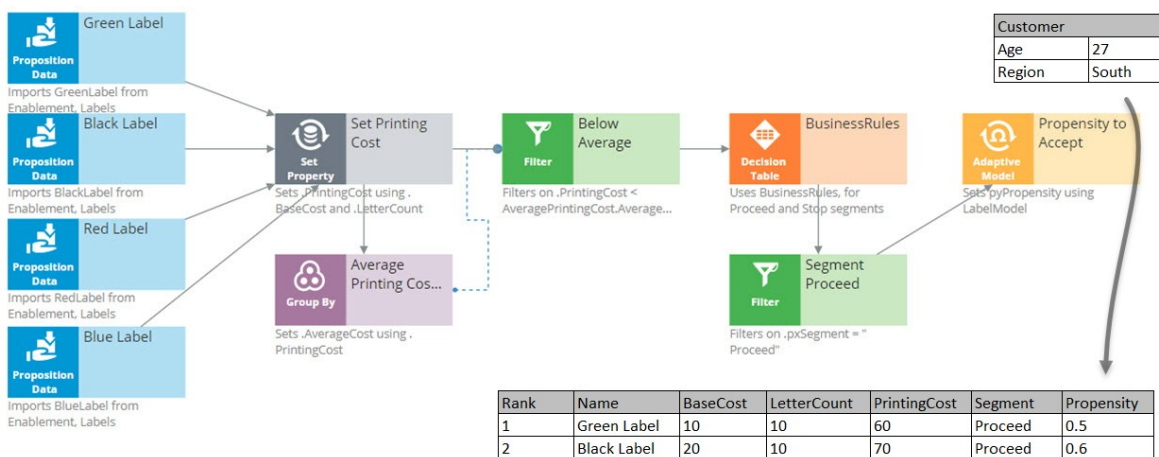
First is the Green Label. Green is allowed through, which means its properties will be available in the new component.

Then the Black Label. It is also allowed through because it also has 'Proceed' in its Segment property.

But the Red Label action is not allowed through, because Red has 'Stop' in its Segment property. Therefore, Red is not part of the output.

The strategy so far has selected two of our original actions, Green and Black.

Now, in the Adaptive Model component, you will use predictive analytics to determine the propensity of each of the remaining actions.



Propensity is the probability that a customer will accept an action, or their likelihood of interest in it.

In order to calculate the propensity, we use an Adaptive Model component. The referenced model is configured to monitor customer characteristics such as Age and Region.

In this case our test customer has an Age of 27 and is from the South Region.

Again, just to keep it simple, we are using a model that makes predictions based on only this information. In reality, models will take into account many more properties.

The Adaptive Model determines the propensity.

First, we supply the action and the customer profile to the Adaptive Model, and the model says: 'Oh, it's the Green Label action; we have some evidence that young people like the Green Label action, but people from the South don't like it.'

Combining both factors, we get an overall propensity of 0.5 for the Green Label action.

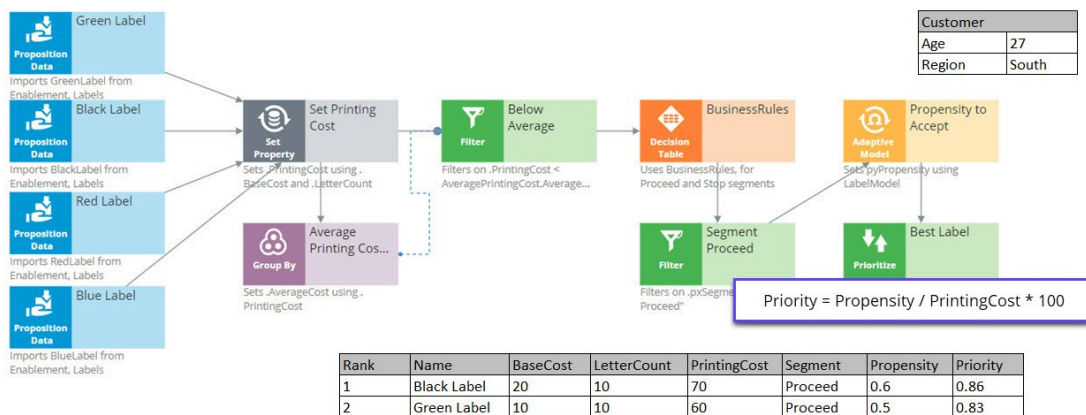
For the Black Label action, the likelihood turns out to be 0.6.

After consulting the Adaptive Model, the Propensity to Accept component sets the Propensity property value for each action.

Remember, the propensity is always a number between zero and 1.

It shows something along the lines of, half of the customers that are like this customer accepted the Green Label action in the past, and 3 out of 5 customers like this customer accepted the Black action last month.

The next component in our chain, called Best Label, is the Prioritize component. This component determines the priority of each action and ranks them. Let's see how this works.



A key element of this component is the priority Expression, which calculates a priority value for each action. According to this Expression, the higher the value, the higher the priority and rank.

In this case, the priority calculation weighs likelihood of acceptance in its equation: 'Propensity divided by PrintingCost times 100'.

When performing this calculation on the Black Label action, we can see that it has a PrintingCost of 70 and a Propensity of 0.6, therefore its Priority is 0.86.

The Green Label action has a lower PrintingCost and a lower Propensity, resulting in a Priority of 0.83.

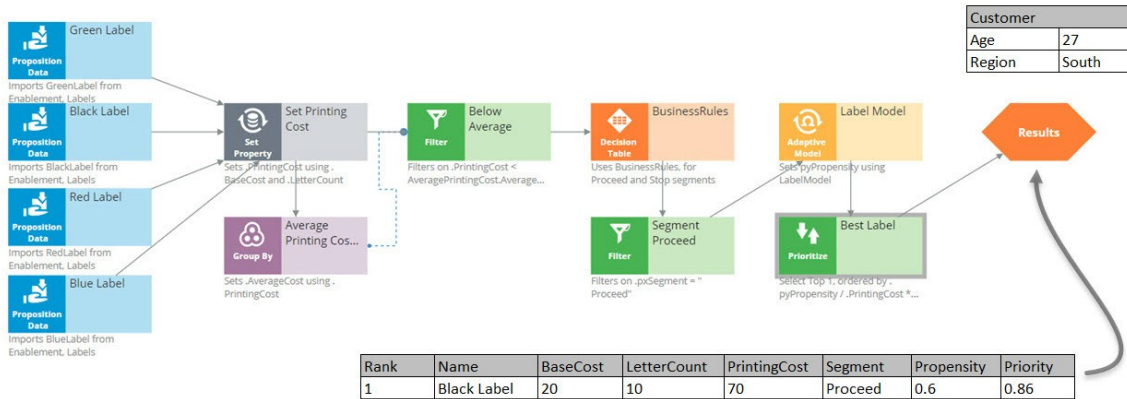
Because 0.86 is higher than 0.83, the Black Label action is now ranked number one.

So, even though the printing cost of the Black Label action is higher than that of the Green Label action, the Black Label action still comes out on top.

In this case, the Priority component reversed the Ranks of the two actions. Black is now the primary action and Green is the secondary action.

The same Prioritization component is also configured to output only the top action.

Therefore, it filters out the Green action altogether, and at the end of our strategy chain, the Black Label is left as our best action.



Creating engagement strategies using customer credit score

Description

A scorecard is a mathematical model that determines a score value based on a set of conditions and a combiner function. The conditions either use customer properties directly, or an expression based on them. The output of a scorecard is the raw score plus a segment, which is obtained by defining a set of cut off values that create a set of score ranges. Learn how a scorecard can be used to determine the customer credit score, and how the credit score can be used in a decision strategy to define suitability rules.

Learning Objectives

- Create a scorecard
- Use the segmentation result and the score value of a scorecard in a decision strategy
- Use a scorecard to determine group-level and action-level suitability

Customer credit score using a scorecard

Introduction

A scorecard is a mathematical model that determines a score value based on a set of conditions and a combiner function. These conditions can either use customer properties directly, or an expression based on them. The output of a scorecard is the raw score and a segment, which is obtained by defining a set of cut-off values that create a set of score ranges.

Transcript

This video explains how a scorecard can be used to determine the customer credit score.

Let's consider a typical loan application scenario for U+ Bank.

The bank wants to automate the home loan requests process by analyzing a customer's credit score based on customer profile information and categorizing it into the right credit score level using a scorecard.

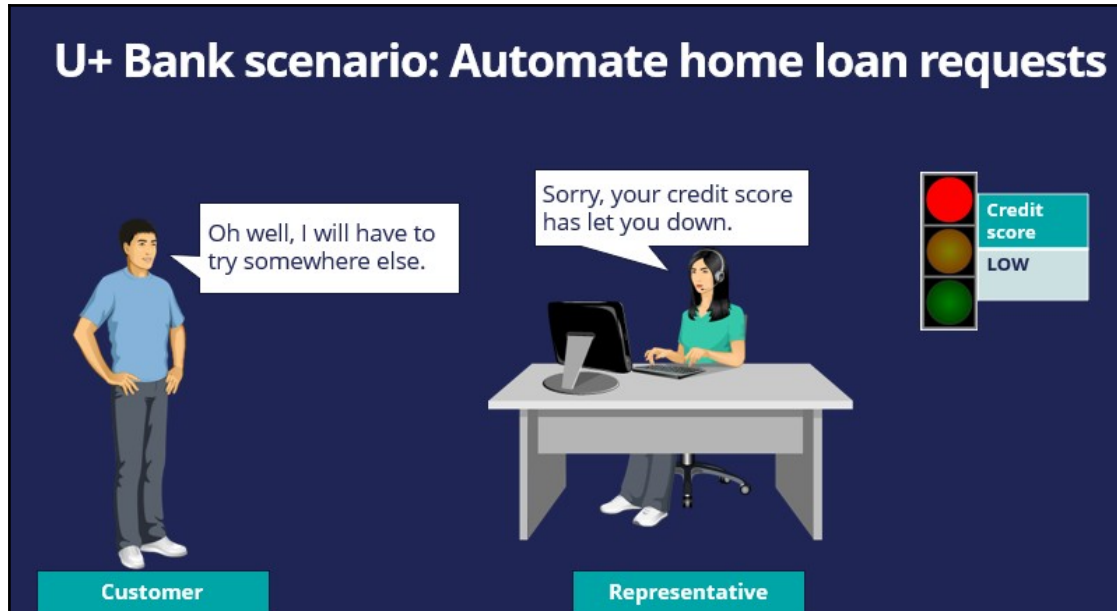
Tom is a U+ Bank customer who visits the bank to apply for a home loan. Iris is a U+ bank loan representative who is responsible for processing loan requests.

Iris gets Tom's profile information. Based on his customer data, Iris calculates his credit score using a scorecard.

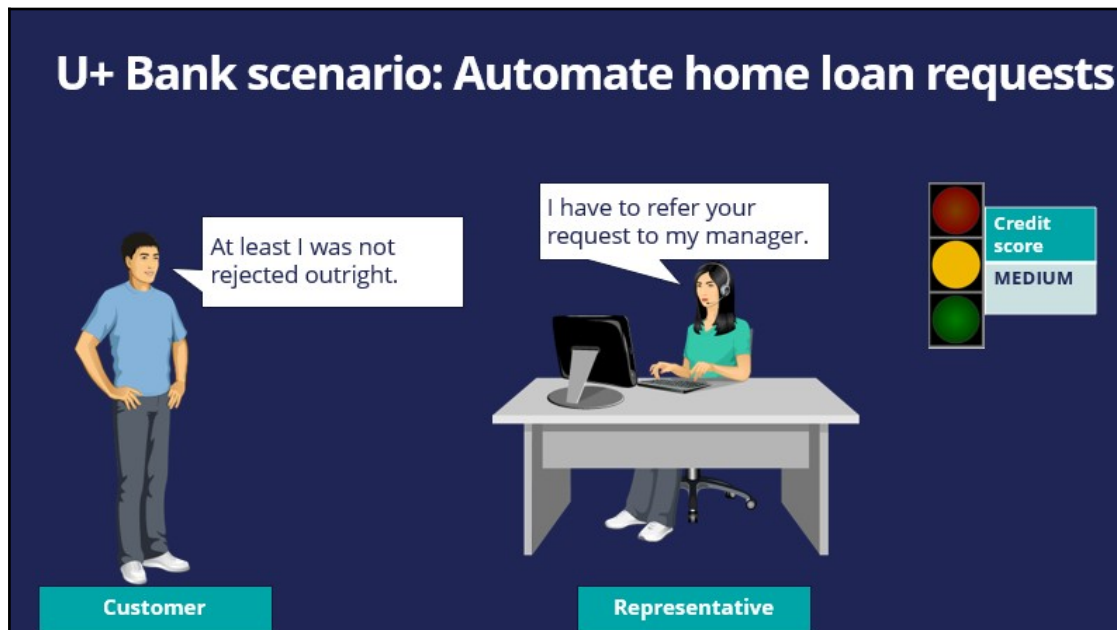
Depending on Tom's credit score, a decision is made on his loan application.

The bank has already defined three credit score levels: **Low**, **Medium** and **High**.

For example, if Tom's credit score is **LOW**, he is not qualified to get a loan, and his application is automatically refused.

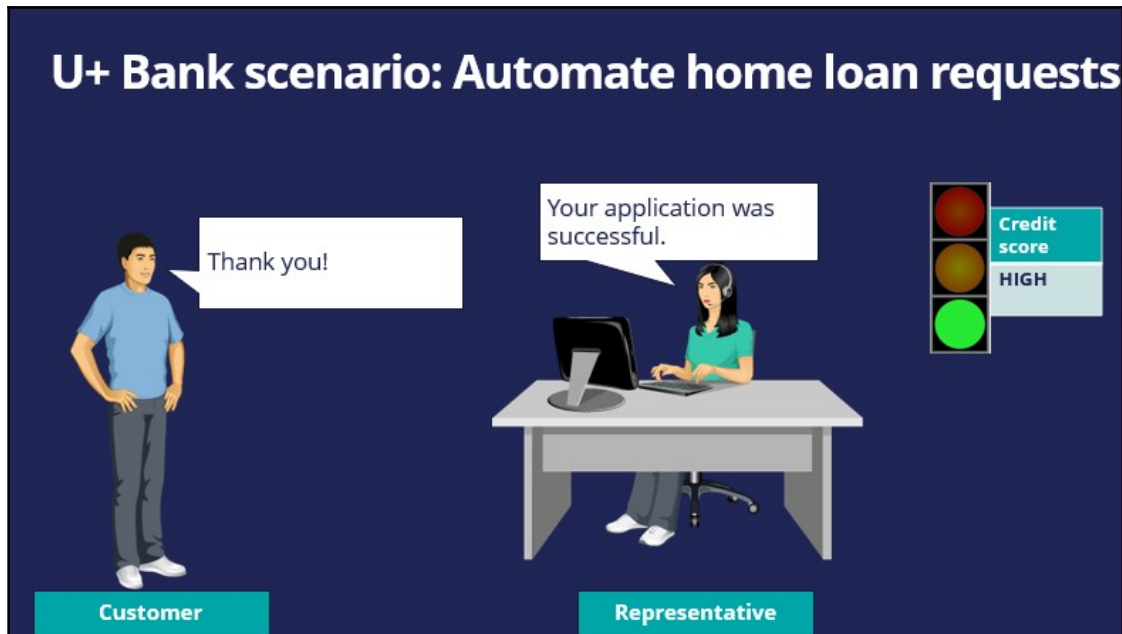


If Tom's credit score is **MEDIUM**, then his application needs to be reviewed further by manager.

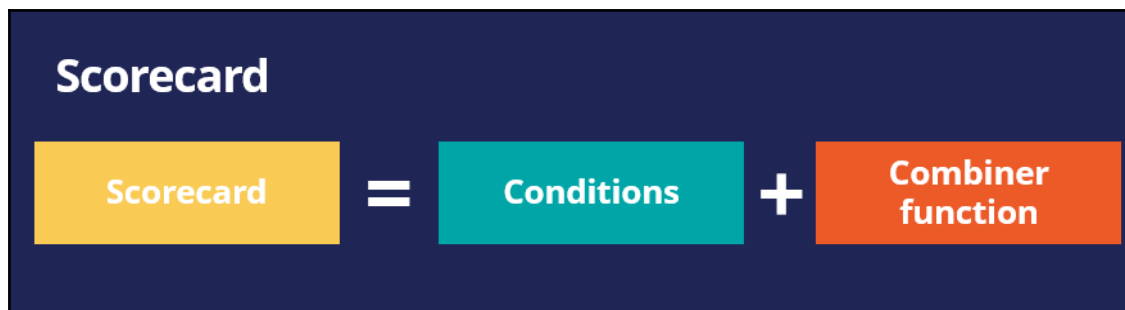


If Tom's credit score is **HIGH**, then he is automatically eligible to get the loan right away.

Let's now learn about the scorecard and how it works by considering the U+ Bank loan application scenario.



A scorecard is a mathematical model that computes a score and outputs a segmentation result based on one or more conditions and a combiner function.



Let's take a look at an example of a scorecard U+ bank could use. Assume that in order to process home loan requests, the bank has identified **three predictors** that can be used in the scorecard calculation. The predictors are **Customer lifetime value, Age, and Account**.

Each predictor has some **conditions** and scores associated with those conditions.

The scorecard enables the bank to assign a **score** to a value or a range of values the predictor can have.

For example, If the **Customer lifetime value** is less than 100, a score of 83 is assigned. If the value is between 100 and 399, a score of 221 is assigned; otherwise a score of 350 is assigned.

If the customers' Age is less than 20, a score of 21 is assigned. If their Age is between 20 and 39, a score of 217 is assigned; otherwise a score of 55 is assigned.

In a similar way, if a customer has an account with U+ bank, a score of 82 is assigned. Otherwise, a score of 10 is assigned.

The combiner function is used to combine the total sum of score values between conditions.

Now that the bank has identified **predictors** and assigned a **score** to a range of values the predictor has, let's see the results of the scorecard.

The output of a scorecard is a **score** and a **segment result**.

First, you will see how to calculate a customer's credit score using the scorecard.

Let's consider an example in which the customer lifetime value is 350, the customer's age is 35, and the customer has a U+ Bank account.

Since the customer lifetime value is 350, he gets a score of 221. For his age, he gets a score of 217, and since he has a U+ bank account, he gets a score of 82. The sum of these three individual scores is 520, representing his final credit score.

The scorecard outputs a **score** as well as a **segment result**.

To process loan requests, the bank has defined three result values. If the credit score is less than or equal to 400, the result is **Not Approved**. If the credit score value is between 401 and 600, the result is **Refer to Manager**. If the credit score is higher than 600, the result is **Approved**. So, the higher a customer's credit score, the better his/her chances of getting a loan approved.

Credit score	Result
< = 400	Not Approved
< = 600	Refer to Manager
Otherwise	Approved

Let's consider the profiles of three customers applying for home loans.

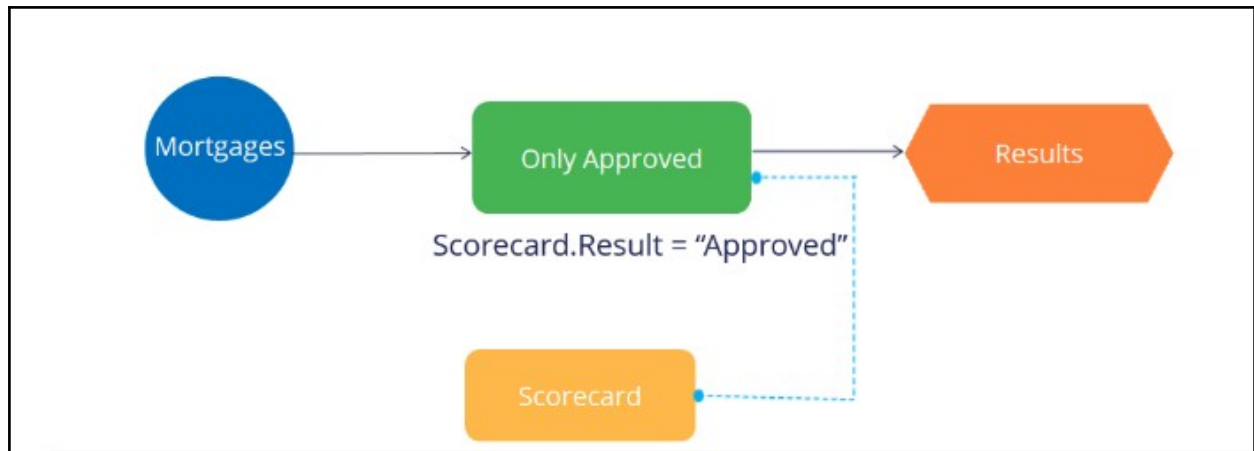
Customer A has a customer lifetime value of 150, their age is 18, and they don't have a U+ Bank account. Therefore, their credit score is 252, and the segment result is **Not Approved**, because their credit score value is less than 400. This means they do not qualify for a loan.

Customer B has a different profile, and their credit score is 415, so the segment result is **Refer to Manager**, because their credit score value is between 401 and 600. This means their application needs further review.

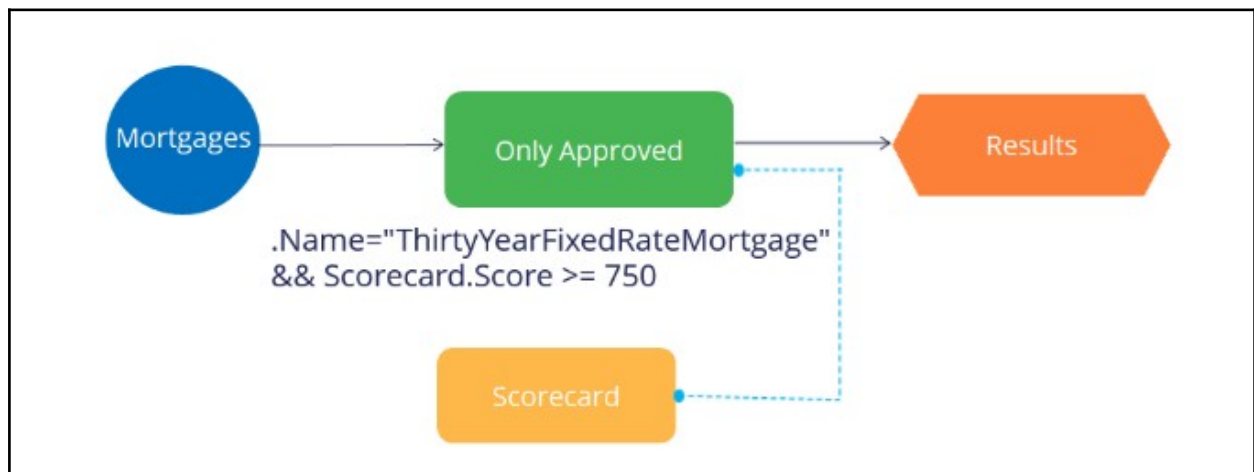
Customer C has a credit score of 649, which is higher than 600, so he is eligible to get a loan right away.

Now that you learned about the scorecard and its results, let's see how to use the scorecard segmentation in a decision strategy.

To use a Scorecard in a decision strategy, use the **Scorecard** component. The **Scorecard** component outputs the result, in this case **Not Approved**, **Refer to Manager**, or **Approved**, which you can use in your decision strategy.



The scorecard also outputs the **raw score value**, this score value can be used directly in the decision strategy.



In summary, the scorecard is a mathematical model that computes a score and outputs a segmentation result based on a set of conditions and a combiner function. The scorecard's score and segmentation result can be used directly in a decision strategy using the Scorecard decision component.

Building a scorecard to calculate the creditscore

Introduction

Scorecard rules are referenced in decision strategies through the scorecard component. Learn how to create scorecard-specific rules to derive decision results from several factors.

Get detailed insight into how scores are calculated by testing scorecard logic using the rule form. Use the test results to view score explanations for all predictors used in the calculation so you can validate and refine the current scorecard design or troubleshoot potential issues.

Transcript

U+ bank wants to build a scorecard that calculates the credit score of a customer to determine if the customer is suitable for a mortgage or not.

The bank has identified three customer properties to use in the scorecard calculation. These are the HasCards (a property that indicates if a customer already has a credit card or not), Income, and Age.

To build a scorecard, navigate to the scorecards landing page and create a scorecard.

Enter a short description for the scorecard.

The Combiner function enables you to select a method for combining scores.



In this scenario, the credit score is the sum of scores attributed to each customer property, so use the Sum method.

The scorecard enables you to assign a score to a value or a range of values the property can have.

For example, the credit card indicator, HasCards property can have the values "Y" or "N".

If a customer has a credit card with U+ bank, assign a score of 100. Otherwise, assign a score of 0.

The screenshot shows a configuration interface for a scorecard rule. It has four columns: Predictor expression, Condition, Score, and Weight. The Predictor expression is ".HasCards". The Condition is "Y". The Score is "100". The Weight is "1". Below this, there is an "Otherwise" section with a score of "0".

Predictor expression*	Condition	Score	Weight*
.HasCards	Y	100	1
	Otherwise	0	

You may also sub-divide values of numeric properties into ranges and assign a score to each range.

The next property is Income. In this case you want to split the values for yearly income into three ranges and assign a score to each range.

The data model contains an Income property, which contains the monthly income of the customer. The scorecard allows you to use this property to build the expression to compute the customer's yearly income.

Once the expression is created, you can start assigning scores to each range. If the customer's yearly income is less than or equal to 25,000, assign a score of 50.

If their yearly income is between 25,001 and 50,000, assign a score of 100; and if their income is between 50,001 and 75,000, assign a score of 150.

If their income is higher than 75,000, assign a score of 200, using the **Otherwise** setting.

The screenshot shows a configuration interface for a scorecard rule. The Predictor expression is ".Income*12". There are three conditions: "<= 25000" with a score of "50", "<= 50000" with a score of "100", and "<= 75000" with a score of "150". Below these, there is an "Otherwise" section with a score of "200".

Predictor expression*	Condition	Score	Weight*
.Income*12	<= 25000	50	1
	<= 50000	100	
	<= 75000	150	
	Otherwise	200	

In a similar way, split the values for Age into five ranges and assign a score to each range. The higher the range, the higher the score you assign.

Once the Scorecard is defined, you can define the results of the scorecard calculation.

When you refresh, the scorecard calculates the Minimum and Maximum scores.

You may define two or more Result values based on the score. In this scenario, you want the scorecard to return a Result value of 'Not Suitable' if the score is less than 250. Otherwise, the Result value is 'Suitable'.

Result*	Cutoff value	Interval
Not Suitable	250	< 250
Suitable	Otherwise	>= 250

Save the configuration.

Now, run the scorecard and verify the results.

You can either fill in the property values, or you can test the result for a predefined test case using a data transform. For example, select the customer Troy.

The end result for Troy is that he is Not Suitable for a mortgage, as his credit score is 200, which is lower than the set threshold.

Execution results		
Result	Score	Combiner function
Not Suitable	200.0	SUM
Interval	Minimum score	Maximum score
< 250	50.0	500.0

You can see the **Execution details** for Troy, which show how his credit score is computed. Since he has no cards, he gets a score of 0; for his income, he gets a score of 150; and for his age, he gets a score of 50. That's 200 in total.

Execution details							
Predictor expression	Value	Operator	Condition	Score	Weight	Points	Lost points
.HasCards	N		Otherwise	0	1	0	100.0
.Income*12	54000.0	<=	75000	150	1	150	50.0
.Age	26.0	<=	35	50	1	50	150.0

Now, test the results for customer Robert.

Robert is suitable for a mortgage, as his credit score is 250.

Execution results		
Result	Score	Combiner function
Suitable	250.0	SUM
Interval	Minimum score	Maximum score
>= 250	50.0	500.0

The newly created scorecard is now available on the Scorecards landing page.

This demo has concluded. What did it show you?

- How to create a scorecard.
- How to assign a score to categorical and numerical property values.
- How to use expressions in scorecards.
- How to define scorecard outputs.
- How to test scorecards.

Creating an engagement strategy

This demo will show you how to build a decision strategy that can be used in Next-Best-Action Designer to implement more complex Engagement Policy rules.

Engagement Policy rules are business rules that describe which Actions are appropriate for a customer, expressed in the form of either Eligibility, Applicability, or Suitability rules.

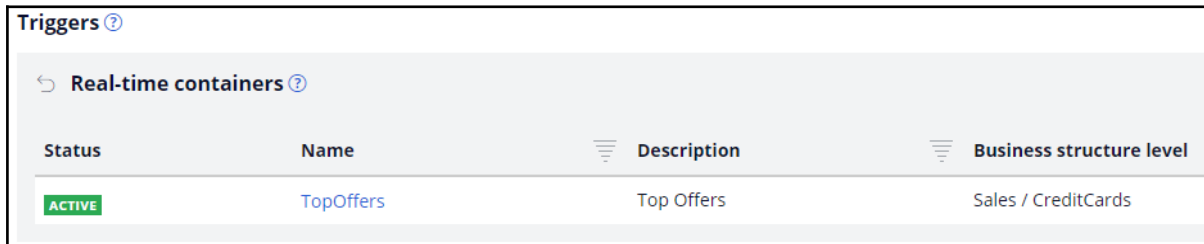
In this video, we will demonstrate the Engagement Policy capability without altering the use case.

Currently, U+ Bank is doing cross-sell on the web by showing various credit cards to its customers. Every time Troy logs in to his accounts page, a credit card offer is shown.

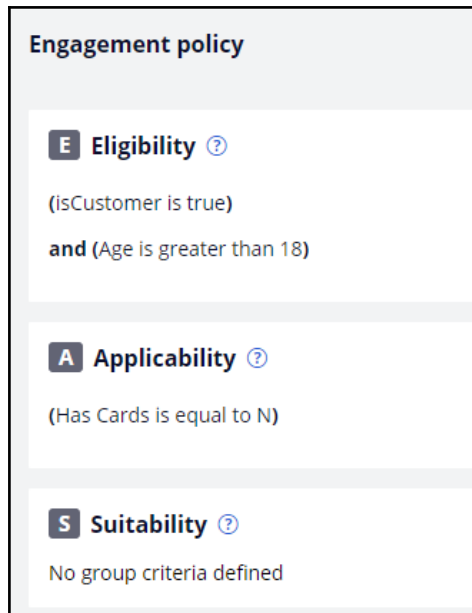
U+ has a new set of Eligibility rules. As a result, Troy only qualifies for the Standard Card offer.

To see the existing configuration, navigate to Next-Best-Action Designer -> Channels.

As you can see, U+ Bank's website invokes the TopOffers real-time container to present offers from the Sales->CreditCards group.



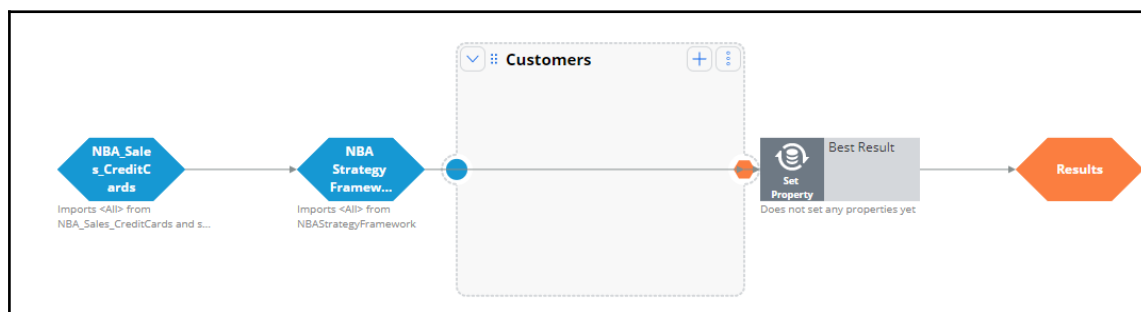
Status	Name	Description	Business structure level
ACTIVE	TopOffers	Top Offers	Sales / CreditCards



In the Engagement Policy, the Eligibility and Applicability rules have been defined to reflect the bank's current requirements using properties in the criteria.

Sometimes, implementing a specific use case requires a decision strategy.

To understand how the decision strategy is used to enforce Engagement Policy rules, open the Trigger_NBA_Sales_CreditCards strategy. This strategy is associated with the TopOffers real-time container. Every time a decision is requested from the U+ Bank website, this strategy is executed.

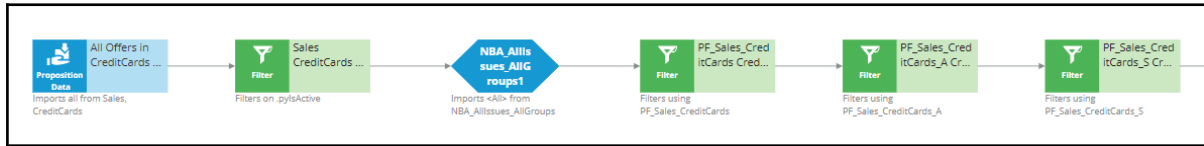


The first component of the strategy is responsible for all Engagement Policy rules.

If you open the strategy, notice that it imports the Actions from the Sales->CreditCards group and then applies the Engagement Policy configuration rules: Eligibility, Applicability, and Suitability.

Each Engagement Policy corresponds to a Filter decision component in the NBA_Sales_CreditCards strategy, and each of these components uses a Proposition Filter rule to implement the rules created in NBA Designer.

A Proposition Filter rule contains the conditions that make certain customers eligible for certain offers.



The decision strategy that implements the new eligibility requirements will be used in this Filter component.

The purpose of this demo is to show the mechanics and required steps for creating an Engagement Strategy, rather than focusing on a concrete use case.

To create a new Engagement Strategy, navigate to the Intelligence -> Strategies and create a strategy with a new canvas.

Enter a short description for the new strategy.

Select the business Issue and Group, and 'Apply to' class. Note that the 'Apply to' class is the Customer class from the Primary Context.

Now, open the strategy.

Enable the External Input component on the canvas to represent all Strategy Results (SR) records that are input into the strategy. The strategy will basically be used as a When rule in a Proposition Filter.

Connect the External Input component to the Results component.



Save and test the strategy with the Troy data transform.

For external input you can use the AllCreditCards strategy, as it imports all Sales->Credit Cards. The test shows that all credit card Actions will reach the Eligibility Strategy, nothing is filtered out by the framework.

Notice that the strategy outputs various credit card offers for Troy.

Complex eligibility rules can be implemented in this decision strategy.

For now, to keep it simple and show the mechanics of the Engagement Strategy, consider that customers qualify only for the Standard Card. Everything else is filtered out.

To implement this, add a Filter component to filter out all credit cards except the Standard Card.

Now, configure the Filter component.

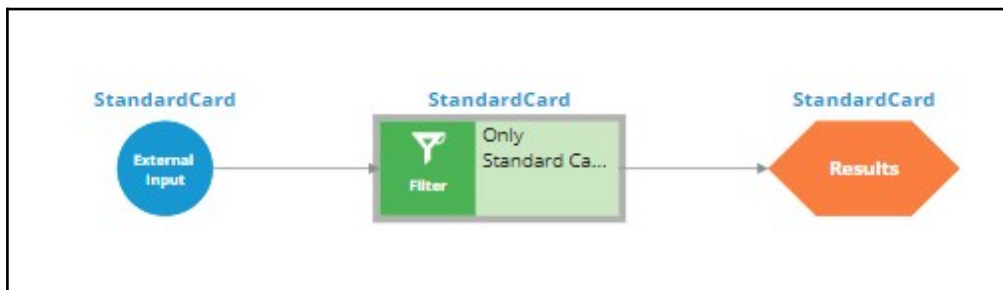
Start by naming the Filter component.

Click the gear icon to open the Expression Editor.

The Filter component should select the Action for which the pyName property is equal to "StandardCard".

Connect the components on the canvas and re-test the strategy.

Examine the output of the Filter component. Now the Filter component outputs only the Standard Card. All other Actions are filtered out.



Make this strategy accessible in Next-Best-Action Designer as an Engagement Strategy.

Once the Record is added, navigate to NBA Designer and open the Engagement Policy to configure this strategy as an Eligibility Strategy for the Group-level Eligibility rules.

The screenshot displays the 'Engagement policy' configuration interface. Under the 'Eligibility' section, three conditions are defined:

- Condition 1: `isCustomer` is true.
- Condition 2: `Age` is greater than 18. A [Select values](#) link is present next to the value 18.
- Condition 3: `Engagement Stra...` has results for `Only Standard Card`.

The condition is: Engagement Strategy has results for Only Standard Card.

Saving this completes the required configurations.

Back on the U+ bank website, log in as Troy to see that the Standard Card offer is displayed. Everything else is filtered out by the Engagement Strategy you just created.

This demo has concluded. What did it show you?

- How to build a decision strategy that can be used as an Engagement Policy rule.
- How to define Eligibility rules using a decision strategy in Next-Best-Action Designer.

Using a scorecard in a decision strategy

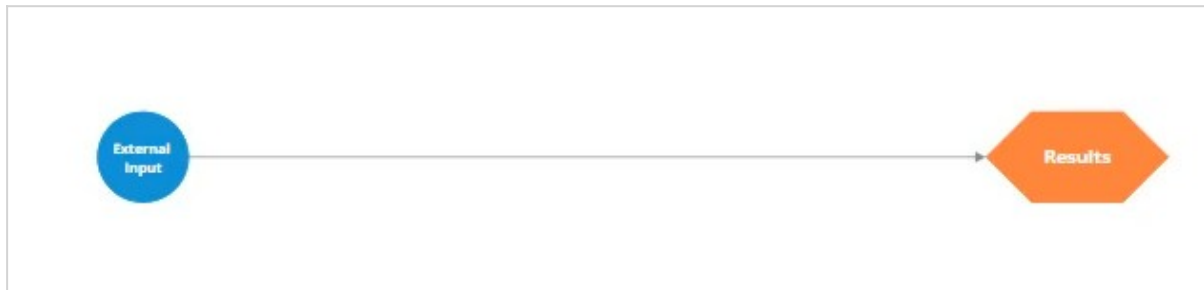
Learn how to use the segmentation result and the score value from a scorecard in a decision strategy. Learn how suitability rules can be defined to reflect the bank's requirements using a decision strategy. Also, learn the basics of strategy optimization.

Transcript

Currently, U+ Bank is cross-selling on the web by showing various credit cards to its customers.

The bank wants to implement a new requirement: All credit cards are suitable only for customers with a credit score greater than or equal to 600.

To implement this requirement, use an engagement policy that includes a decision strategy.



U+ Bank created a Scorecard that computes the credit score of the customer. To use the scorecard in the decision strategy, add a Scorecard component to the canvas and configure it.

Select the *DetermineCreditScore* scorecard model, which U+ Bank has already created.

If you open this scorecard, you can see how the system computes the credit score. Note that the three customer properties are in use, and a score is assigned to the value or range of values that the property can have.

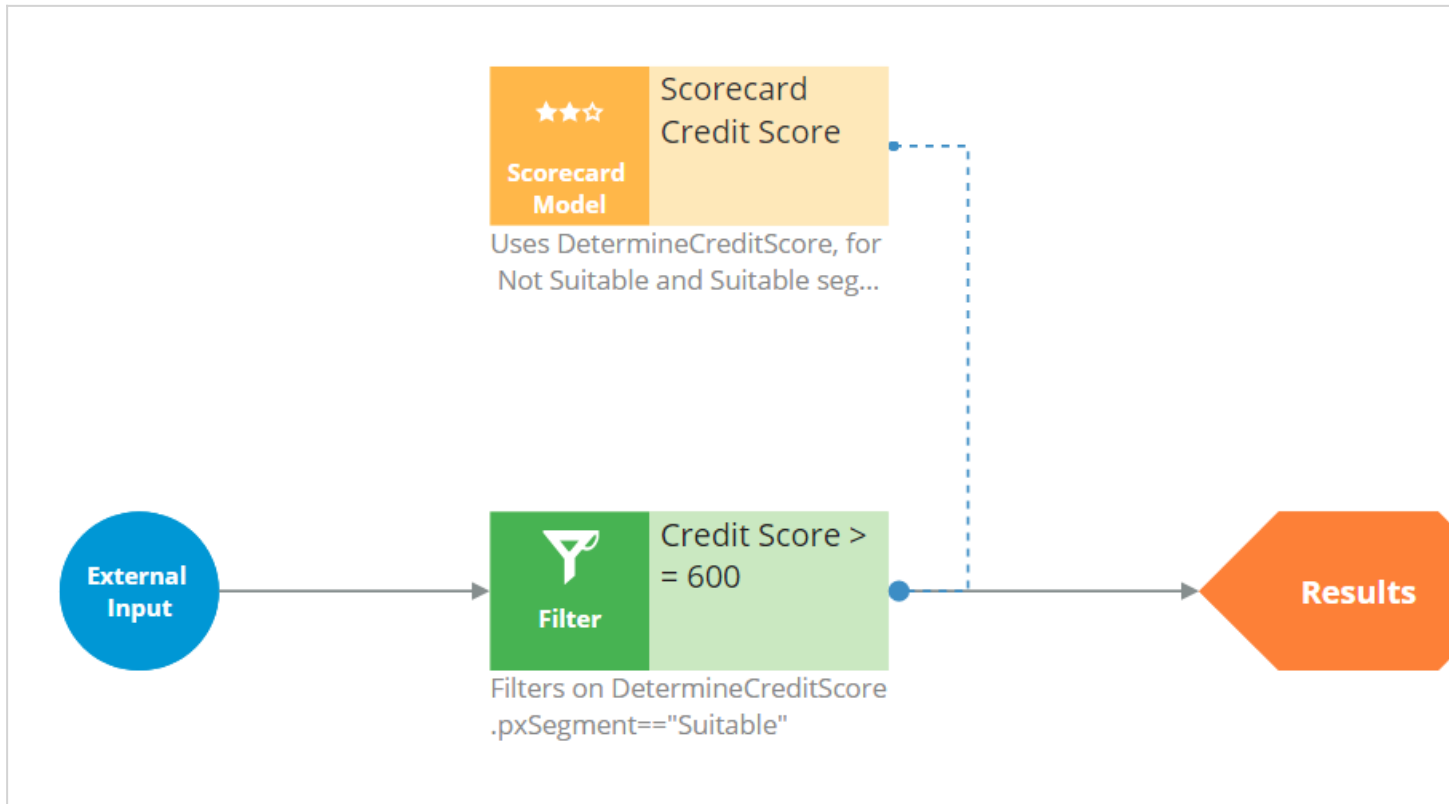
You can also see the segment results. The scorecard returns *Not Suitable* result if the credit score is less than 600. Otherwise, it outputs a *Suitable* result.

To implement the new requirement, use a Filter component to express the condition under which the Actions are suitable.

You want this strategy to output something only if the result of the scorecard is "Suitable". The system stores the result of the scorecard in the *pxSegment* property. Therefore, set the filter condition to *DetermineCreditScore.pxSegment=="Suitable"*.

If the result of the scorecard is *Not Suitable*, this strategy has no results.

Note that to reference a property from a component that is not connected to the Filter component, use the `<ComponentName>.<PropertyName>` construct.



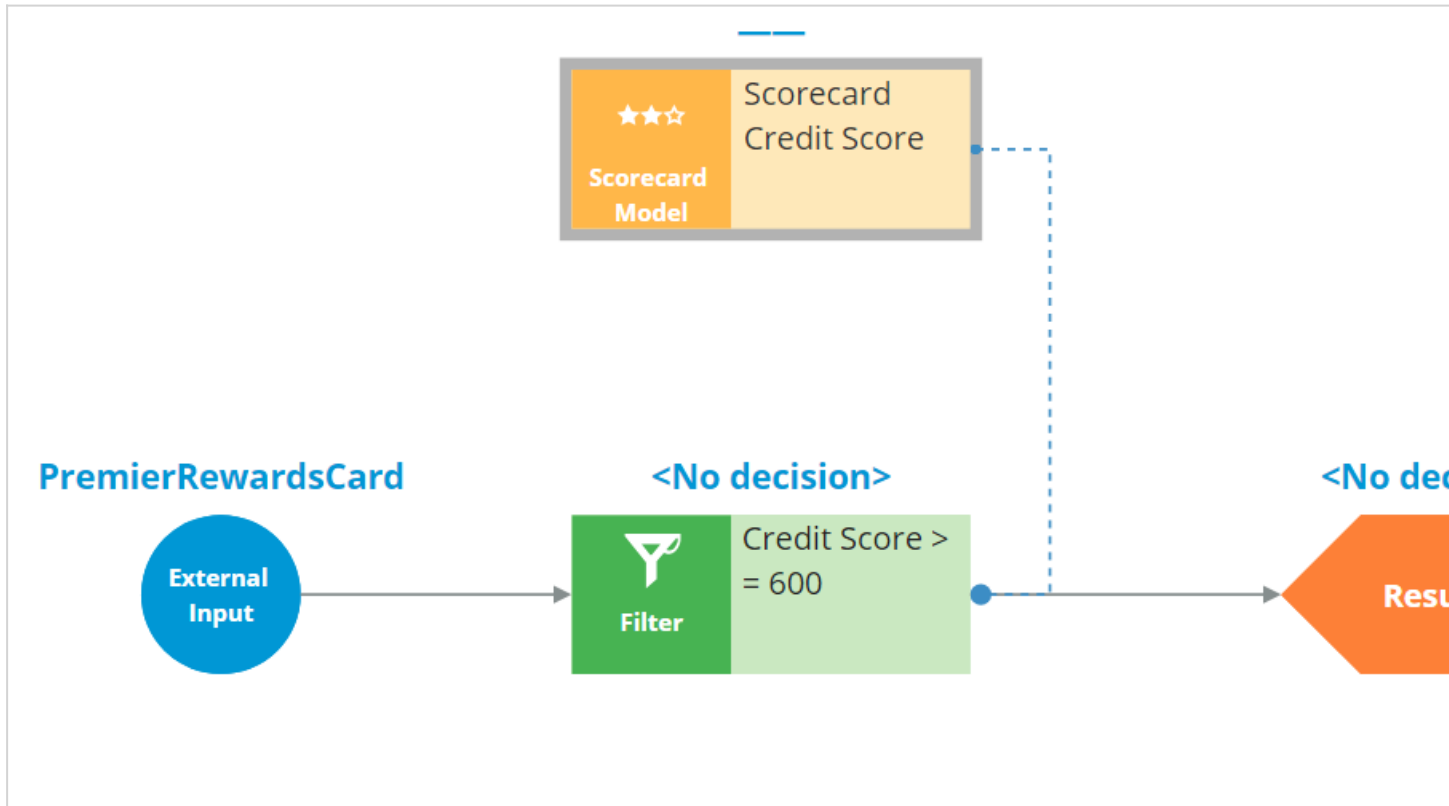
By using the External Inputs component, you can also create more complex conditions, where Action attributes are also in use.

Save the configurations.

Now, test the strategy by using Data Transform for Troy and Robert.

For external inputs, consider all credit cards available.

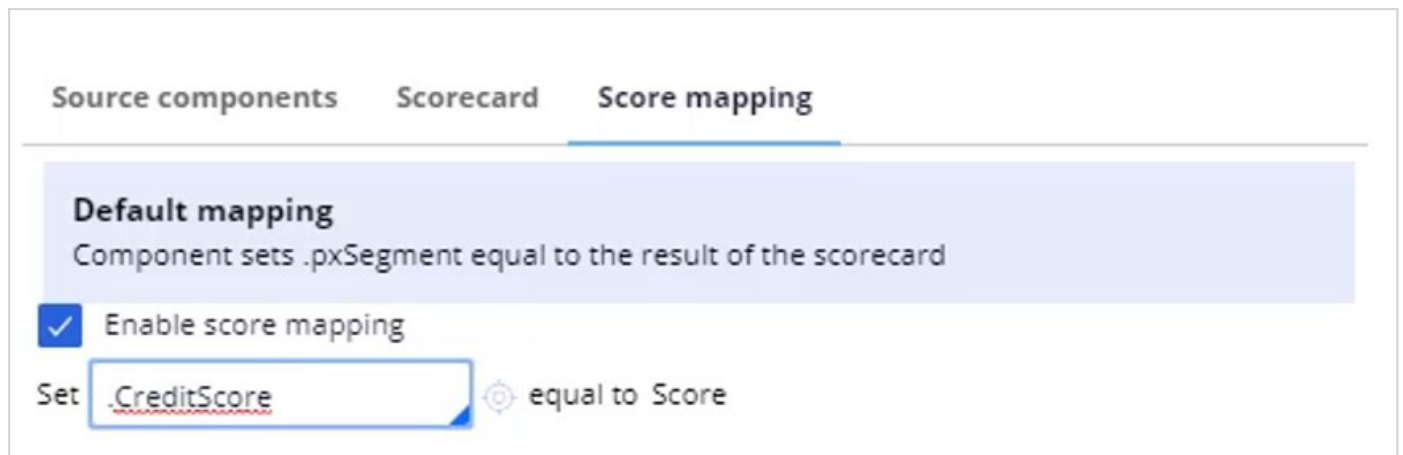
The result of the scorecard is *Not Suitable*. Therefore, the strategy did not output any results for Troy.



Now, repeat the test to verify the results for the Robert Data Transform. The strategy is also not producing any results for Robert, as the segment value is *Not Suitable*.

Now, assume the credit score value for customers is already computed and available in the *Customer* Data Model. However, this value is not set for certain customers, so you want to use the credit score determined by the scorecard itself. To implement this requirement, you must have both scores available and use a switch component to select the right credit score.

To make these adjustments, open the Scorecard component and then map the score computed by the scorecard to the *CreditScore* property. This component now outputs the scorecard calculation in the *CreditScore* property.

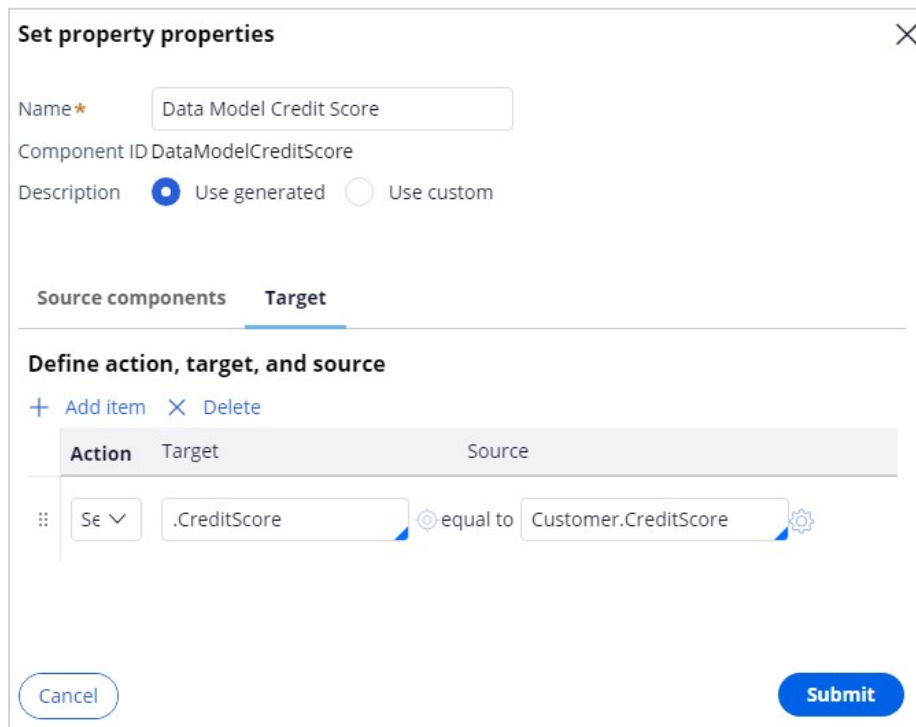


Then, add a **Set Property** component, given that the credit score is already available for certain customers.

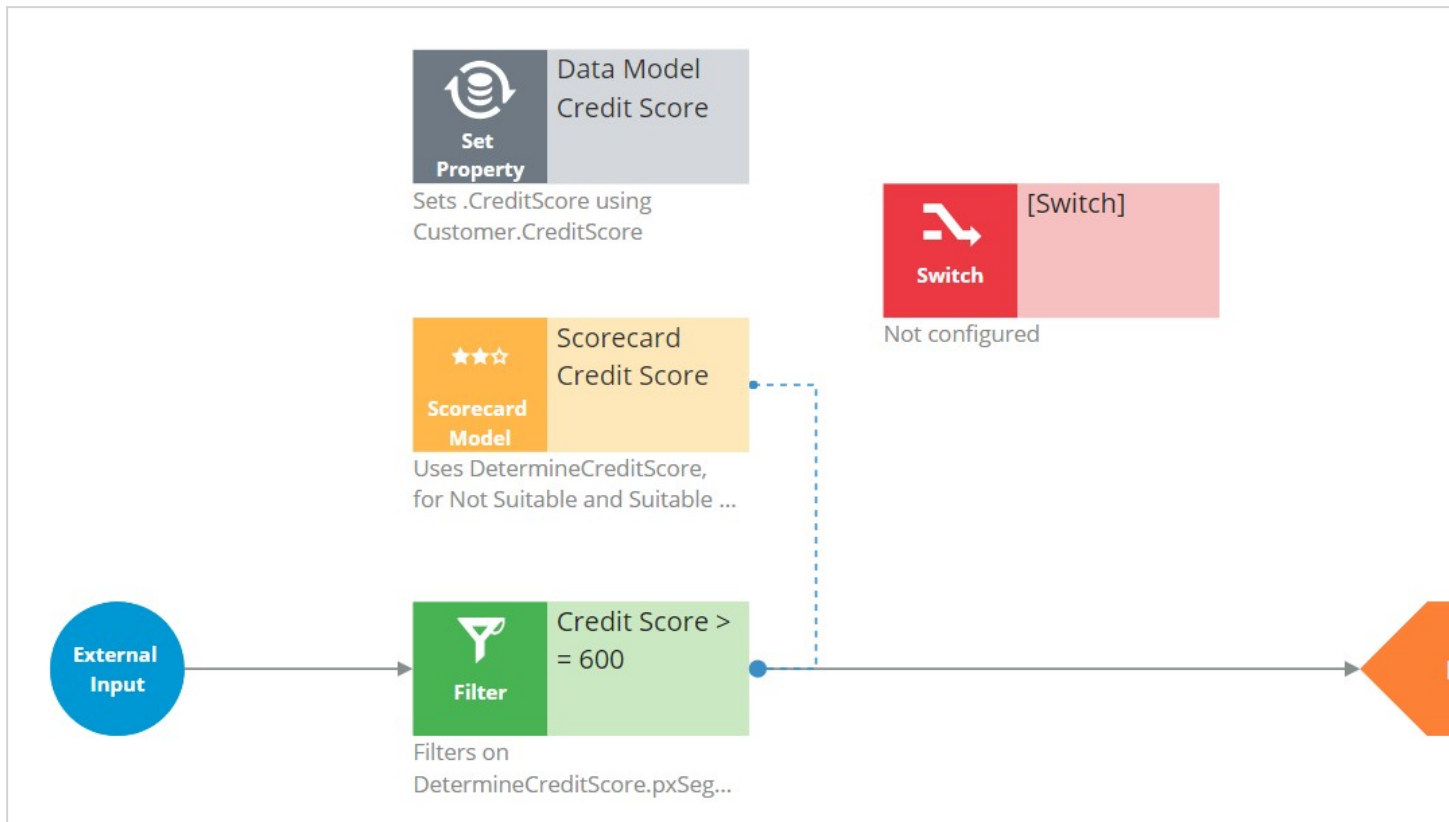
Use **Add item** to set the same *CreditScore* property to the available credit score value from the *Customer* Data Model.

Define the expression as *Customer.CreditScore*.

If set, this expression results in the credit score from the *Customer* Data Model.



To select between the two credit scores, add a Switch component.



Set the name of the Switch component to *Final Credit Score* and configure the condition. To build the condition, use the Expression Builder. The condition reads:

Select Data Model Credit Score if @PropertyHasValue(Customer.CreditScore). Otherwise select Scorecard Credit Score.

@PropertyHasValue is a function that checks if a property has a value. In this case, it checks whether the *CreditScore* property of a customer has a value in the *Customer* Data Model.

Switch properties ✕

Name*

Component ID FinalCreditScore

Description Use generated Use custom

Switch

[+ Add item](#) [✕ Delete](#)

Component	Condition
Select... <input type="text" value="Data Model Credit Sco"/> ▾	if <input type="text" value="@PropertyHasValue(Customer.CreditS"/> ⚙️
Otherwise <input type="text" value="Scorecard Credit Score"/> ▾	

With this configuration, the Switch component selects the *CreditScore* Data Model only if the *CreditScore* property of a customer has a value. If the *CreditScore* property is empty or does not have a value, the Switch component continues to run the scorecard to calculate the *CreditScore*.

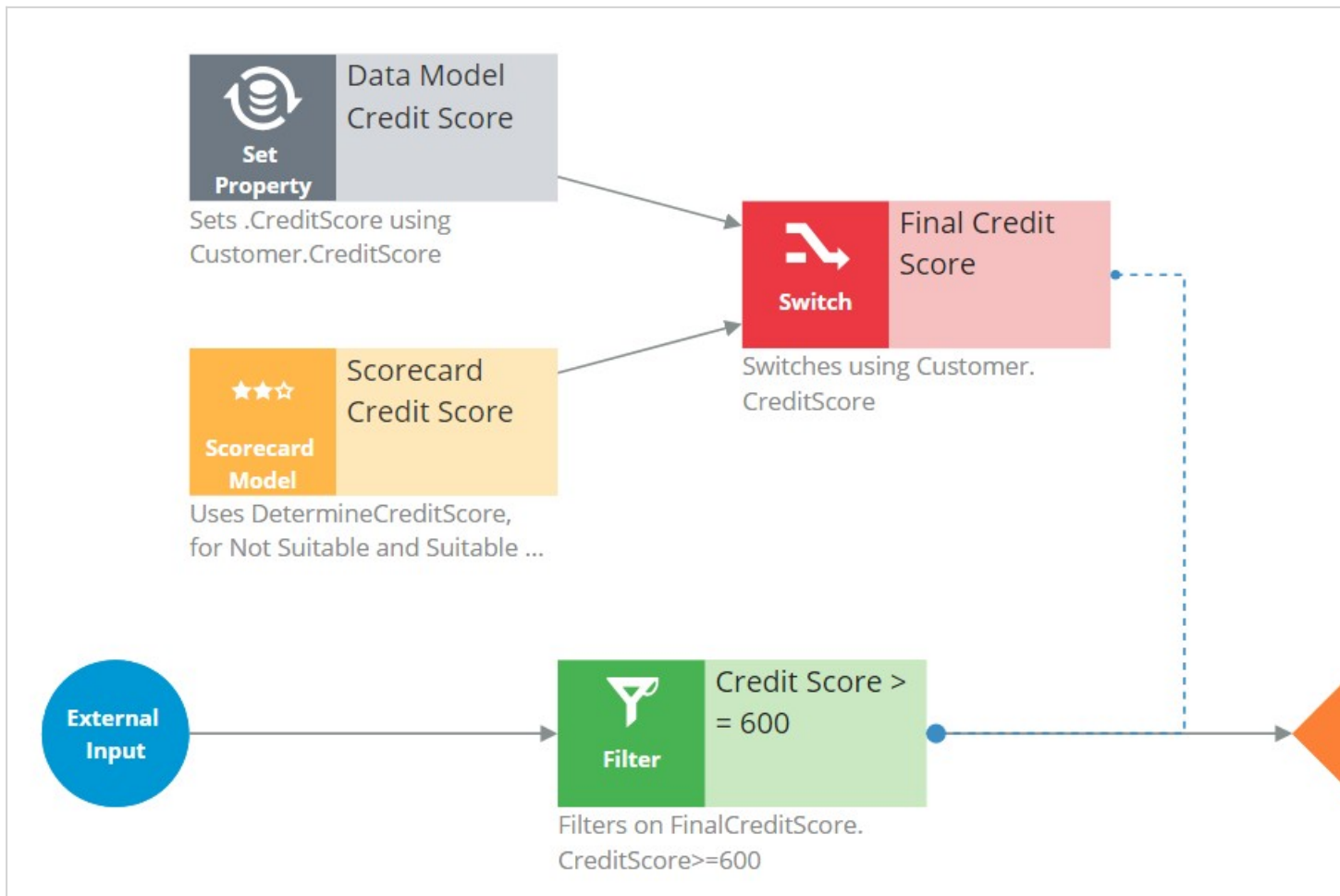
This configuration ensures that the Switch component optimizes the strategy by running the scorecard only for customers without a credit score in the external customer data. It effectively avoids redundant calculations and improves the efficiency of the decision strategy.

The arrows from Set Property and Scorecard Model components point to the Switch component. The system adds them automatically to ensure that the *CreditScore* values determined by these components are copied to the Switch component.

After you configure the Switch component, open the Filter component properties to modify the Filter condition. In this scenario, the bank decides to present various credit cards to suitable customers who have credit scores greater than or equal to 600.

So, open the Expression Builder to modify the condition as *FinalCreditScore.CreditScore*>=600.

Notice that the Filter component now references data from the Switch component.



Save the configurations.

Re-test the strategy for the Troy and Robert data transforms.

Note that the strategy outputs results for Troy because his credit score, 625, is greater than 600.

If you open the Troy Data Transform, note that the CreditScore in the Data Model is 625.

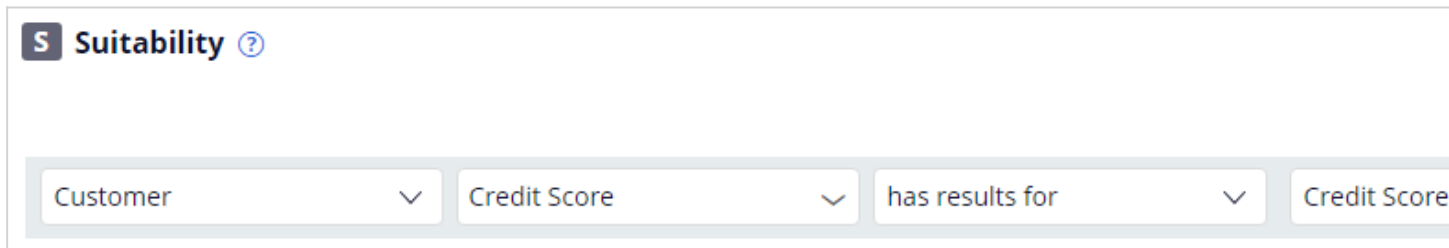
The Switch component picks up the credit score value available in the Data Model (that is, CreditScore = 625) and prevents unnecessary calculation by the scorecard.

The strategy also outputs results for Robert because his credit score is 675. The Switch component also picks up the credit score value available in the Data Model for Robert.

Because you configured the strategy, check it in so that the strategy becomes available on the U+ Bank website.

Now, mark the strategy as a relevant record and associate it with the Suitability category to make it available in the Next-Best-Action Designer engagement policy configuration.

You can now configure this strategy in the Next-Best-Action Designer engagement policy as a suitability condition for the group-level Suitability Rules.



Select the strategy. The condition is *Credit Score has results for the Credit Score >=600*.

Save to complete all the required configurations.

On the U+ Bank website, if you log in as Troy, notice that a credit card offer is displayed. The offer is shown because credit cards are suitable for Troy.

Now, if you log in as Robert, notice that the credit card offer is also displayed.

You have reached the end of this video. What did it show you?

- How to use the segmentation result and the score value of a scorecard in a decision strategy.
- How to use the *PropertyHasValue* function to check whether a *Customer* property contains a value.
- How to optimize the decision strategy by using a Switch component.

Using predictive models

Description

Predictive analytics requires historical data that contains the customer behavior you want to predict. A predictive model is used to predict an outcome based on customer behavior such as offer acceptance and churn based on customer characteristics such as credit risk, income, product subscriptions, etc. A predictive model component references a predictive model, which in turn predicts customer behavior.

Learning objectives

Describe predictive analytics

Describe how to use a predictive model in a decision strategy

Arbitrate between business issues using applicability rules in Next-Best-Action Designer

Predictive models drive predictions

With the decision management capability of Pega Platform™, you can enhance applications to help optimize business processes, predict customer behavior, analyze natural language, and make informed decisions to better meet customers' needs and to achieve positive business outcomes.

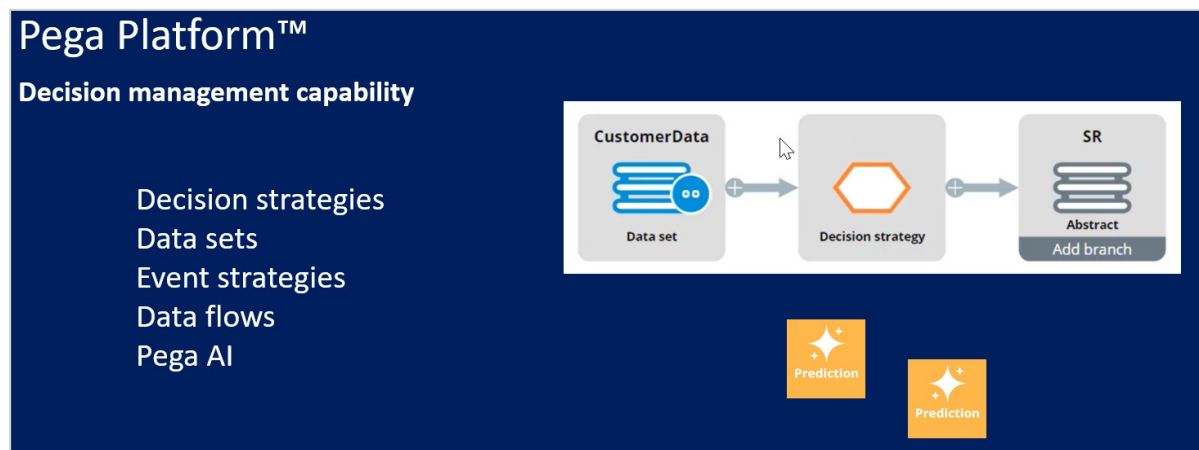
Transcript

This video introduces you to Pega AI, a feature of the decision management capability of Pega Platform™.

Other decisioning features of the Pega Platform include:

- Decision strategies to improve customer experience and deploy intelligent processes based on behavioral and operational data and data sets to read and write the data used in the decision strategies.
- You can use event strategies to detect patterns in data streams and react to them.
- And to ingest, process, and move data from one or more sources to one or more destinations, you can configure data flows as scalable and resilient data pipelines.

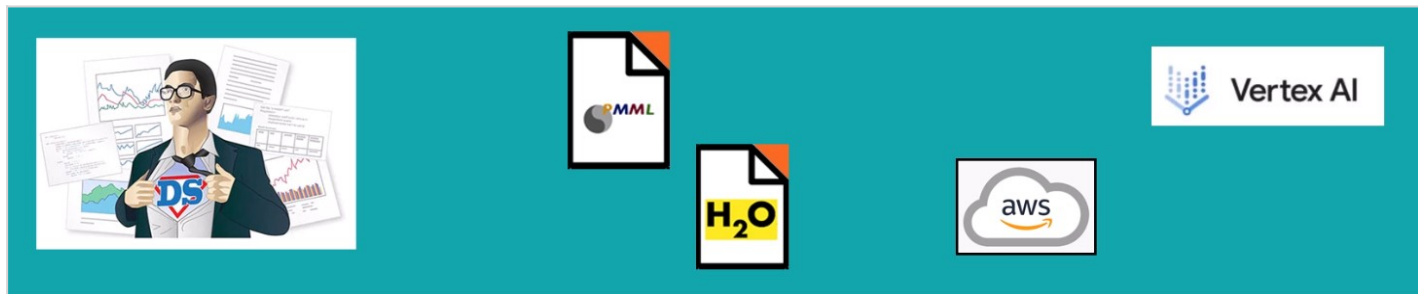
Decision management uses Pega AI to make predictions about customer behavior, successful case completion, the topic of an incoming message, or other subjects to make the decisions more relevant.



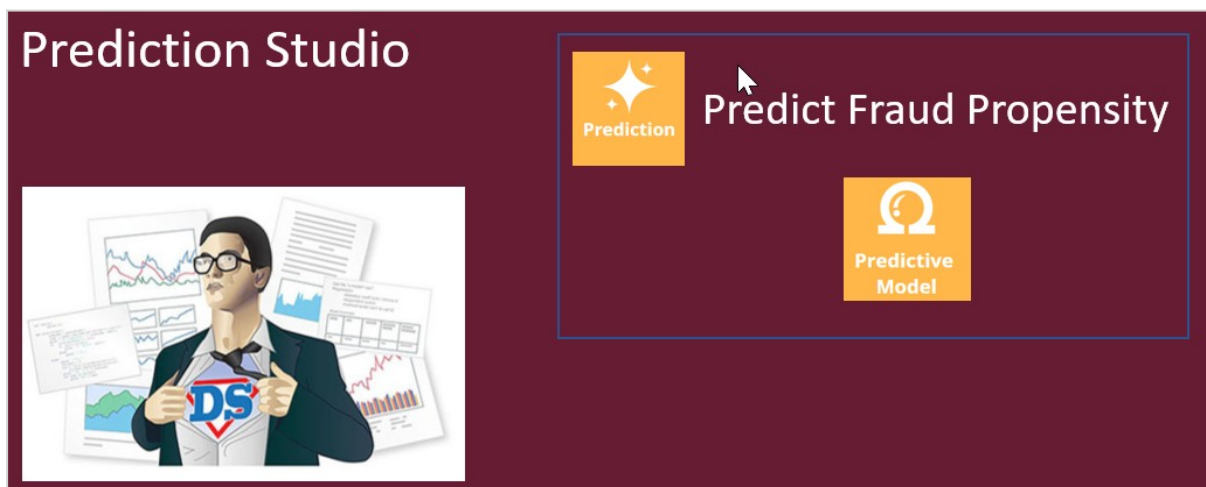
Decision management is a Pega Platform capability. You can apply decision management to any application that is built on Pega Platform.

Predictions differ to suit the domain they are used in, but one or more predictive models drive them all.

A data scientist can create a predictive model in Pega Platform or an external environment that can export the model as a PMML or H2O file. Another option is to connect to a machine learning service such as AWS SageMaker or Google Vertex AI.



If an insurance company wants to use Pega Process AI™ to route incoming claims that might be fraudulent to an expert based on the outcome of a predictive model the data scientist creates a fraud model to drive a new case management prediction in Prediction Studio.



Prediction Studio is the dedicated workspace where you manage the life cycle of predictive models and the predictions they drive.

A prediction is a hand-off to an application developer, who can then use the prediction in a decision step in the case type to route cases more accurately. This strengthens the separation of concerns.

You can use Pega Customer Decision Hub™ to make next-best-action decisions for your customers.

Customer Decision Hub predictions can predict customer behavior, such as which customer is about to churn or predict the likelihood that a customer clicks on a web banner to support the decision on which banner to show to a customer.

Pega Adaptive Decision Manager (ADM), a key component of the decision management capability allows a data scientist to configure self-learning, adaptive models that continuously improve predictions about business processes and customer behavior.

An adaptive model rule typically represents many adaptive model instances because each unique combination of the model context generates a model.

In Customer Decision Hub, adaptive models drive many predictions that come with the product out of the box, such as the Predict Web Propensity prediction that predicts the likelihood that a customer clicks a web banner.

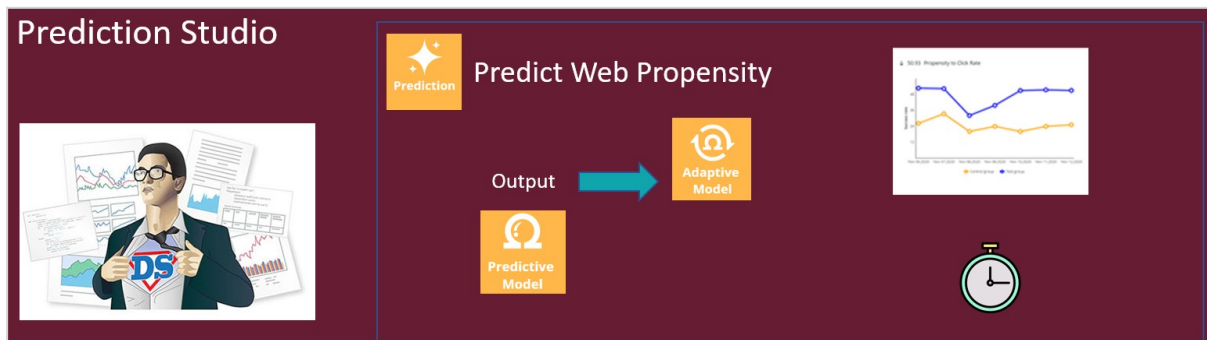
Customer Decision Hub predictions have several features specific for the domain such as a control group for which the prediction outputs a random propensity instead of the propensity that is generated by the adaptive model instance.

Comparison of the control group and the model propensity-based group allows you to measure the lift in a success rate that the AI generates, an important business metric.

Also, Customer Decision Hub predictions feature a response timeout setting. After the timeout expires, a negative response is recorded.

The response timeout setting depends on the use case. For example, in a web use case, several minutes suffice while in an outbound email campaign, the response timeout is set to several days to allow customers enough time to respond.

You can further enhance the prediction by using the output of a predictive model as a predictor in the adaptive model.



The Pega Customer Service™ application uses the natural language processing capability of decision management to analyze incoming text and route the messages based on the topics and entities detected.

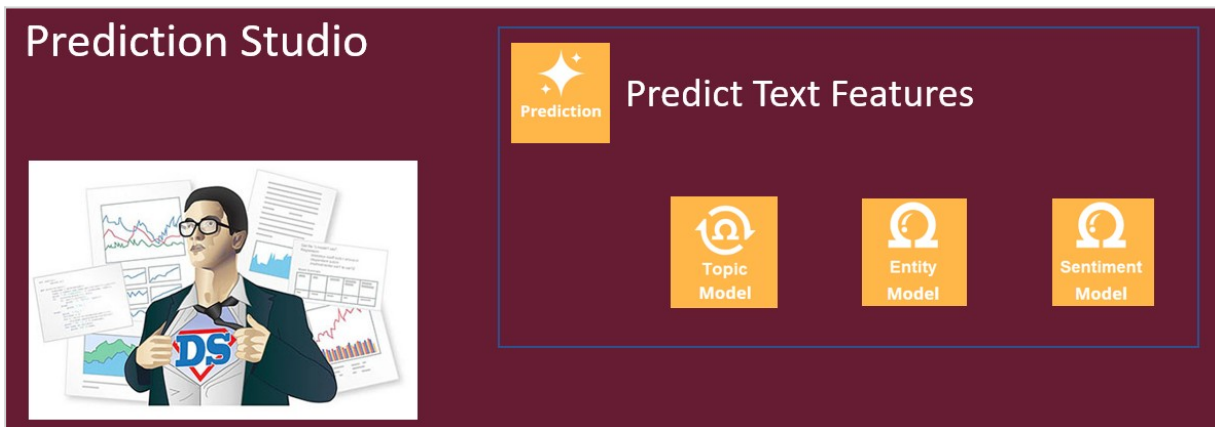
Pega Customer Service uses Text analytics predictions that are distinctly different from both case management predictions and Customer Decision Hub predictions.

Text analytics predictions use predictive models to detect the topic of an incoming message that the application can use to optimize the routing of the message to the relevant department.

Secondly, text analytics predictions use entity extraction models that qualify text as, for example, an account number, a postal ZIP code, or an address.

The application can use this information to fill relevant fields in a case automatically.

Finally, the text analytics predictions come with a sentiment model that can route or prioritize negative messages to improve the customer experience.



Feedback on the detected topics, entities, and sentiment by CSRs improves the performance of the text analytics prediction over time.

This video has concluded. What did it show you?

- Pega AI allows you to improve business processes and customer engagement by using predictions.
- Predictive models drive the predictions.
- The predictive models can be static or adaptive.
- Predictions are managed in Prediction Studio.

Arbitrating across business issues

Pega Customer Decision Hub™ combines analytics, business rules and customer data to make intelligent decisions. Every next best action weighs customer needs and business objectives to optimize decisions.

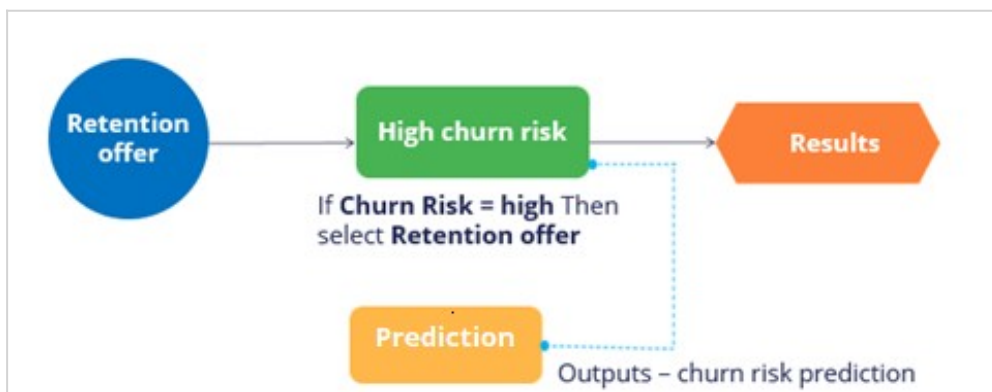


To arbitrate across multiple business issues, you create a decision strategy and use this strategy to define applicability conditions in Next Best Action Designer.

For example, consider a bank who is already doing sales. Now it wants to proactively offer retention offers for customers with high churn risk.

To predict the churn risk, a data scientist creates a churn prediction that calculates the likelihood that a customer will soon leave the bank.

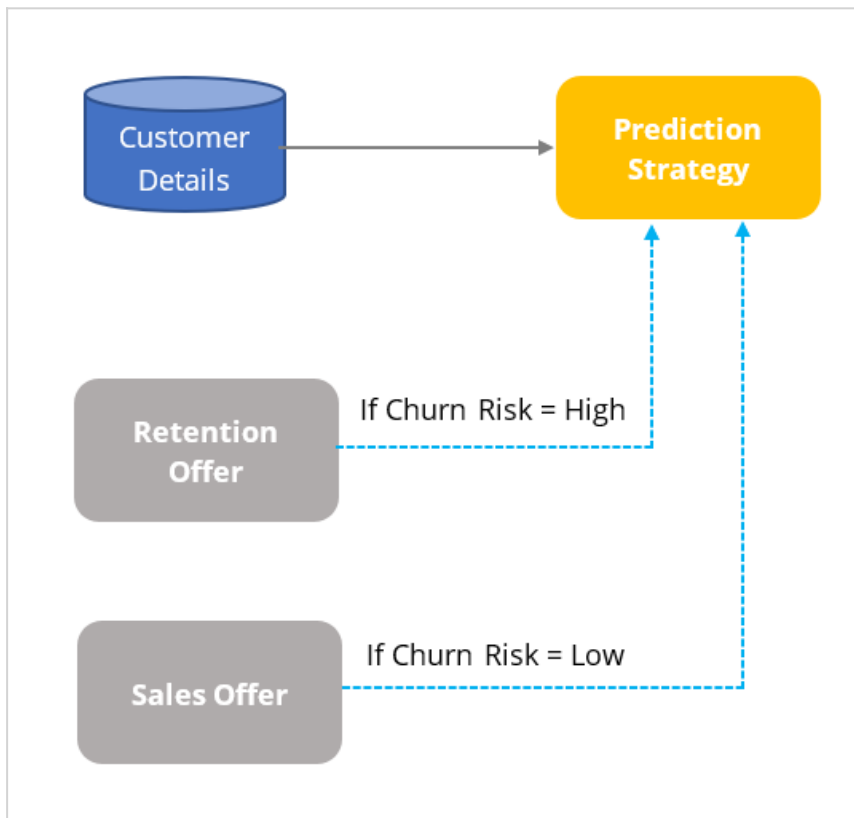
You create a decision strategy that differentiates between high risk and low risk customers that references the churn prediction.



In Next Best Action Designer, you use the decision strategy to define applicability rules for the sales offers and the retention offers.

When the customer has a low churn risk, sales offers are applicable.

Conversely, when the customer has a high churn risk, retention offers are applicable.



Using predictions in engagement strategies

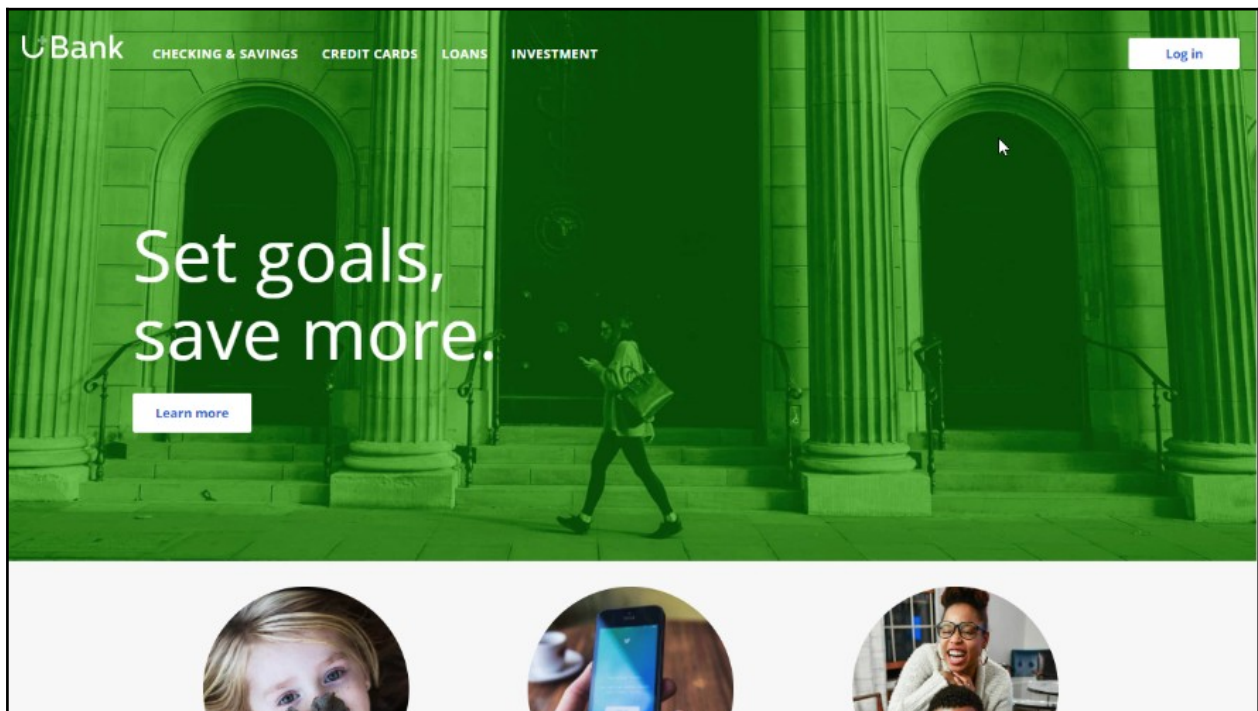
Introduction

A prediction is used to predict customer behavior such as offer acceptance and churn based on characteristics such as credit risk, income, and product subscriptions. Learn how to arbitrate between different groups of actions to display more relevant offers to customers. Gain experience using a prediction in a decision strategy and learn how applicability rules can be defined to reflect the bank's requirements in a decision strategy.

Transcript

This demo shows you how to use a prediction in an engagement strategy to determine customer applicability for a retention offer.

Currently, U+ Bank is cross-selling on the web by showing various credit cards to eligible customers who log in to its website. The bank now wants to show a retention offer, instead of a credit card offer, to customers who are likely to churn in the near future. The credit card offers are shown only to loyal customers.



To meet this business requirement, a decisioning administrator has already set up the taxonomy by defining a new business issue called Retention, and an offer group.


Taxonomy

Business structure

Business structure

Issues / Groups

Retention

 ExtraMiles

Sales

 CreditCards

This ExtraMiles group contains a retention offer, Extra Miles 5K.

Actions

Search

by name or description

Issue / Group

Retention / ExtraMiles ▼

Showing 1 of 1 results

[Extra miles 5K](#)

ExtraMiles5K

The next step is to create an applicability condition that makes a customer qualify for a retention offer when there is a high likelihood that the customer might churn. A data scientist has created a prediction that identifies these high-risk customers. When you open the prediction in Prediction Studio, notice that the possible response labels are **Churn** and

Loyal to predict customer behavior. The result of the prediction is stored in the pxSegment property.

Response labels

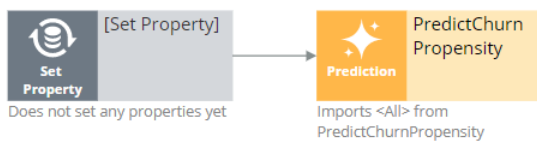
Labels for the possible values of the responses.

Churn

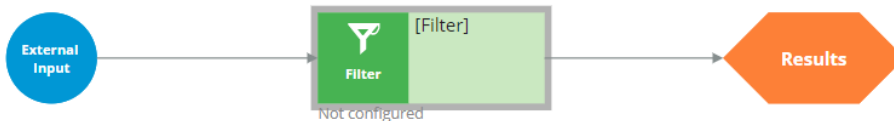
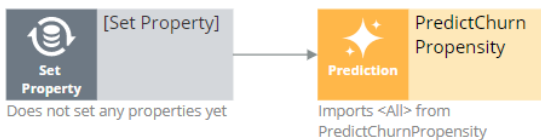
Target label Alternative label

Churn Loyal

To define the applicability condition, you create a decision strategy to output a retention offer only if the response label of the prediction is **Churn**. Add a **Prediction** component to the canvas and configure it to reference the churn prediction. Add a **Set Property** component and connect it to the Prediction component. You can configure the Set Property component at a later point to accommodate parameterized fields.



Next, add a filter component to filter out the loyal customers and pass retention offers to high churn risk customers only.



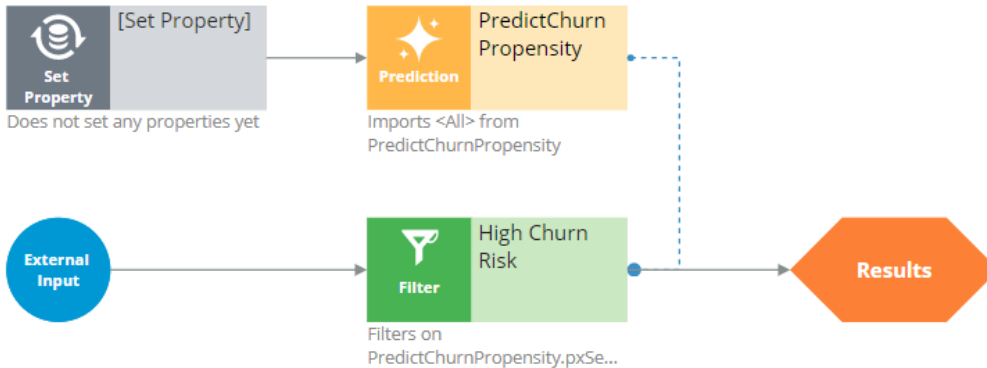
The filter condition is defined to output a retention offer when the pxSegment property of the prediction is equal to **Churn**.

Expression builder

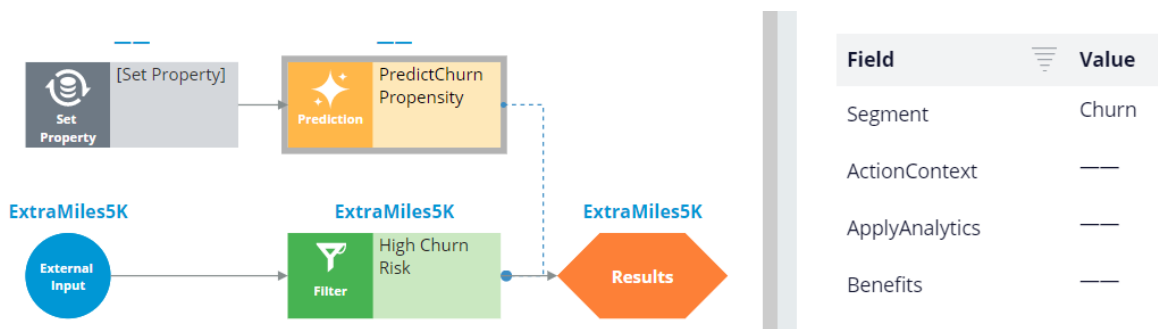
Browse

Test

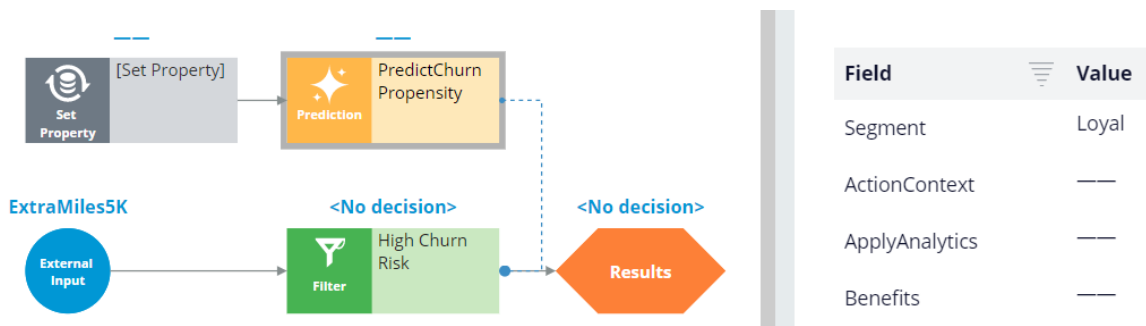
```
1 PredictChurnPropensity.pxSegment="Churn"
```



Next, test the strategy using two customer profiles, **Troy** and **Barbara**. For external inputs, consider all available retention offers. The strategy outputs a result for **Troy** because the result of the prediction is **Churn**.



The strategy does not have a result for **Barbara**, because the Segment value is **Loyal**.



By checking in the strategy, you commit your changes so that they go into effect. You can now use this strategy in the Next-Best-Action Designer engagement policy as an applicability condition.

The first business rule you need to implement is that the **ExtraMiles** group is applicable only to high churn risk customers. To implement this rule, in the **Applicability** section, define a condition for the customer field. Select the **RetentionStrategy**. The condition is: the RetentionStrategy has results for the High Churn Risk component.

The second business rule you need to implement is: U+ Bank wants to show credit card offers to low-risk customers only; meaning the **CreditCards** group is not applicable for high-risk customers. To implement this rule, modify the **Applicability** section of the **CreditCards** group. The condition is: the RetentionStrategy doesn't have results for the High Churn Risk component.

Once the applicability conditions are defined, you need to amend the **Channels** configuration. Because U+ Bank introduced a new group, **ExtraMiles**, which belongs to a new business issue, **Retention**, you need to select the results from the appropriate business structure level. In this case, the bank wants to arbitrate between two different business issues: Sales and Retention. Therefore, select All Issues/All Groups from the business structure level. Saving the configuration implements the business requirement.

On the U+ Bank website, when you log in as **Troy**, notice that the retention offer is displayed because **Troy** is predicted to churn in the near future.

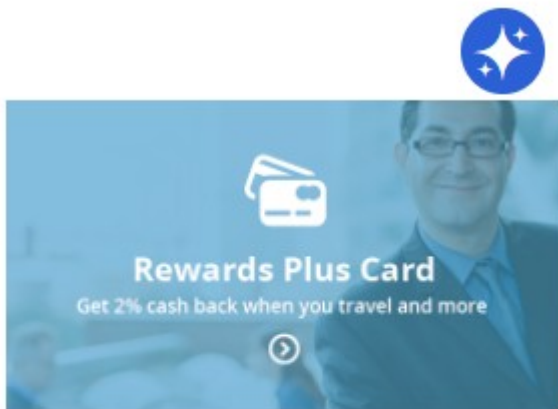


Extra miles 5K

5,000 extra miles

[Learn more](#)

Now, when you log in as **Barbara**, notice that the credit card offer is displayed because she is predicted to remain loyal for now.



Rewards Plus card

Get 2% cash back when you travel and more

[Learn more](#)

You have reached the end of this demo. What did it show you?

- How to use a prediction in a decision strategy
- How to arbitrate between different groups of actions to display more relevant offers to customers
- How to define applicability rules using a decision strategy in Next-Best-Action Designer

Creating parameterized predictors

Pega Customer Decision Hub™ uses adaptive models that use a wide range of predictors to optimize customer interactions. Input fields that are not directly available in the customer data model can be made accessible to the models by configuring these fields as parameterized predictors. Learn how to create parameterized predictors for expressions, offline model scores, and online model scores.

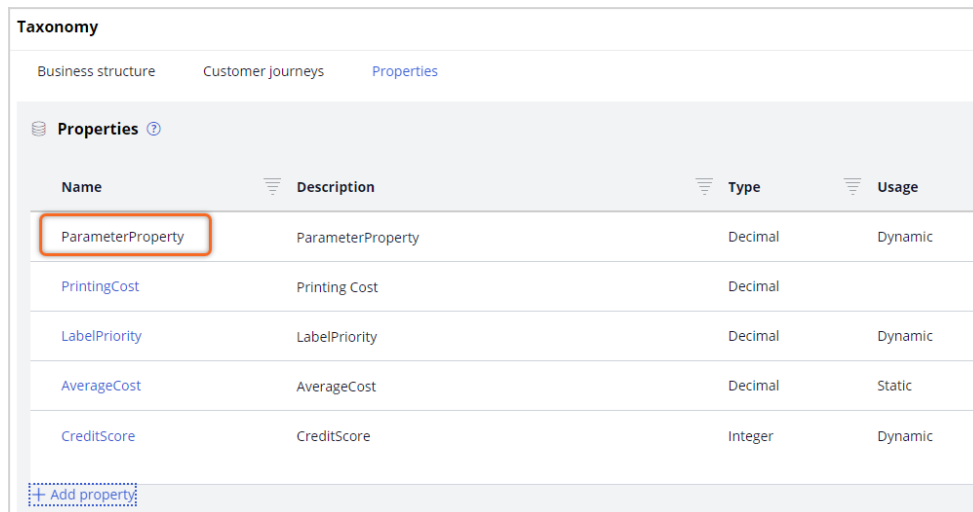
Video

@@@

Property to store a parameter

This demo shows you how to create parameterized predictors for expressions, offline model scores, and online model scores in Customer Decision Hub. In all three cases, the Data Scientist creates an SR property to store the parameter value, adds the property as a parameterized predictor to the prediction, and configures the pre-processing extension strategy.

In NBA Designer, you create a new property as part of the taxonomy. In this demo, we repurpose a generic Parameter Property. The system automatically maps the property to the *CDH-Strategy Result* (or *CDH-SR*) class.



Name	Description	Type	Usage
ParameterProperty	ParameterProperty	Decimal	Dynamic
PrintingCost	Printing Cost	Decimal	
LabelPriority	LabelPriority	Decimal	Dynamic
AverageCost	AverageCost	Decimal	Static
CreditScore	CreditScore	Integer	Dynamic

Mapping of the parameter

Next, you map a new parameterized predictor in the prediction to *ParameterProperty* in Prediction Studio. The Predict Web Propensity prediction calculates the likelihood that a customer clicks on a web banner. Like *ParameterProperty*, the prediction results are saved to the *CDH-SR* class.

Prediction properties ✕

Prediction name

Outcome

Prediction output
 Predicting Propensity to Click for each combination of

	<input type="text" value=".pyIssue"/>	
and	<input type="text" value=".pyGroup"/>	
and	<input type="text" value=".pyName"/>	
and	<input type="text" value=".pyDirection"/>	
and	<input type="text" value=".pyChannel"/>	
and	<input type="text" value=".pyTreatment"/>	

[+ Add](#)

Save results to

CDH-SR

Data-pxStrategyResult

You then create a new parameterized predictor and map it to *ParameterProperty*.

Edit Parameters ✕

Parameterized predictors (5) Pages (1)

Map fields to parameterized predictors to supply this data to this model. < 1 2

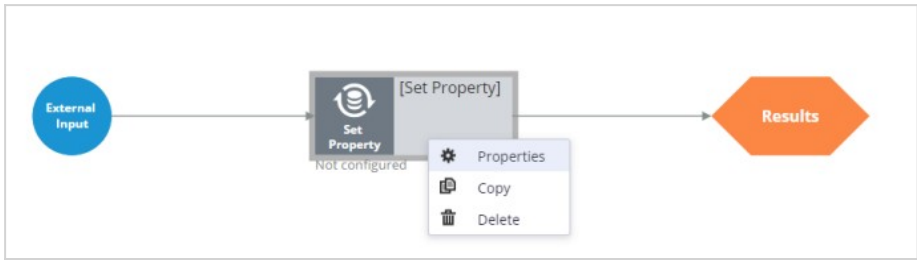
Predictor	Data type	Predictor type	Field
NewParameter	Decimal	numeric	.ParameterProperty

[+ Add parameterized predictor](#)

Configuring the extension point

To populate the parameter, you configure the NBA pre-processing extension point strategy. The pre-processing strategy runs before the prediction strategy so that the output of the extension strategy is available as input to the prediction. You save the strategy in the correct *Apply TO* class and then set the strategy result class to CDH-SR.

The first example of a parameterized predictor is an expression. You add a Set Property component to set the parameter propensity to be equal to the ratio of customer visits to a web page in the last 30 days and in the last 90 days.



This expression returns a value of zero when the number of Investment page visits in the last 90 days is zero.

Set property properties

Name*

Component ID Calculation

Description Use generated Use custom

Source components **Target**

Define action, target, and source

+ Add item X Delete

Action	Target	Source
Se	.ParameterProperty	equal to @if(DecisionRequestCDH.Ct

Expression builder Browse Test

```

1 @if(DecisionRequestCDH.Customer.FSClickstream.I
nvestmentPageVisitsLast90Days=0,0, divide(Decis
ionRequestCDH.Customer.FSClickstream.Investment
PageVisitsLast30Days, DecisionRequestCDH.Custome
r.FSClickstream.InvestmentPageVisitsLast90Day
s))

```

Otherwise, it returns the ratio of the Investment page visits in the last 30 days to the last 90 days. A high value may indicate the increasing interest of the customer in the content of this page.

The second example of a parameterized predictor is offline model scores. The data scientist team might produce many product group scores for each customer and each channel. A Decisioning Architect creates a database table to accommodate the data. A data set, associated to the customer context, makes the data available as a data source. Notice that each customer can have multiple scores for different categories or different channels.

To add the model scores as potential predictors to the adaptive models, you add a Data Join component to the extension strategy.



The component joins the Offlinescores page that contains the model scores to the available data.

Data join properties

Source components **Join** Properties mapping

Join source components with

Type Pages

Name* Offlinescores

Class CDH-SR

To output only relevant scores, you specify the channel and the product category to match the prediction.

Join when all conditions below are met

+ Add item × Delete

Offline scores	Offlinescores
When .pyChannel	is equal to "Web"
and When .Category	is equal to .Category

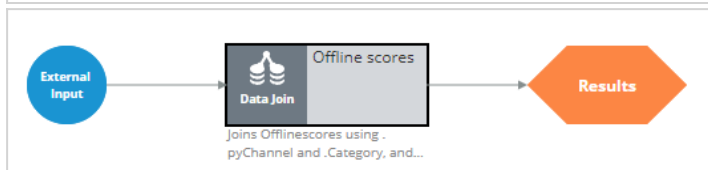
Exclude source component results that do not match join condition.

You then configure the component to populate the Parameter Property with the model scores.

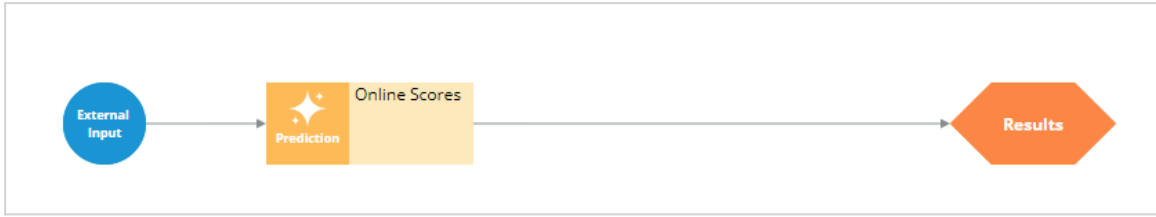
Define mapping

+ Add item × Delete

Target	Source (Offlinescores)
:: Set .ParameterProperty	equal to Offlinescores.Score



The third example of a parameterized predictor is the score of a churn prediction running in Customer Decision Hub. The data scientist team develops predictive models based on the latest insights and data that drive the churn prediction. To make the churn score available to the adaptive models as a parameterized predictor, add a prediction component that references the churn prediction to the pre-processing extension strategy.



Set the Parameter Property to equal the propensity calculated by the churn prediction.

Define action, target, and source

+ Add item × Delete

Action	Target	Source
Se	.ParameterProperty	equal to .pyPropensity

Run the strategy to verify that Parameter Property is populated.

Outcome weight	0.0
Parameter Property	0.72017587 60057334
Placement type	Tile
Previous component	Online Scores
Priority	25.8181156 1903224
Product category	---
ProductID	---
Propensity	0.72017587 60057334

To summarize, creating parameterized predictors for adaptive models involves:

- Creating an SR property to store the parameter value
- Adding the property as a parameterized predictor to the prediction
- Configuring the pre-processing extension strategy

Defining and creating customer journeys

Description

Customer journeys are the sum of the experiences that your customers go through when interacting with your organization. Rather than looking at just part of an experience or transaction, the customer journey documents the full spectrum of the customer experience. Customer journeys give you a familiar mental model to manage and work with their creative content. You can create customer journeys directly in Next-Best-Action Designer and map customer actions to different journey stages.

Learning objectives

- Define customer journeys in Next-Best-Action Designer.
- Assemble and manage customer journeys.
- Define stage entry criteria for customer journeys.
- Test and monitor customer progress through customer journeys.

Pega Customer Journeys

Customer journeys are a sum of the experiences that your customers go through while interacting with your enterprise. Rather than reviewing only part of an experience or transaction, the customer journey documents the full customer experience.

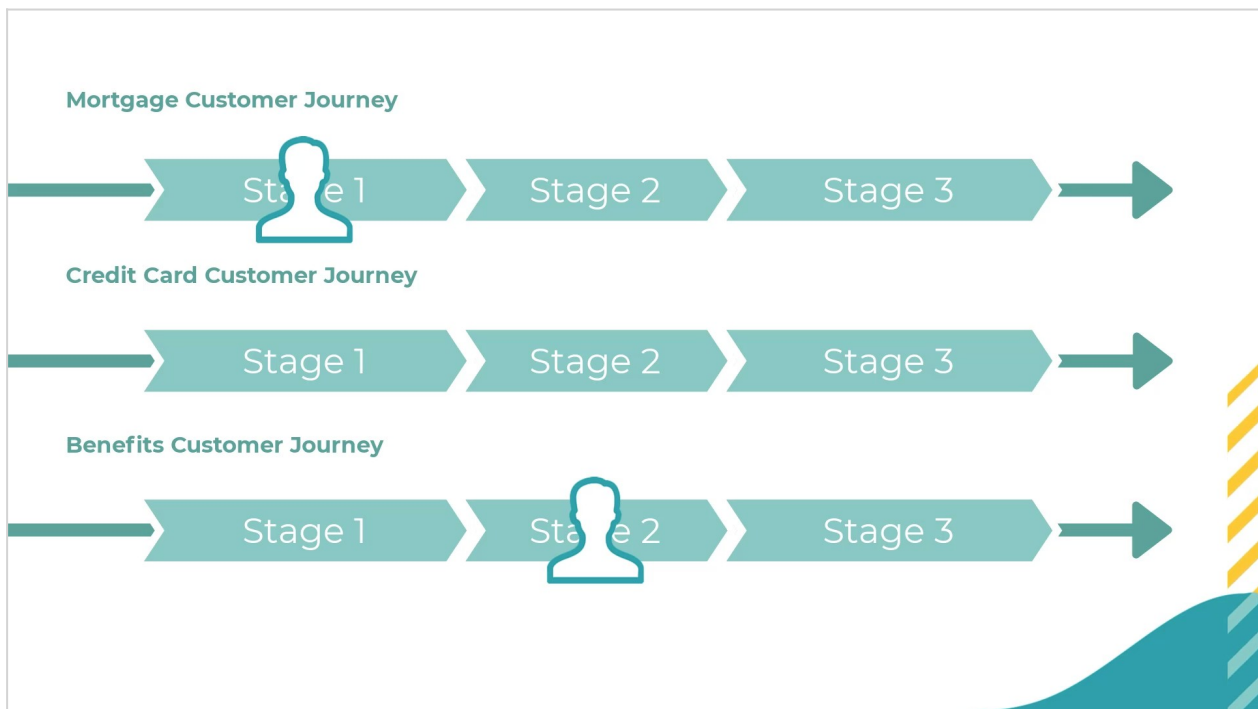
Transcript

Industry standard customer journeys are used to script the customer experience, which often results in a customer getting stuck on a path of offers that does not lead to a relevant outcome. Pega Next-Best-Action Customer journeys take a transformational approach to the industry standard tool that are the customer journeys.

Pega customer journeys and actions in them are prioritized using real-time AI, which provides you the following benefits:

- Allows you to build meaningful customer journeys
- Stays true to the Next-Best-Action paradigm
- Delivers a more relevant customer experience

With Pega customer journeys, customers might find themselves in multiple journeys or switching from one to another depending on what is most relevant to them at that point in time.

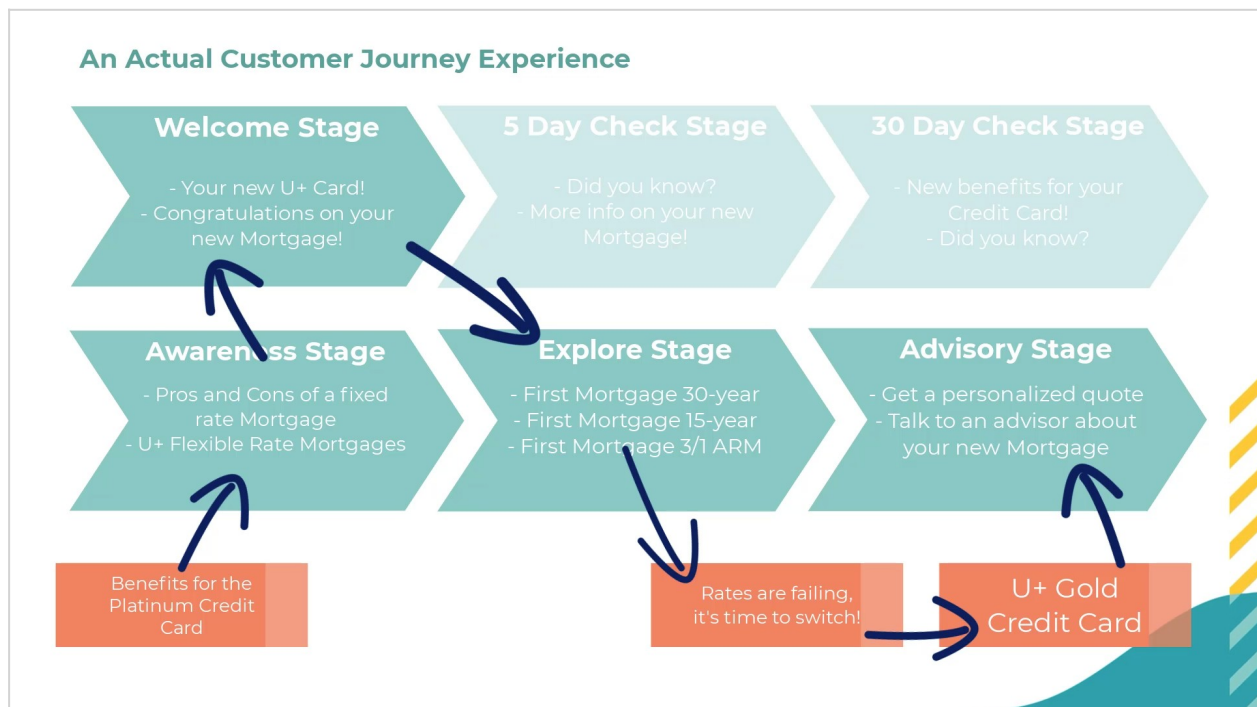


Customer journeys give you a helpful mental model and allow you to influence the customer experience while remaining true to the Next-Best-Action paradigm. Instead of scripting the entire customer experience offer-by-offer, you can add multiple journeys and stages with actions to your system and let AI leverage real-time data to determine which journey or action in a stage is most relevant to the customer at that time.

You can apply customer journeys to multiple business issues, for example:

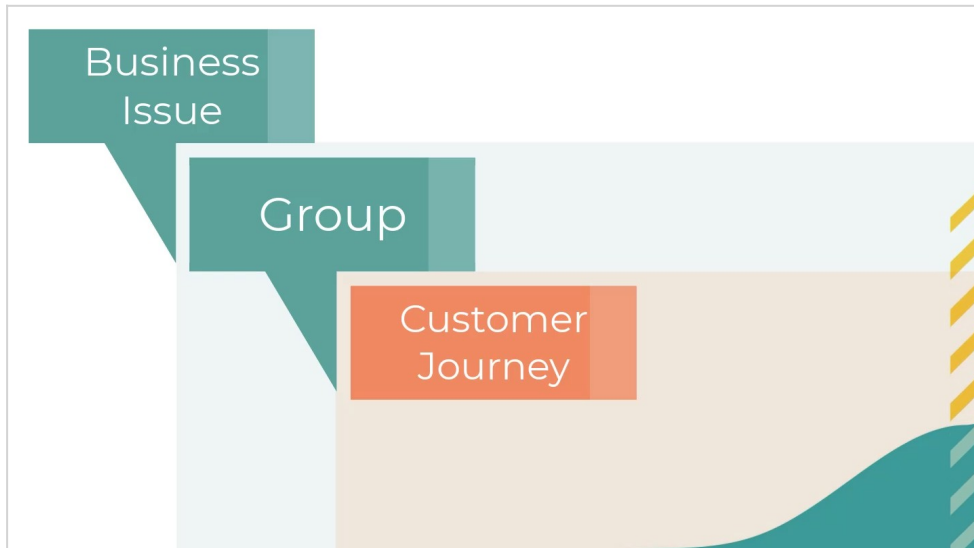
- Periodic check-ins during the onboarding period. You can have an initial welcome stage, followed by a five-day check stage, and then finally, a 30-day check stage.
- Familiarizing and leading the customer to a new mortgage. In this journey, we first raise awareness about mortgages, and then once familiar with the concept, we invite the customer to explore various mortgage offers. Finally, once engaged in the explore phase, we can provide an advisor to help the customer to develop a detailed mortgage offer.

Customers can be a part of many journeys, which are continuously prioritized. Customers can be in one stage of each of given journey, and then progress through the stages sequentially. The system ensures that the customer receives the most relevant action, even if an action is not a part of any journey, which means the actual customer journey might look very different from what you originally had in mind.



You define customer journeys through Next-Best-Action Designer. There, you can define a customer journey and the business issue and group that it will be nested under, which means that this journey inherits the Engagement Policies of that business issue and group.

You can also set your journey to cover all issues and all groups for a broader scope of actions.

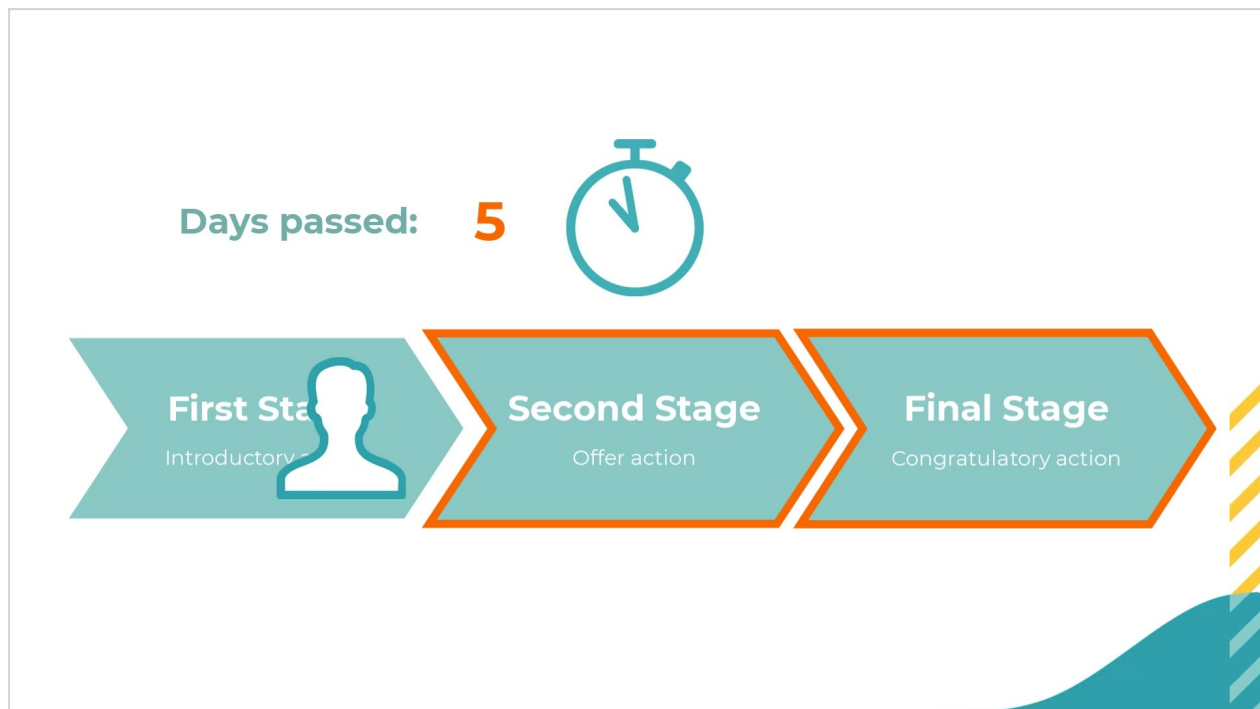


After setting up the journey, you define the number of stages and their names. The customer will progress through these stages in a sequential order. You then add relevant actions to each stage. The actions that you have added come with their available treatments. Customers in each stage see and interact with these actions, and the system recognizes these actions as a part of a particular stage in a customer journey.

If you add an action to a customer journey, you can decide if you present the action as only part of the journey, or both as a part of the journey and as a standalone action. The system records interactions for the action separately.

Before customer progresses from one stage to another, they need to satisfy the entry criteria of the next stage. It is not recommended to use very rigid entry criteria with multiple conditions, as the AI can guide the customer through different journeys and standalone actions.

Entry criteria can be as simple as 30 days passing since the customer has joined the bank and as complex as a customer having engaged with a set of mortgage offers in the last week, showing interest in talking to an advisor.



Take a look at an example of how entry criteria work and where they start to take effect in the prioritization. Say your system has 30 actions, and it is time to present the best three of them to a particular customer. The 12 orange actions are a part of various customer journeys.

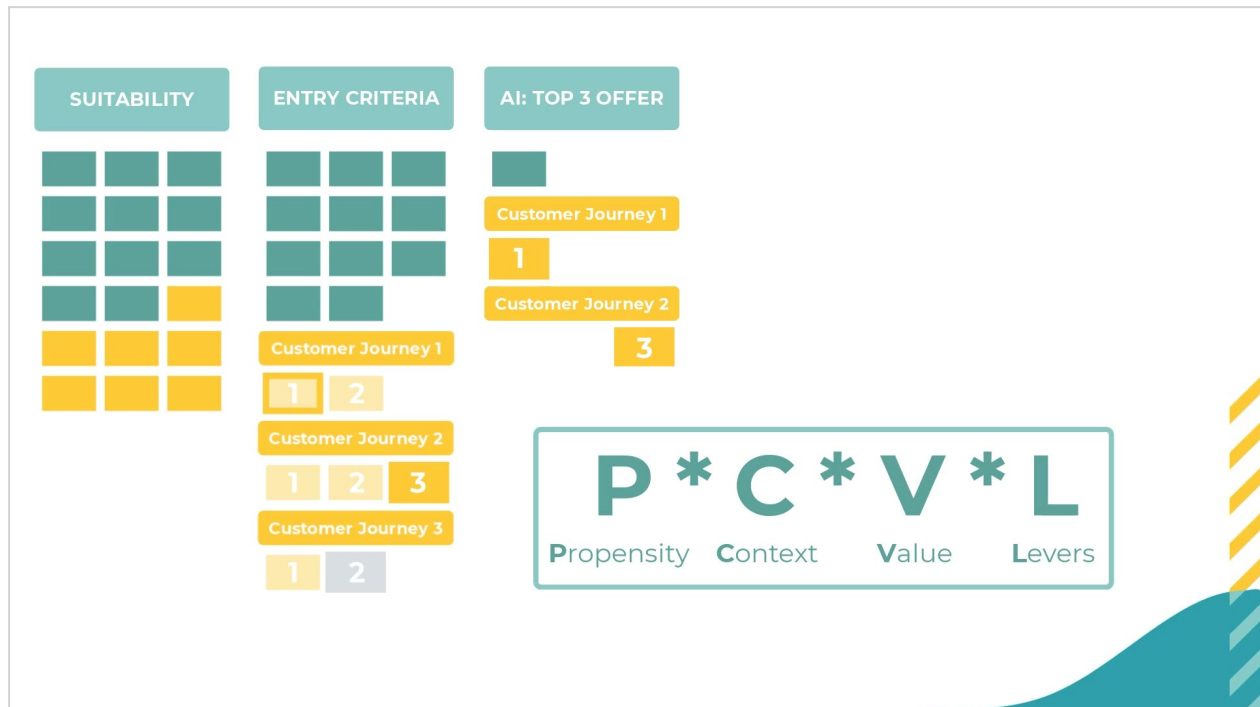
First, the system applies the standard engagement policies. Consequently, the eligibility, applicability, and suitability policies of the business issue, group, and actions are applied, which leaves us with 18 actions, out of which, seven are part of various customer journeys.

Now let's expand to see that the customer journey actions are a part of different journeys, and that each action is a part of a different stage. As you can see, different actions are relevant for the customer in the various stages due to the entry criteria set up by the Decisioning Architect. In this example, the customer:

- Passes the entry criteria to the first stage in the first journey
- Is in the third stage of the second journey, and it passes its entry criteria
- Is in the second stage of the third journey, but it does not pass its entry criteria

From the seven actions, which progressed through the suitability and are part of a journey, the customer actually qualifies for only the two highlighted actions.

At this point, AI selects among the 13 actions the best three actions to show to the customer based on the PCVL calculation (propensity * context * value * levers).



In summary, each component of the policy filters the initial set of 30 actions:

- 24 eligible
- 21 applicable
- 18 suitable

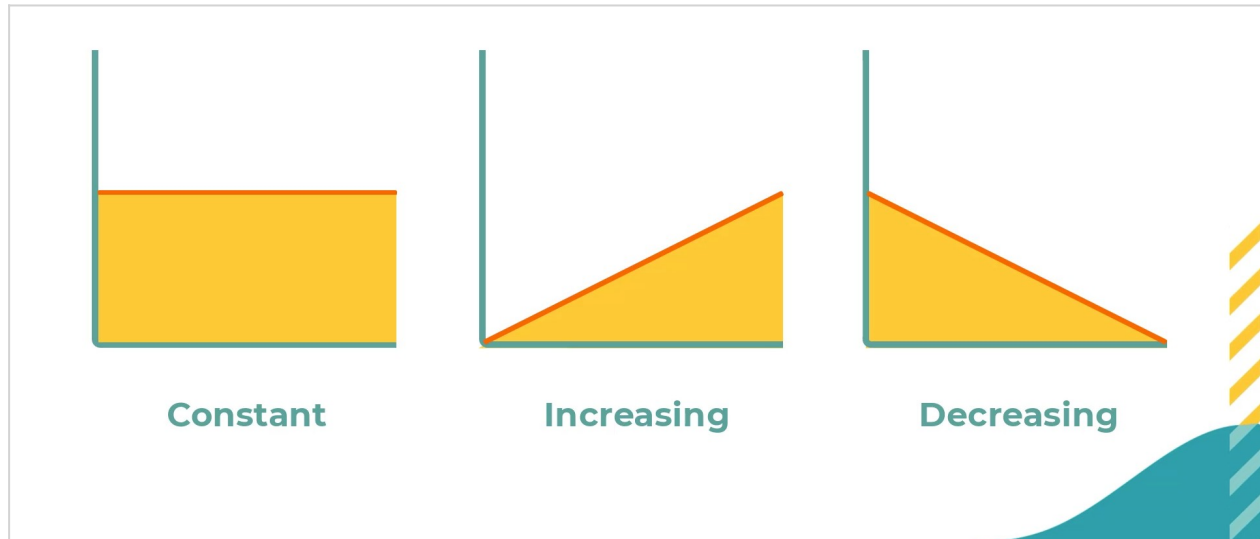
From the 18 suitable actions, 11 are standalone, not part of any journey and two actions from the remaining seven pass the entry criteria from the various journeys and stages defined. The remaining 13 actions, which pass the conditions are prioritized by the AI, and then, the best three are selected. The selected actions can be one of the standalone actions, an action from a journey, or an action from a journey that is already in progress.

Customer journeys can influence the PCVL formula in two ways.

The first way to influence the propensity occurs passively as you add customer journeys to your system. Customer journeys come with predictors that begin to function as your customers interact with the journeys that you have designed for them. These predictors will influence the propensity directly.

The second way is to use stage upweighting. Stage upweighting applies a numerical upweight value to actions in that stage and is used to increase or decrease actions promotion level. You can apply it to any stage, and it is a part of the Business levers in the Arbitration.

Upweighting can be constant, increasing or decreasing over time. For example, you might want to upweight actions of a stage for the first 10 days that the customer is in this new stage. During the first 10 days, the upweight will decrease steadily to avoid annoying the customer with a single offer.



In summary, customer journeys provide a good mental model for managing content and influencing next-best-action decisions to present suitable actions seamlessly for your customers.

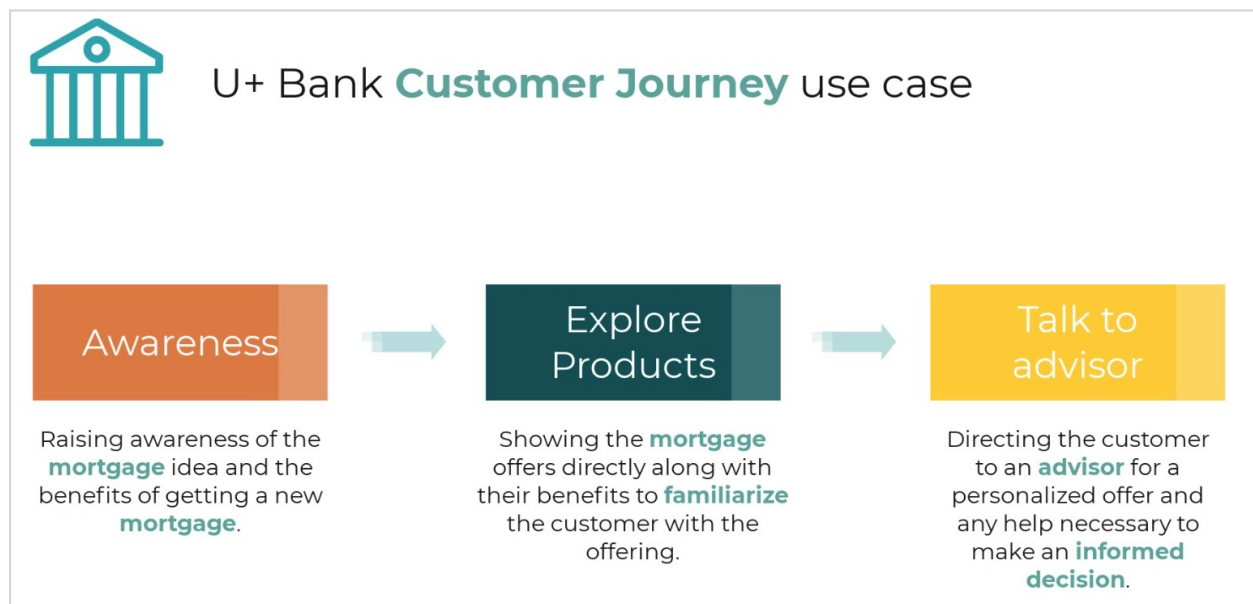
Defining Customer Journeys

Define and maintain customer journeys through the Next-Best-Action Designer to let the NBA Designers shape and monitor the experience that customers have with an entire process, not just one action or treatment. For example, the customer journey can guide a customer to first gain awareness of a product, see the offers themselves, and then finally speak to a representative. This sequential offer structure creates a streamlined and more thoughtful approach to presenting a product. Other examples of such processes include applying for a new credit card, refinancing a loan, or onboarding a new customer.

Transcript

This video demonstrates how to define and assemble a fully functioning customer journey in Customer Decision Hub.

U+ Bank wants to use customer journeys for its mortgage offering. The bank plans to first make customers aware of applying for a mortgage and its benefits. After the customer shows interest, U+ Bank wants to enable customers to explore some of the mortgage offers directly. Finally, U+ Bank wants to connect customers to an advisor to make a personalized offer and to ensure that customers make an informed decision on their mortgage. By using customer journeys, U+ Bank puts actions and treatments into stages in a sequential format that guides customers toward mutually beneficial outcomes.



To implement this use case, you first define the customer journey on the **Taxonomy** tab of **Next-Best-Action Designer**.

Add customer journey ✕

Name * Customer journey identifier

Discover and Apply for a Mortgage DiscoverAndApplyForAMortgage

Examples: Discover & Apply for the Platinum Card; Learn about Family Plans

Description

Issue / Group ?

Grow / Mortgages

Availability

Always


Never

Within a defined time period

Cancel
Submit


On the **Taxonomy** tab, you name the customer journey, and then specify the business issue and group to which your customer journey belongs. This designation is important because customer journeys inherit the engagement policies of the business issue and group that they occupy. In the following example, the actions that are assigned to this journey inherit the engagement policies of the **Grow** business issue and **Mortgages** group.

Next-Best-Action Designer




Taxonomy

Define your Next-Best-Action business structures and customer journeys




Constraints

Set outbound channel limits and suppression policies



Engagement policy

Capture business rules which define when actions are appropriate



Blurred

Taxonomy

Business structure Customer journeys Properties

+ **Customer journeys** ?

v **Grow / Mortgages**

Discover and Apply for a Mortgage ACTIVE

Awareness

Explore products

Talk to advisor

After you create your journey, you can add stages to it. At this point, you define only the names of the stages and the sequence in which the customer progresses through them. In this example, you create stages that represent the three engagement stages with the customer in the process of getting a new mortgage:

- **Awareness**
- **Explore products**
- **Talk to advisor**

Business structure	Issue	Group	Customer journey
	Grow	Mortgages	Discover and Apply for a Mortgage ACTIVE
<ul style="list-style-type: none"> ▼ All issues ☰ All groups ▼ Acquire <ul style="list-style-type: none"> ☰ Credit cards ☰ Auto loans ☰ Deposit accounts ☰ Mortgages ▼ Grow <ul style="list-style-type: none"> ☰ Credit cards ☰ Deposit accounts ☰ Mortgages 	<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p>Awareness ⋮</p> <ul style="list-style-type: none"> > Pros and Cons tile </div> <div style="width: 30%;"> <p>Explore products ⋮</p> <ul style="list-style-type: none"> > Thirty-year mortgage tile > Home equity LOC tile > Fifteen-year mortgage tile </div> <div style="width: 30%;"> <p>Talk to advisor ⋮</p> <ul style="list-style-type: none"> > Talk To An Advisor tile </div> </div>		

After you define the journey and the stages, you then add actions to the corresponding stages. When you add actions to each stage, the application adds all their available treatments. The order in which you add actions to the stage is irrelevant. The AI uses the propensity of the customer to decide which action is the best to display in the stage or out of the stage. The customer journeys do not interfere with the next-best-action paradigm.

Edit Action: Pros and Cons tile [Available]
 Grow • Mortgages • ProsAndConsTile CDH-Artifacts:01-01-01

Details **Engagement policy** Treatments Flow Test History

Action context ⓘ Intended recipient ⓘ
 Customer ▾ Eligible contact ▾

Customer journey stage ⓘ
 Discover and Apply for a Mortgage: Awareness ▾ This action is only applicable when in a customer journey stage ⓘ
 + Add customer journey stage

Eligibility ⓘ

Inherited from Mortgages Apply ⓘ

All actions
 (Customer: Relationship length in days is greater than 1
 and Customer: Credit score is greater than 600
 and Customer: Is in collections is false)

U+ Bank wants to ensure that the application displays a given action to a customer only as part of a journey. Configure this in the action itself to prevent it from displaying as a standalone action.

Next, you set the entry criteria. Entry criteria determine which conditions customers must satisfy to move to the next stage. Each stage can have its own criteria that check for various business conditions to match. For example, criteria can check the following conditions:

- The maturity of the customer relationship with U+ Bank.
- The level of interest that a customer shows in a specific category of products.
- Whether it is appropriate to connect customers with an advisor.

To implement entry criteria, use simple conditions that apply properties or use furthermore complex conditions that apply decision strategies.

For the **Discover and Apply for a Mortgage** journey, U+ Bank decides to use the following conditions:

The **Awareness** stage is relevant for customers that currently are not part of this journey.

The **Explore products** stage is relevant only if customers engage in the **Awareness** stage.

The **Talk to advisor** stage is relevant only if customers engage with an offer in the **Explore products** stage.

Edit entry criteria: Awareness ✕

Current stage details

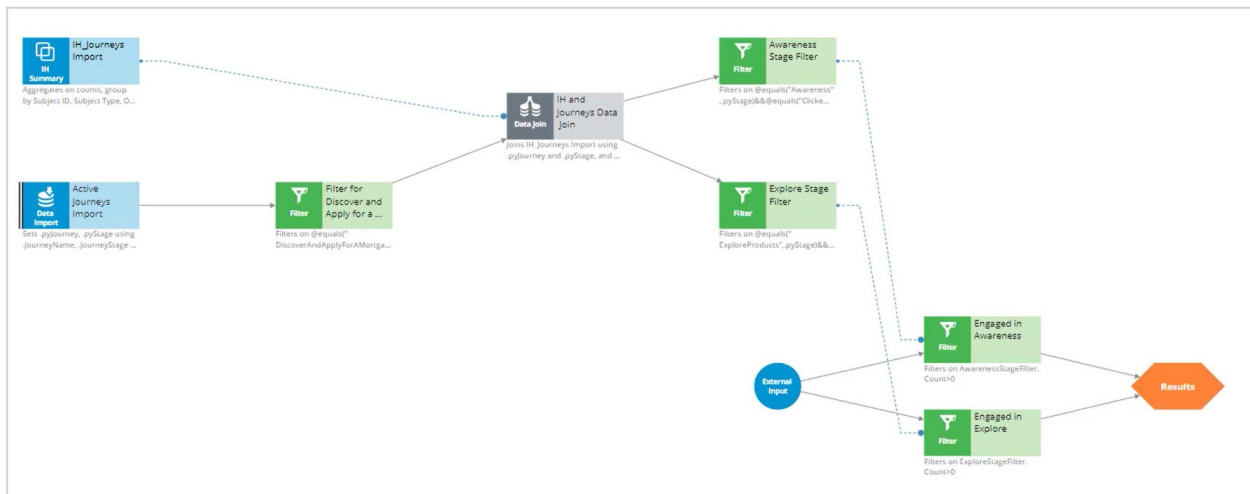
Journey ID: DiscoverAndApplyForAMortgage Stage ID: Awareness

Customer is in this stage when: ?

[+ Add criteria](#)

Cancel Submit

Start with the first stage, **Awareness**. U+ Bank wants to configure the **Awareness** stage to ensure that customers qualify because the goal of this stage is to bring attention to the mortgage offer. As a result, you leave the entry criteria for this stage blank. Customers who qualify for actions from the **Mortgages** group are in this stage of the journey.



Next, set the entry criteria for the two remaining stages by using the decision strategy. The strategy that you use here is customized for this journey and is not a part of the out-of-the-box experience.

The entry criteria expression uses two filter components:

- **Engaged in Awareness**
- **Engaged in Explore**

Therefore, they have to be connected to the Results component.

The **Engaged in Awareness** filter component outputs a record if the customer clicks at least one action from the **Awareness** stage. To implement it, use an interaction history summary component, which imports journeys and stages that a customer has ever initiated. This data is joined with the current journey data to find out the current active stages of this journey. Next, you find out if any click events are registered in the awareness stage. Finally, the U+ Bank definition of engaged in awareness: if at least one or more click events are detected on any action from the awareness stage. The Engaged in explore condition follows a similar pattern. The strategy can implement more complex conditions, but for this scenario, U+ Bank uses this strategy to define its customer engagement in a stage.

Edit entry criteria: Explore products [X]

Current stage details

Journey ID: DiscoverAndApplyForAMortgage Stage ID: ExploreProducts

Customer is in this stage when: [?]

Group ANDs [v] [⋮]

Customer [v] Entry Criteria Discov... [v] has results for [v] Engaged in Awareness [v] [↗] [+] [🗑️]

[Cancel] [Submit]

With this decision strategy, in the **Explore products** stage, you set the condition to check if the strategy has results for **Engaged in Awareness**. The condition determines whether the customer engaged positively with one of the offers in the **Awareness** stage, and the strategy returns a result for the **Engaged in Awareness** filter.

Edit entry criteria: Talk to advisor [X]

Current stage details

Journey ID: DiscoverAndApplyForAMortgage Stage ID: Complete

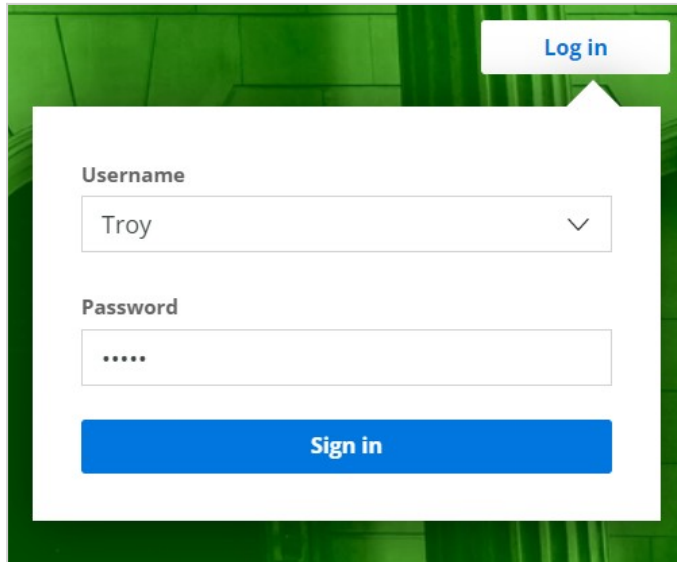
Customer is in this stage when: [?]

Group ANDs [v] [⋮]

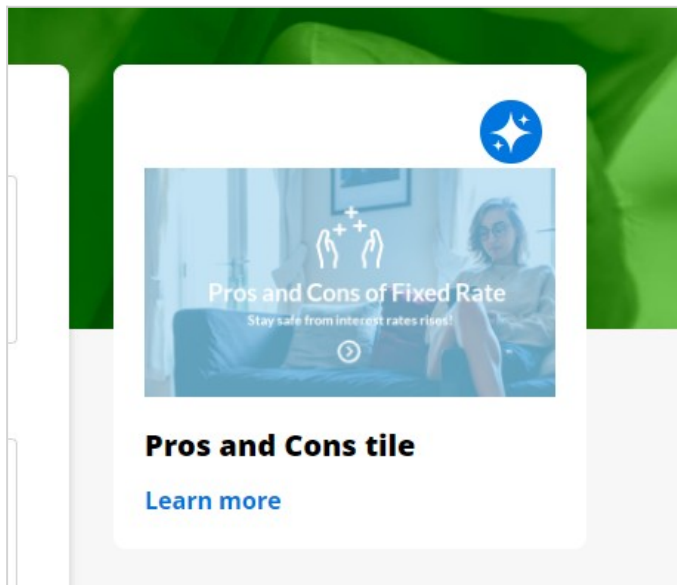
Customer [v] Entry Criteria Discov... [v] has results for [v] Engaged in Explore [v] [↗] [+] [🗑️]

[Cancel] [Submit]

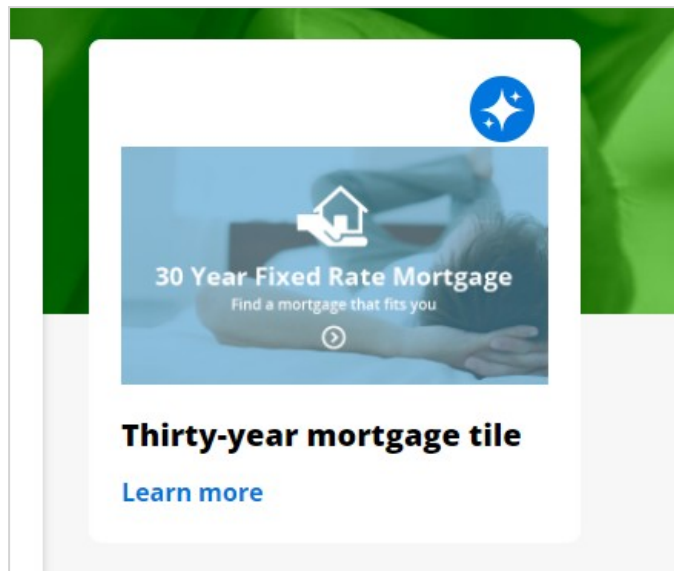
Next, you specify an analogous condition for the **Talk to advisor** stage to check whether the customer engages positively with one of the offers in the **Explore products** stage. After you set up these entry criteria and save them, you can test the customer journey.



Test the first customer journey for U+ Bank on the bank website. You can follow the progress of the U+ customers in the Customer Profile Viewer, as shown:



Follow Troy as he first logs in to the U+ Bank website and begins to engage with the Discover and Apply for a Mortgage customer journey. Troy sees the first web tile for one of the **Awareness** stage actions. This triggers the recording of the impression. As you can see in the **Customer journey** section of the Customer Profile Viewer, Troy is now in the **Awareness** stage.




Troy clicks the banner, meaning he is engaging the awareness stage. This action makes Troy eligible for the next stage of the journey. As a result, the next time he logs in, he can see a mortgages action, as prioritized by the AI, from the possible actions from the current Explore stage. You can see the results of this interaction in Customer Profile Viewer that shows that Troy is now in the **Explore Products** stage.

Later, Troy logs back into the U+ Bank website and shows interest by clicking the offer. As a result, he advances to the **Talk to advisor** stage and the next time he logs in, he sees the **Talk to Advisor** tile. You can see the progress in Customer Profile Viewer.

Customer Profile Viewer Actions

Type: Customer ID View

 **Troy Murphy** 14
 Email address: customer1@enablement.com
 Lifetime value: 900 (3 green dots)
 Churn risk: Medium (2 orange, 1 grey dot)
 Credit risk: High (3 red dots)

[Overview](#) | [Next best actions](#) | [Decision history](#) | [Interaction history](#)





Demographics

Gender	Age
M	26
Income	—

Recent interactions

All outcomes | Last 7 days

Wed May 18, 2022

- 09:52 AM  ThirtyYearMortga... IMPRESSION
- 09:51 AM  ProsAndConsTile CLICKED
- 09:51 AM  ProsAndConsTile IMPRESSION
- 09:50 AM  ProsAndConsTile IMPRESSION

Customer journeys

Discover and Apply for a Mortgage

✓ Awareness > Explore products > Talk to advisor

Entered stage: Explore products on 5/18/22 9:52 AM

Suppressed actions

No suppressed actions for this customer

Troy moves through the entire journey and can now make an informed decision on which mortgage he wants to purchase after the advisor contacts him. You see now how the underlying decision strategy works in the background for entry criteria to trigger the customer progress through the **Discover and Apply for a Mortgage** customer journey.

You have reached the end of this video. What did it show you?

- How to define and assemble a customer journey.
- How to set stage entry criteria.
- How to test customer journeys.
- How to monitor customer journey progress in Customer Profile Viewer.

Essentials of always on outbound

Description

Learn about the principles of always-on outbound customer engagement and its similarities and differences with traditional outbound marketing. Gain an overview of how the brain works in outbound customer engagement. Study a typical always-on outbound business scenario.

Learning objectives

- Describe the principals of always-on 1:1 customer engagement
- Compare and contrast traditional marketing concepts with always-on outbound
- Describe a typical always-on business use case in the outbound channel

Working of the always-on brain in outbound

What do customers expect from business?

Customers are more empowered than ever before. As a result, they have very high expectations of the experiences they receive from their service providers. These experiences must be meaningful, consistent, and personalized across every channel they interact with.



In alignment with customer expectations, Pega Customer Decision Hub™ is designed to deliver the right message, to the right customer, at the right time, on the right channel.

This approach not only provides a better experience for customers, it also helps the business improve its customer relationships over the long term.

The always-on brain

Pega Customer Decision Hub is the “always-on brain” that acts as a single, centralized decision authority.



The always-on brain carries out 1:1 customer engagement across channels.

“Always-on” means that the brain constantly monitors each customer’s context across channels. When it detects a need, it selects and delivers the appropriate offer, retention message, or service recommendation in the customer’s preferred channel.

The brain uses data about the customer, including past behaviour, profile information, and contextual data that may, for example, stream from their device, as input. It leverages advanced AI techniques to predict customer context, propensities, and relevance. It also uses stored decision strategies to trigger messages and actions at the moment the customer will be most receptive to them. As a centralized resource, the brain enables you to deliver consistent and personalized Next-Best-Actions across all channels.

Customer engagement channels

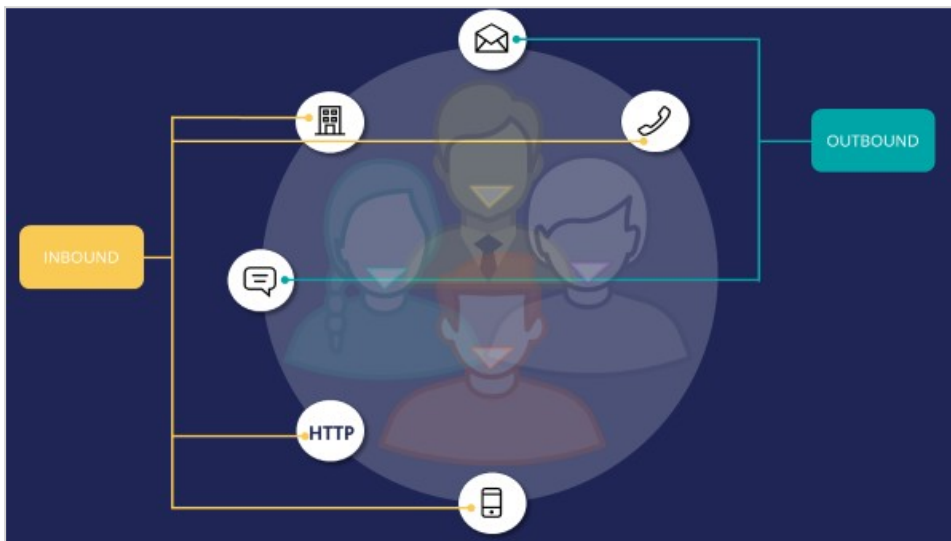
Customer interactions take place on various channels, which can be broadly classified as “owned” and “paid”.



Owned channels are the channels owned by the business, such as a self-service web portal, contact center, or mobile application. In these channels the business has more control over customer engagement activities.

Paid channels are owned by a third party, for example social media platforms such as Google and Facebook. Businesses have to pay to use these channels for customer engagement activities.

Owned channels can be further classified as inbound and outbound channels.



An inbound channel is used when a customer approaches the business, via, for example, a self-service web portal.

An outbound channel is used by the business to send a message to the customer, via, for example, email or SMS.

The flow of information and the mechanics of delivering Next-Best-Actions is different in inbound and outbound channels.

Inbound interactions

When a customer interaction takes place on an inbound channel, the channel identifies the customer, then the brain evaluates the Next-Best-Action for that customer and sends the result back to the same channel, in real-time. The customer's response to the action is recorded by the system, and the cycle repeats.



Outbound interactions

Always-on outbound means that the brain sends outbound messages, via email or SMS, to customers only when it's appropriate.

There are two types of outbound interactions: always-on actions and one-time actions.

Always-on actions

Always-on actions are typically the actions that are available for the Next-Best Action framework (Always-on brain) to select and deliver the appropriate communication to the customer.

This can happen in three ways:

- a. Promotional actions (Default):** The brain is configured to proactively send out messages on a scheduled basis when relevant. It evaluates the Next-Best-Action for each customer identified in a segment, which contains the potential list of customers to whom you want to send messages. In most cases, customer responses arrive after the message has been sent. When customers respond, the responses are recorded in the system and used to evaluate the next-best action. The always-on

brain considers all the engagement policy conditions, applies arbitration, outbound action/channel limits, and contact policies while determining the appropriate action and it delivers the top-ranked action. Promotional actions are the default method of communication.

b. Transactional actions: When a customer event is detected that is significant to the business, for example, a customer abruptly ends an online transaction, Customer Decision Hub receives the data from the event stream. The brain then evaluates the next-best action and sends the resulting action, such as a notification email with a link to continue the incomplete transaction, to the customer. In this case, the system triggers the outbound message only for the customer for whom the event occurred. Transactional actions are regulated by engagement policies, arbitration factors, action limits, and contact policies. However, the outbound channel limits are disregarded, as long as it is one of the qualified actions.

c. Mandatory actions: In certain situations, it might be necessary to inform certain customers about changes such as an increase in interest rates or network outages. These types of communications are referred to as mandatory actions, that must be sent out promptly. These actions are regulated by engagement policies and contact policies. Arbitration, outbound channel limits, and outbound action limits do not apply to these actions. However, contact policies are still applied to these actions to ensure the same actions are not delivered repeatedly.

Here’s an illustration of which policies and constraints apply to different outbound interactions.

		Eligibility/ Applicability/ Suitability conditions	Arbitration	Outbound channel limits	Outbound action limits	Contact policies
Always-On actions	Promotional (Default) actions	✓	✓	✓	✓	✓
	Transactional actions	✓	✓	✗	✓	✓
	Mandatory actions	✓	✗	✗	✗	✓

One-time actions

One-time actions are typical batch communication that all or a subset of customers receive at a specific date and time. These actions are executed in a Business Operations Environment. No engagement policies or constraints are applied to these actions, and everyone receives the same action.

a. One-time communication: In exceptional circumstances such as a planned maintenance activity in certain areas, you want to contact all or a subset of your customers outside of the regularly scheduled updates. In such cases, you can initiate a one-time communication, which does not coordinate with the outbound schedule. This type of communication typically applies to urgent or time-sensitive matters and does not repeat automatically.

		Eligibility/ Applicability/ Suitability conditions	Arbitration	Outbound channel limits	Outbound action limits	Contact policies
Always-On actions	Promotional (Default actions)	✓	✓	✓	✓	✓
	Transactional actions	✓	✓	✗	✓	✓
	Mandatory actions	✓	✗	✗	✗	✓
One-time actions	One-time actions	✗	✗	✗	✗	✗

Traditional marketing versus always-on outbound

Overview

As more organizations switch from traditional marketing approaches to 1:1 customer engagement, there is a need to compare and contrast the terminologies used in the two approaches.

Here are some key terminologies and concepts used in traditional marketing, translated into concepts in the always-on, Next-Best-Action approach.

Marketing campaigns and programs

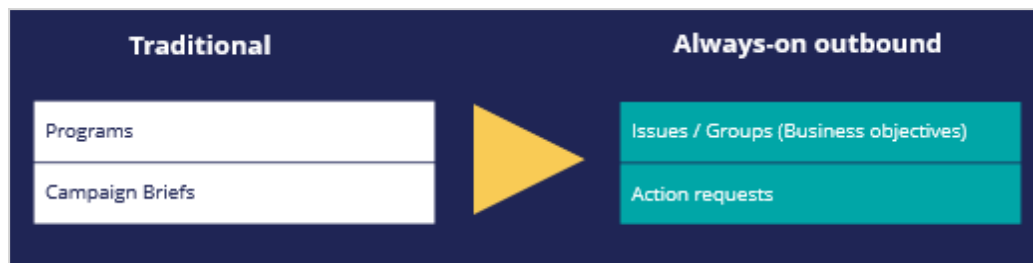
In traditional marketing, a marketing program is a set of activities designed to meet marketing objectives by grouping campaigns into logical business outcomes.

In the always-on approach, marketing programs translate into hierarchies of **Issues** and **Groups**. You can use your existing marketing objectives and desired outcomes to define the Issue/Group hierarchies that best meet your needs.

For example, a marketing program could be designed to address the business objective of decreasing dormancy in credit card usage. In this case, you would design a series of marketing campaigns that target infrequent credit card users, encouraging them to use their cards with various offers such as additional bonus points or discounts. In the always-on approach, this translates into defining the Issue as, in this case, *Dormancy*, and the Group as *Promotions*. Under this group, you would then create actions called *10% additional bonus points*, or *5% discount*, and so on.

As a marketing operations person, you receive campaign briefs, which are documents you can use to create **Actions**.

A campaign translates into an action or a set of actions delivered to qualified customers. In the always-on approach, you put an action into or out of play by changing its availability setting. The advantage in the always-on approach is that any adjustment to an action will propagate across all Issues and Groups, which means that you are optimizing across all programs and campaigns at once.



Segmentation

Traditionally, you would use the term segment, audience, or population interchangeably to refer to the target audience of a Campaign, and you would define granular criteria to identify this audience.

In the always-on approach, audience segmentation translates into engagement policies combined with Artificial Intelligence (AI).

The image compares traditional segmentation criteria with always-on outbound policy rules. On the left, 'Traditional' segmentation is shown as a series of intersecting filters: 'Onboarded' (253 customers), 'Average income' (53 customers), 'High balance' (23 customers), and 'Young' (9 customers). On the right, 'Always-on outbound' shows four policy rules: 'Eligibility' (checked), 'Applicability' (checked), 'Suitability' (checked), and 'Targeting' (unchecked).

Traditional	Always-on outbound
<p>Onboarded Start with : CUSTOMERS : 253</p> <p>A LifeCyclePeriod contains 'Onboard'</p>	<p>E Eligibility <input checked="" type="checkbox"/> (LifeCyclePeriod is equal to Onboard)</p>
<p>Average income Intersect : CUSTOMERS : 53</p> <p>A Income between '1000' and '3000'</p>	<p>A Applicability <input checked="" type="checkbox"/> No group criteria defined</p>
<p>High balance Intersect : CUSTOMERS : 23</p> <p>A AverageBalance greater than '1000'</p>	<p>S Suitability <input checked="" type="checkbox"/> (AverageBalance is greater than 1000)</p>
<p>Young Intersect : CUSTOMERS : 9</p> <p>A Age between '25' and '45'</p>	<p>T Targeting <input type="checkbox"/> (Age is greater than or equal to 25) or (Age is less than or equal to 40) and Income is greater than or equal to 1000) or ((Income is less than or equal to 3000)</p>

By understanding the purpose of traditional segments, you can translate them into eligibility, applicability, suitability and contact policy rules.

When selecting targets, any criteria that are not based on clear business rules should be avoided. For example, targeting a specific demographic such as an age range or income level based on market research or intuition is not necessary. In the always-on approach, it's

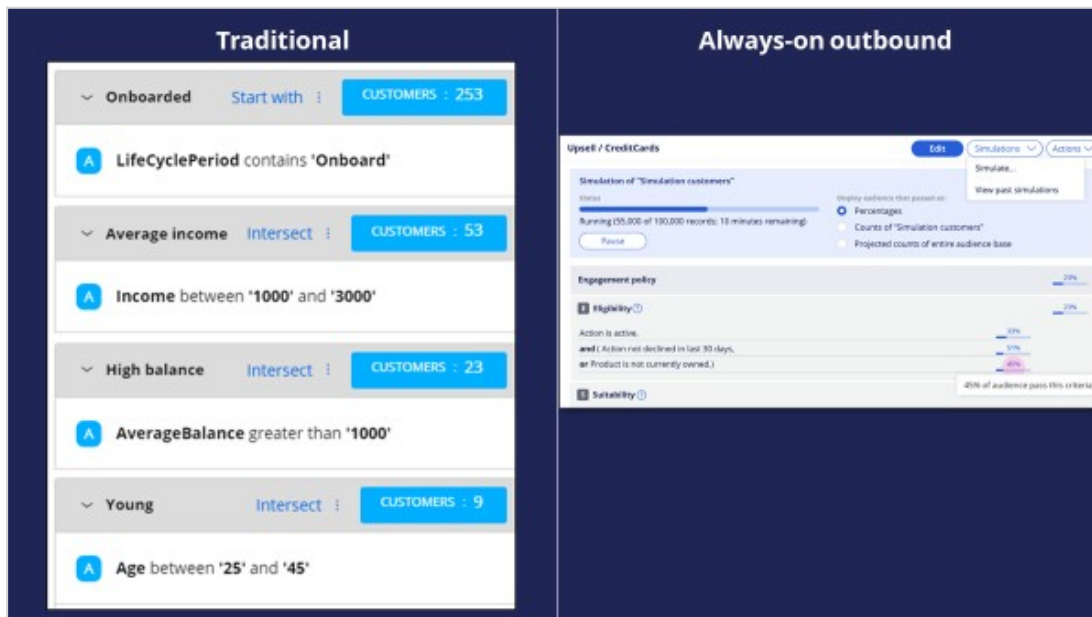
the job of the AI to choose the right action for each customer, once all disqualified actions are filtered out.

In the above example, the lifecycle period criterion is translated into an eligibility condition as it is a definite requirement of the business. The Average balance criterion is translated into a suitability condition as the business thinks the group of actions will be appropriate for customers who have such a balance. However, targeting an age range or income-based demographic are not clear business requirements. These conditions in the traditional approach are coming more from intuition and some offline analysis. Therefore, in the new always-on outbound approach, these conditions are not part of engagement conditions but are left to the AI. The AI will take care of finding the right customer segments that are more likely to have a positive response to the configured offers.

Segments, in the new approach, are used only to identify the starting population for an always-on outbound schedule (a multi-level campaign which initiates outbound actions). The starting population contains the list of potential customers to whom you want to send messages. For example, the starting population could be all customers who have opted in to receive promotional messages.

Segmentation reports

Traditionally, you might have used segments to generate reports on customers who match specific criteria.



In the always-on approach, you use reports generated by the simulation facility in Next-Best-Action Designer to:

- Understand customer counts.
- Validate the effect of changes to engagement policies and AI controls.
- Visualize the action mix.

Volume caps

When dealing with any form of outbound communication, there is a need to limit the number of communications both on a per-customer basis and in terms of total messages generated by the system over a given time period.

In the always-on approach, you use volume constraints, contact limits and suppression rules to set caps on communications.

Volume constraints

Customer contact limits

Channel	Contacts per customer	Duration
Email	1	Weekly
SMS	1	Weekly

Contact policy library

- > 7-day action impressions: Track Impressions for the action over the past 7 days
- > 7-day group clicks: Track Clicks for all actions in the group over the past 7 days
- ∨ 30-day sent count: Track Pending for all actions in the group over the past 30 days
If there are 2 Pending for Email treatments, suppress the action for 60 days

How should multiple actions for a customer be treated?

- Individually for each action
- As a group for the context (output all actions)
- Return any action that does not exceed constraint

Reset constraint counts

- Individually
- All at once

Maximum 50 with Channel: Email, Action: StandardCard

Maximum 100 with Type: Master

+ Add definition

Scheduling

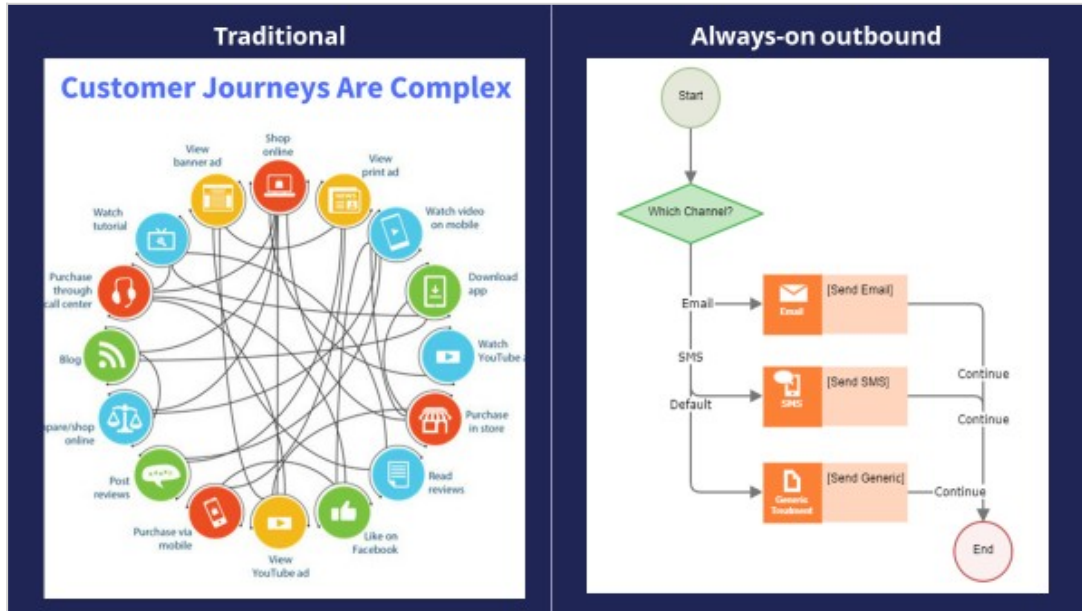
In a traditional marketing approach, there are several options to control the campaign schedule. For example, you can run a campaign as a one-off, or as recurring daily, weekly or monthly.

In the always-on approach:

- The schedule is always-on, it runs at a set frequency, for example on a daily basis.
- Actions are marked as in or out of play using the action 'availability' option.
- Emergency schedules can be set-up as secondary schedules.
- Contact policies help prevent communication fatigue in customers.

Journey guidance

Traditional campaign journeys can be long and complex. They are designed as deterministic journeys in which one campaign triggers the next, which triggers a third, and so on.



The always-on approach turns managing complex journeys into managing experiences:

- Logic in the channels should be minimized to focus on the overall experience. The number of steps involved in outbound communication, represented by action flow shapes, should be limited to 2 or 3 per channel.
- Next-Best-Action combines many actions to optimize experiences. Let the Pega brain decide the next step.
- Action flows shouldn't contain decision logic (that's in the brain) and should not try to adapt to customer or business changes.
- Use business hierarchies (Issues/Groups) to organize the business outcomes and actions that collectively make up experiences.

Business use case: Always-on outbound customer engagement

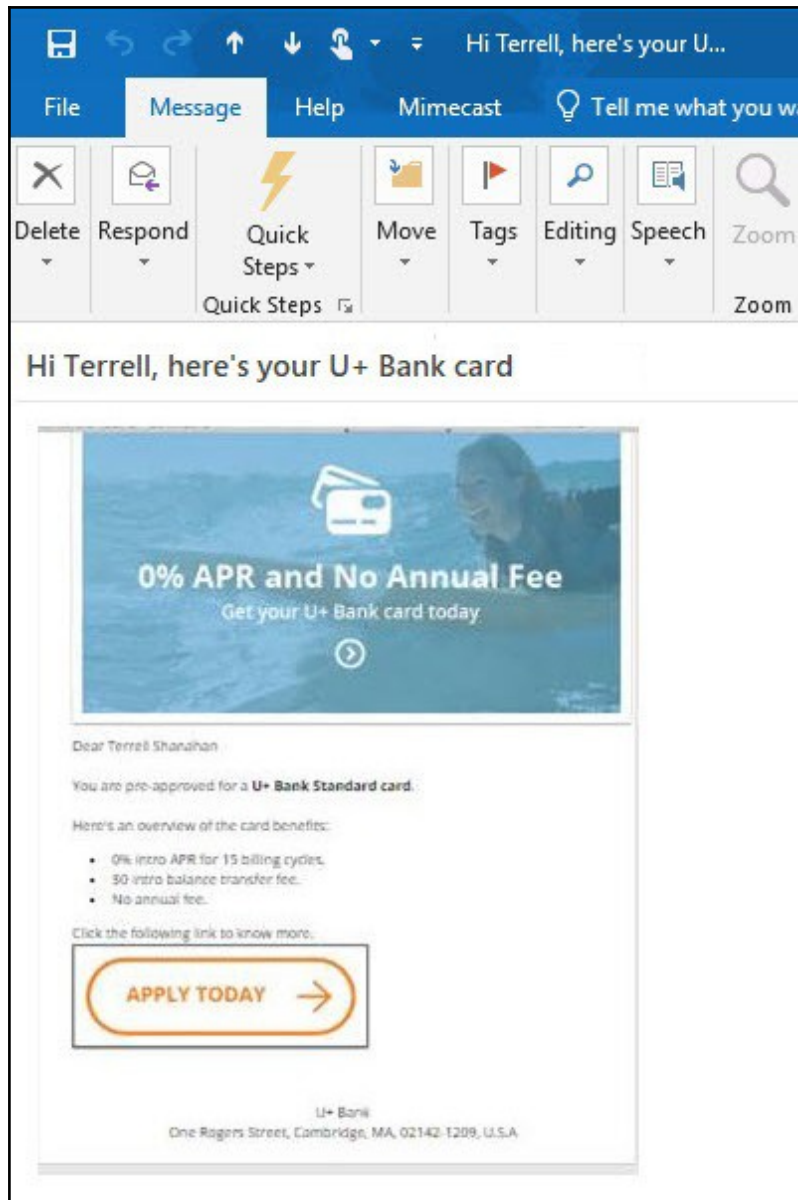
Introduction

U+ Bank has recently implemented the cross-sell on the web microjourney with the goal to increase the engagement with their customers on the web channel. The bank uses Pega Customer Decision Hub™ to recommend relevant banner ads to its customers when they visit their personal portal.

Customer Decision Hub combines customer information, information about their past communications with the bank, current contextual information, business rules and the predicted customer behavior, as calculated by Artificial Intelligence (AI) to select the Next-Best-Action, which in this case is the best sales offer for each customer.

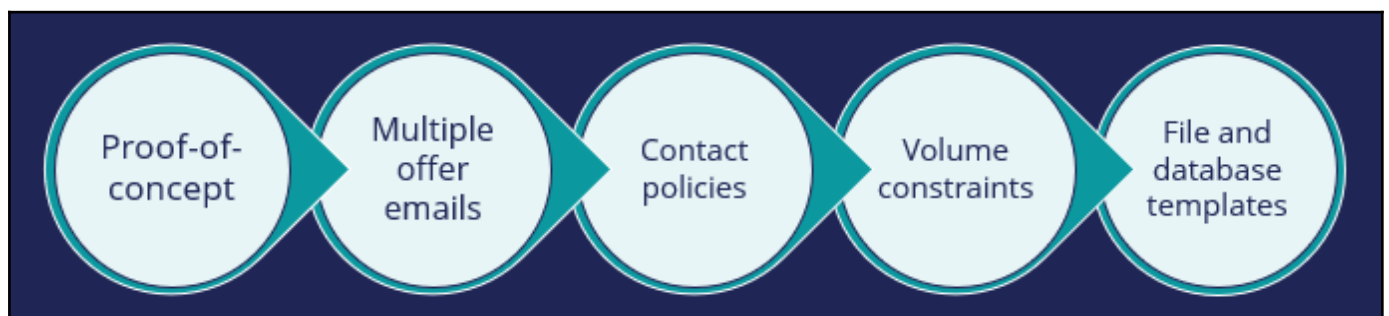
Satisfied with the results of the first microjourney, the bank now wants to extend these Next-Best-Action recommendations to be delivered via one of their outbound communication channels, email.

In this microjourney, the main intent of U+ is to increase the customer engagement on email channel. This can be measured by the click-through rate. A click-through metric is recorded when the customer clicks on the 'APPLY TODAY' button in the email received.



Implementation phases

The marketing operations team plans to implement the requirement in multiple phases.



The first phase is a proof-of-concept phase. In this phase, the goal is to send a credit card offer via email to qualified customers. This requires adding an email treatment and configuring the flow to an existing action, enabling the email channel, and configuring the always-on outbound schedule.

As a result of this phase, the Standard card offer details will be sent to qualified customers in an email. For example, if a customer, Stephen John, qualifies for the Standard card offer, the always-on outbound will send him an email with the offer details. He can click on the 'APPLY TODAY' button, it takes him to the credit cards page on the bank's website where he can read more details about the offer and apply for it, if interested.

In the second phase, the bank plans to include more credit card offers to be promoted via email. For example, if a customer is eligible for more than one credit card offer, he will be presented with the top offer he qualifies for. The prioritization uses AI and other factors such as business weights and levers to select the top offer for a customer.

In the third phase, the bank would like to introduce contact policy rules to ensure that a customer does not receive the same offer message twice within a period of time. For example, a customer will not receive an offer email for 60 days if one was sent in the past 30 days.

In the fourth phase, the bank would like to apply some limitations on the outbound volume due to certain operational constraints. The IT department of the bank has imposed a restriction on the number of emails that can be sent in a day. The marketing team has been asked to apply some limitations on the outbound volumes of the email.

The fifth phase is to share action details with a third-party email distributor. To comply with the security and spamming regulations, the bank has decided to use a third-party email delivery service instead of using the inhouse email server. As a result, the action details need to be written to a database table. The action details will then be shared with a third-party email distributor.

In summary, you will learn more about how U+ bank will leverage the always-on outbound capability in this phase of the project.

Sending offer emails

Description

Learn how to send offer emails to qualified customers. Reuse existing actions and customer engagement policies configured in Next-Best-Action Designer and extend them for outbound customer engagement in the email channel.

Learning objectives

- Configure the always-on brain to send offer emails

Sending offer emails

To market your offers, you can send offer emails to qualified customers. Pega Customer Decision™ Hub enables you to re-use existing actions and customer engagement policies configured in Next-Best-Action Designer and extend them for outbound customer engagement in the email channel.

Transcript

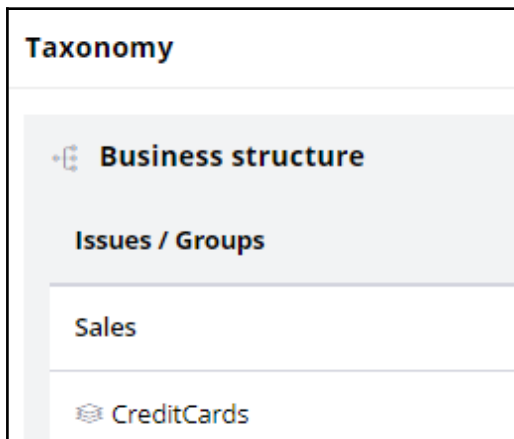
This demo shows you how to send offer emails to qualified customers.

U+, a retail bank, wants to use Pega Customer Decision Hub to send emails related to a credit card offer to qualified customers.

The email contains the offer details and a call to action. When a customer clicks the **APPLY TODAY** call to action in the email, they are directed to the credit cards page on the bank's website, where they can learn more about the offer and apply if interested.

This is the Pega Customer Decision Hub portal. Before configuring an action to send offer emails to qualified customers, look at the Next-Best-Action Designer configured by U+.

The **Taxonomy** tab contains the business hierarchy of issues and groups. Every action presented to the customer is classified within an issue/group hierarchy. In this case, the bank wants to do cross-sell/upsell of credit cards. As a result, the bank created one Issue called **Sales** and a Group called **CreditCards**.

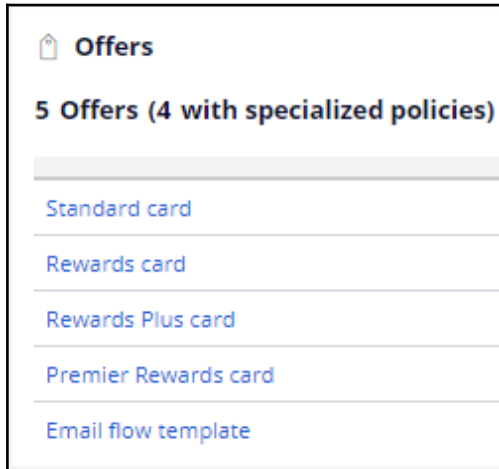


Now, navigate to the **Engagement policy** tab to view the business rules and actions defined for the CreditCards group.

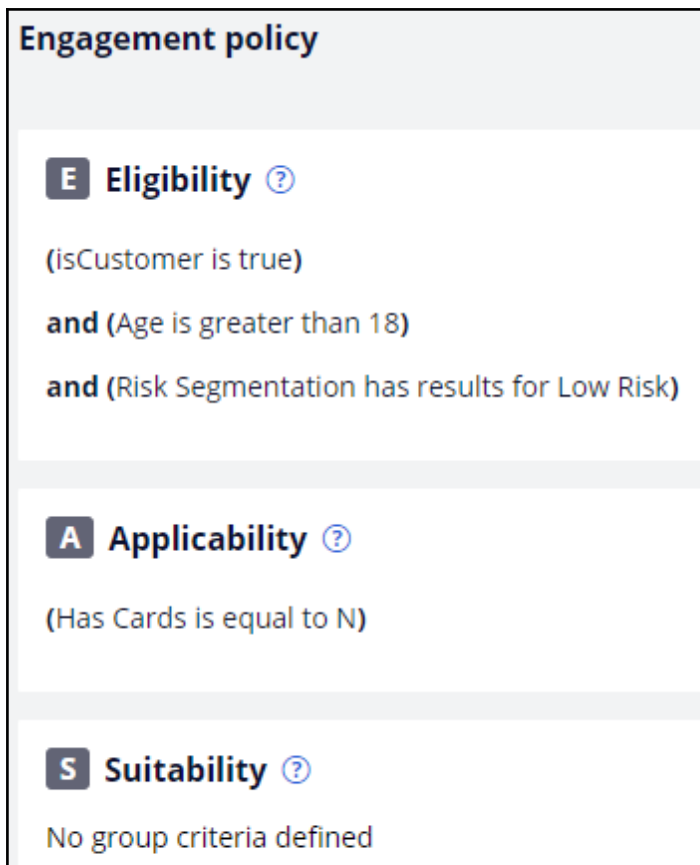
These engagement policy conditions determine which customers qualify for the actions. For example, the eligibility condition **Age is greater than or equal to 18** implies that only customers who are equal to or over the age of 18 are eligible for the credit card offers.

Also, credit card offers are applicable to customers only if they do not currently own a credit card.

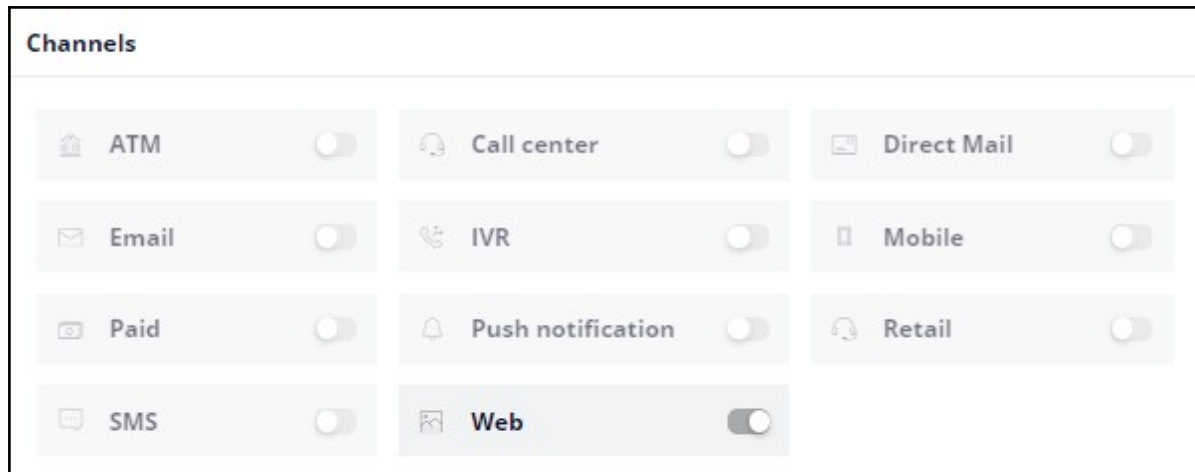
Notice that a consultant has already created a few actions under the Sales/CreditCards hierarchy.



These actions correlate with the credit card offers that the bank wants to promote.

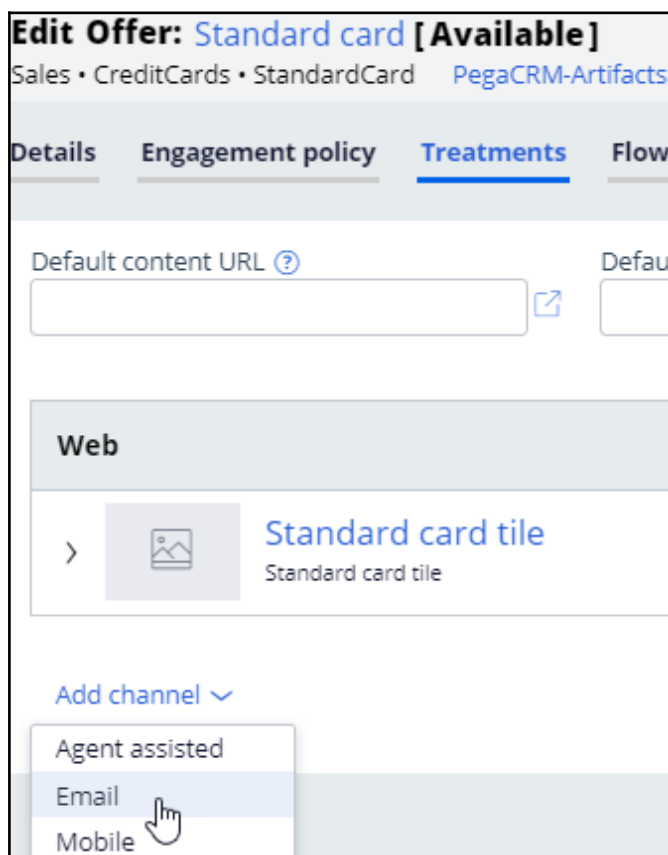


Currently, the bank promotes credit cards on the web channel only. The bank wants to extend this promotion to the email channel.

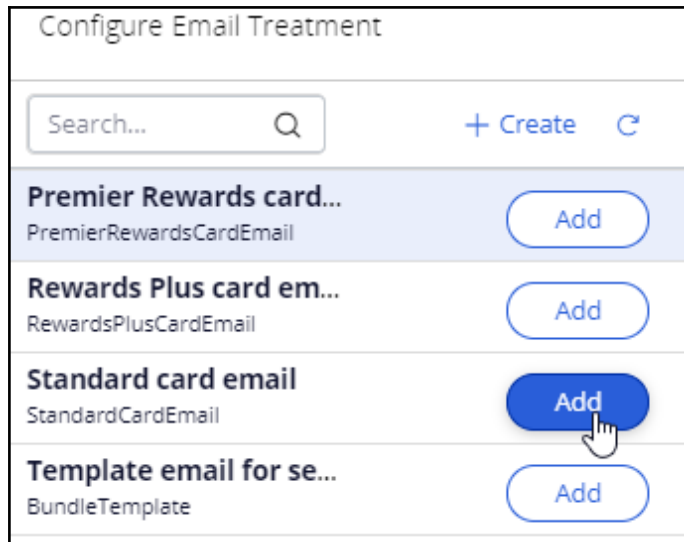


The bank wants to begin by sending the Standard card offer emails to qualified customers. To send emails in an always-on outbound environment, you need to add an email treatment to the action.

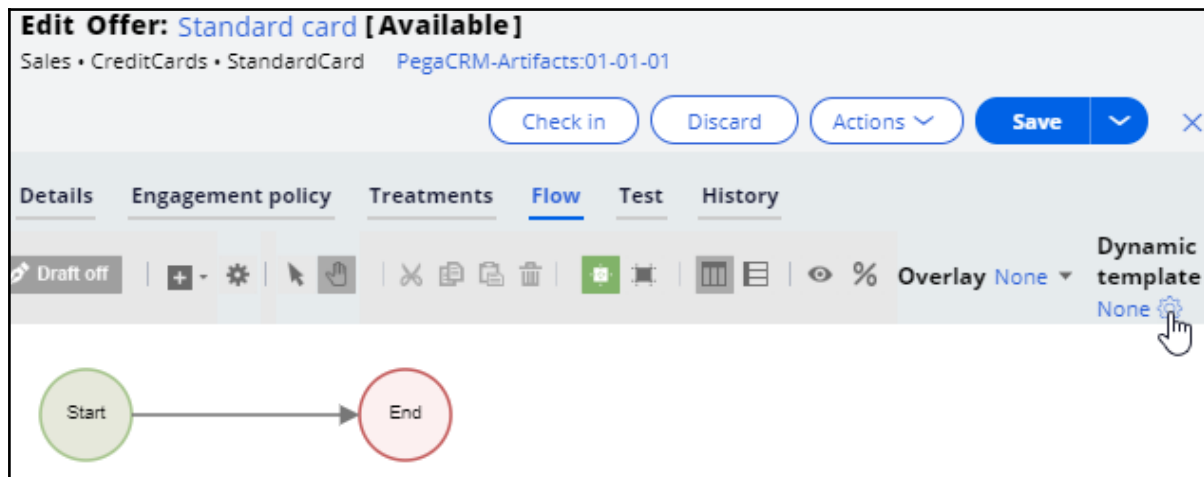
Open the **Standard card** action to add an email treatment.



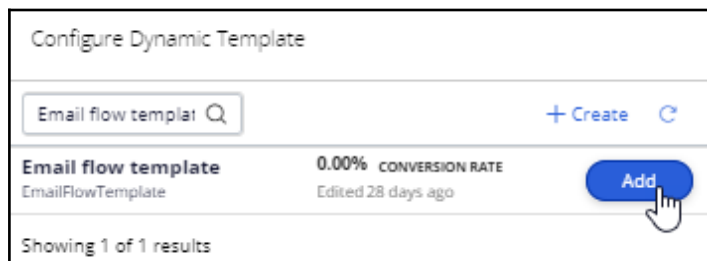
You can use an existing treatment or create a new one. In this case, you use an existing email treatment.



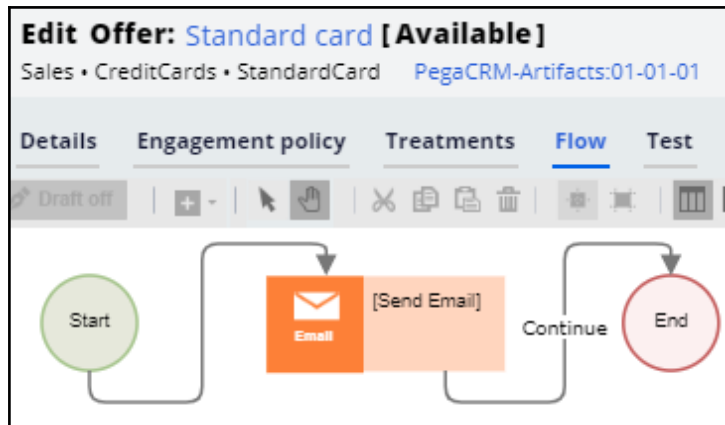
Next, you must configure the action flow. The action flow is a flow diagram that contains the sequence of steps carried out during the lifecycle of an action. You can create action flows once and reuse the flows across actions by using the **Dynamic template** option.



Now, click **Dynamic template** to add an action flow template.

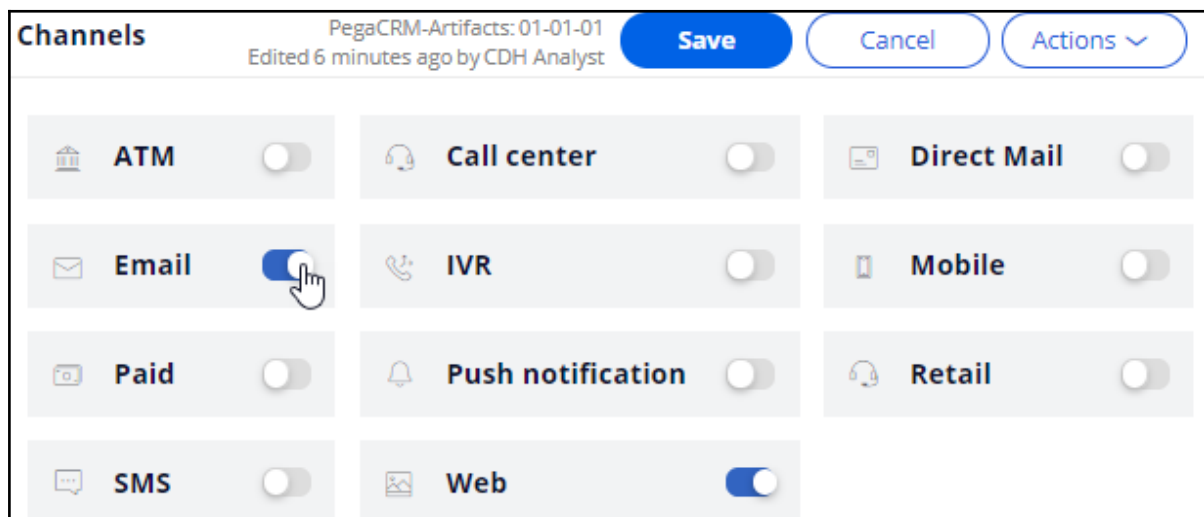


Notice that the flow has only one key shape: the **Send Email** shape. It represents the outbound email delivery.



Now, on the **Channels** tab, you can enable the channels and triggers that invoke a Next-Best-Action evaluation.

In this case, as U+ wants to send out emails, enable the **Email** channel.



In the always-on outbound paradigm, the brain evaluates the next best action for each customer identified in a segment, which contains the potential list of customers to whom you want to send messages. The system can send these messages at a regular frequency (for example, daily). You configure the **Primary schedule** to set the frequency.

The schedule has two mandatory configurations: **Recurrence** and **Starting population**.



To configure an outbound schedule, first, set the required **Recurrence** frequency. In this case, retain the default frequency, *Daily*. Then, select the **Refresh the audience** check

box. Enabling the **Refresh the audience** option ensures that the latest customer information is always used when running the schedule.

Configure outbound schedule

Recurring by*

Daily

Every weekday

Every day or every nth day

Start

3/12/2020 2:10 AM

End

No end date

Set number of occurrences

End by a date

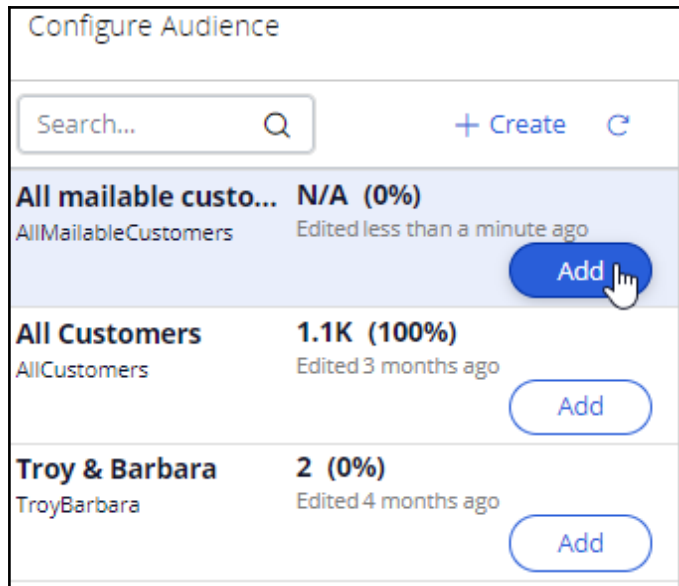
Refresh the audience

Refresh the audience for each scheduled occurrence

Refresh the audience only for the first scheduled occurrence

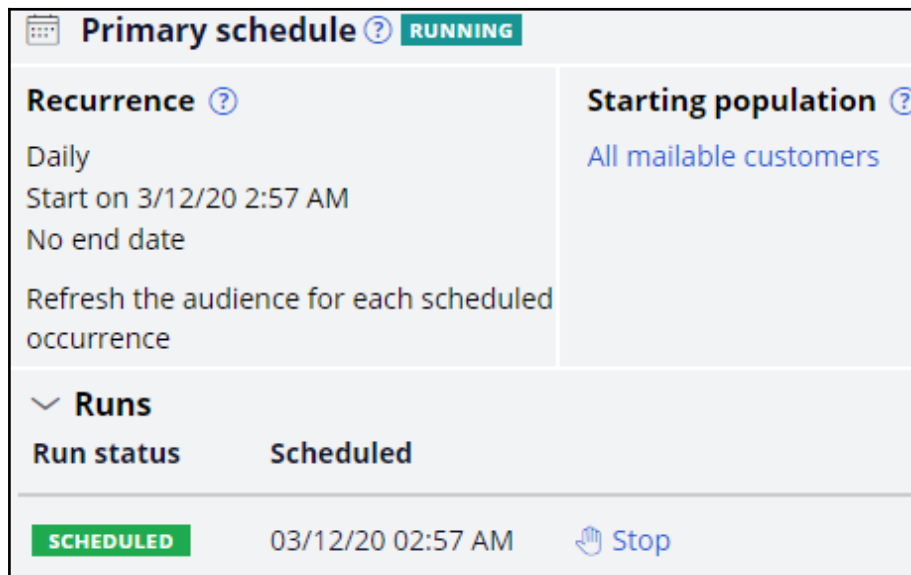
Write results using a database template

Next, select the segment that contains the list of customers to be considered by the brain for outbound messaging. Typically, this is a large population of all customers to whom you potentially want to reach out. In this case, select the **All mailable customers** segment.



With that, all the necessary configurations for this scenario are complete.

Run the schedule. When the date and time for a scheduled run are reached, an entry is displayed with the status **Scheduled**, which indicates that the run is about to begin.



When you refresh, the status changes to **Running** once processing begins.

< Runs
 Run status Scheduled
 RUNNING 03/12/20 02:57 AM Stop
 > Upcoming runs

You can refresh to see the latest status. Once the run is complete, you can see the results of the run.

In the completed run results, you can see the number of customers selected in the **Audience** stage, which is the count of customers selected by the Segment.

You can also see the number of actions identified by the strategy, which is the count of customers selected by the brain.

The count in the **Initiated** section represents the number of actions initiated for outbound messaging. The count in the **Result** section is the number of actions that have completed all the steps defined in their action flow.

You can also see the actions in the current run, which the brain selected.

< Runs
 Run status Scheduled
 COMPLETED 03/12/20 02:57 AM
 Audience 769 customers > Strategy 191 action results > Initiated 191 actions initiated > Result 191 completed
 03/12/2020 03:00 AM 03/12/2020 03:06 AM 03/12/2020 03:06 AM
 Action Count
 StandardCard 191

Qualified customers receive the email. When they click the **APPLY TODAY** call to action, they are taken to the preconfigured URL, which, in this scenario, is the Credit Cards landing page on the bank's website.

This demo has concluded. What did it show you?

- How to enable the email channel in Next-Best-Action Designer.
- How to add an email treatment to an action to have an email offer available for the outbound run.
- How to schedule and run an outbound run.

Defining the starting population

Description

The starting population is the potential list of customers you want the brain to consider for always-on outbound customer engagement. Learn how to create a segment and configure it with basic criteria to identify the starting population for Next-Best-Action outbound runs.

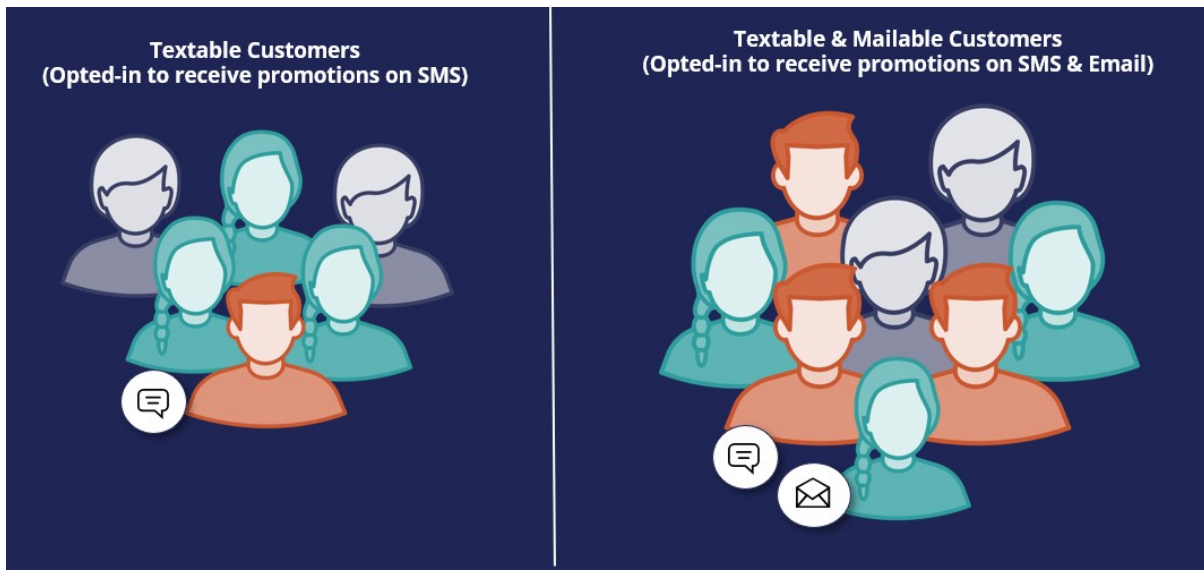
Learning objectives

- Describe the concept of segmentation
- Explain how to create a segment for the starting population of an outbound run

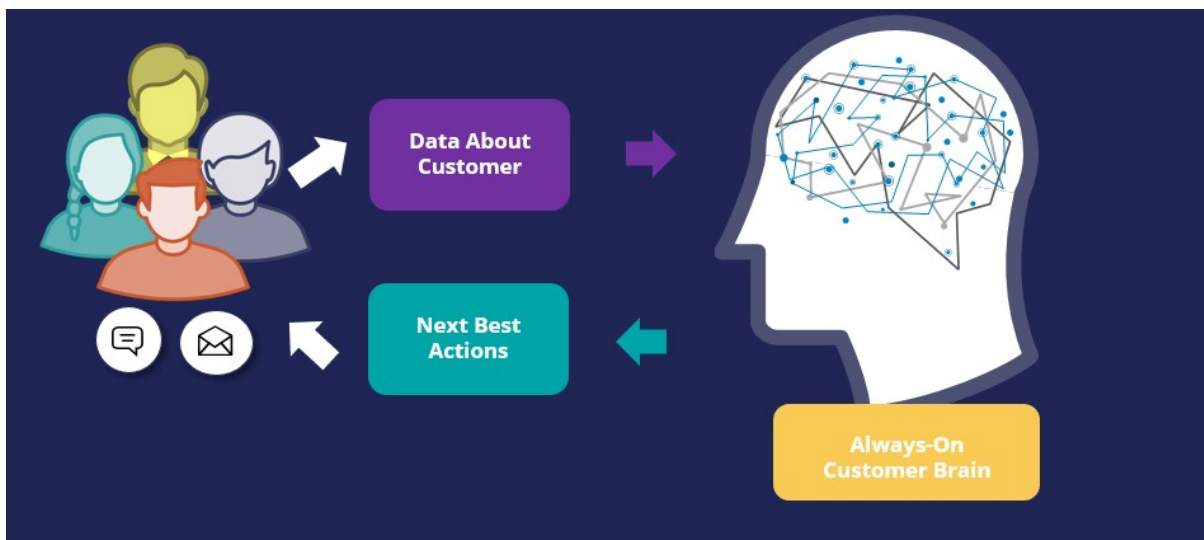
Segmentation

What are segments?

In Pega Customer Decision Hub™, the audience is a list of potential target customers, implemented using the segment component. A segment is a reference to a database table containing a list of customer IDs that match specific criteria.



In the always-on outbound paradigm, the Pega brain evaluates the Next-Best-Action for each of the customers identified in a segment, which contains the list of potential customers to whom you want to send messages.



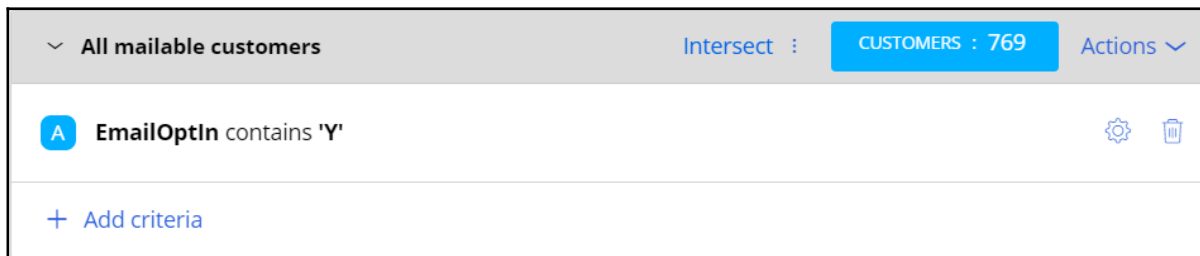
This set of customers is known as the starting population. The customers in this group will be further qualified through eligibility, applicability, and suitability criteria configured in Next-Best-Action Designer.

Criteria in a segment

To select a starting population, you apply a few high-level criteria to a segment. It's best practice not to get too granular at the segment level, rather use eligibility, applicability, and suitability rules configured in Next-Best-Action Designer to fine-tune your criteria once the starting population has been selected.

You can define one or more conditions that must be true for a customer to be included in a segment.

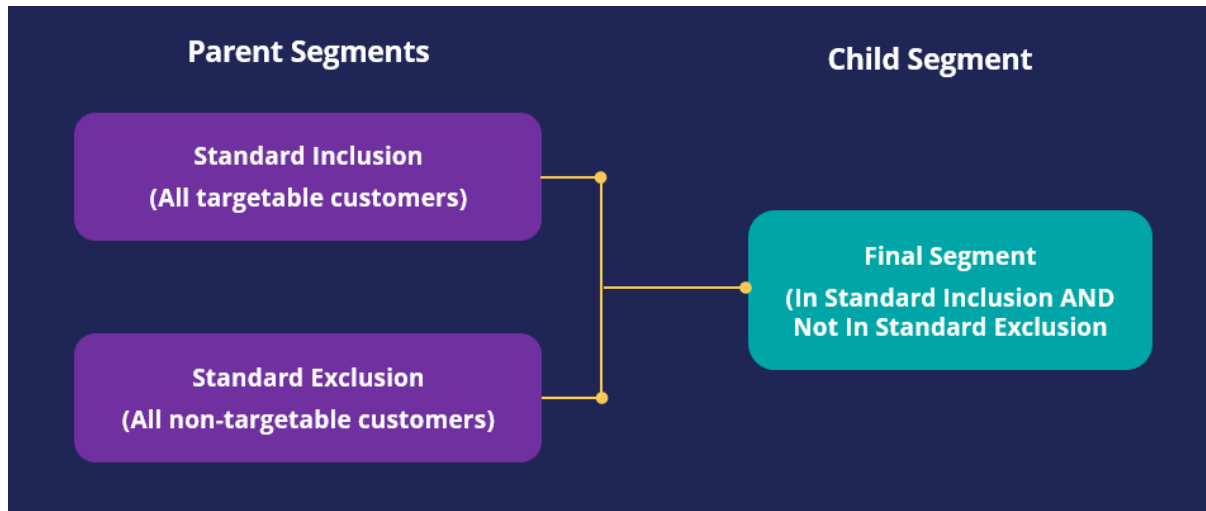
For example, to send emails related to a credit card offer to a target audience, you should first define a criterion that will filter for customers who have opted-in to receive promotional emails.



Segments hierarchy

A segment can reference a “parent” segment, enabling you to create a hierarchy of segments.

For example, you can create top-level segments that identify common lists of customers for exclusion or inclusion in various “child” Segments. Standard top-level segments allow global communication rules to be implemented by forming the basis of customer lists that automatically adhere to these rules. The rules can include, for example, standard exclusions, which are the core customers who should not be contacted, or standard inclusions, which are the core customers who may always be contacted. The resulting customer lists can be used as the starting point for creating the final segment.



If changes are made to global rules, the definitions of the standard top-level segments can be modified accordingly, and as a result, every segment that references them will automatically be re-aligned to the new rules.

Creating a segment

Business scenario

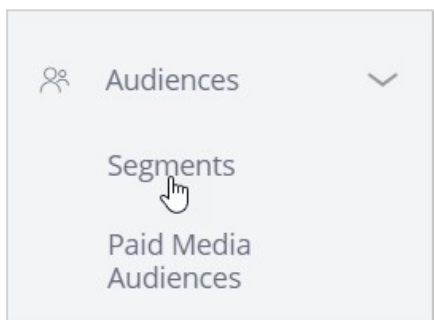
U+, a retail bank, wants to use the always-on outbound approach to send promotional emails to qualified customers. The bank wants to include all customers who have opted in to receive promotional emails as the starting population for the outbound schedule.

Creating a segment






To meet the business requirement, you need to create a segment with two criteria.

1. A criterion to reference the existing top-level segment, **All Customers**, which contains a list of all customers in the customer base.
2. A criterion to check if customers have opted in to receive promotional emails.

Segments are managed on the Segments landing page, which can be accessed from **Audiences** -> **Segments** in the navigation menu.



You can open existing segments or create a new segment from here.

Segments		 Create	Help 
Issue / Group	Type		 Filter
All issues / All groups	All types		
Showing 4 of 4 results			
	All Customers AllCustomers	1.1K (100%) customers <small>Refreshed 1 month ago</small>	Criteria Edited 4 month ago
	Troy & Barbara TroyBarbara	2 (0%) customers <small>Refreshed 3 months ago</small>	Criteria Edited 4 month ago

Provide a short description for the new segment, and the identifier is automatically populated when you tab out.

Create Segment
[Create and open](#)

Segment Record Configuration

All Mailable Customers

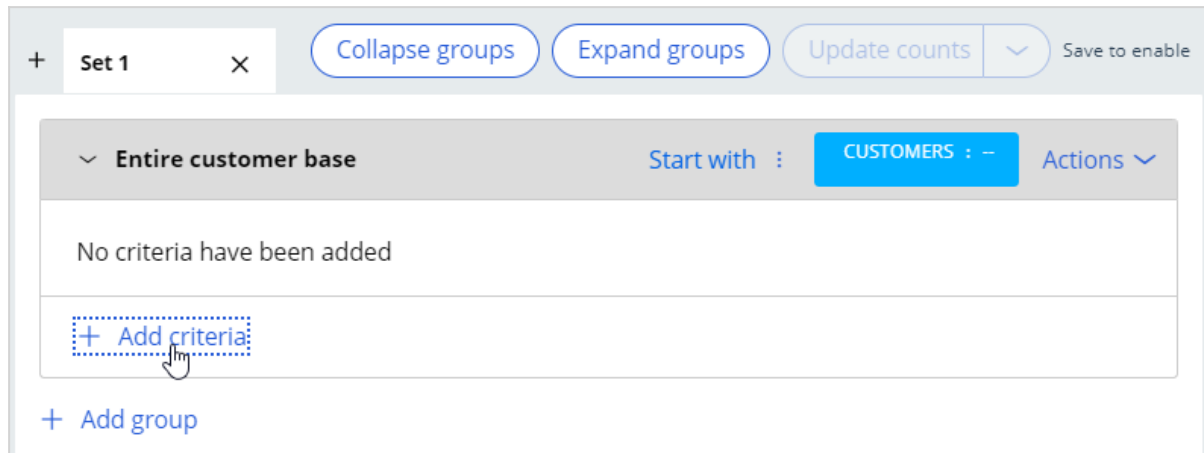
Identifier
 AllMailableCustomers [Edit](#)

A short description or title for this record

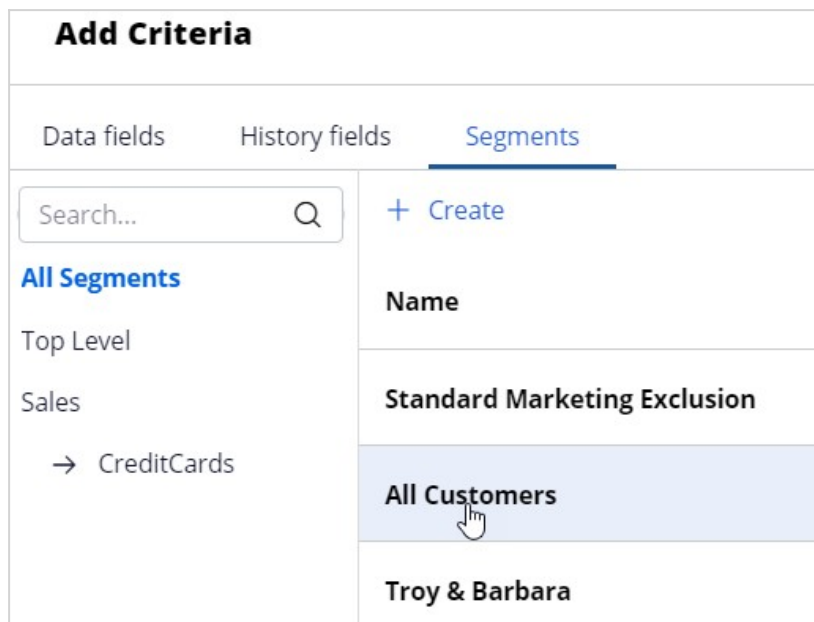
Entering the segment criteria

The **Design** tab is where you assemble the criteria required to identify customers of interest.

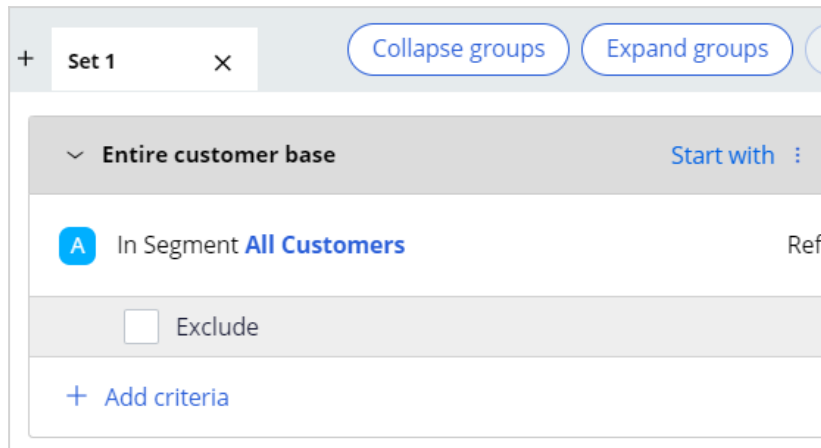
Segment criteria are grouped for ease of reading. So, before you can add the first criteria, add a criteria group.



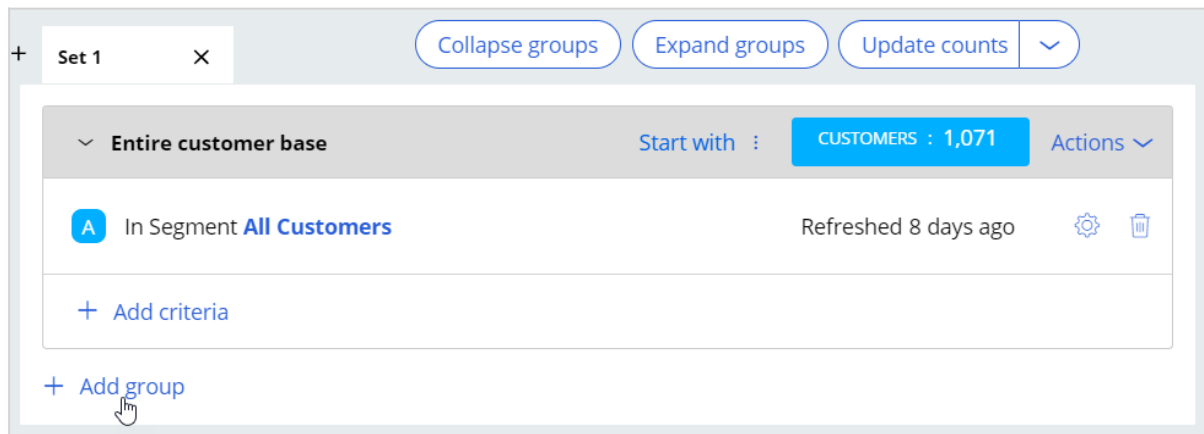
In the **Add Criteria** window, the **Segments** tab lists all available segments. Select the existing **All Customers** segment.



The **All Customers** segment is added as the first segment criterion. The **Exclude** option allows you to exclude the result of a criterion.



The next step is to add another criterion to identify customers who have opted in to receive promotional emails. You can add the new criterion in the existing criteria group or in a new group.



In the **Add Criteria** window, the **Data Fields** tab lists all available customer properties. You can either select the field directly or search for it by entering the partial name in the search field. Then, select the desired field from the search results. In this system, the field **EmailOptIn** contains the information about whether a customer has opted in to receive emails or not.

Add Criteria

Data fields History fields Segments

EmailOptIn

Customer
→ History

Name ▲

Customer.EmailOptIn

Once the field is selected, you can configure it to meet the business requirement. In this system, the value **Y** indicates that the customer has opted in, and **N** indicates that he/she has not opted in to receive promotional emails.

The screenshot shows a user interface for creating customer segments. At the top, there is a tab labeled 'Set 1' with a close button. Below the tab are three buttons: 'Collapse groups', 'Expand groups', and 'Update counts' with a dropdown arrow. To the right of these buttons is the text 'Save to enable'. The main area contains two criteria groups. The first group is titled 'Entire customer base' and is set to 'Start with : CUSTOMERS : -'. It contains one criterion: 'In Segment All Customers', which was 'Refreshed 1 month 13 days ago'. Below this criterion is an 'Exclude' checkbox and an 'Add criteria' button. The second group is titled 'All mailable customers' and is set to 'Intersect : CUSTOMERS : -'. It contains one criterion: 'EmailOptIn Not yet configured'. Below this criterion are three input fields: a dropdown menu set to 'Exactly', a text input field containing 'Y', and three checkboxes: 'Exclude', 'Include missing values', and 'Ignore case'. Below these fields is an 'Add criteria' button. At the bottom left of the interface is an 'Add group' button.

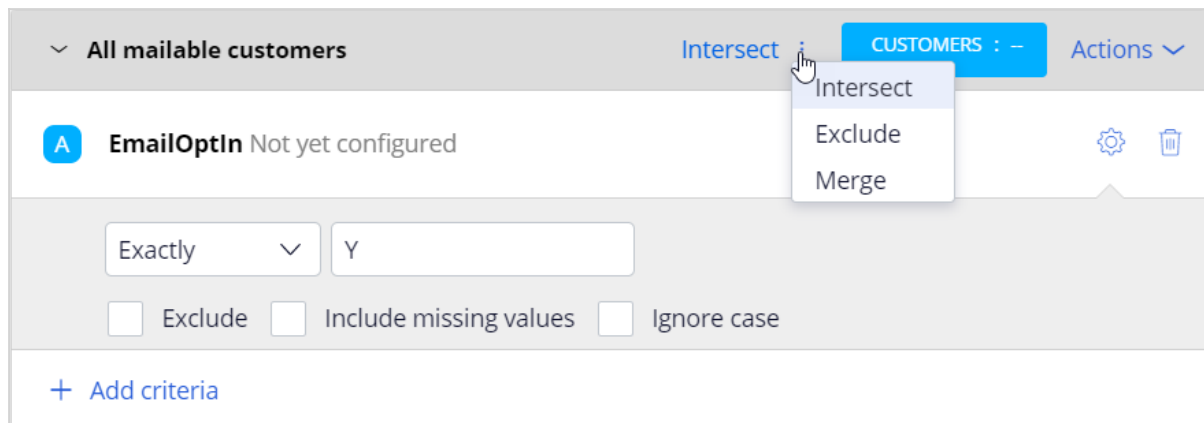
The **Include missing values** option is used to include results that do not have a value specified for the selected field.

The **Ignore case** option is used to perform a case-insensitive search. You can leave both options unselected for this scenario.

The **Update Counts** option is used to update the population count of the segment while you are building it.

Note: The **Update Counts** option is enabled when you save the segment.

You can combine the result of a criteria group with the result from the criteria group above it using one of the three options:



Intersect selects the customers the two results have in common.

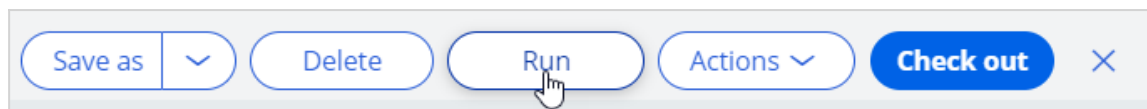
Exclude excludes the result of the current criteria group from the result above it.

Merge combines the results of both criteria groups.

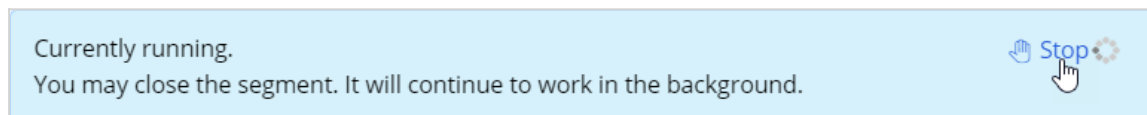
Save and run the segment

When you are done defining all the segment criteria, you can save and run the segment. Running the segment will populate the number of customers in a segment out of the total number of customers in your application.

Click **Run** to run the segment.



While the segment is running, you can stop the execution by clicking the **Stop** button.



Once the run completes, the number of customers in the resulting segment is displayed as **Population count**.

Design Options and Schedule Paid sync History

Population count: **769** of **1,071** [Show Customers](#)

More ▾



+ Set 1 × Collapse groups Expand groups Update counts ▾

▾ Entire customer base Start with : CUSTOMERS : 1,071
 A In Segment **All Customers** Refreshed 16 days ago

▾ All available customers Intersect : CUSTOMERS : 769
 A EmailOptIn exactly 'Y'

There are **1071** customers in the **All Customers** Segment, out of which **769** customers have opted in to receive promotional emails.

Click the **Show Customers** link to preview the list of customers selected for this segment.

Population count: **769** of **1,071** [Show Customers](#)  

The list is shown in a separate window.

CustomerID* ↑	ACCOUNT ID	Title	Full Name	Gender	Age	Language	MaritalStatus
14			Troy Murphy	M	26		Married
15			Barbara Stockton	F	32		Single
16			Joanna Williams	F	25		Married
17			Robert Walton	M	26		Single
18			John Smith	M	45		Single
19			Arnold Green	M	56		Single
CON-1			Layne McLaughlin	F	19		Single
CON-10			Ezekiel Funk	M	32		Single

Key segment configurations

On the **Options and Schedule** tab, there are two key settings under **Data refresh** that control how data in the segment is refreshed at run-time and what level of control a segment can exert over other segments.

Design
Options and schedule
Paid sync
History

Data refresh

Enable transactional refresh
Select this option for segments that are used for exclusion lists or transactional notifications to customers. Transactional refresh ensures correct segment population in the case of concurrent refreshes.

Enable refresh by parent segment
Select this option to allow other segments that have this segment as part of their criteria to automatically refresh this segment when they are run.

Enable child segment refresh
Select this option to allow the segment to automatically refresh its refreshable child segments when run.

Key Audience ⓘ

Make this segment a Key Audience (0 used of 20)
Select this option to allow this segment to be filtered against in various Discovery Tools.

Sampled segment

Select a sample of criteria results

When the **Refreshable Segment** option is enabled, the segment will be refreshed when the other segments that reference it are run.

The **Refresh Child Segments** option allows you to control the cascading effect of the refresh operation. The segment can automatically refresh data in its child segments if they are marked as refreshable.

Marking a segment as a **Key Audience** makes it accessible in various Discovery tools like Action Performance Tracker. This allows for the action performance analysis of outcomes, related KPIs, and metrics, providing insights into what worked and potential opportunities.

Note that due to computing resource constraints, a maximum of 20 Key Audiences are allowed.

The **Sampled Segment Options** allows you to select a random sample of a segment result to be selected as the final segment result.

Segment run history

On the **History** tab, you can view the details of previous execution runs. The history lists the runs triggered manually by a user, indicated by **Populate Segment Requested by: <user name>**. It also lists runs triggered by the outbound schedule in Next-Best-Action Designer, indicated by **Populate Segment Requested by: Program run**.

Design	Options and Schedule	Paid sync	History
Current run			
This segment is not currently running.			
Completed runs			
Run status	Information	Last run ↓	Count
COMPLETED	Populate Segment Requested by: Program Run	03/27/20 05:37 AM	769
COMPLETED	Populate Segment Requested by: CDH Analyst	03/27/20 05:20 AM	769

Defining an action for outbound

Description

A Next-Best-Action is selected by the brain after it considers customer engagement policies, business objectives and recommendations from the AI. An action can be presented to the customer on various inbound and outbound channels. Learn how to configure an action to make it suitable for outbound customer engagement and design a visually appealing email message using easy-to-use templates. Also, learn how to configure an action flow, which contains the steps involved in delivering an action to a customer.

Learning objectives

- Create an email treatment
- Design visually appealing emails using templates
- Configure actions with re-usable action flows

Email Treatments

Offer emails play a crucial role in attracting customers as part of a campaign over the email channel. With Pega Customer Decision Hub™, you can use email treatments to design visually appealing email messages, and then send your offers to qualified customers in an always-on outbound environment.

In Pega Customer Decision Hub™, email treatments can be Internal or External.

Internal email treatments

You create internal email treatments and email accounts for initiating email distribution within Pega Customer Decision Hub. You can design the email content either in Pega Customer Decision Hub or in an external HTML editing software, such as Dreamweaver, and Stripo. However, the final email assembling happens in Pega Customer Decision Hub.

Pega Customer Decision Hub provides two common patterns for designing an internal email treatment:

- You can design the email content directly within the email treatment in Pega Customer Decision Hub. Use this pattern only when the email content is relatively simple and does not require any complex HTML design.
- Marketing departments can outsource email content designs to professional designing agencies who do not have direct access to Pega Customer Decision Hub. These agencies use HTML editors, such as Dreamweaver, to build visually appealing content. You can import the HTML with the externally designed email content directly into an internal email treatment.

For both scenarios, your Pega Customer Decision Hub instance requires a direct integration with an SMTP server to send the emails. When you run an outbound schedule and the system triggers emails, Pega Customer Decision Hub constructs the email content by replacing the placeholders with personalized data and then sends the content directly to the SMTP server.

With internal email treatments, you can use the email treatment forms to view the content, map placeholders, send test emails, and track impressions and clicks. You then internally assess the performance, throttling, and maintenance.

External email treatments

You create external email treatments in Pega Customer Decision Hub too, but the email content is mapped to a file or database template. Pega Customer Decision Hub writes an outbound schedule run results to a file or database table that is shared with a third-party email service provider for sending the outbound emails.

For external email treatments, Pega Customer Decision Hub does not store email content. In this common case, Pega Customer Decision Hub does not have direct integration with an SMTP server, but a third-party email service provider handles the service. These providers typically rely on file and database integrations and have a defined format of the data that they expect. Their environments host the email design.

When you use external email treatments you create output data from Pega Customer Decision Hub in the format that the integration demands. Typically, the output of the file or table consists of personalization data, a type of identifier that links the email treatment with a template that the service providers use, and additional identifiers to process the email responses that Pega Customer Decision Hub receives from the provider.

With external email treatments, Pega Customer Decision Hub only generates the data that is required to send the emails. The provider designs and sends the email messages and manages the performance and scaling of the whole process. Pega Customer Decision Hub requires additional integration to process the responses, update the interaction history, and train adaptive models.

Creating internal email treatments

Internal email treatments are created within Pega Customer Decision Hub. The email accounts used to initiate email sending are also configured in Pega Customer Decision Hub.

Internal email treatments can be designed and personalized in Pega or in an external tool.

Email content design and personalization in Pega

This demo shows you how to create an internal email treatment and send emails using Pega to present offer details in an email.

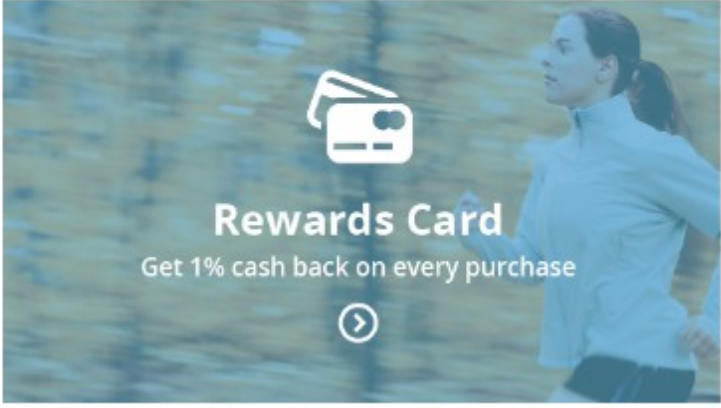
U+, a retail bank, wants to promote their Rewards credit card via email. They want to send the email to qualified customers in an always-on outbound environment.

The email is personalized and contains the offer details and a call to action. When customers click the **APPLY TODAY** call to action in the email, they are directed to the credit cards page on the bank's website to learn more about the offer and apply for it if interested.

Enjoy cashback on every purchase!

U+ Bank

Get 1% cashback on every purchase



Rewards Card
Get 1% cash back on every purchase

Dear Stephen Gold

You are pre-approved for a **U+ Bank Rewards card**.

Here's an overview of the card benefits:

- \$200 cash rewards after you spend \$500 on purchases in the first 3 months from account opening.
- 1% cash back on every purchase.
- No annual fee.

Click the following link to know more.

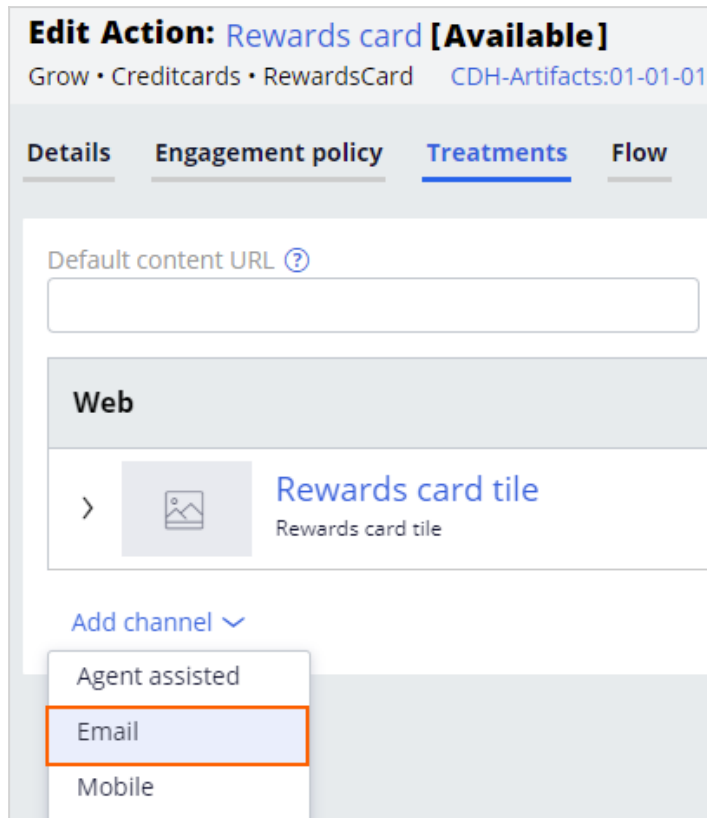
APPLY TODAY →

U+ Bank
One Rogers Street, Cambridge, MA, 02142-1209, U.S.A

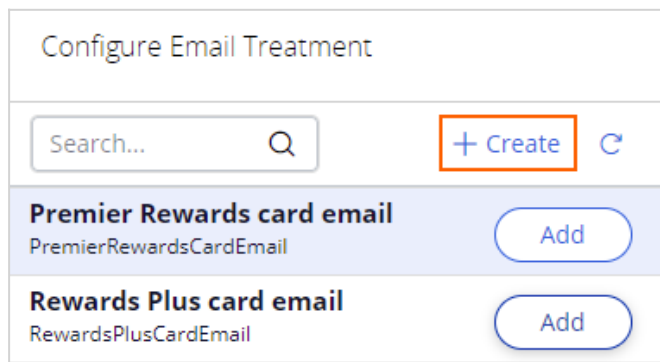
This is Pega Customer Decision Hub™ portal. To send emails in an always-on outbound environment, you need to add an email treatment to every action that you want to promote through email. An email treatment contains the message that you want to communicate to customers.

In this case, you create and add an email treatment to the Rewards card. Open the Rewards card to add new treatments to the **action**.

You can manage treatments associated with an action on the **Treatments** tab. To associate a treatment with an action, click **Add channel**.



You can use an existing email treatment or create a new one. In this case, you want to create a new internal email treatment.



Now, enter an appropriate **short description** to create the new treatment. It is a best practice to create the treatment in the same issue/group hierarchy in which you plan to use it; this will make the treatment easier to access and manage.

Create Email Treatment **Create and open**

Email Treatment Record Configuration

Identifier
A short description or title for this record RewardsCardEmail [Edit](#)

Categorization

Issue: Group:

Then, enter the email content.

Edit Email Treatment: Rewards Card Email [Available]

[Content](#) [Details](#) [Test email](#) [Prompts](#) [Pages & Classes](#) [Security](#) [History](#)

Subject

Logo

Title

Section content

Footer links

First, enter the email subject. You can include both static text and dynamic placeholders in the subject.

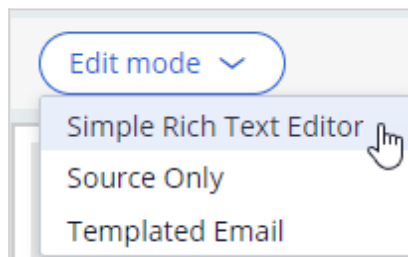
To access dynamic placeholders, start typing dot followed by Customer to access customer properties or OfferData to access action properties.

Dynamic placeholders allow you to personalize the subject. You can construct the placeholders with both customer and offer attribute properties.

At runtime, the properties are replaced with the real values. For example, if the customer's first name is *Stephen*, the generated subject line reads: *Hi Stephen, here's your U+ Bank card*.

Then, in the **Details** tab, enter an appropriate keycode. When the **Track Email Open Rate** option is enabled, the system records an impression when a customer opens an email.

In the **Content** tab, you can design the email using one of the three edit modes: Simple Rich Text Editor, Source Only, and Templated Email.

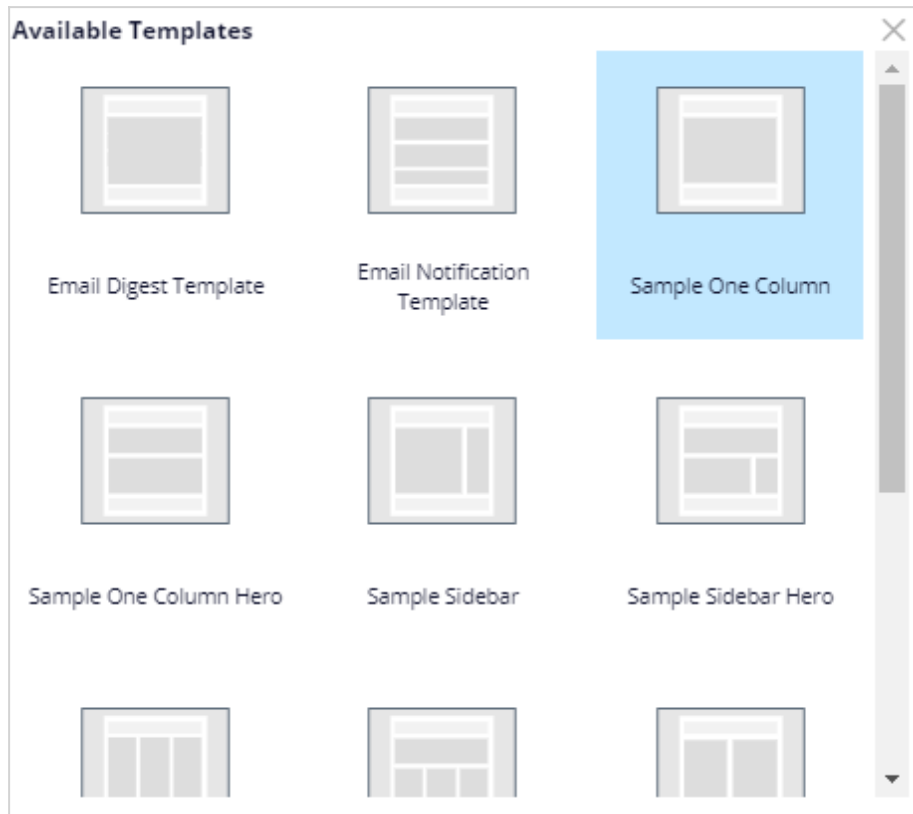


The **Simple Rich Text Editor** provides a blank page with a rich text editor, which allows you to compose an email manually.

The **Source Only** mode allows you to work with HTML content only. If the content is designed in an external tool, you can copy and paste the HTML content in this mode.

The **Templated Email** mode provides you with several easy-to-use templates for designing appealing emails; this is also the default mode.

These are the available template options.



In this case, you use the default **Simple One Column** template.

The **Simple One Column** template contains four sections: Preheader, Header, Body, and Footer.



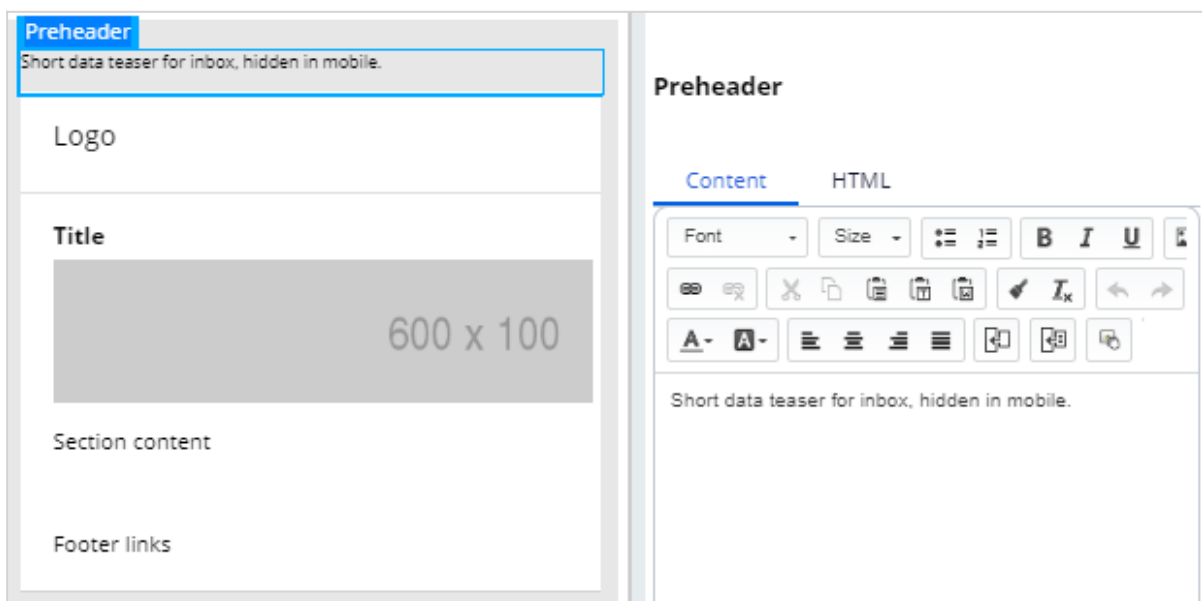
The Preheader section can contain a short **data teaser for inbox** that is intended to persuade customers to read the email further. This text is hidden on a mobile device.

The Header section can contain the organization's logo.

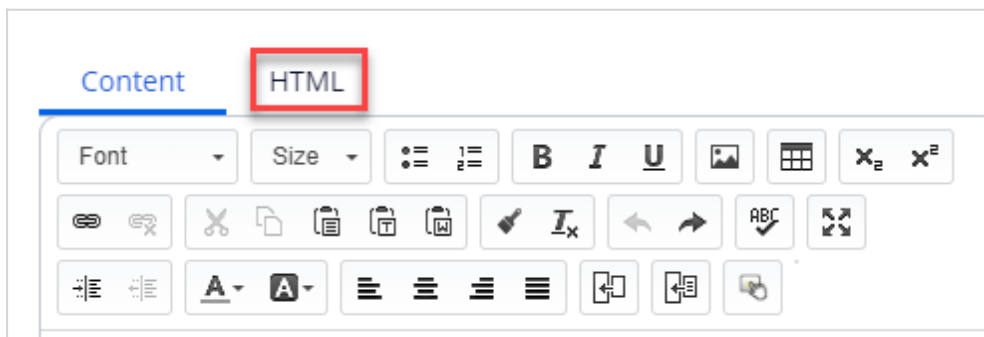
The Body section contains a placeholder for a title, image, text, or buttons as required. This section is where the main content goes.

The Footer section can contain contact details or links, if any.

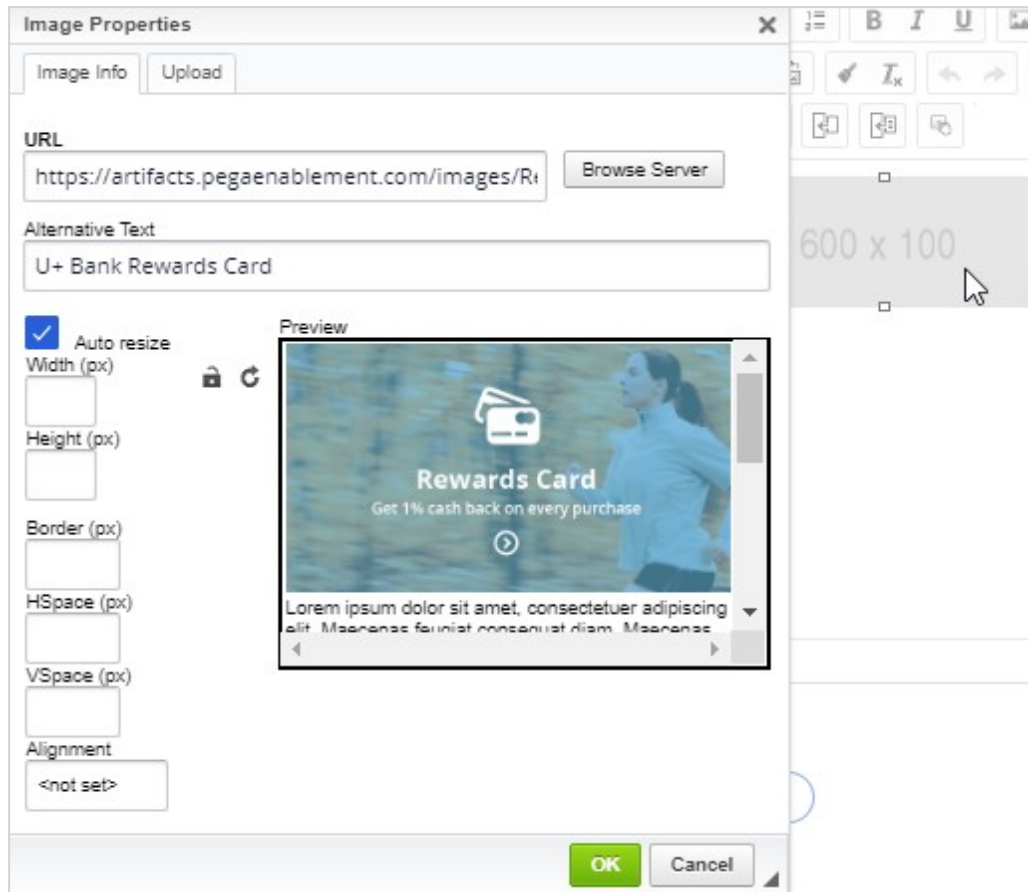
When you click on the different placeholders or sections in the template, an editing pane is displayed on the right side of the screen.



This editor also allows you to copy and paste HTML content from other tools using the **HTML** tab.



For example, here is how you can create an HTML version of the U+ Bank logo by using HTML code.



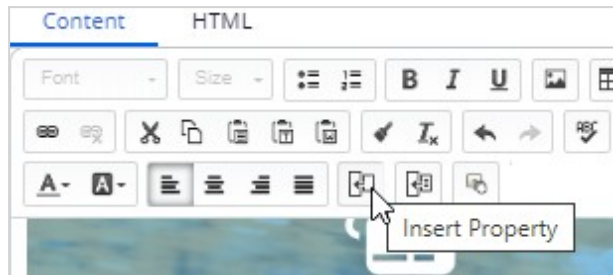
It is best practice to host an image on an external server and reference it in the email. Do not embed images directly in the email, as it makes the email bulky.

In the email content, you can add static content and dynamic placeholders to personalize the message.

For example, the property **.Customer.FullName** can be added as a placeholder. The customer's real full name will replace the placeholder at the time of email delivery.

Dear <<.Customer.FullName>>

To add a customer or offer attribute as a dynamic placeholder, use the **Insert Property** button.



 A screenshot of the 'Property Parameters' dialog box. It has a close button (X) in the top right corner. The 'Name' field contains the text '.Customer.pyFullName'. Below it are 'Format' and 'When' fields, both currently empty. At the bottom, there are 'Cancel' and 'Save' buttons.

Any property that you add in an email treatment has an equivalent Pega markup tag (**pega:reference**) to reference customer or offer attributes. After you insert the property, you can view the tag (which is automatically added) in the **HTML** tab. When you design the email in an external application, you can use these tags to add dynamic placeholders, so the personalization works when you import the HTML into the email treatment.

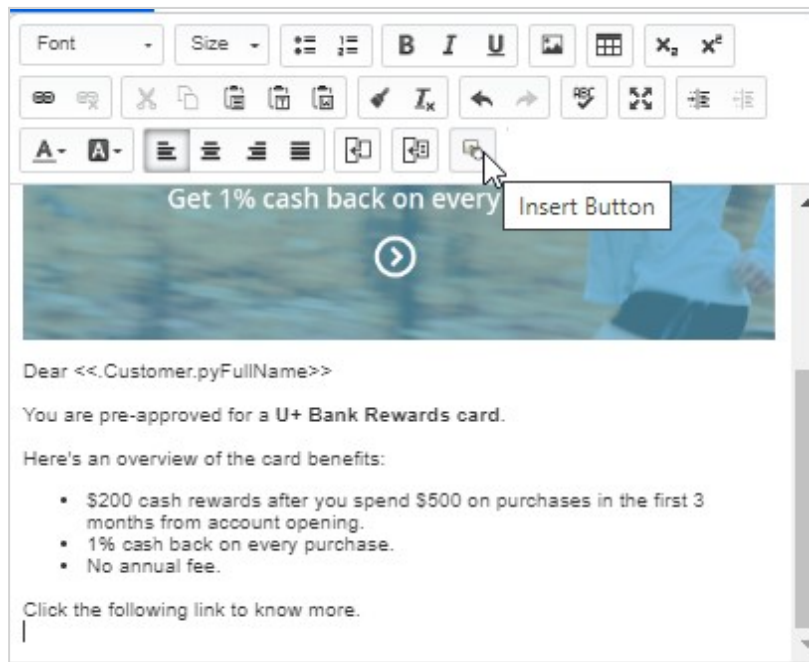
```

1 
2 <p>Dear&nbsp;<pega:reference name=".Customer.pyFullName"></pega:
  reference>&nbsp;<br />
3 <br />
4 You are pre-approved for a&nbsp;<strong>U+ Bank Rewards&nbsp;&nbsp;car
  d</strong>.<br />
  
```

Note that to add an offer attribute, you can add a dynamic placeholder, which is then replaced with the offer attribute at the time of email delivery. For example, you can use <<.OfferData.pyLabel>> to reference the offer attribute.

In this case, you may want to replace U+ Bank card names with an offer attribute by using the <<.OfferData.pyLabel>> placeholder.

You can also add links or buttons that the customer can click to respond to an offer or view more details. In this case, you add an **APPLY TODAY** call to action button.



In the **Button Parameters**, the **Function** drop-down menu lists the possible functions that you can assign to a button.

Button Parameters

Function
Accept action

Redirect to
 Response screen
 External URL

Response screen
DefaultEmailResponse

Track click rate

Style options

Button text
Accepted

Button image URL

Custom style

Cancel Save

Now, select **Open external URL** and provide an external URL that opens when the customer clicks the button. Then, provide the button image URL.

Button Parameters ✕

Function
Open external URL ▾

URL
https://pegasystems.github.io/uplus-wss/retail_bank/cate

Include action data

Track click rate

Style options

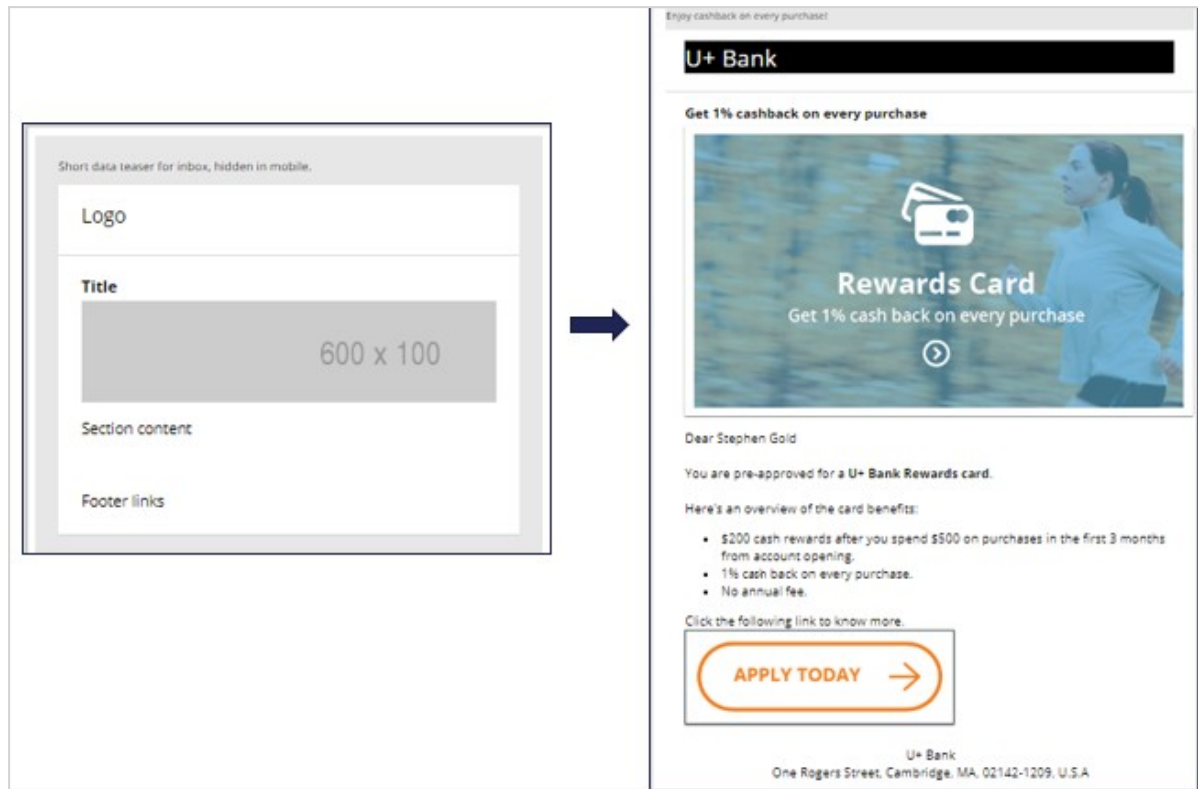
Button text

Button image URL
<https://artifacts.pegaenablement.com/images/ema>

Custom style

Cancel Save

A composed email looks like this.



Now, save the treatment and add it to the Rewards card action.


You can send the email to test accounts or to yourself to test the visualization and rendering of the content on various email clients.

To test the treatment, switch to the **Test email** tab. In this tab, you can specify the target email addresses and the email subject specifically for the test.

To:

Test recipients

Seed list

customer1@enablement.com 



+ Add

Send from

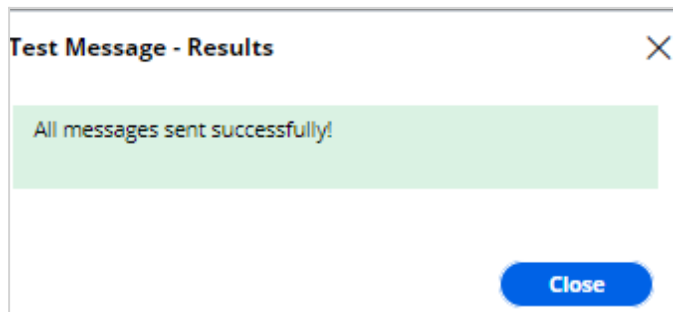
Default email account

Other email account

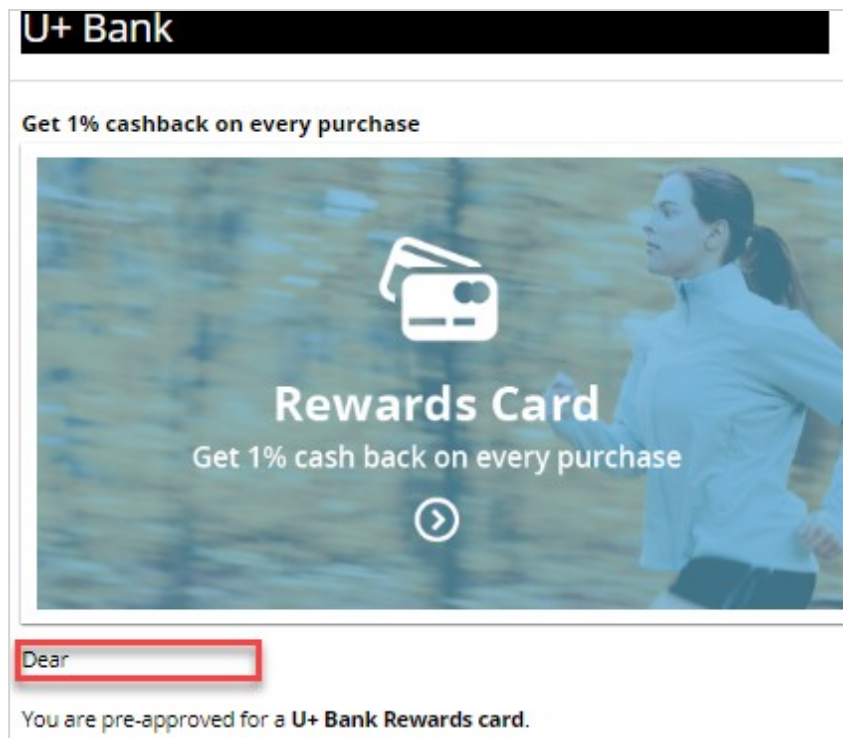
Subject

"This is a test message"  

To test the treatment, click **Send test email**. Once the test is complete, you can check the emails in the email client.



Note that none of the attribute placeholders are populated when you manually enter the recipients' email address. However, you can test the functionality of the buttons and links.



To populate the attribute placeholders, you can use a **seed list** as test input.

To:

Test recipients

Seed list

Seed list: [View all](#)

Send from

Default email account

Other email account

Subject

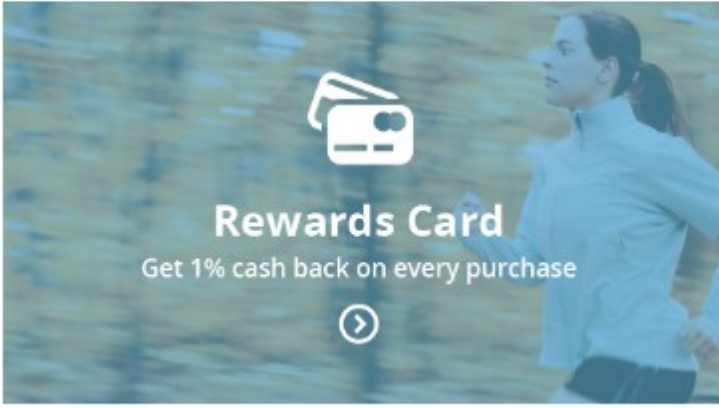
A seed list contains test customer data. It contains values for customer attributes, including email addresses.

When using a seed list to test the Treatment, the dynamic placeholders are replaced with customer information from the seed list.

Note that the customer name placeholder is replaced with the data from the seed list.

U+ Bank

Get 1% cashback on every purchase



Rewards Card
Get 1% cash back on every purchase

Dear Stephen Gold

You are pre-approved for a U+ Bank Rewards card.

Here's an overview of the card benefits:

- \$200 cash rewards after you spend \$500 on purchases in the first 3 months from account opening.
- 1% cash back on every purchase.
- No annual fee.

Click the following link to know more.

APPLY TODAY →

This demo has concluded. What did it show you?

- How to add an email treatment to an action
- How to use dynamic placeholders to populate the customer and offer attributes in an email treatment
- How to use HTML code to generate content in an email treatment
- How to use an image from an external server in an email treatment
- How to add a button to an email treatment
- How to send emails to test accounts to test the visualization and rendering of the email content
- How to use seed lists to test the dynamic placeholders

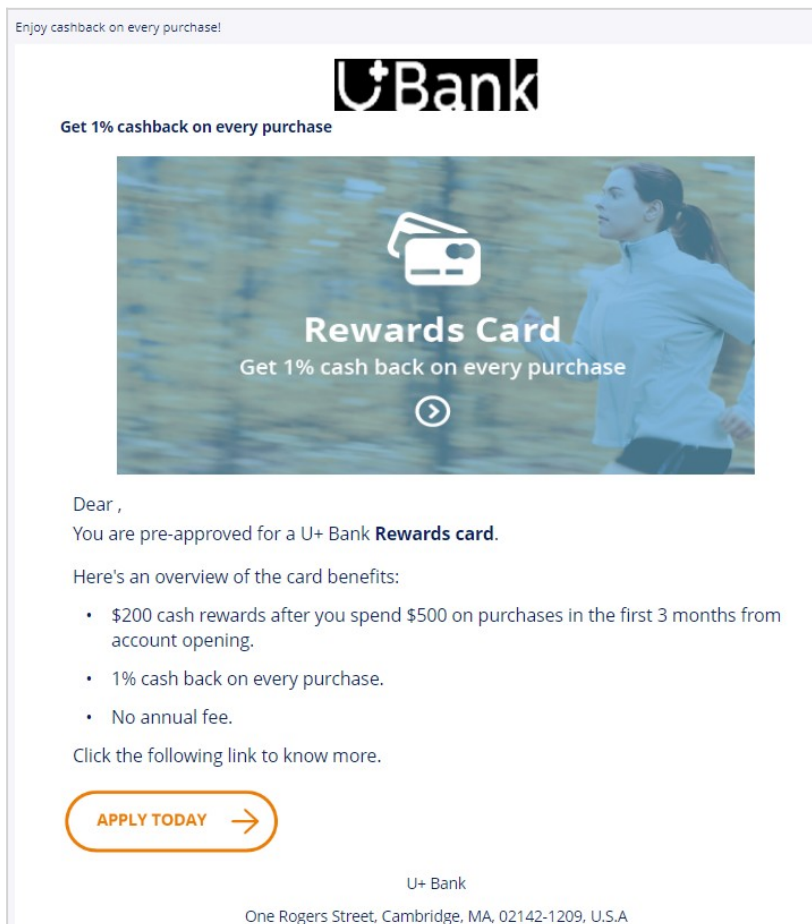
Email content design and personalization in an external tool

This demo shows you how to personalize email content that you create in an external tool and use an SMTP server to send emails from Pega Customer Decision Hub.

Pega Customer Decision Hub comes with the capabilities to create and personalize email content to present offers to customers. However, your organization can use an external tool, such as Dreamweaver, to design and customize email content based on your requirements.

In this example, you will learn how to personalize an email treatment in an external tool and use an SMTP server to send emails from Pega Customer Decision Hub™.

This is the email treatment that was designed in an external tool and the corresponding HTML code.



When you design email in an external tool, you can use Pega markup tags to add dynamic placeholders so that your personalizations work when you import the HTML into Pega

Customer Decision Hub. In this example, the *Pega:reference* markup tag refers to a customer property, Customer FullName.

```
Dear <pega:reference name=".Customer.FullName"></pega:reference>,
```

After completing the email treatment design and personalization, you can create a new internal email treatment for an action, and then load the email content that you designed externally into that treatment. Now, save the treatment and add it to the action.

You can send the email to test accounts or to yourself to test the visualization and rendering of the content on various email clients. Check the email in the email client. Note that the data from the seed list replaces the customer name placeholder.

You have reached the end of this demo. What did it show you?

- How to reference Pega Platform properties in email content that you design in an external tool.
- How to import email content from an external tool into Pega Customer Decision Hub.

Defining the action flow

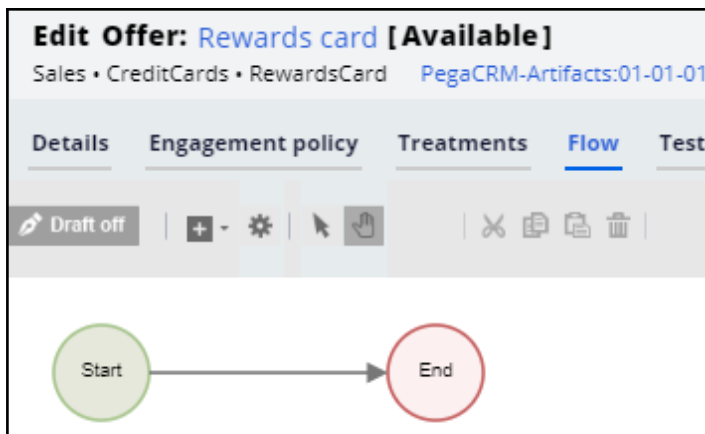
Business scenario

U+, a retail bank, has designed a new email treatment to promote the Rewards Card offer. Now, an action flow must be created to ensure that the email is delivered when this action is selected for a customer.

Action Flow

The action flow is a flow diagram that contains the sequence of steps carried out during the lifecycle of an action. These steps include delivering the action message via an outbound channel, waiting for a response, updating the Interaction History, sending a follow-up message, and so on. In a flow, each of these steps is represented by a variety of shapes. Best practice is to keep the flow short and simple, preferably not more than 2 or 3 steps per channel. Every additional step taken in the action flow is a missed opportunity to let the Pega brain decide the Next-Best-Action.

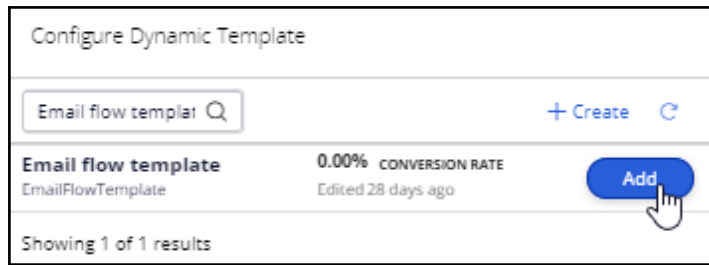
A new action contains only the default **Start** and **End** shapes. To send emails in an always-on outbound environment, you need to add a **Send Email** shape to the action flow.



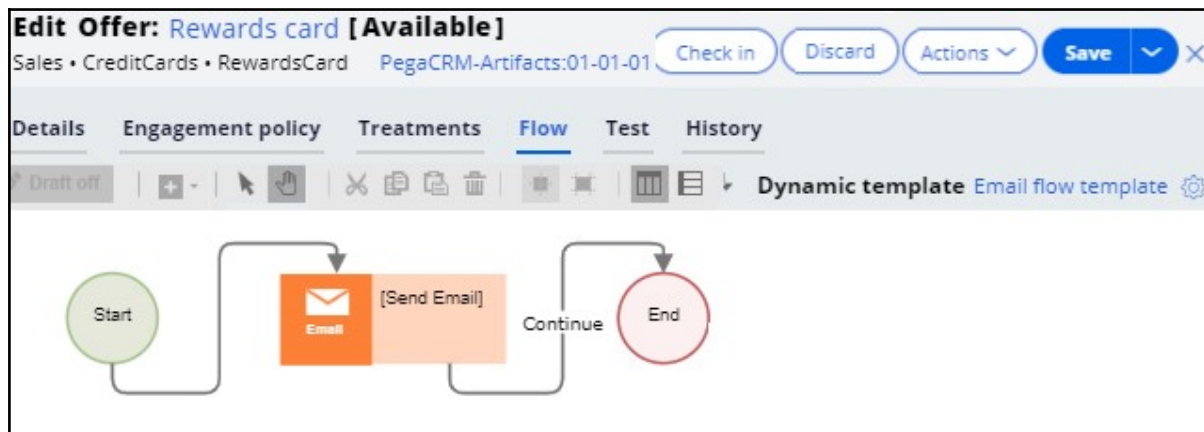
When there are several actions that need the same flow pattern, you can create that flow pattern once and reuse it for multiple actions using the **Dynamic template** option.



In the **Configure Dynamic Template** window, select the action that contains the desired flow pattern.



When you use the **Dynamic template** to configure the flow, you cannot change the flow in the referenced action. To edit the flow, you must edit the original action in which the flow was defined.



Basic email flow template

Let's look at a basic flow pattern for email delivery.

A flow template is just a normal action with the flow configured. In the following example, an action is used as a flow template within other actions to enable them to send emails.

Thus, the **Availability** of the action is set to **Never** to prevent it from being presented to customers.

Offer: Email flow template [Available]
 Sales • CreditCards • EmailFlowTemplate PegaCRM-Artifacts:01-01-01

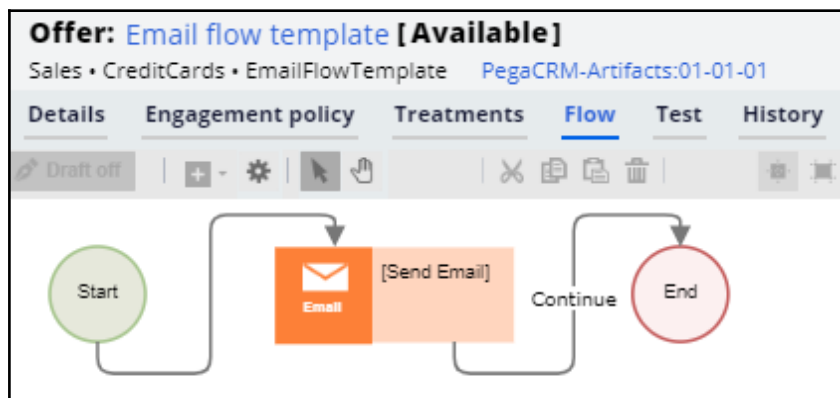
Details Engagement policy Treatments Flow Test History

Key code: EMAILFLOWTEMPLATE Variant:

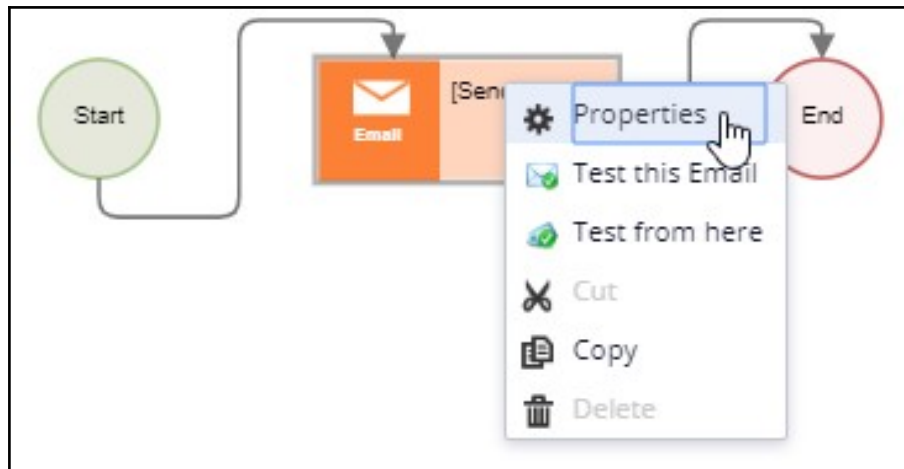
Availability

Always
 Never
 Within a defined time period

The **Send Email** shape in the flow typically references the email treatment that contains the email content to be sent.



However, when you want the flow to be used as a template, you want the treatment belonging to the individual action to be used rather than a standard email treatment. In the **Send Email** shape, set the **Treatment** option to **Use treatment from strategy** to ensure that the treatment from each action, as selected by the Next-Best-Action decision, is used to send emails, instead of a standard treatment.



Send Email properties

Send Email:

Configure the content of the email and its delivery.

Treatment
What email treatment should be sent?

Use associated Treatment Specify Treatment ?

Wait after sending

Advanced
What else can I do?

▶ **Email account**

▶ **Tickets**

Cancel
Submit

This ensures that the email treatment is independent of the flow template associated with the action. For example, when the Rewards Card uses the **Email flow template** as its **Dynamic flow** template, the Rewards Card email treatment will be used for sending the email.

Avoiding overexposure of actions on outbound

Description

Contacting a customer too frequently can have a negative impact on the customer's attitude toward further actions by your company. Learn how to define contact limits and use suppression policies to pause actions for a given time period based on past customer behavior.

Learning objectives

- List the types of contact policies
- Explain how to avoid overexposure of actions using contact limits and suppression policies

Contact policy types

Types of constraints

Too many contact attempts over a short period of time can have a negative impact on a customer's attitude toward further actions by your company. To maximize the lifetime value of every customer relationship, organizations must prevent outreach fatigue by optimizing the number of actions taken.

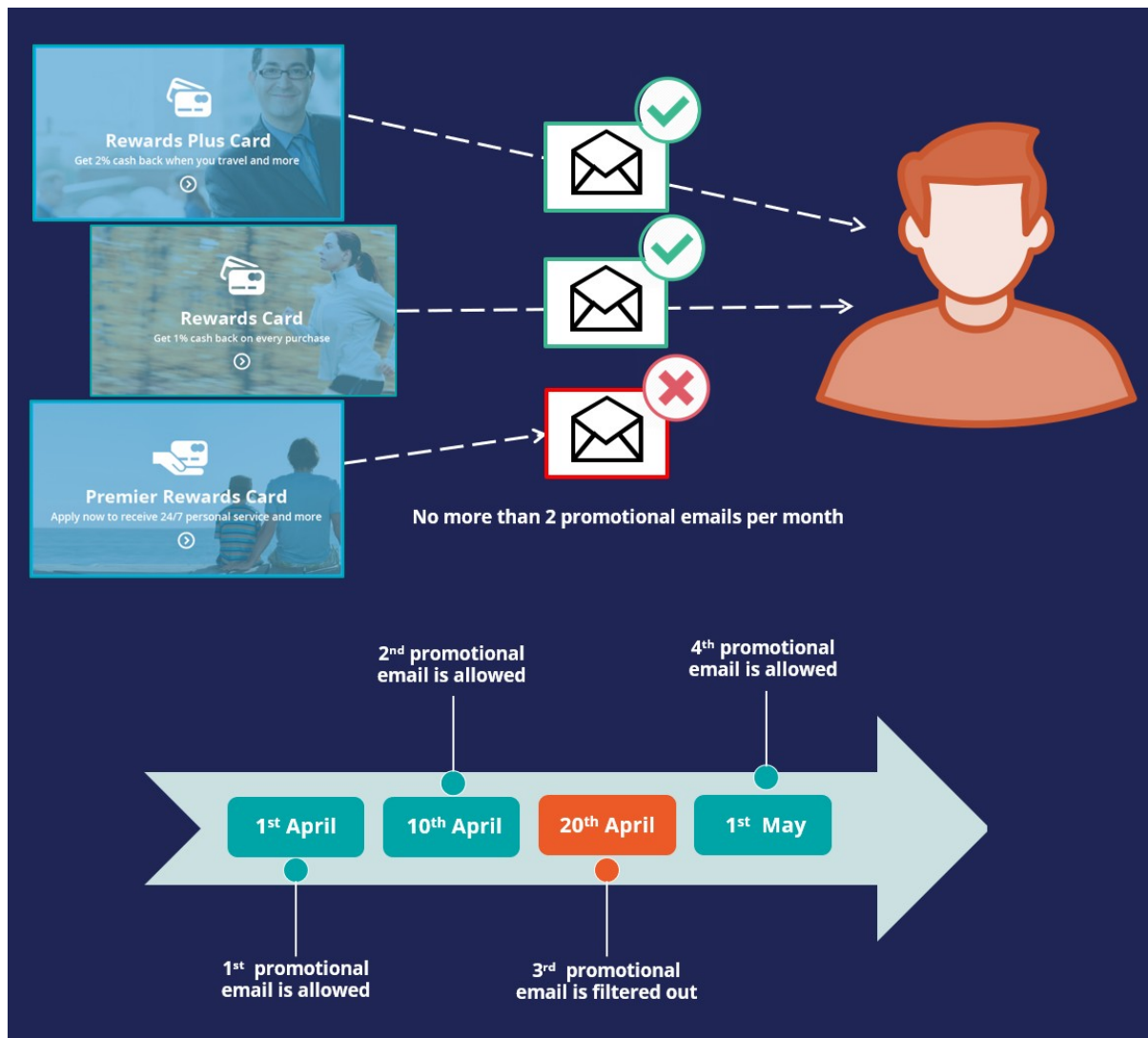
In Pega Customer Decision Hub™, there are two types of contact limits that allow you to limit actions over a given period of time.

1. Customer contact limits
2. Suppression policies

Customer contact limits

Customer contact limits allow you to limit the number of actions that a customer can receive over a given period of time on a specific channel. These customer contact limits prevent an action from reaching a customer on a specific channel, irrespective of past responses to that action by the customer.

For example, you can decide that you do not want your customers to receive more than two promotional emails per month.

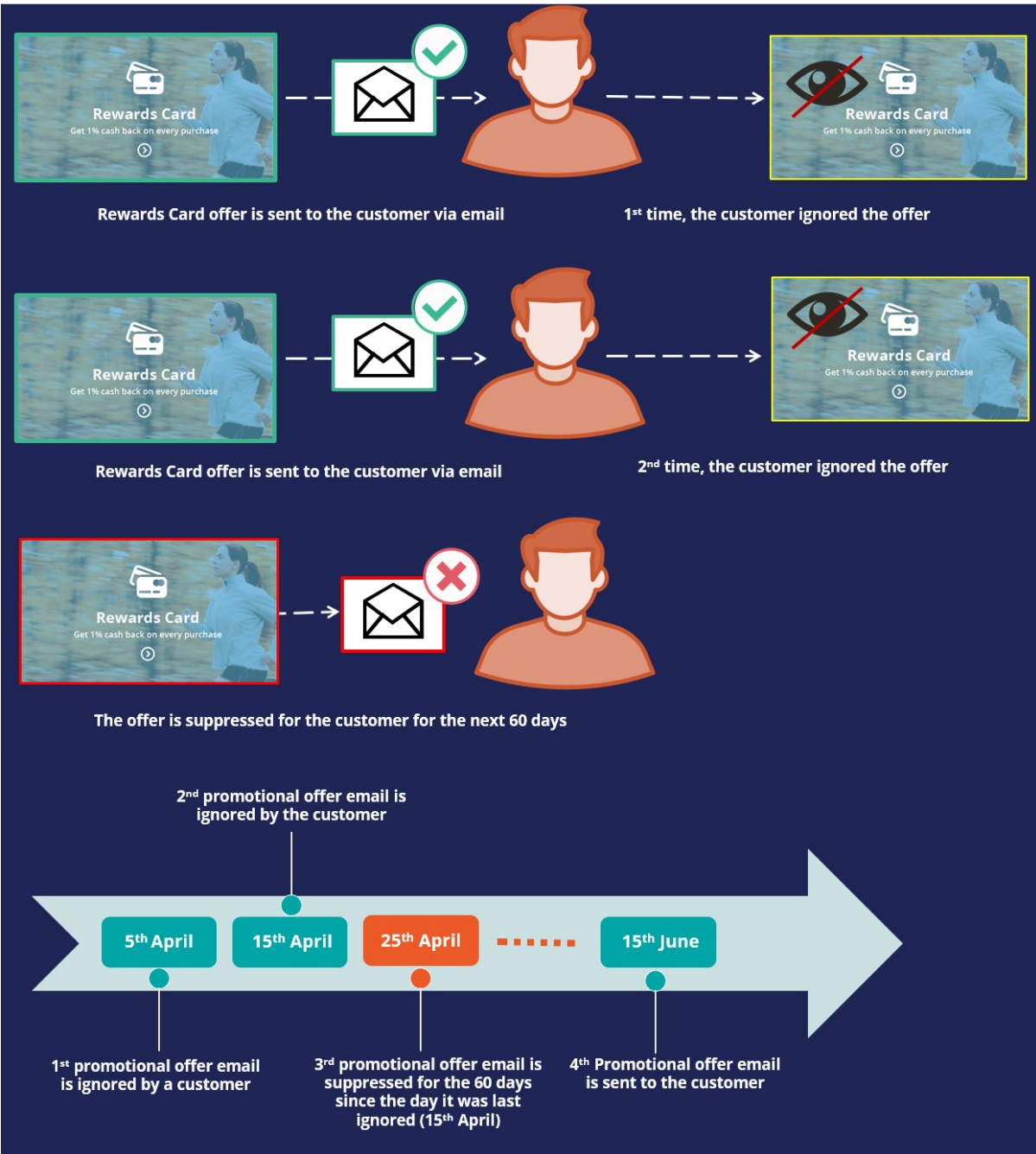


Suppression policies

Suppression policies determine when and for how long an action or group of actions should not be shown to a customer. These suppression policies put an action on hold after a specific number of outcomes are recorded for some or all channels.

Suppressing or pausing an action prevents oversaturation by limiting the number of times a customer is exposed to the same action.

For example, if a customer ignores a credit card promotional offer email twice in the previous 30 days, then the same credit card offer should not be sent to the customer again for the next 60 days.



Avoid sending the same action for a time period

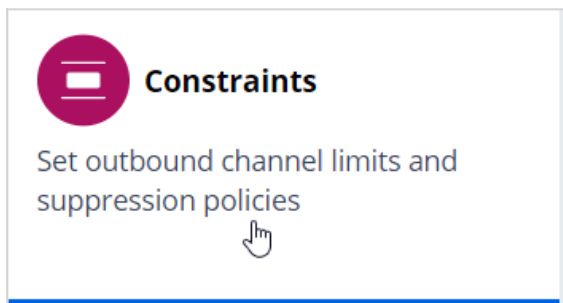
Business scenario

U+ Bank's marketing department currently promotes various credit card offers by sending emails to qualified customers.

Now the bank wants to implement a new set of requirements: they want to limit the number of emails sent to their customers to two per month. Also, the bank wants to avoid sending a credit card offer via email for 60 days if one was sent in the past 30 days.

Updating constraints

On the Next-Best-Action Designer **Constraints** tab, you configure outbound channel limits and suppression policies.



Edit **Constraints** to modify the limits.

Channel	Limit
All outbound	1 limit defined
Retail	1 limit defined
Email	1 limit defined
Push notification	1 limit defined
SMS	1 limit defined

Updating customer contact limits

To limit the number of outbound interactions that a customer can receive over a given period of time on a specific channel, you need to set customer contact limits.

↑ Outbound limits ?	
> All outbound	1 limit defined ⋮
> Retail	1 limit defined ⋮
👉 Email	1 limit defined ⋮
> Push notification	1 limit defined ⋮
> SMS	1 limit defined ⋮

In this scenario, the bank wants their customers to receive no more than two emails per month. So, you need to update the existing customer contact policy for the email channel to two emails per month.

↑ Outbound limits ?	
> All outbound	1 limit defined ⋮
> Retail	1 limit defined ⋮
∨ Email	1 limit defined ⋮
Maximum number of actions Within time period	
<input type="text" value="2"/>	<input type="text" value="30 days"/> ∨

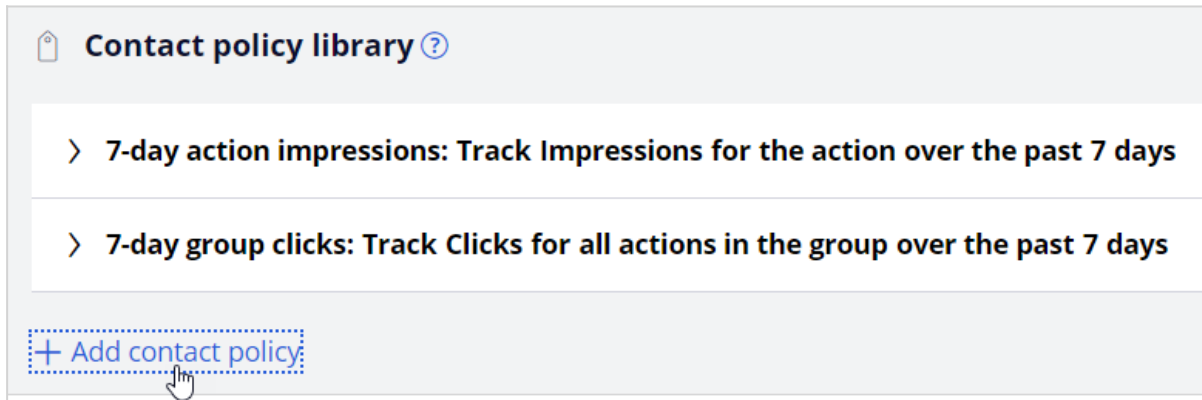
In the contact policy, you can update current limits or add new channel limits.

Note: You cannot repeat a channel in a contact policy.

Defining suppression policies

To suppress a single action or a group of actions for a limited time period, you need to define suppression policies.

In this scenario, the bank wants to avoid sending promotional emails related to a credit card offer for 60 days if one was sent in the past 30 days. So, you need to add a contact policy to put an action on hold for 60 days.



When an email is sent, a record is added to the Interaction History with the outcome value **Pending**.

Provide a name for the new contact policy and then select the type of outcome it will track; in this case it will track outcomes with the value **Pending**.

You can specify whether the responses are tracked for one specific action, or for all actions in the group. Track the responses at the action level, since you want to suppress only credit card promotions in email.

You can select the time period over which the responses should be tracked. In this case, responses should be tracked for the past 30 days.

A screenshot of a modal form titled "Add contact policy" with a close button (X) in the top right corner. The form has the following fields: a "Name *" field containing "30-day action pending"; a "Scope" section with "Track" set to "Pending" (dropdown), "for" set to "the action" (dropdown), "within the past" set to "30" (dropdown), and "days"; and two buttons at the bottom: "Cancel" and "Submit".

The business requirement is to suppress the action for 60 days if an email was sent in the past 30 days. In other words, if there is even one pending response for the email treatment

over the past 30-day period, the action must be suppressed. So, you need to enter the number of responses required to fulfill the suppression criteria as 1.

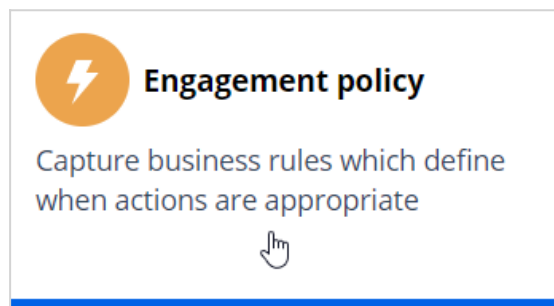
Then you need to select the channel for which the responses are tracked and enter the number of days for which an action should be paused when the suppression criteria are met.

30-day action pending: Track Pending for the action over the past 30 days

IF Pending for THEN suppress

Associating a contact policy with an issue and a group

On the **Engagement policy** tab, you can associate contact policies with an issue and group. The contact policies are reusable across all issues and groups.



As the bank wants to suppress promotional emails related to credit card offers, you need to edit the **CreditCards** group to add the newly defined contact policy to it.

Business structure

PegaCRM-Artifacts: 01-01-01
Edited 4 minutes ago by CDH Analyst

Edit
Delete
Actions

All groups

Sales

CreditCards

Engagement policy

E Eligibility
?

(isCustomer is true)

and (Age is greater than 18)

A Applicability
?

(Has Cards is equal to N)

S Suitability
?

No group criteria defined

C Contact policy
?

- > 7-day action impressions: Track Impressions for the action over the past 7 days
- > 7-day group clicks: Track Clicks for all actions in the group over the past 7 days
- v 30-day action pending: Track Pending for the action over the past 30 days
 If there are 1 Pending for Outbound / Email, suppress the action in all channels for 60 days

Limiting action volume on outbound

Description

In outbound customer engagement, you need to cater to different kinds of operational limitations. These may include limits on the number of email communications that can be sent in a day, or the total number of communications that can be sent regarding a particular offer. Learn how to define volume constraints to limit the volume of actions in an outbound run.

Learning objectives

- Limit outbound action volumes using volume constraints

Volume constraints

In Pega Customer Decision Hub™, you can limit the volume of actions that the system delivers to customers through always-on outbound.

Volume constraint modes

Within a volume constraint definition, you can define the combination of constraints for one or more actions with one or more channel types.

In situations where a customer is eligible for multiple actions, you can apply volume constraint with one of three modes, depending on the business requirement. You might specify whether to apply volume constraints to actions as a group (for which a customer qualifies) or evaluate each action for which a customer qualifies separately.

Individually for each action

In this mode, the top action for which a customer qualifies passes through the set of defined volume constraints, and the system validates each action against each constraint type. If the action passes through all the constraints, the system selects the action and delivers it to the customer. The customer always receives a maximum of one action, even though the limits for the other low-priority actions is not reached. The top action is selected based on priority calculations ($p*c*v*I$) defined in Next-Best-Action Designer arbitration.

For example, consider a customer who is eligible for the **Standard card**, **Rewards card**, and **Rewards Plus card** credit card actions, and the **Standard card** is the top priority card. Also, assume that the remaining volume for the top action (Standard card) and priority 3 action (Rewards Plus card) is above zero, and the remaining volume for the priority 2 action (Rewards card) is zero. When you set volume constraints mode to **Individually for each action**, the customer sees only the top priority action, **Standard**

card, as shown in the following figure:

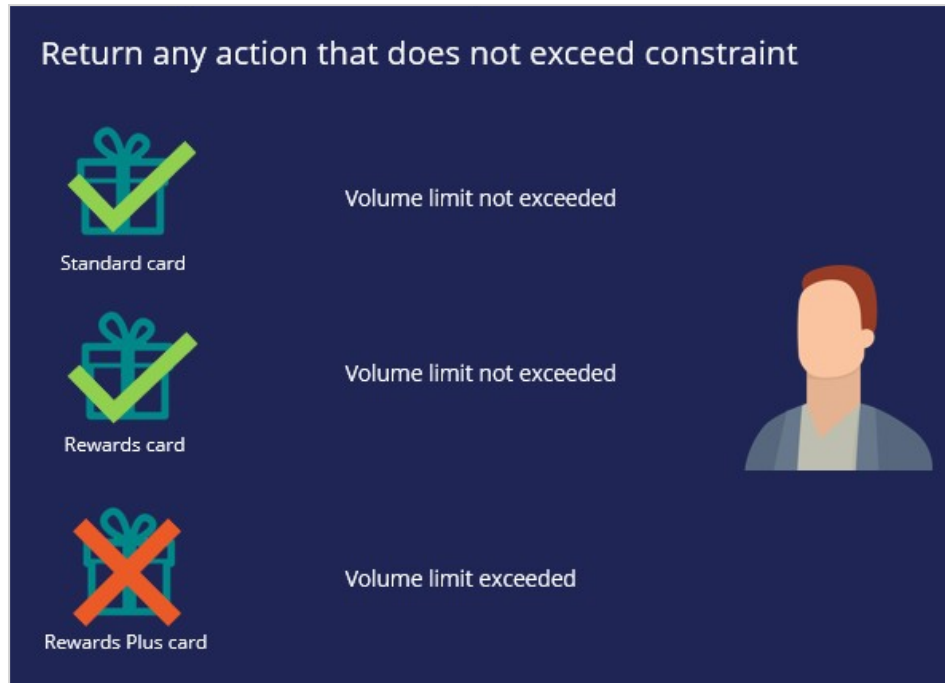


If the volume remaining for the top priority action is zero, then the next priority action is validated against the constraint types.

Return any action that does not exceed constraint

In this mode, each action a customer qualifies for passes through the set of defined volume constraints, and the system validates each action against each constraint type. If an action passes the set of constraints, the system delivers the action. If an action does not satisfy a constraint, the system does not deliver an action. The system considers all actions for which a customer qualifies, and the order in which the check occurs is irrelevant. The result is that a customer receives all the actions for which they qualify provided that the actions do not reach the volume limit. The **Return any action that does not exceed constraint** mode is the recommended approach, even though the system does not consider priority during the volume constraint phase. This mode ensures that the limit handling is separated from arbitration logic.

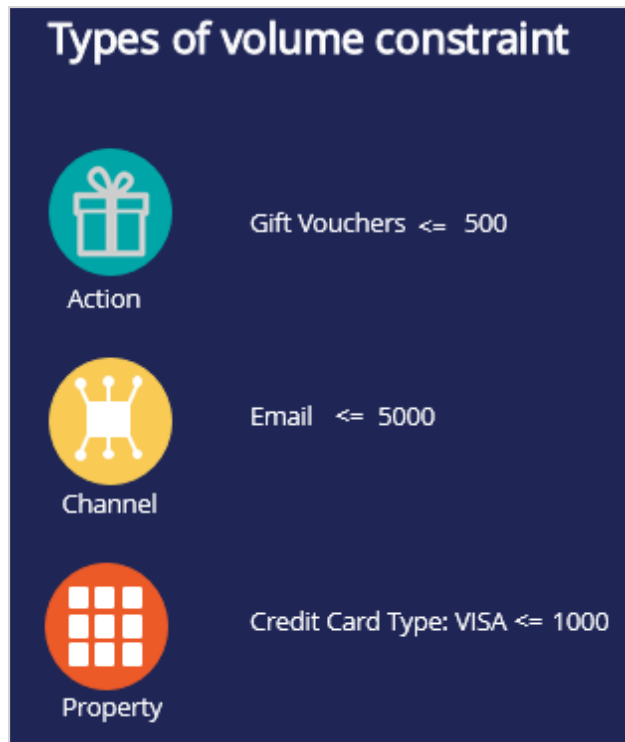
For example, consider a customer who is eligible for the **Standard card**, **Rewards card**, and **Rewards Plus card** actions. Also, the remaining volume for **Standard card** and **Rewards card** is above zero, but the volume of the Rewards Plus card action reached zero. When you set volume constraints to **Return any action that does not exceed constraint**, the customer receives **Standard card** and **Rewards card** but not the **Rewards Plus** card, as shown in the following figure:



When the system evaluates each action separately, Pega Customer Decision Hub can present all actions that pass the engagement policy conditions to the customer.

Volume constraint types

Volume constraints fall into three categories depending on the level at which you apply the constraints: action, channel, or property, as shown in the following figure:



Action constraint

An action constraint limits the volume of actions that the system can send to customers.

An action constraint is useful when you have a limited number of actions that you can send to customers, for example, 500 gift vouchers. Configuring an action constraint ensures that you do not offer more actions than are available.

Channel constraint

A channel constraint limits the volume of actions that the system can deliver through a certain channel.

You can use a channel constraint for channel capacity planning. For example, assume that your call center can only handle up to 1000 inbound calls daily due to limited capacity. If your company sends out 50,000 emails per day to customers, it's reasonable to anticipate that 2-3% of the recipients may follow up with an inbound call, which translates to approximately 1000-1500 calls daily. This volume of calls would exceed the capacity of your inbound call center. Therefore, to prevent the call volume from surpassing the handling capacity, it would be necessary to restrict the number of outbound emails sent to 35,000 per day.

Property constraint

By default, actions fall into the issue-group business hierarchy. By using a property constraint, you can limit the volume of a set of actions across different issues and groups by grouping them using a property value.

For example, you can group all credit card actions by setting a property, *CreditCardType*, to VISA. Then, you can define a constraint to limit the action volume where you do not want to send more than 500 VISA credit cards in a month.

Outbound run scenario

In an outbound run, the system considers a segment of the audience for an outbound communication. When customers are eligible for multiple actions, the system applies volume constraints across customers to limit the volume of actions that it delivers. After the next-best-action logic runs, the system executes the volume constraint limits in a non-predefined order of the customers.

Consider a simplified example of five customers to better understand how the limit constraint engine works. The following table shows the set of customers from the audience, the actions for which they qualify, the channel for which a treatment is defined, and the priority of the action. The priority represents the final priority calculation ($p*c*v*I$) as defined in the Next-Best-Action Designer arbitration.

Interaction ID	Action	Channel	Priority
CUST-01	Gold Card	Push	50
CUST-02	Gold Card	Email	25
CUST-03	BOGO	Push	75
CUST-03	Platinum Card	Email	50
CUST-03	Free Shipping	SMS	25
CUST-04	Free Shipping	Direct mail	50
CUST-05	BOGO	SMS	50
CUST-05	Free Shipping	Push	25

The following table shows the volume constraints configured in the system. There are two types of volume constraints configured:


1. A channel constraint with a limit of 2 actions for the **Push** channel.
2. An action constraint with a limit of 1 for the **BOGO** action.

Volume Constraints

Key	Value	Maximum
Channel	Push	2
Action	BOGO	1

Mode 1: Individually for each action

Interaction ID	Action	Channel	Priority
CUST-01	Gold Card	Push	50
CUST-02	Gold Card	Email	25
CUST-03	BOGO	Push	75
CUST-03	Platinum Card	Email	50
CUST-03	Free Shipping	SMS	25
CUST-04	Free Shipping	Direct mail	50
CUST-05	BOGO	SMS	50
CUST-05	Free Shipping	Push	25



Interaction ID	Action	Channel	Priority
CUST-02	Gold Card	Email	25
CUST-03	BOGO	Push	75
CUST-04	Free Shipping	Direct mail	50
CUST-05	Free Shipping	Push	25

If there are no volume constraints configured for a particular channel, action, or property, the system does not apply limits, and the actions pass through for outbound communication.

Therefore, *CUST-02* and *CUST-04* pass through as there are no limits defined for these actions (**Gold Card** and **Free Shipping**) or channel (Email, Direct mail).


The system considers the rest of the customers in a non-predefined order. Assume that it picks *CUST-03* and *CUST-05* first. Note that for *CUST-03*, the system chooses the BOGO action with higher priority. In this mode, the system delivers only one action for each customer and ignores the other two actions.

CUST-05 qualifies for two actions: **BOGO** (priority 50) and **Free Shipping** (priority 25). Because the system picked the BOGO action for *CUST-03* and exhausted the action limit, it selects the next action, Free Shipping, for *CUST-05*.

Because the system selected two **Push** channel actions and one **BOGO** action, both reach their volume constraint limits, and *CUST-01* receives no action.

Mode 2: Return any action that does not exceed constraint

Interaction ID	Action	Channel	Priority
CUST-01	Gold Card	Push	50
CUST-02	Gold Card	Email	25
CUST-03	BOGO	Push	75
CUST-03	Platinum Card	Email	50
CUST-03	Free Shipping	SMS	25
CUST-04	Free Shipping	Direct mail	50
CUST-05	BOGO	SMS	50
CUST-05	Free Shipping	Push	25



Interaction ID	Action	Channel
CUST-02	Gold Card	Email
CUST-03	Platinum Card	Email
CUST-03	BOGO	Push
CUST-03	Free Shipping	SMS
CUST-04	Free Shipping	Direct mail
CUST-05	Free Shipping	Push

CUST-02 and *CUST-04* pass through as no limits are defined for these actions.

Assume that the system picks *CUST-03* and *CUST-05* first. In this mode, *CUST-03* receives all the actions for which they qualify because the actions did not reach the limit.

CUST-05 receives only the **Free Shipping** action and not the **BOGO** action, as the BOGO action reaches the volume constraint limit when the system offers the action to **CUST-03**.

Finally, *CUST-01* does not pass through because the Push channel reaches the volume constraint limit.

Configuring a volume constraint

Business scenario

U+ bank has been running always-on outbound and now they would like to apply some limitations on the outbound volume. They have the following three different kinds of operational constraints to implement.

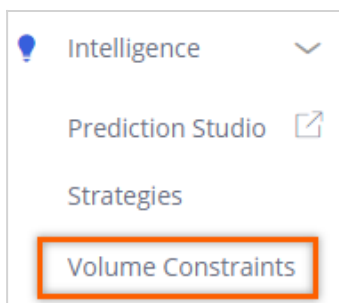
- The IT department has enforced a limit of 1000 emails per week to avoid being classified as spammers. Hence, the marketing team wants to adhere to the limits.
- The bank would like to restrict the number of Standard card offer emails to only 100 per week due to a business requirement.
- The bank would also like to limit offer emails related to credit cards of type "Master" to only 400 per week due to a logistical issue with their fulfillment vendor.

Creating a volume constraint

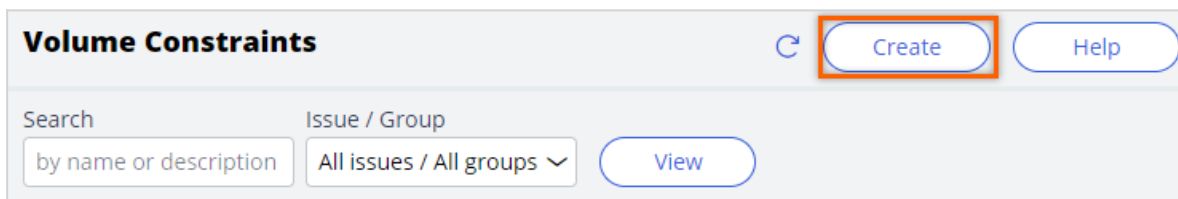
A constraint is an optional configuration element of always-on outbound. When configured, it will limit the volume of actions that will ultimately be initiated.

To limit action volume on the email channel, you need to configure a volume constraint for the always-on outbound.

The volume constraint landing page can be accessed from **Intelligence** -> **Volume Constraints** from the navigation menu.



You can add a new volume constraint from this page.



Provide a short description for the new volume constraint and the identifier is automatically populated when you tab out.

Note: As a best practice, the volume constraint must be created at the top level of your hierarchy for it to be used in the Next-Best-Action Designer outbound schedule.

Create Volume Constraint Create and open

Volume Constraint Record Configuration

Identifier
A short description or title for this record LimitOffers [Edit](#)

Categorization

Issue Group

Configuring a volume constraint

The **General** tab is where you configure the different types of volume constraints.

Edit Volume Constraint: Limit Offers [Available] Actions ▾ Save ▾

Top Level • LimitOffers CDH-Artifacts:01-01-01

General History

How should multiple actions for a customer be treated?

?

Individually for each action


As a group for the context (output all actions)

Return any action that does not exceed constraint

Reset constraint counts

Individually

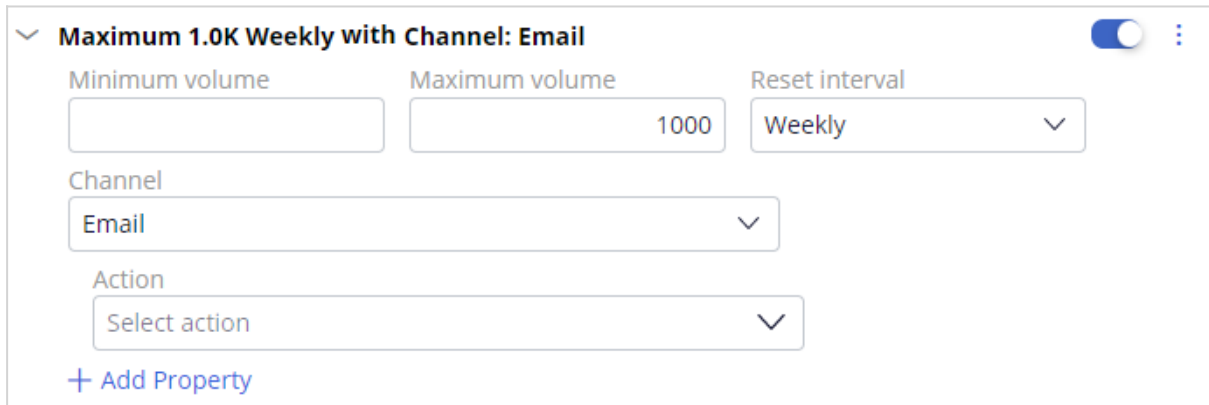
All at once



Constraint definitions have not been added.

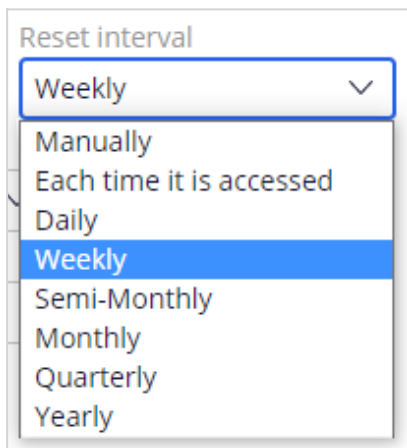
Add definition

In this scenario the bank wants to implement three types of volume constraints. As a first step, add a channel constraint that limits the number of outbound messages to 1000/week on the email channel.



The screenshot shows a configuration panel for a volume constraint. The title is "Maximum 1.0K Weekly with Channel: Email". There are three input fields: "Minimum volume" (empty), "Maximum volume" (set to 1000), and "Reset interval" (set to Weekly). Below these are two dropdown menus: "Channel" (set to Email) and "Action" (set to Select action). A "+ Add Property" link is at the bottom left.

The **Reset interval** option determines how frequently the volume limit is reset to the full quota. There are a number of reset interval options available, including: *Manually*, which will allow you to reset the constraint limits manually using the **Reset** button.



The screenshot shows a dropdown menu for "Reset interval". The options are: Weekly (selected), Manually, Each time it is accessed, Daily, Weekly (highlighted), Semi-Monthly, Monthly, Quarterly, and Yearly.

The other options in this list can be used when you want the system to automatically reset the constraint. For example, *Weekly* means that the system will reset the limits the first time they are applied in a particular week, Sunday being considered the first day of the week.

The remaining quota of the volume constraints, if any, in a given reset interval will be used in the following scheduled run. For example, if only 300 Rewards card emails can be sent in a week, the scheduled run might send 250 emails based on action prioritization. The remaining 50 emails will be sent in the same week during the next scheduled run.

In this case, configure the **Reset interval** as *Weekly* for both channel and action constraints.

Next, add an action constraint that will restrict the Standard card offer emails to only 100 per week.

> **Maximum 1.0K Weekly with Channel: Email**
1,000 remaining
 ⋮

∨ **Maximum 100 Weekly with Channel: Email, Action: StandardCard**
 ⋮

Minimum volume	Maximum volume	Reset interval
<input type="text"/>	<input type="text" value="100"/>	Weekly <input type="button" value="v"/>
Channel	Action	
Email <input type="button" value="v"/>	StandardCard <input type="button" value="v"/>	

[+ Add Property](#)

Finally, as per the requirements, add a property constraint that will limit offer emails related to credit cards of type "Master" to only 400 per week.

> **Maximum 1.0K Weekly with Channel: Email**
1,000 remaining
 ⋮

> **Maximum 100 Weekly with Channel: Email, Action: StandardCard**
 ⋮

∨ **Maximum 400 Weekly with Channel: Email, Type: Master**
 ⋮

Minimum volume	Maximum volume	Reset interval
<input type="text"/>	<input type="text" value="400"/>	Weekly <input type="button" value="v"/>
Channel	Action	
Email <input type="button" value="v"/>	Select action <input type="button" value="v"/>	
Property	Value	
Type <input type="button" value="v"/>	Master <input type="button" value="v"/>	

[+ Add Property](#)

Configuring volume constraint in the Next-Best-Action Designer

To implement this business requirement, you need to configure the Next-Best-Action Designer to perform always-on outbound customer engagement with the volume constraint configured.

On the **Channels** tab, configure the **Primary Schedule** to achieve this.

Primary schedule ?
Schedule cannot be run. Confirm start date is not in the past, and starting population is selected.
⋮

Recurrence ? ⚙ Not yet configured	Starting population ? ⚙ Not yet configured	Volume constraint (optional) ? ⚙ Not yet configured
---	--	---

The schedule has two mandatory configurations, **Recurrence** and **Starting population**. To include the volume limits on outbound, you must configure the **Volume constraint**.

Primary schedule ? DRAFT		
Recurrence ? ⚙️ Daily Start on 4/2/20 1:33 PM No end date Refresh the audience for each scheduled occurrence	Starting population ? ⚙️ All mailable customers	Volume constraint (optional) ? ⚙️ Limit Offers

With that, all the necessary configurations for this scenario are complete.

Run the schedule. Once the primary schedule has run, you can check the remaining volume limits for the constraints. The remaining quota of the volume constraints, if any, in a given interval will be used in the following scheduled run.

How should multiple actions for a customer be treated? ?	Reset constraint counts
<input checked="" type="radio"/> Individually for each action	<input checked="" type="radio"/> Individually
<input type="radio"/> As a group for the context (output all actions)	<input type="radio"/> All at once
<input type="radio"/> Return any action that does not exceed constraint	

> Maximum 50 Daily with Channel: Email	0 remaining	<input type="checkbox"/>	⋮
> Maximum 20 Daily with Channel: Email, Action: StandardCard	1 remaining	<input type="checkbox"/>	⋮

Sharing action details with third-party distributors

Description

Sending outbound communications via email and SMS has its challenges, including various operational, security and legal requirements. Sometimes, it's easier to let an external email service provider handle message delivery for you. Learn how to share action details with an external service provider using files and database templates.

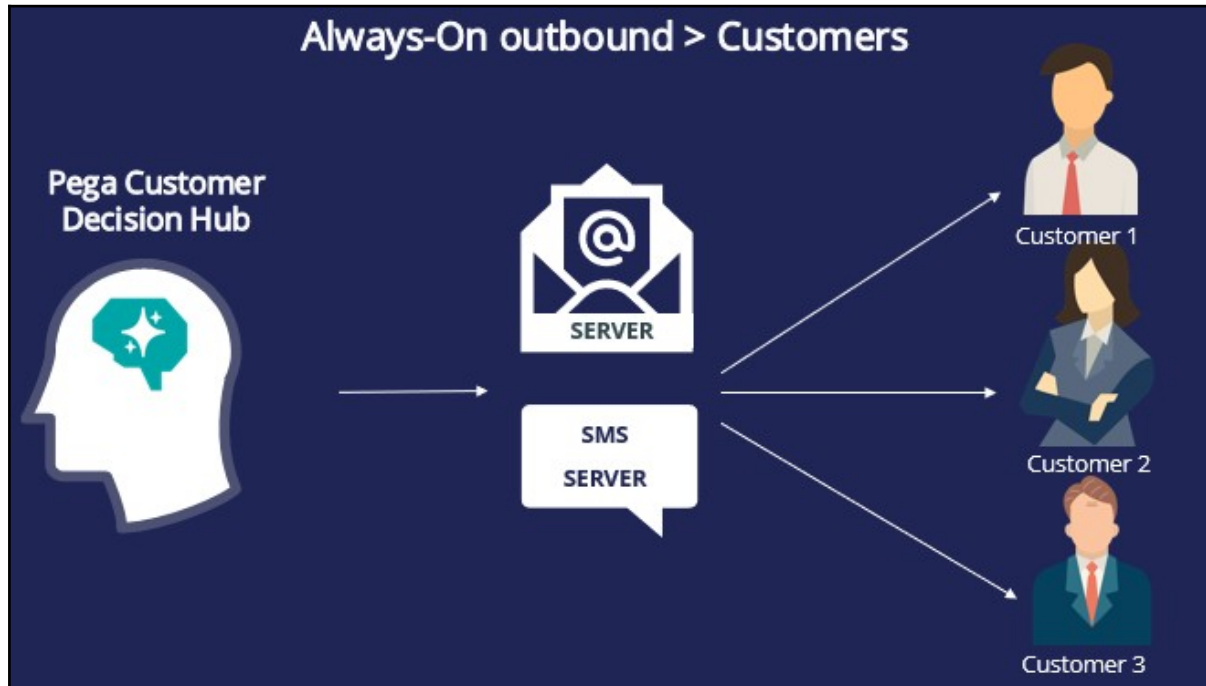
Learning objectives

- Explain outbound file and database templates
- Write action details to files or database tables to send to third-party email service vendors

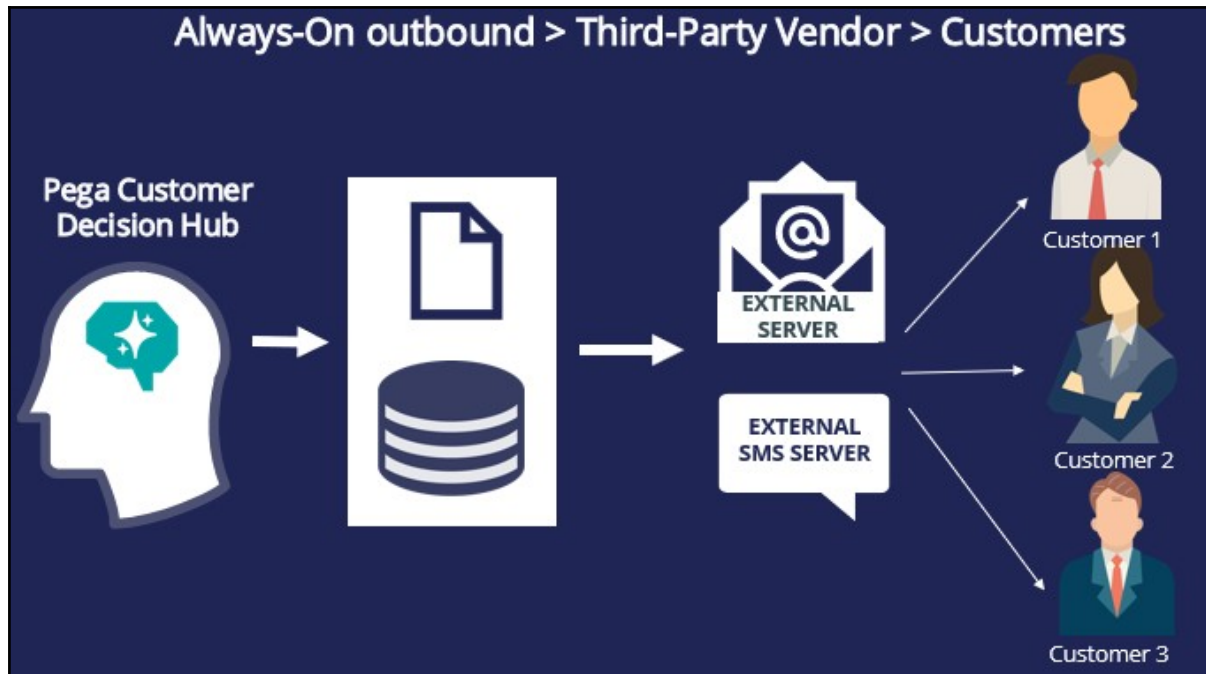
File and database templates

Why do you need outbound templates?

In outbound customer engagement, Pega Customer Decision Hub sends action messages to customers directly via email or SMS. That is, Pega Customer Decision Hub directly communicates with the email or SMS servers to send out outbound messages to qualified customers.



However, sometimes the necessary infrastructure to send action messages directly to customers may not be available. In such cases, you need to use a third-party email/SMS distributor. You need a medium to transfer the action and customer details, to a third-party distributor, which is typically a file or a database table.



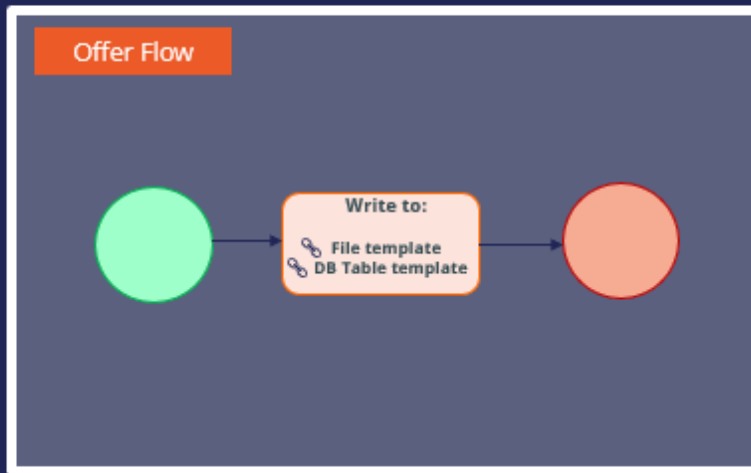
Pega Customer Decision Hub enables you to write the file in one of the following cloud-based destinations as well as the local filesystem.

- JFrog Artifactory
- Amazon S3
- Microsoft Azure

Writing action details to a file or database

Pega Customer Decision Hub enables you to easily write action details to a file on the local system or on a cloud-based destination such as Amazon S3 or to a database table, which can be sent to a third-party vendor or application for distribution. To achieve this, you need to configure an action flow to include a Send shape with the file or database template option enabled.

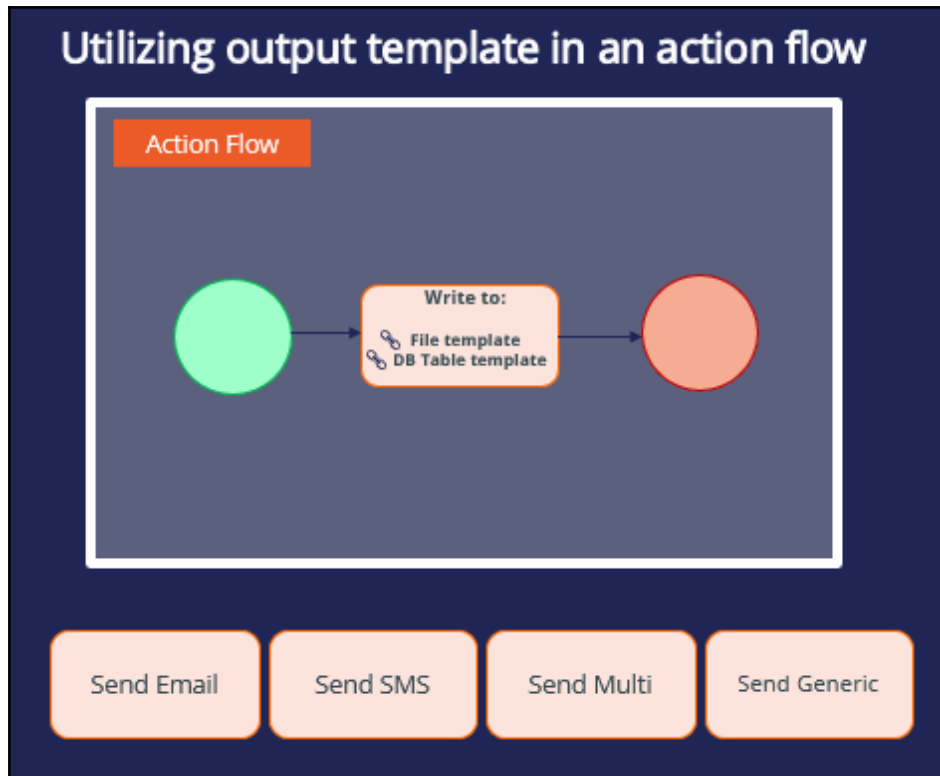
Utilizing output template in an action flow



The four available Send shapes that can be used for this purpose are: **Send Email**, **Send SMS**, **Send Multi**, and **Send Generic**. Any one of these shapes can be used to write action details to a file or a database table. However, when you know that the action messages will be sent eventually via Email or SMS by the third-party vendor, it is better to use the corresponding shape, as this will clearly illustrate the purpose of the component and the channel will be recorded as an Email or SMS in the Interaction History.

If you are unsure of the communication channel, or if it is a channel that Pega Customer Decision Hub cannot directly integrate with, use the Send Generic shape.

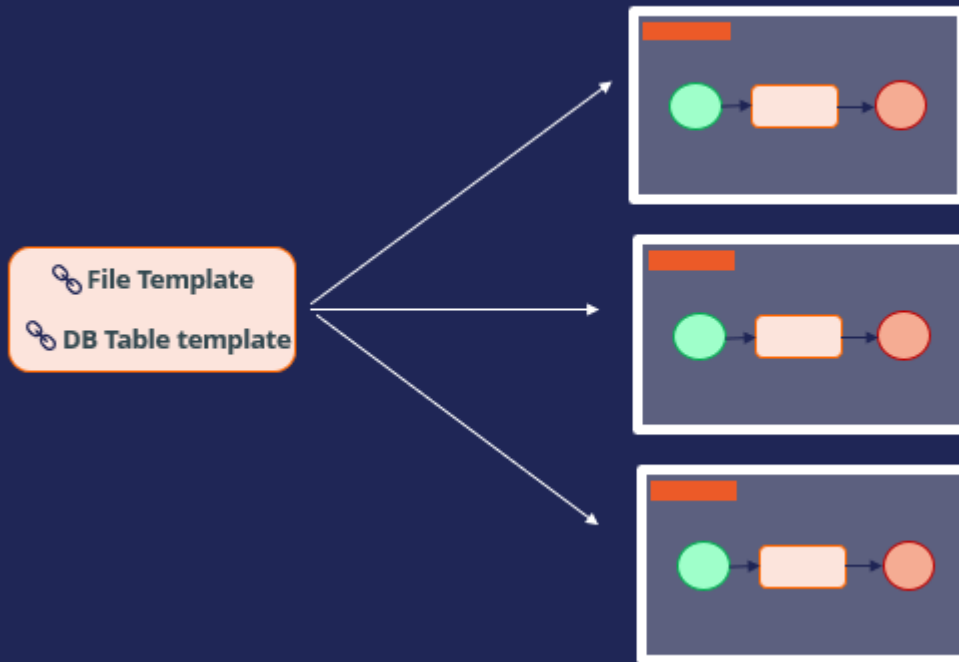
Utilizing output template in an action flow



Multiple action details in a single location

Details of multiple actions can be written to the same file or database table. To do this, you need to configure the same file/database table template in multiple action flows. This functionality is useful when *the **Next-Best-Action outbound schedule** includes multiple actions*. *This approach consolidates multiple action details into a single location.*

One file template > Multiple action flows



Creating an external treatment to share action details in a file or database

Outbound emails are sent either directly by Pega Customer Decision Hub or by a third-party email service provider. Use external email treatments to send the necessary customer data to email service providers in the form of a file or a database table. When you use external email treatments, Pega Customer Decision Hub™ does not store email content. In this case, Customer Decision Hub does not have direct integration with an SMTP server, but a third-party email service provider handles the service. These providers rely on file or database integrations and have a defined format of the data that they expect.

In such cases, create the file or database integrations in Customer Decision Hub with all the information you want to share with the third-party email service provider. Then, associate the file or database template with an external treatment. With external email treatments, Customer Decision Hub generates the data that the third-party email service provider requires to send the emails.

Business scenario

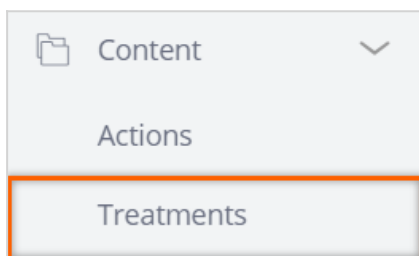
U+ Bank, a retail bank, sends credit card offers to eligible customers by email. However, to comply with the security and spamming regulations, the bank wants to use a third-party email delivery service instead of using the in-house email server. As a result, the offer details need to be written to a file that can be shared with a third-party email service provider and mapped to an external email treatment.

Creating a file template

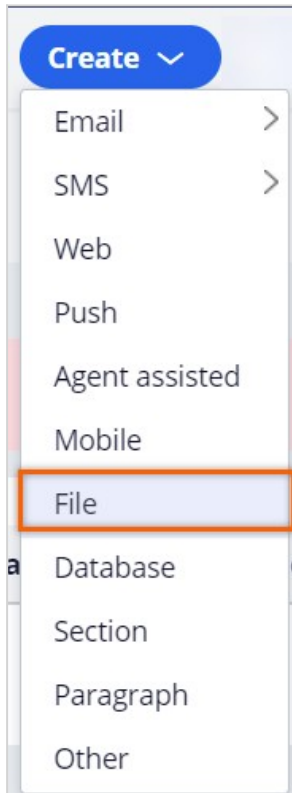
Pega Customer Decision Hub™ enables you to write action details to a file that can be sent to a third-party vendor or application for distribution.

To implement this business requirement, you need to configure a file template. The file template landing page can be accessed from the navigation pane, **Content** >

Treatments.



You can create a new file template from this page.



Provide a short description for the new file template, and the identifier is automatically populated when you tab out.

A screenshot of a web form titled 'Create file output template'. At the top right is a blue 'Create and open' button. Below the title is a section titled 'Output Template Record Configuration'. It contains a text input field with the value 'Generic File Template' and a label 'Identifier' with the value 'GenericFileTemplate' and an 'Edit' link. Below this is a 'Categorization' section with two dropdown menus: 'Issue' (with 'Select Issue' selected) and 'Group' (with 'Select Group' selected).

Creating a repository


When configuring a file template, provide details on an existing repository or create a new repository where the file needs to be stored.

In this case, you create a new repository.

Edit file output template: [Generic File Template \[Available\]](#)
 Top Level • GenericFileTemplate • File CDH-Artifacts:01-01-01

Definition Fields Finalize History

Repository name *

Local File System 

Provide a short description and a name for the repository to create one.

Create Repository **Create and open** Cancel ?

Short description *

Local file repository

Repository name

Local file repository

Select a **Repository Type** where you want to store the file.

Repository type

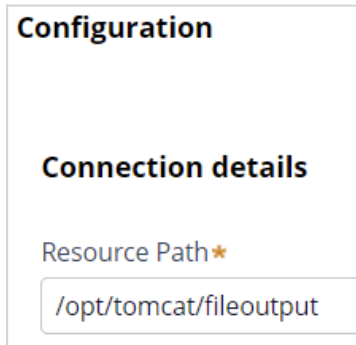
Select

Pega Customer Decision Hub supports various cloud-based destinations as repository types, such as **JFrog Artifactory**, **Amazon S3**, and **Microsoft Azure**, in addition to the local file system.

Select repository type

				
Pega	Artifactory	S3	File system	Azure

When you select the **File system** repository type, in the **Resource Path**, provide a directory path in the file system to where you want to store the file.



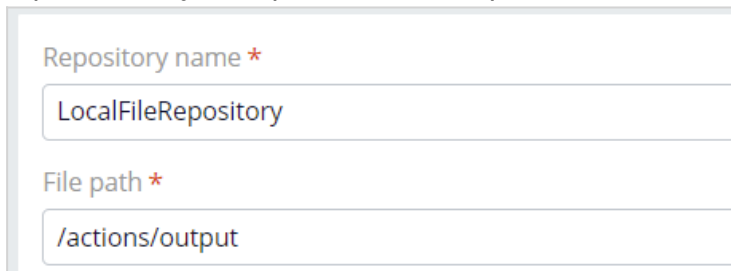
The screenshot shows a configuration window titled "Configuration". Under the "Connection details" section, there is a label "Resource Path*" followed by a text input field containing the path "/opt/tomcat/fileoutput".

Configuring a file template

In the **Edit file output template** page, continue configuring the file template.

Select the repository and provide a path where the file needs to be stored. The location is a sub-directory under the directory specified in the file system repository. The format for the file path and name is folder/filename.

For example, if the repository path is */opt/tomcat/fileoutput* and the file path is *actions/output/mailling-list.csv*, then the file will be created under */opt/tomcat/fileoutput/actions/output*.



The screenshot shows a configuration form with two input fields. The first field is labeled "Repository name *" and contains the text "LocalFileRepository". The second field is labeled "File path *" and contains the text "/actions/output".

Finalizing the output template

Finalization is the technical process of writing data from an internal staging area to the final file. Finalized data is ready to be consumed by a different system. The system can trigger finalization manually or automatically based on a schedule. Select the **Finalize template after Action Initiation** option to trigger the finalization after each run.

By finalizing a template at the end of each run, you ensure that all data in the file is the result of a single run. That is, the final output includes all data related to the run that triggered the finalization. If you want the data from multiple outbound runs to be written to the same file, leave this option unselected. When you leave this option unselected, you must configure a separate finalization schedule in the **Finalize** tab or finalize the file manually in the **Treatments > File** landing page.

If you choose to finalize at the end of each run, select **Include run ID in file name** to include the run ID for easier identification of the files generated in each run.

Repository name *
Local file repository
File path *
/opt/tomcat/fileoutput/actions/output
<input checked="" type="checkbox"/> Finalize template after Action initiation ?
<input checked="" type="checkbox"/> Include run ID in file name
<input checked="" type="checkbox"/> Include finalization date-time in file name

In the **Fields** tab, you can configure the information elements that you want to output.

You can configure the customer and action details that you want to share with the third-party email service provider. In the **Name** field, you enter a name for the field. In the **Content** field, you select the corresponding property. Use the *.Customer.* construct to access customer properties and *.OfferData.* construct to access action properties. As shown in the example, adding the treatment name as an information element helps to map the actions with the corresponding external treatments.

Definition				Fields	Finalize	History
Name	Type	Content	Format			
Customer_ID	Data F	.Customer.Custoi	n/a			
Customer_Name	Data F	.Customer.FullNa	n/a			
Customer_Email	Data F	.Customer.Prima	n/a			
Offer_Key_Code	Data F	.OfferData.KeyCo	n/a			
Offer_Name	Data F	.OfferData.pyNar	n/a			
Treatment_Name	Data F	.OfferData.pyTre	n/a			

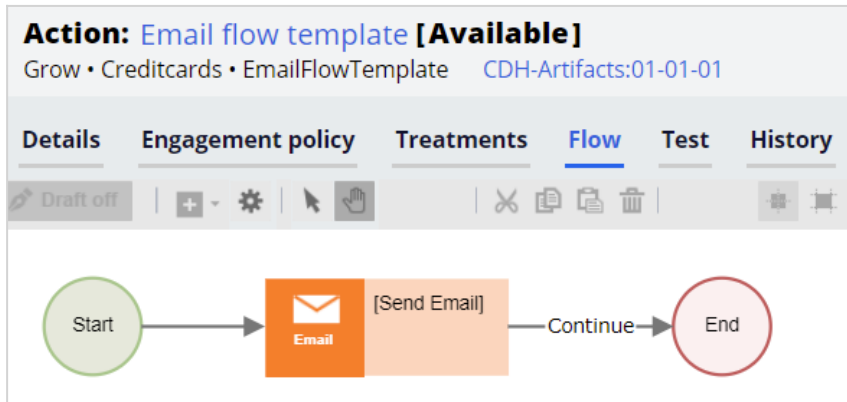
On the **Finalize** tab, you can configure further finalization options.

Definition	Fields	Finalize	History
Schedule			
<input type="checkbox"/> Enable schedule			
Delivery options			
<input type="checkbox"/> Send Email notifications when finalized from the Landing Page			

The **Advanced** section can be used for additional processing during the finalization process. For example, you can use it to initiate the file transfer to a particular location.

Using the file template in an action flow

You need to configure one of the **Send*** shapes, that is, Send Email, Send SMS, Send Generic or Send Multi shapes, in the action flow to reference the file template so that the action details are written to the output file during an outbound run.



To write action details to a file template, select the **Use associated Treatment** option. This option ensures that the treatment is associated with the action.

Send Email properties

Send Email: [Send Email]
Configure the content of the email and its delivery.

Treatment
What email treatment should be sent?

Use associated Treatment Specify Treatment Wait after sending

Advanced
What else can I do?

- ▶ Email account
- ▶ Tickets

Cancel Submit

Creating an external email treatment

Once you integrate the customer and action details into the file template, create an external email treatment. With external email treatments, Customer Decision Hub generates the data that the third-party email service provider requires to send the emails.

For this scenario, you create a new external email treatment, **Standard card email**. Add a short description and an appropriate **Keycode** for the email treatment. Then, select the file template to write the action details. You use the file template that you created earlier, which the system uses to write the action details that you want to share with the third-party email service provider. Next, add an email subject. You can specify a specific email subject that you want the third-party email service provider to use while sending the outbound emails. You add the `.OfferData.pyEmailSubject` property to the file or database template to include the subject as part of the details of the shared actions.

Details

Name *	Description
<input type="text" value="Standard card email"/>	<input type="text" value="Standard card email"/>
Type	Language
External	<input type="text"/>
Availability	
<input checked="" type="radio"/> Always	
<input type="radio"/> Never	
<input type="radio"/> Within a defined time period	
Channel eligibility	Treatment eligibility
<input type="text"/> <input checked="" type="checkbox"/> Apply ?	<input type="text"/>

Key code

Content

Template type * File template ⚙️

GenericFileTemplate

Subject

 ⚙️

Running an outbound schedule

To write the action details to a file, run an outbound schedule. The file with action details is stored in the specified repository.

```
Customer_ID,Customer_Name,Customer_Email,Offer_Key_Code,Offer_Name,Treatment_Name
CUST1605,Larissa Schuster,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST682,Anahi Torphy,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST5388,Sedrick Kutch,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST8488,Walton Daugherty,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST4058,Abby Russel,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST2954,Armani Hessel,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST1758,Asa Buckridge,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST10887,Madisyn Ondricka,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST7548,Arvilla Bartell,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST8634,Rodger Borer,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST9863,Bethel Greenfelder,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST11084,Morris Littel,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
18,John Smith,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST5918,Daija Walker,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST6540,Maeve Bauch,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST982,Kobe Waelchi,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST4119,Quinton Herman,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST3444,Tomas Schneider,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST4322,Myra Donnelly,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST7168,Oceane Muller,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST8997,Janessa Zboncak,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST4205,Nathaniel Block,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST3526,Angelica Collins,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST8695,Isabelle Predovic,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST844,Icile Trantow,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST3105,Jalen Rolfson,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST3404,Kyleigh Stroman,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST7290,Eldora Schneider,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST6031,Clotilde Balistreri,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST4730,Verna D'Amore,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST4185,Laron Price,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST4321,Howard Feeney,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST1186,Norma Bechtelar,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST3265,Omari Bogan,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST3351,Bart Durgan,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST2066,Unique Marquardt,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST6392,Kaitlyn Spinka,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
CUST9109,Javonte Rodriguez,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard,Rewards Plus card email
CUST2722,Charlene Conn,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST1287,Gilbert Bergnaum,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST8053,Ransom Adams,customer1@enablement.com,STANDARDCCARD,StandardCard,Standard card email
CUST9195,Dejon Little,customer1@enablement.com,PREMIERREWARDSCARD,PremierRewardsCard,Premier Rewards card email
CUST6161,Vallie Kling,customer1@enablement.com,REWARDSCARD,RewardsCard,Rewards card email
```

The file contains the fields as configured in the file template.

Writing action details to a database

Like a file template, you can map an external email treatment to a database template and make the outbound run write details to the database table. The process of creating a database template is similar to creating a file template. Instead of the repository, file path, and file name configurations, you must provide the **Database** name and **Table** name.

Edit database output template: Generic Output Template [Available]

Actions ▾ Save ▾

Definition Fields Finalize History

Database *
 ⚙

Table *

Finalize template at the end of each outbound run

Writing option
 Overwrite existing data
 Append to existing data

Bypassing action flow processing


In the always-on outbound scenario, you can bypass the action flow processing and write action details to a database table directly with a database template. Use this feature when you do not need to perform multiple steps in the life cycle of an action (such as send an action message, wait for a response, or take appropriate action upon response).

To use this feature, go to **Next-Best-Action Designer, Channel**. In the **Primary schedule** section, click the **Gear** icon next to **Recurrence** to access the outbound schedule configuration. Enable the **Write results using a database template** option to add a database template.

Note: Here, you can configure only database templates that have the **Writing option** set to **Append to existing data**.

Configure outbound schedule

Every weekday
 Every day or every nth day

Start
4/23/2020 1:03 AM 

End
 No end date
 Set number of occurrences
 End by a date

Refresh the audience
 Refresh the audience for each scheduled occurrence
 Refresh the audience only for the first scheduled occurrence

Write results using a database template
GenericDBTemplate

[+ Add template](#)

Capturing responses

When you use an internal email treatment and send the email content directly to the SMTP server, Customer Decision Hub captures the customer response (clicks and impressions). Because Customer Decision Hub does not have direct integration with the SMTP server in the case of external email treatments, the third-party email service provider receives the responses to all emails. In this case, an integration is necessary to send the captured responses to Customer Decision Hub to ensure that the system updates the interaction history and trains the AI models with the received responses. External systems can send the responses using a REST integration, calling the **Capture Response API**, or creating a data flow to process the responses from a file, stream, or database table.

Business agility in 1:1 customer engagement overview

Description

Learn the processes involved in creating a new action in a business operations environment and propagate the changes to the production environment.

Learning Objectives

After completing this module, you should be able to:

- Describe business agility and its importance in one-to-one customer engagement
- Distinguish between business-as-usual and enterprise changes
- Describe the steps involved in managing business changes in one-to-one customer engagement

Agility in a customer engagement project

What is agility?

Agility means that your organization is able to react quickly to changes in the marketplace, an important capability in marketing. In the digital age, things change constantly, and taking weeks or even months to translate a good idea into an actionable plan is bad for business.

Pega 1:1 Operations Manager increases your agility in the digital channels, giving business users everything they need to quickly transition to new content or messaging as needs change or trends emerge.

Business value of agility

Generally, changes in an enterprise application require IT involvement and a full IT development cycle. But business users often want to change things like the description of a product or its expiration date. In most enterprise applications, even these kinds of changes can be costly.

Pega 1:1 Operations Manager empowers business users to make these sorts of 'business-as-usual' changes themselves. This not only preserves resources, it enables business teams to be more responsive to changing business needs and trends in the marketplace.

Business changes vs enterprise level changes

Business-as-usual changes are small, day-to-day updates that result from normal business operations in digital touchpoints. It typically falls to the business operations team to satisfy ad-hoc requests from business teams to make these types of incremental changes to optimize existing artifacts. These are usually smaller pieces of work that need fast turnaround, such as:

- Adding a new Action (such as offer or message) with treatments to production
- Removing or disabling an Action in production
- Modifying a group-level Engagement Policy
- Enabling an integrated channel
- Fixing a typo in a call center message provided by Pega Customer Decision Hub™

Enterprise changes tend to require larger pieces of strategic development work and usually disrupt normal operations. These tasks may introduce something new or implement a change in the way something existing is presented or works.

- Adding new data attributes to the customer profile
- Integrating a new channel
- Implementing an approval flow for change requests

Pega 1:1 Operations Manager delivers business agility

Pega 1:1 Operations Manager provides a guided process for managing changes to decisioning artifacts. Business users determine what must change in a collaborative and unified way, and the system helps the user implement the changes by:

- Simplifying change processes and reducing administrative tasks
- Providing an intuitive, task-driven interface that guides the user from task to task
- Facilitating collaboration between operations, business, and technical teams

Change management process

Introduction

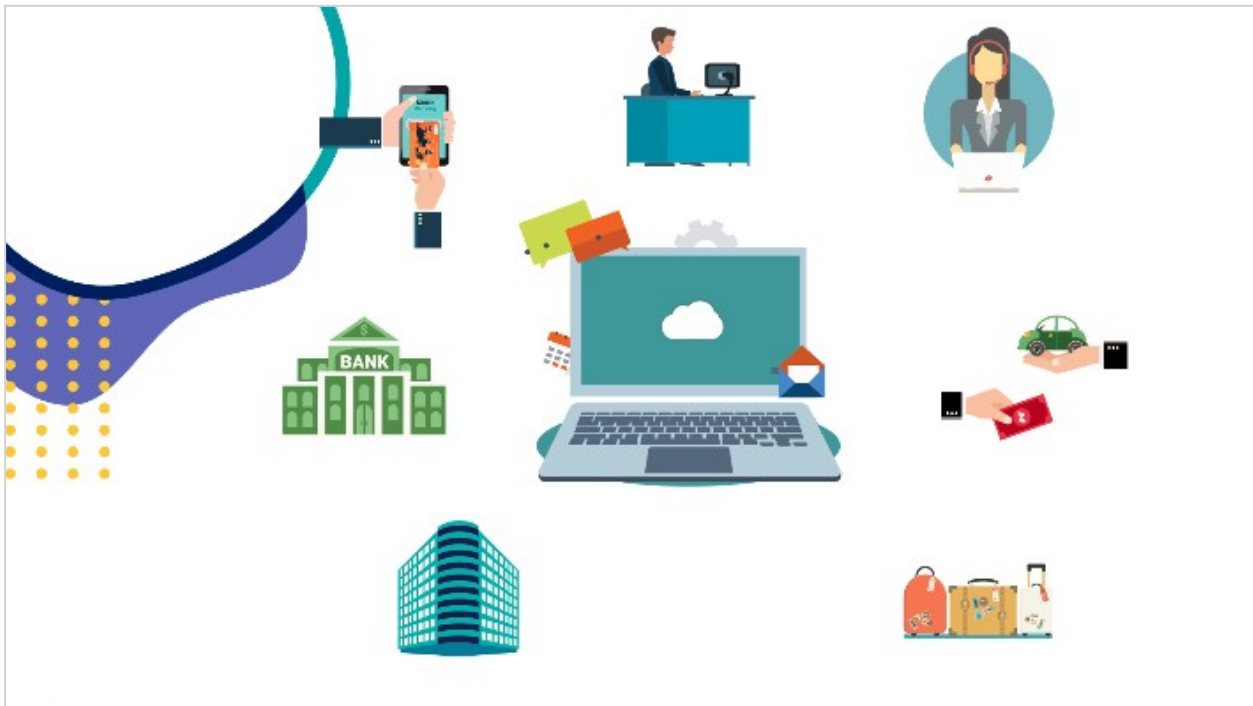
When business requirements change rapidly, the software development process needs to be more agile, while still producing high-quality, reliable software. This topic covers the importance of the change management process, and how it works in a one-to-one customer engagement project for Pega Cloud® customers.

Transcript

This video describes the change management process in a one-to-one customer engagement project and the Pega cloud environments that support the process.

Let's start by understanding some background on change management in enterprise software development projects.

In recent decades, organizations have been using computer software to automate many traditionally manual tasks.



Business processes need to constantly evolve as customer behavior and market conditions change. Naturally, the software that supports business practices needs

to evolve as well. As the pace of change in business requirements increases, the software development process needs to be more agile, while still producing high-quality and reliable software.

Here's a simplified view of an enterprise software development cycle. It consists of four high-level stages.

Developers develop new software or update existing software.

The work from several developers is merged into a single system in the integration phase.

The new software version goes through testing, and the final, approved software (or a software change) is deployed into production.

This cycle repeats for new as well as incremental updates to existing software.

A software development process is supported by different environments.

Let's study the environments available to Pega Cloud® customers in support of a one-to-one customer engagement project using Pega Customer Decision Hub™.



- A development environment is one in which developers create new versions of the application by adding enhancements or fixing issues. It is referred to as a system of record (SOR) for Pega applications.
- A staging environment is used for various testing such as functional testing, unit testing, and user acceptance testing.
- The business operations environment (BOE) is a replica of the production environment. However, it contains only a sample of the production data. This

is where the business operations team creates and tests new business artifacts and conducts simulations.

- The production environment is the main system that propagates the next best actions to external channels, collects customer responses, and is where the AI learning happens. It is also used for live monitoring of key performance indicators.

In a one-to-one customer engagement project, changes to the application can be classified into two categories: enterprise changes and business changes.

Enterprise changes are the changes that developers make to the Pega application. An example of enterprise changes are extensions to the core Pega application and its integration points with external systems.

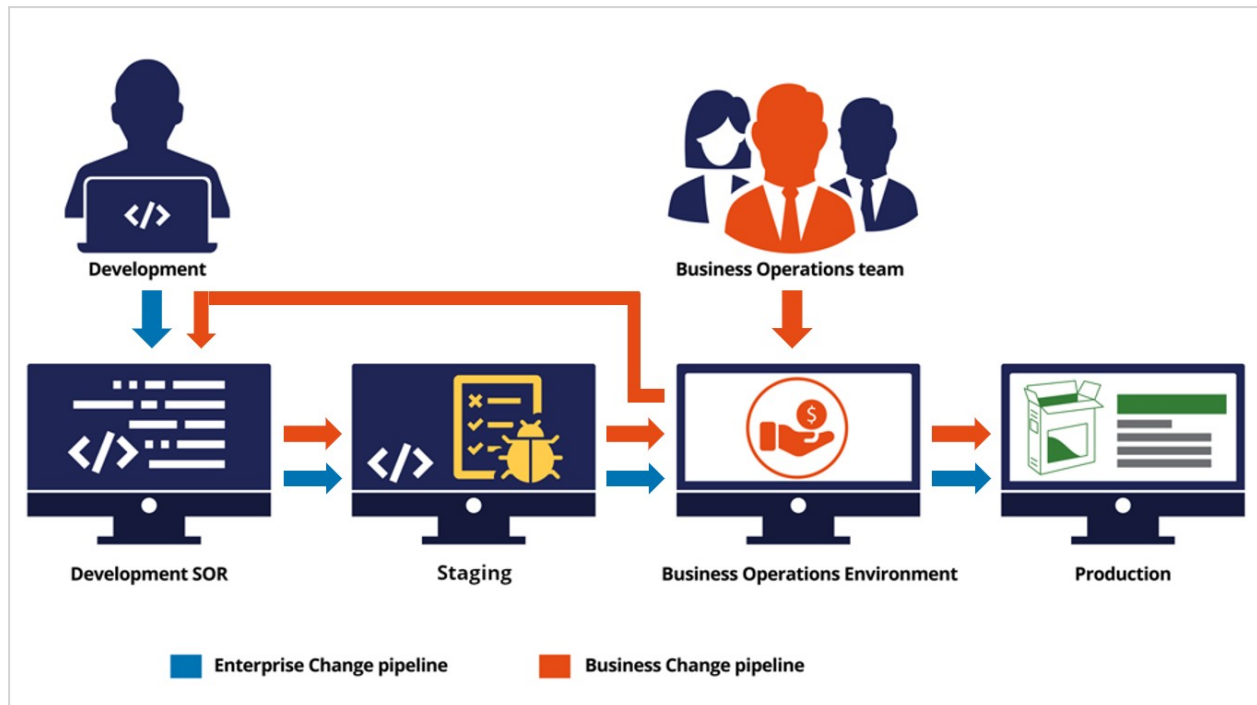
Developers make these changes in the development environment. Changes to the application are pushed to other environments through the enterprise-change pipeline managed by the Pega Deployment Manager™.

Examples of business changes include creating a new action or updating an existing action with new treatments or engagement policies. Also, this environment is used to carry out various simulations and analyses, for example, to test if there is an ethical bias in the decisions made by the next-best-action strategy framework. Business changes are made by the business operations team in the business operations environment.

The business content team uses the 1:1 Operations Manager portal to initiate changes in the business operations environment.

In the BOE, the changes are carried out in an application overlay. The application overlay defines the scope in which business users can change the application (for example, by modifying actions, treatments, decision strategies, and so on) to accommodate the constantly changing business conditions and requirements.

Changes from the business operations environment are pushed to the development environment and from there to other environments through the business change pipeline.



You have reached the end of this video which showed you:

- The importance of the change management process in an enterprise software development project.
- The high-level software development cycle.
- The cloud environments provided by Pega for a one-to-one customer engagement project using Pega Customer Decision Hub.
- The flow of enterprise and business changes through the enterprise and business change pipelines.

Pega 1:1 Operations Manager overview

Description

Learn how Pega 1:1 Operations Manager helps streamline the change management processes that affect one-to-one customer engagement orchestrated by Pega Customer Decision Hub™.

Learning Objectives

After completing this module, you should be able to:

- Explain the lifecycle of a change request
- List of the types of change requests
- Describe the various personas involved in the change management process

Building your business operations team

Business operations team

The goal of the business operations team is to bring agility to your change management so your organization can achieve excellence in one-to-one customer engagement.

It is best practice to have a single team for managing business changes at the organization level, typically known as the business operations team. This team includes three types of users.

Business users

Business users create or submit a new business change requirement, known as a change request in the Pega 1:1 Operations Manager application.

Responsible for:

- Planning new offers and messages as next best actions for customers.
- Leveraging next best action to drive performance and meet company goals.

Top activities:

- Planning customer engagement activities.
- Reviewing and approving content and offer distribution.

Examples:

- At a bank, the following job profiles represent business users:



 **Sheryl Williams**
Digital marketing manager
(Credit card division)
U+ Bank
www.uplusbank.com



Ben Stokes

Digital marketing manager
(Personal loan division)

U+ Bank

www.uplusbank.com



Robert Hill

Digital marketing manager
(Owned media-web/mobile)

U+ Bank

www.uplusbank.com

Team lead

The team lead reviews and manages priorities for business change requests.

Responsible for:

- Conducting the agile ceremonies, unblocking issues, and managing the day-to-day activities required to support the scrum team. The team lead works with IT to deploy revisions.

Top activities:

- Run daily stand up, show and tell, and sprint retrospective meetings
- Remove blockers impeding team progress
- Ensure compliance with change management best practices and policies
- Work with IT on deployments

Example:

- Team leads are often scrum masters on a business operations team.



It is a best practice to include one team lead for managing your business operations team.

Next-best-action specialists

Next-best-action specialists work on tasks to implement the business changes requested by business users.

Responsible for:

- Implementing and testing offers or messages in the business operations environment.

Top activities:

- Supporting the definition and assessment of business requirements.
- Translating business needs into next-best-action artifacts and strategies.
- Building and testing artifacts to understand their impact.

Examples:

- Next-best-action specialists are often known as the marketing operations team. They have hands-on experience with marketing automation and related software. Next-best-action specialists understand the functions of various marketing artifacts such as actions, treatments, engagement policy, and decision strategies. Next-best-action specialists also have good working knowledge of how next-best-action decisioning works in Pega Customer Decision Hub™.



Davide Achebe

Operations team member

U+ Bank

www.uplusbank.com



Maggie Square

Operations team member

U+ Bank

www.uplusbank.com



Betsy Smith

Operations team member

U+ Bank

www.uplusbank.com

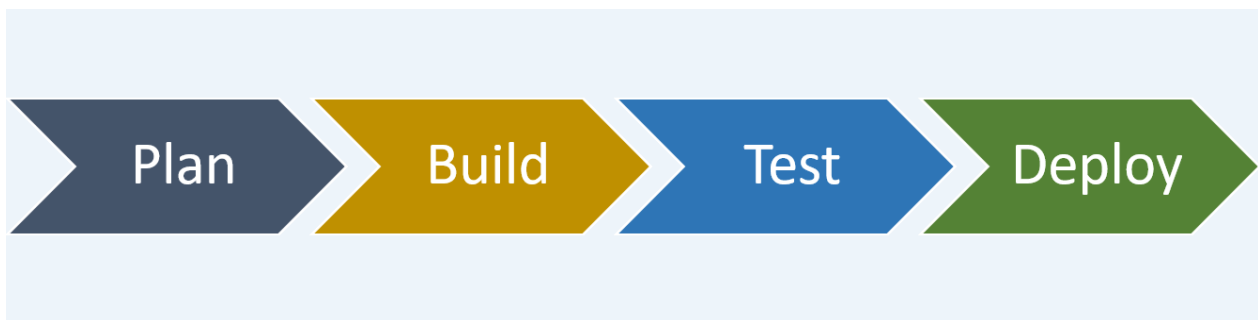
You can scale up operations by adding more team members to this group.

Lifecycle of a change request

Change request stages

In Pega 1:1 Operations Manager, a change request represents a change to one or more next-best-action artifacts in Pega Customer Decision Hub™. Together, business users determine and agree upon what needs to change, and the system guides the next-best-action specialist to make those changes.

A change request goes through four stages in its lifecycle. Each stage encapsulates the various tasks required to carry out the change effectively. In each stage, a designated user or set of users carries out the required tasks. Once the necessary tasks are complete, the system automatically manages the movement of the change request from one stage to another.



The lifecycle of a change request contains the following four stages:

- Plan
 - A business user creates the change request.
 - **Key** information about the change is captured from the business user directly into Pega 1:1 Operations Manager.
 - The team lead quickly **plans** and prioritizes the request so that the operational team can focus on more high-value work.
- Build
 - The operations team **builds** the next-best-action artifacts using a simple, guided process.
 - Next-best-action specialists translate the change request details into corresponding next-best-action artifacts in Pega Customer Decision Hub.
 - The system automatically creates or updates the corresponding artifacts.
- Test

- The system recommends that the next-best-action specialists verify and test the generated or modified artifacts to ensure that the change meets the business' expectations.
 - Changes can be tested in-context to reduce QA cycles.
- Deploy
 - The team lead approves the change request and transfers it to the team responsible for deploying the changes to the production environment.
 - Pega Deployment Manager enables the deployment team to safely **deploy** the changes to a production pipeline.

Change request types

Introduction

Business requirements change rapidly and typically have a different release cycle compared to the enterprise release cycle.

To ensure logical separation between technical and non-technical changes in change management, business changes or non-technical changes are made in the Business Operations Environment (BOE), according to business requirements.

You can make two types of business changes to your Customer Decision Hub application from the BOE:

- High-volume day-to-day business changes, such as creating new actions and treatments, and updating existing action and treatment details.
- Low-volume changes such as changes to taxonomy, arbitration, and group-level contact policies.

As a business user, you create a change request to initiate the change management process in 1:1 Operations Manager to make business-as-usual changes to the Pega Customer Decision Hub.

Three types of change requests are currently supported. Selecting the right change request category initiates the appropriate change management process flow.

Create new action

Use the **Create new action** category when submitting a request to create a new action.

Create Action

OUTBOUND

OUTBOUND

OUTBOUND INBOUND

Re-engagement
Promotional

Re-engage customers who have expressed interest. Examples are a shopping cart abandonment or an indication of interest.

OUTBOUND INBOUND

Retention
Promotional

Engage with customers to ensure they remain loyal, to prevent them from leaving, or win them back. Examples are a one-time credit of \$50 or a discount on your existing services.

OUTBOUND INBOUND

Sales
Promotional

Promote a new or upgraded product or service. Examples are marketing or promotional offers for a new credit card or phone upgrade.

OUTBOUND INBOUND

Announcement
One-time

Send a message to a specific group of customers on an alternate schedule. Examples are the closing of a retail location or message from the CEO.

OUTBOUND

Emergency
One-time

Send a message to a specific group of customers on an alternate schedule. Examples are a service outage or other disruption of service.

OUTBOUND

Change request name *
Promote new card on the web - cashback 5%

Completion date *
8/8/2023

Description
Create new leads from web channel for new Cashback 5% credit card.

Attach files
Filename

Remaining: 934 characters
Are you making this change to fix a production defect? [?](#)
 Yes No

Update existing action

Use the **Update existing action** category when submitting a request to update an existing action.

Update existing Actions

Name *

Completion date *

Are you making this change to fix a production defect? ?

Yes No

Business issue ?

Group ?

Description

Remaining: 937 characters

Attach files

As part of updating an existing action, you can request one or more changes at the action level. The available tasks enable you to add a new treatment, modify the

action-level engagement policy and modify the action details.

- Edit attributes
- Edit availability
- Edit bundle attributes
- Edit journey details
- Edit engagement policy
- Edit treatments
- Edit other attributes

Request type Other

If the change request is neither creating a new action nor updating an existing action, then you can select the **Other** category. This case type is best suited to making changes that are not related to a single action, such as editing engagement

policies, changing arbitration rules, and so on.

New operations change request

Name *
Add applicability condition to Credit cards

Select type *
 Other

Completion date *
8/8/2023

Are you making this change to fix a production defect? ?
 Yes No

Business issue ?
Grow

Group ?
Credit cards

Description
Credit cards are applicable only for the customers with income more than 10000 USD.

Remaining: 917 characters

Attach files
Filename

A change request with **Other** as its category is assigned to a team lead, who then adds one or more tasks from the predefined list of objectives according to the

business requirement. These tasks are typically assigned to an NBA designer.

T Add Task
Due 2 days from now

Objective *

Select...


- New internal email treatment
- New internal SMS treatment
- New when rule
- New external treatment
- Add or edit business taxonomy
- Manage Real-time containers, Events and Outbound schedules
- Manage group-level Engagement policies
- Manage prioritization settings (Arbitration)
- Edit Contact policies, Channel and Action limits
- Edit existing internal email treatment
- Edit existing internal SMS treatment
- New audience
- New DB or File treatment

Revision Management

If the required business change is beyond the scope of the three change request types, such as edit a strategy, or edit a scorecard, then the team lead initiates the change by creating a new revision in the revision management section of the 1:1 Operations Manager.

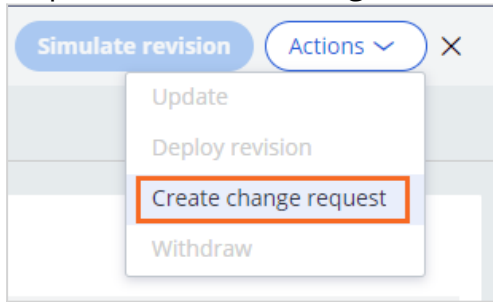
CDH standard release

Update business rules

 R-1
Due 08/08/2023 • **IN PROGRESS** • change requests 1

Then, in the Customer Decision Hub portal, the team lead creates a change request within the revision created in the 1:1 Operations Manager. The team lead includes all rules that are required to carry out the change and typically assigns the change

request to an NBA designer.



Fast-track change request

You implement business changes in the business operations environment (BOE) by using revisions. Each revision consists of one or more change requests. After individually implementing these change requests, when they are ready for deployment, the Revision Manager deploys all the change requests together as a release. This process is a standard release.

Fast-track change requests offer a means to efficiently implement urgent or unplanned business demands and deploy them immediately.

Mark the change request as a production fix to convert it into a fast-track change request. This step ensures the change request is added to fast-release revision.

Update existing Actions — ✕

Name *

Completion date *

Are you making this change to fix a production defect? [?](#)

Yes No

Business issue [?](#)

 ▼

Group [?](#)

 ▼

Description

 to "for"."/>

Remaining: 923 characters

Attach files

Once the change requests in a revision are individually implemented and ready for deployment, the Revision Manager deploys all the change requests using business pipelines. The standard revision uses the Standard business pipeline while the fast-release revision is deployed using the Fast Release pipeline.

Launching a new offer on web

Description

Learn the processes involved in creating a new action in a business operations environment and propagate the changes to the production environment.

Learning Objectives

After completing this module, you should be able to:

- Submit a request for a new action
- Manage change request priorities
- Build the action with the appropriate details
- Move the changes to the production environment

Using Automated insights when creating treatments in 1:1 Operations Manager

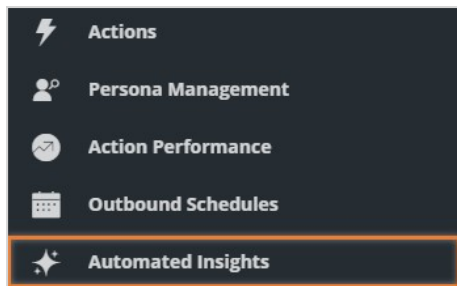
Introduction

Pega Infinity™ '23, introduces new features that use the power of Pega GenAI™ to simplify your daily tasks in Pega 1:1 Operations Manager. One of the main generative AI features available in 1:1 Operations Manager is Automated insights.

Transcript

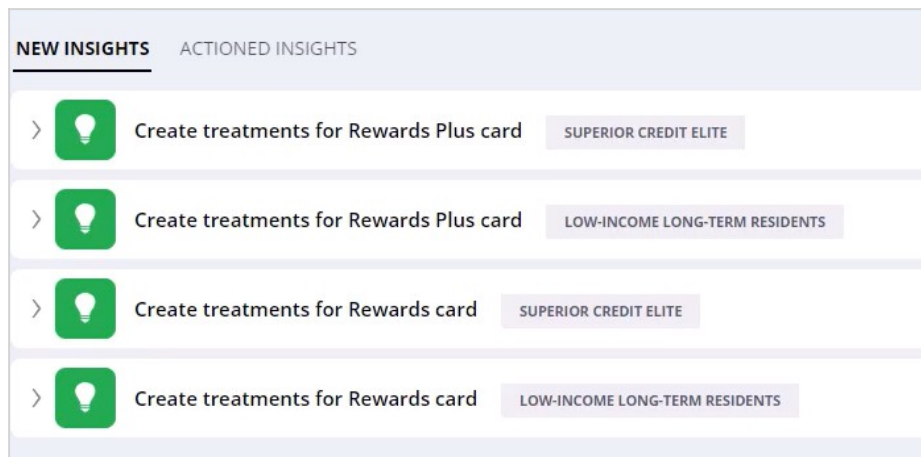
This video shows you how to use **Automated insights** to make the change request creation process more efficient and effective.

Automated insights combine the power and features of Pega Value Finder analysis with Pega GenAI, to detect underperforming Next-Best-Actions and provide actionable recommendations for improving actions and treatments. The suggestions are available in 1:1 Operations Manager, in the **Automated Insights** section.

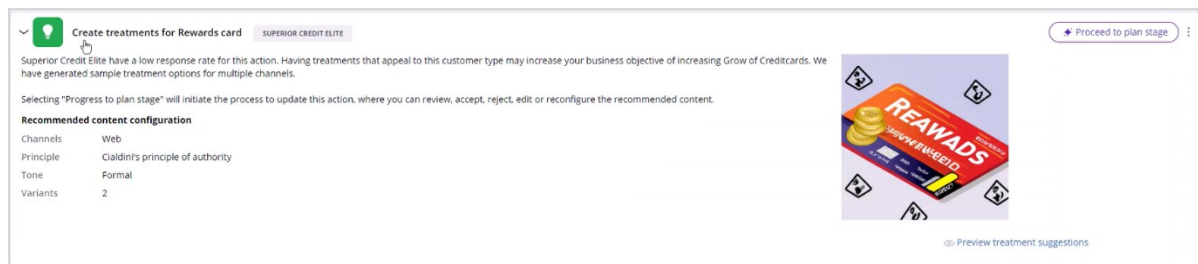


U+Bank has several active credit card offers. Based on a Value Finder analysis, the treatments associated with the actions do not bring the maximum business value. The bank aims to increase the accept rate of existing actions.

On the **Automated insights** tab, you can explore the recommended solutions to enhance actions and boost business value.



Expand the single recommendation to view its description and details. In the **Recommended content configuration** section, you can review the recommended treatment settings and check how many variants of the treatments are preconfigured. Click **Preview treatment suggestions** to see the two options. You will be able to select one of these options at a later stage.



In this case, a particular audience exhibits a low response rate for the Rewards card. Presenting this offer to the **Superior Credit Elite** customer type would likely result in a higher accept rate.

Selecting **Proceed to a plan stage** initiates the process of updating the action with the recommended treatment. The newly created change request includes Pega GenAI-generated suggestions, providing the opportunity to modify or reject them based on business needs.

New change request: Update action treatments ✕

Generated with Pega GenAI™
 This change request will include AI-generated treatment suggestions. You will have the opportunity to decline or modify this content, or add your own treatments

Change request name *

Action
 Rewards card

Completion date *

Description

Remaining: 947 characters

Attach files

In the **Define channels and treatments** stage, you can observe two treatment recommendations for the action. U+ Bank aims to maximize the interest in the action, so proceed with both recommended options.

Each recommendation provides two variants of predefined treatments with Pega GenAI-generated images. Additionally, you can also observe the initial **Rewards card tile** treatment. Remove the variants that are less appealing, then set the **Placement type** to **Tile** and continue.

Automated insights


Create treatments for Superior Credit Elite [Use and refine](#)

Superior Credit Elite have a low response rate for this action. Having treatments that appeal to this customer type may increase your business objective of increasing Grow of Creditcards. We have generated sample treatment options for multiple channels.

Selecting "Progress to plan stage" will initiate the process to update this action, where you can review, accept, reject, edit or reconfigure the recommended content.

Recommended content configuration

Channels	Web
Principle	Cialdini's principle of authority
Tone	Formal
Variants	2



[Preview treatment suggestions](#)

[>](#) **Create treatments for Low-income long-term residents** [Use and refine](#)

After you successfully create the change request as a Business User, log in as a Team Lead and prioritize the change request.

The next step is to complete the action build task. Log in as an NBA Specialist and check if you see the Design web treatment tasks for each of the newly created treatments. Assign the Design web treatment tasks to yourself and complete them one by one. As an NBA Specialist, you can also regenerate Pega GenAI-generated treatment images if they don't meet the business requirements. Pega GenAI-generated images can be used as an inspiration. However, the final decision about the content rests with the operations team.

From this point, the NBA Specialist, the Team Lead and the Revision Manger can finish the process until the new changes are deployed to production.

You have reached the end of this video. What did it show you?

- What Automated Insights are in 1:1 Operations Manager.
- How to use Automated Insights recommendations to make the Change Request creation process more effective and efficient.

Submitting a request for a new action

Introduction

Pega 1:1 Operations Manager streamlines the business change management process. With the Pega 1:1 Operations Manager, users introduce changes to Pega Customer Decision Hub™ by using a safe and guided process, test the changes, and conduct simulations before pushing the changes to the production environment.

Transcript

This video explains how to submit a change request in 1:1 Operations Manager.

The marketing team at U+ Bank, a retail bank, uses the web self-service portal as a marketing channel. The bank wants to promote using credit cards for utility bill payments to all customers who log in to the self-service portal.

The bank wants to introduce a new cashback offer of 5% to the customers who make bill payments with the credit card.

Business users and next-best-action specialists use Pega 1:1 Operations Manager to work collaboratively in a business operations environment.

A business user first creates a change request in Pega 1:1 Operations Manager to initiate adding a new action. A change request represents a change to one or more next-best-action artifacts.

As a business user, you create a change request to capture the business context of the required change.

The change request category indicates the type of change. In this case, you want to introduce a new offer, so click **Create >Action**.

In the action creation window, you can select the pre-defined action type, that most closely matches the use case or you can create the custom action change request.

Opting for the predefined action type automatically populates certain data within the change request.

The bank wants to introduce a new offer, so select the category Sales.

In the change request form, enter a name that describes the purpose of the request.

As a business user, you provide the completion date as the expected timeframe in which the next-best-action specialists complete the request.

You can use the **Description** section to provide any additional business details that are useful for the next-best-action specialists.

Create the new change request. The new change request has a unique ID.

The change request is currently in the **Plan** stage.

Observe that the first task to be completed is **Define attributes**. The purpose of this task is to capture additional details from the business user about the new action to help the next-best-action specialists who work on it later.

In the **Define attributes** section, enter a name for the new action. In this case, the `Cashback 5 card`.

Select the appropriate Business issue and Group if you know it. These details can be changed later in collaboration with the next-best-action specialist.

Continue to define the availability.

In the **Define availability** section, you can provide details on action availability.

Continue to define the journey stage.

In the **Define journey stage**, you can mark the action to be available only when a customer is on a specified customer journey stage.

Click continue to skip this step and define the engagement policy.

In the **Define engagement policy** section, there are two options to define the engagement policy details.

You can describe the requirements using business language. The NBA Specialist is responsible for implementing these requirements later in the build stage.

You can also select the **Use condition builder** option to see four categories to define the criteria.

Eligibility rules are rules that qualify a customer for an action. For example, the age of the customer must be greater than or equal to 18.

Applicability rules are factors that determine if an action is appropriate at this point for this customer. For example, the customer does not own any credit cards yet.

Suitability rules are conditions that determine if an action is deemed appropriate and is in the best interest of the customer. For example, customer's debt-to-income ratio should be less than or equal to 48.

Contact policy conditions help avoid customer fatigue through over-communication. For example, hide if an offer is already accepted.

In this case, you add an eligibility rule: `Annual Income is greater than 10000`.

Continue to define the channels.

Finally, in the **Define channels** section, you can select the channel and add details about the channel in which the action is presented. As you opted for the predefined "Sales" action type at the beginning of the change request creation process, two channels—E-mail and Web—are preselected at this stage. In this case, the bank wants to display this offer on the web channel. Therefore, delete the E-mail channel group. In the **Web channel group** window, you can provide details for one or more web treatments (for example, the availability date and the placement type such as hero or tile). In this case, a tile.

Finish the task.

Notice that there is a new task in the **To do** section: **Rank change request**. As a business user, you are unable to work on these tasks as are intended for a team lead.

You have reached the end of this video. What did it show you?

- How to submit a change request in 1:1 Operations Manager as a business user.

Prioritizing change requests

Introduction

Prioritizing the available change requests helps the NBA Specialists to plan their day and work on the most important tasks first.

Transcript

This video explains how to prioritize a change request and submit it to the next-best-action specialist (or, NBA Specialist) in 1:1 Operations Manager.

A business user submits a change request to create a new action intended to promote using credit cards for utility bill payments. The business user has provided details for the action attributes by specifying all the details for the new action.

As a team lead, you are responsible for two tasks in the **Plan** stage.

The first task is to prioritize the available change requests so that the NBA Specialists can plan their day and work on the most important tasks first.

The second task is to assign peer reviewers to verify the engagement policy defined in the **Build** stage.

You must first rank the change request.

By default, the rank of a new change request is 0. In the **Rank** field, provide a numerical value that represents the relative priority of the change request compared to others in the work pool.

Proceed to the next task.

Now assign a peer reviewer. A peer reviewer can be any operator who can verify the change request details given by the business user against the configuration made by the NBA Specialist. In this case, the peer reviewer is the business user.

Complete the task.

Notice that there are new tasks that correspond to the **Build** stage of the change request in the **To do** section.

In this case, because the change request is a **Create new action** type, the following tasks are created: Define action details, Configure engagement policy, and Define web treatment.

The NBA Specialists are responsible for the tasks.

You have reached the end of this video. What did it show you?

- How to prioritize a change request and assign a peer reviewer to the request as a team lead.

Completing the action build tasks

The build stage is where an NBA Specialist completes the various build tasks generated by the system to create an action. This is the step in which you map the business requirements to specific settings in Pega Customer Decision Hub™ so that the system can auto-generate the relevant artifacts.

Transcript

This video shows you how to complete the action build tasks in 1:1 Operations Manager as a next-best-action specialist, or NBA Specialist.

A business user has recently submitted a change request to create a new action to promote credit cards for utility payments. The team lead has assigned a high priority to the change request so that you can work on it as an NBA specialist.

As an NBA Specialist, you can access the change request that is in the queue on the **Change requests** landing page. Open the request. As an NBA Specialist, you have various build tasks to complete that the system generated to implement the business requirements that the business user defined. In this case, you have three tasks. You need to assign any task to yourself before beginning work so that the system prevents a second person from picking up the same task.

The **Define action details** task corresponds to defining the basic action attributes. Assign the task to yourself and continue. The fields are prepopulated with values based on the information that the business user provided when creating the change request. However, you can change the prepopulated values in this **Build** task.

The fields correspond to the action attributes in Pega Customer Decision Hub™. For example, **Name** is the action name, and **Business issue** and **Group** decide the categorization of the action in the business hierarchy. Ensure that the **Availability>Always** is selected. Complete the task.

The **To do** list now has just two tasks remaining. The next task is to define the journey stage of the action. Assign the task to yourself.

On the **Define journey stage** tab of the task pane, you can mark the action to be available only when a customer is on a specified customer journey stage. Click continue to skip this step and define the engagement policy.

On the **Configure engagement policy** tab, you can view the engagement policy that the business user defined per the business requirement.

In this case, the business user defined the eligibility rule: **Annual income is greater than 10000**. Continue the task.

The next step is to verify if the engagement policy is correct. In this case, select Joanna, a customer who is supposed to receive the **Cashback 5 card** action, as her annual income is greater than 10000, the defined eligibility value. Run the validation and continue. Add a relevant comment and sent the task for peer review.

The last task is to design a web treatment for the action. Enter the respective details and complete the task.

The change request creation is now completed. Because the team lead assigned a peer reviewer to this change request, you now need to send it to the reviewer. In this case, the reviewer is the business user.

Once the business user reviews and approves the change request, you need to review all the tasks and resolve the change request as an NBA Specialist. When all **Build** tasks are complete, refresh the change request page. Finally, validate and approve all the artifacts. Alternatively, if you want to change any attributes of the newly created artifacts, you can click **Return to build** to change the values and then regenerate the artifacts.

Now, as the Team Lead, open the change request to verify it. Approve the change request to promote it to the **Deployment** stage.

You have reached the end of this video. What did it show you?

- How to complete the action build tasks in 1:1 Operations Manager as an NBA specialist.
- How to review a task as a business user.
- How to, as a team lead, validate tasks that the NBA Specialists developed.

Deploying changes to production

In this stage you push the changes made by the business operations team to the pipeline for deployment.

Transcript

This video shows you how to deploy changes that are made in the business operations environment to the production environment.

To create a new action that promotes credit cards use for utility payments, a next-best-action specialist, or NBA Specialist, has completed all the **Build** tasks and validated the generated artifacts, such as actions and treatments.

Then, the team lead promoted the change request to the revision manager to further push the change request to deployment.

As a revision manager, you need to push the changes made by the business operations team to the pipeline for deployment. Access the change request by clicking **Revision Management** in the navigation pane.

You can see the change request in this revision. Now, push the request to deployment.

Observe that the revision changes are first queued and then get merged to the deployment pipeline. Note that this process might take a while to load.

This pipeline enables a seamless experience to merge changes to the development system of record, version the application, and generate a release artifact.

Deployment Manager then deploys the artifacts to the production environment.

In the production environment, you can select a subset of users to test the modifications in a production environment, and then roll out the changes to all users when the test results are acceptable. In this case, you activate the change request right away.

Complete the revision.

You have reached the end of this video. What did it show you?

- How to deploy changes made in the business operations environment to the production environment.

Creating treatments with GenAI

Introduction

Pega software suite, Pega Infinity™ '23, introduces new features that use the power of Pega GenAI™ to simplify your daily tasks in Pega 1:1 Operations Manager. One of the main generative AI features available in 1:1 Operations Manager is treatment creation.

Transcript

This video shows you how to create a treatment in 1:1 Operations Manager, using Pega GenAI features.

A treatment represents a message to be communicated to a customer. With Pega GenAI, designing persuasive web and e-mail treatments becomes easier and more efficient.

To enable the image generation functionality, you must configure your own generative AI image provider and set a provider key. Without this key, the application will not generate images.

U+Bank plans to announce a new, Travel Rewards card offer to its customers, through web and e-mail channels.

To begin, create a new change request of the **Create > Action** type, give it an appropriate name and description, and select the completion date.

Create Action
— ✕

Select the pre-defined Action type that most closely matches your use case. These types pre-populate certain attributes, including if it's an Always-on or One-time Action.

Change selection

💡

Sales
Promotional

Promote a new or upgraded product or service. Examples are marketing or promotional offers for a new credit card or phone upgrade.

OUTBOUND
INBOUND

Change request name *

Completion date *

Description

Unlock a world of possibilities with the Travel Card - the ultimate companion for your globetrotting adventures! Elevate your travel experience and turn every journey into a memorable adventure with exclusive perks and benefits.

Remaining: 772 characters

Are you making this change to fix a production defect? ?

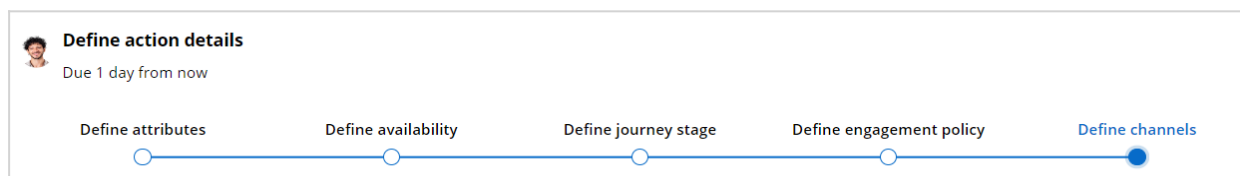
Yes
 No

Attach files

Attach

Cancel
Create

In the Plan stage of the change request completion process, you can define the attributes, availability, journey stage, and engagement policies. Once completed, you proceed to the **Define channels** stage.



In the **Define channels** stage, two treatment creation windows are open: e-mail and web. In this case, these treatment types are predefined because you selected the Sales action type at the beginning of the process.

The Pega GenAI feature is available in both web and e-mail treatment creation windows, after you click the purple **Polaris** icon.

When creating an email treatment, Pega GenAI assists in generating a message to a customer, based on the provided input details.

You then select one of seven Cialdini's Principles of Influence to specify the aim of the message.

Depending on the selected principle, the generated message will be authoritative, empathetic, or focused on the scarcity of a particular offer.

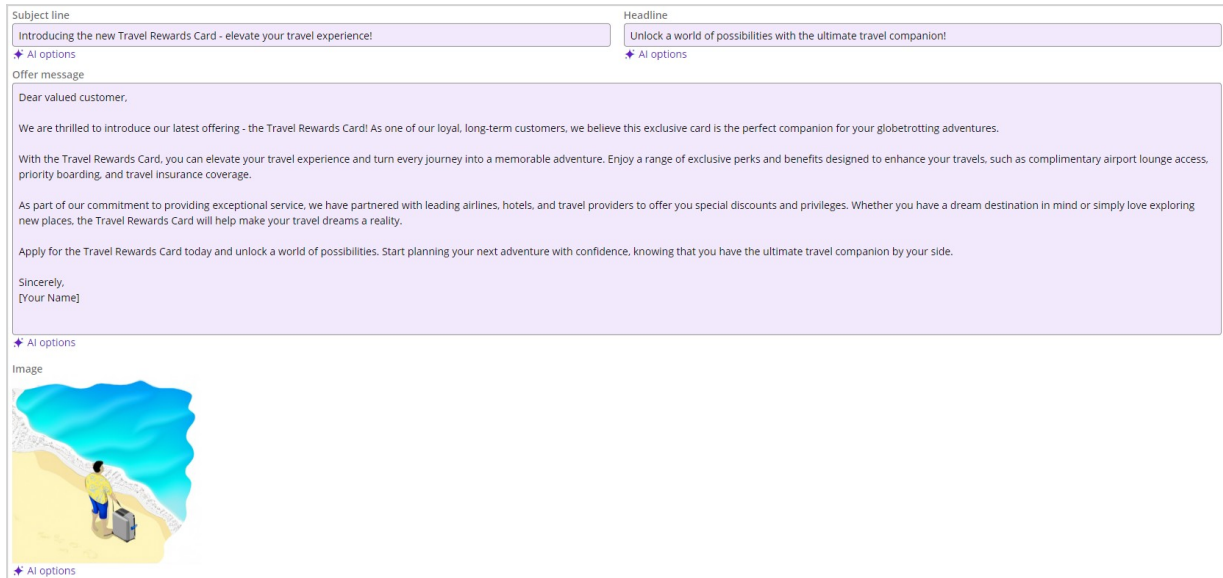
After selecting the principle, determine the tone of voice for the message. Finally, outline the specific items you wish to emphasize in the treatment.

The screenshot shows a configuration window titled "Treatment 1" with a subtitle "Cialdini's principle of social proof • Formal". Inside the window, there is a "Configure AI" section. Under "Principle", a dropdown menu is set to "Cialdini's principle of social proof". Under "Tone", a dropdown menu is set to "Formal". Below this, there are checkboxes for "Generate fields": "Headline", "Subject", "Body", and "Image", all of which are checked. A "Generate treatment" section contains a text box with the text: "U+Bank would like to present the new, exclusive, travel rewards card offer to their loyal, long-term customers." At the bottom of the window, there are three buttons: "Cancel", "Preview prompt", and "Generate".

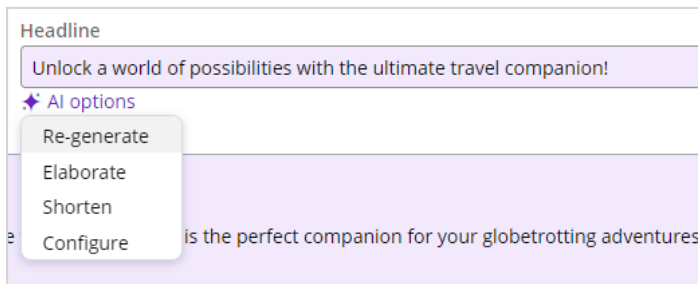
Before generating the e-mail, click on the Preview prompt button, to view the instructions and the details used by Pega GenAI to generate the final message.

The screenshot shows the Pega GenAI interface. On the left, there is a sidebar with a search bar and a list of items. The main area is titled "New, Travel card offer" and shows a "Details" section with fields for "Rank", "Status", "Request type", "Completion date", "Business issue", "Group", "Created", and "Updated". The "Details" section is expanded to show "Pulse" and "Artifacts". In the center, there is a "Channels" section with a "Treatment" configuration window. This window is titled "Cialdini's principle of social proof • Formal" and contains the same configuration as the previous screenshot. Below the configuration window, there is a "Subject line" field with the text "Provide a standard email subject line" and a "Headline" field with the text "Provide a headline to be show in email". At the bottom, there is an "Offer message" field.

After clicking Generate, Pega GenAI creates a formal e-mail within seconds. You can review the generated content and make updates if necessary,

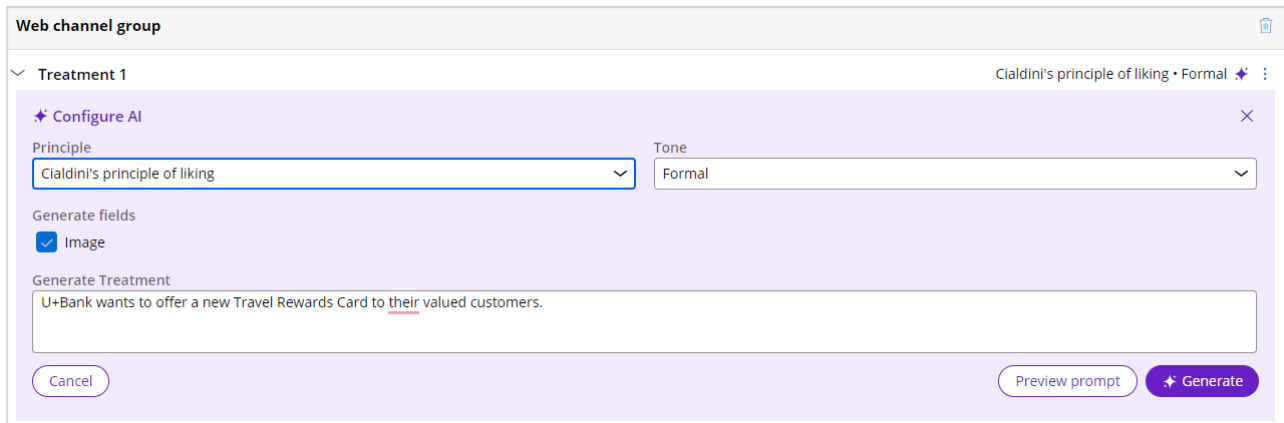


Each part of the message, such as the **Subject line**, the **Headline**, the **Offer message**, and the **Image**, can be re-generated or edited separately.



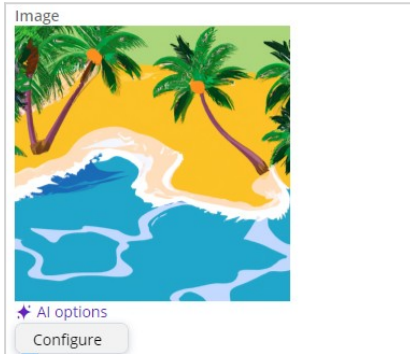
The other elements of the e-mail treatment, for example **Name**, **Availability** and **Business weight** must be set manually.

For the web treatment, which is the visual presentation of the offer on the website, Pega GenAI can generate an image that matches the offer, based on the given details.



You can select the **Principle** and the **Tone**, write the description in natural language and click Generate to see the AI-generated image.

If the image does not meet the business requirements, you can regenerate it by clicking the **Polaris** icon and then clicking **Configure**. In the **Configure AI** window, you can write a new description and generate a new image.



Set the remaining treatment elements manually and **Finish** the task.

After you successfully create the change request as a Business User, log in as a Team Lead to prioritize the change request.

The next step is to complete the action build task. You log in as an NBA Specialist and check for tasks related to the New, Travel card offer.



Assign the tasks to yourself and complete them one by one.

Finalize the first task by submitting the action details. Then complete the second task without adding a journey stage or engagement policies.

In the Design web treatment task, review the details of the treatment. As an NBA Specialist, you can also regenerate the image using the Pega GenAI feature if it doesn't meet business requirements. AI-generated images can be used as an inspiration. However, the final decision about the content rests with the operations team.

In the Design e-mail treatment task, review the details of the treatment and if needed, regenerate them using Pega GenAI. As an NBA Specialist, you must also complete the Destination field. Select Email output template from the dropdown list and then complete the task.

Business user and NBA Specialist, both have the right to use Pega GenAI capabilities and the Team Lead can review the contents created by Pega GenAI and approve it. It ensures that a chain of people checks the content generated by the AI.

Following the submission of updates by the NBA Specialist, log in again as a Team Lead to approve the updates and promote them for deployment.

The final step in the process is for the Operations Manager to deploy the changes to production.

You have reached the end of this video. You have learned:

- How to streamline the process of creating treatments using Pega GenAI features.
- The benefits of using Pega GenAI features.

Update existing actions

Description

Pega 1:1 Operations Manager currently supports three types of change requests. Choosing the right change request category initiates the appropriate change management process flow. In this module, you learn about the change management process to update existing actions.

Learning objectives

After completing this module, you should be able to:

- Update existing actions using 1:1 Operations Manager.

Update existing actions

Introduction

In 1:1 Operations Manager, there are three types of change requests currently supported. Selecting the right change request category initiates the appropriate change management process flow. In this topic, you will learn the change management process of updating existing actions.

Transcript

This video explains the concept of updating existing actions by using Pega 1:1 Operations Manager™.

In 1:1 Operations Manager, there are three types of change requests that are currently supported. Selecting the correct change request category initiates the proper change management process flow.

In this topic, you learn how to update existing actions in 1:1 Operations Manager.

You can request one or more changes at the action level when updating an existing action. Depending on the type of update required, you can select the task from the available list.

For example, if the two credit card actions you want to update are the Rewards Card and Rewards Plus Card, you can modify only the action details for the Rewards card, and you can modify the engagement policy and action details for the Rewards Plus card action.

Any change that a business user initiates goes through the change management process flow that majorly involves four roles and five different steps.

1. The business user starts the process by defining the change request.
2. Then, the team lead prioritizes the change requests that are received.
3. The NBA Specialist then configures the change request and submits it back to the team lead.
4. The team lead promotes the change request for deployment.
5. And finally, the revision manager deploys the revision.

As a business user, you define the change request for the NBA Specialist to understand the changes to make.

In this step, you can choose the actions that you want to update and select the tasks that you want to modify to those actions.

Notice that multiple actions can be modified as part of one change request, and each action can have its own set of changes. For example, while updating Action 1, you can change only the Engagement policy, while for Action 2 you can change the Web treatment and Engagement policy.

In the change management process flow, the second step is to prioritize the change request created by the business user. In this stage, the team lead can also assign a peer reviewer to review the engagement policies created. However, this task is optional.

Once the change request is assigned a rank, a team of NBA Specialists gets to work on that change request. As an NBA Specialist, you can see all the tasks related to all the actions of the change request once you start working on it. You can start working on them by assigning any of the tasks to yourself.

In 1:1 Operations Manager, while updating a group of actions, modifying the configuration of each action is an independent process. As an NBA Specialist, once you assign a task related to one action to yourself, you can see only the tasks related to that action. You need to verify the artifacts and then complete configuring all the tasks related to that action before you work on the next action of that change request.

Finally, once you complete all the tasks of that change request, you validate all the artifacts configured and submit the change request back to the team lead.

In the process of change management flow, the next step as a team lead is to verify the change request configured by the NBA Specialist and approve it.

Once approved, the changes are made available or submitted to the revision manager, who can deploy the changes to production and complete the revision.

You have reached the end of this video. What did it show you?

- How to update existing actions using 1:1 Operations Manager.

Updating existing actions in bulk

Introduction

One of the change requests currently supported in 1:1 Operations Manager is the **Action update** change request. Initiating this request triggers a process for updating the properties of single or multiple actions. Within the **Action update** change request, the **Bulk Edit** option is available, streamlining the creation of change requests aimed at updating several actions with one specific modification.

Transcript

In 1:1 Operations Manager, there are three types of change requests that are currently supported:

- Create Action
- Create Action update
- Create Other

In addition to single action updates, the **Action update** change request type enables you to perform bulk updates on always-on actions. A bulk edit feature is a valuable option when you need to implement a specific change across multiple actions. For instance, if a business requirement entails modifying the availability of 100 actions, instead of creating change requests and modifying settings separately for each action, you can make a unified modification. The bulk update simplifies the process of creating change requests for multiple actions.

Consider the following scenario: U + Bank, a retail bank, wants to limit the availability of two actions: the Rewards Card and Rewards Plus card, and make them available only for the next two weeks.

To meet the business requirement, you need to design a bulk edit change request. As a Business user, create the **Action update** change request and select the credit card actions

that require editing.

Update existing Actions

Name *
Update the availability

Completion date *
8/17/2023

Are you making this change to fix a production defect? ?
 Yes No

Business issue ?
Grow

Group ?
Credit cards

Description
Update the availability of the Rewards and Rewards Plus cards: Availabile within a defined period of time from 10.08.2023 to 24.08.2023

Remaining: 865 characters

Attach files
Filename

In the **Detail Updates for Selected Actions** stage, the **Bulk edit** option becomes accessible when you choose two or more actions from the list. In this case, select the Rewards Card and Rewards Plus Card, and proceed with the bulk update. The bulk edit option supports several predefined tasks. In this case, modify the action availability.

Detail updates for selected Actions

Due 6 days from now •

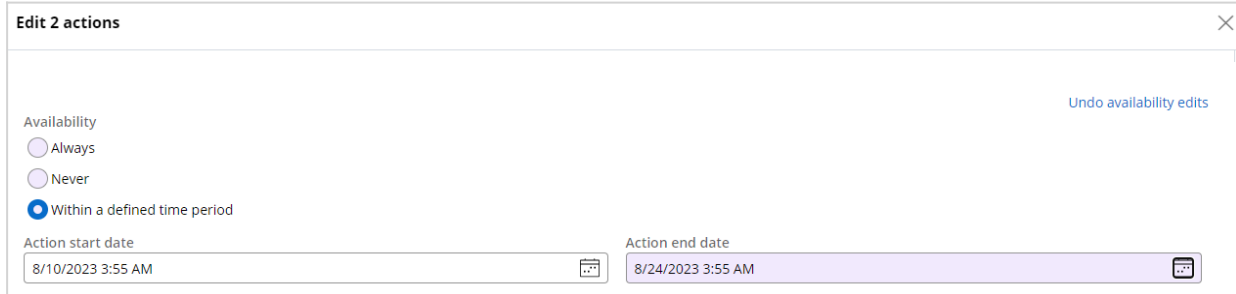
Select actions to update Detail updates for selected actions

Search... 2 results

<input checked="" type="checkbox"/>	Action	Con	Issue	Group	Treatments
<input checked="" type="checkbox"/>	Rewards card	Cust	Grow	Creditca...	1
<input checked="" type="checkbox"/>	Rewards Plus card	Cust	Grow	Creditca...	1

- Edit attributes
- Edit availability**
- Edit bundle attributes
- Edit treatments
- Edit other attributes

Within the **Edit 2 actions** window, you can set the actions as available at all times (always visible to customers), unavailable (never active), or within a defined time period (visible temporarily). Adjust the availability timing according to the requirement, and submit the change request.



The screenshot shows a window titled "Edit 2 actions" with a close button (X) in the top right corner. On the right side, there is a link labeled "Undo availability edits". Under the heading "Availability", there are three radio button options: "Always", "Never", and "Within a defined time period". The "Within a defined time period" option is selected. Below these options are two date and time pickers. The first is labeled "Action start date" and shows "8/10/2023 3:55 AM". The second is labeled "Action end date" and shows "8/24/2023 3:55 AM". Both pickers have calendar icons to their right.

After you successfully create the change request as a Business User, log in as a Team Lead to prioritize the change request.

The next step is to complete the action build task. Log in as an NBA Specialist and check if you can see two tasks, one for each action. Despite having created a single bulk edit change request for multiple actions, as an NBA Specialist, you should complete tasks individually.

Assign the first task to yourself and finalize the task by reviewing the availability setting. Given that the Business User has updated availability during the planning stage, the build stage only requires verification and confirmation of the changes. Upon completing the build stage, review and submit the task. Repeat the same process for the second task and then submit the changes.

Following the submission of updates by the NBA Specialist, log in again as a Team Lead to approve the updates and promote them for deployment.

The final step in the process is to deploy the changes to production using Operations Manager.

Implementing a business change using Revision Management

Description

Business requirements change rapidly and typically have a different release cycle than the enterprise release cycle.

Change request types in the 1:1 Operations Manager portal support most business changes. However, for some low-volume business changes, such as changing decision strategies and editing decision data rules, you use the revision management capabilities in Pega Customer Decision Hub.

Learn how to make such low-volume business changes in the BOE environment and deploy them to production.

Learning objectives

- Initiate a change request for low-volume business change in the BOE.
- Implement the change request as an NBA Designer.
- Approve and deploy the changes to all environments using the business change pipeline.

Modifying a decision strategy

Business requirements change rapidly and typically have a different release cycle than the enterprise release cycle.

To implement any business change, you create a change request in 1:1 Operations Manager. Such changes include creating and modifying actions, treatments, group-level eligibilities and so on.

Every change request requires an associated revision. When you create a new change request, the system automatically associates the change with an existing revision. If an existing revision is not available, a new revision is created within revision management. All the changes made to the change request reflects in the revision management.

The change request types in the 1:1 Operations Manager portal support most business changes. However, for some low-volume business changes, such as changing decision strategies and editing decision data rules, use the revision management capability to create a new revision and implement the business request. You can implement such changes using revision management in the Customer Decision Hub portal.

As a best practice, use 1:1 Operations Manager portal to initiate the change requests from the existing change management flows to avoid creating conflicts between the two portals.

Transcript

This demo shows you how to create and implement a change request using Revision Management.

U+ Bank uses Pega Customer Decision Hub™ and cross-selling on the web by showing various credit card offers to its customers.

The bank wants to modify the credit score decision strategy that segments customers into two categories based on the customer credit score: **Suitable** and **Not Suitable**.

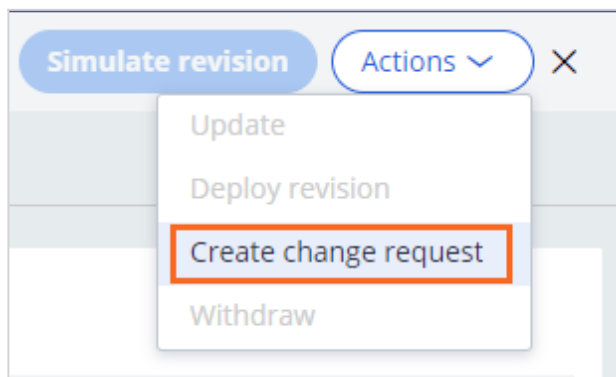
Such a low-volume business change cannot be achieved by the change request types available in 1:1 Operations Manager. So, you must initiate and implement

such change requests using the revision management capabilities of Pega Customer Decision Hub. To carry out the business change, as a team lead, create a revision manually in the Revision Management section of 1:1 Operations Manager.



Note that only one revision can be active at a time, and you cannot create a revision when there is an existing revision in an open state.

Once you create the revision, you switch to Customer Decision Hub to access the revision created in 1:1 Operations Manager. Now, within the same revision, you create a new change request using the Revision Management feature in the Customer Decision Hub portal.



In this case, you create a new change request to edit the engagement strategy and assign the change request to NBA designer.

When you create a change request in 1:1 Operations Manager, the required rules are automatically added to the revision and are available for the NBA designer to edit based on the objective of the task.

When the change request is created in the Customer Decision Hub portal, you must manually add the rules that are necessary to complete the new change request to

the revision. This step is to ensure ensures that the rules are available for the NBA designer to implement the change.

For this change request, add the **Credit Score** rule of the Strategy type and submit the change request.

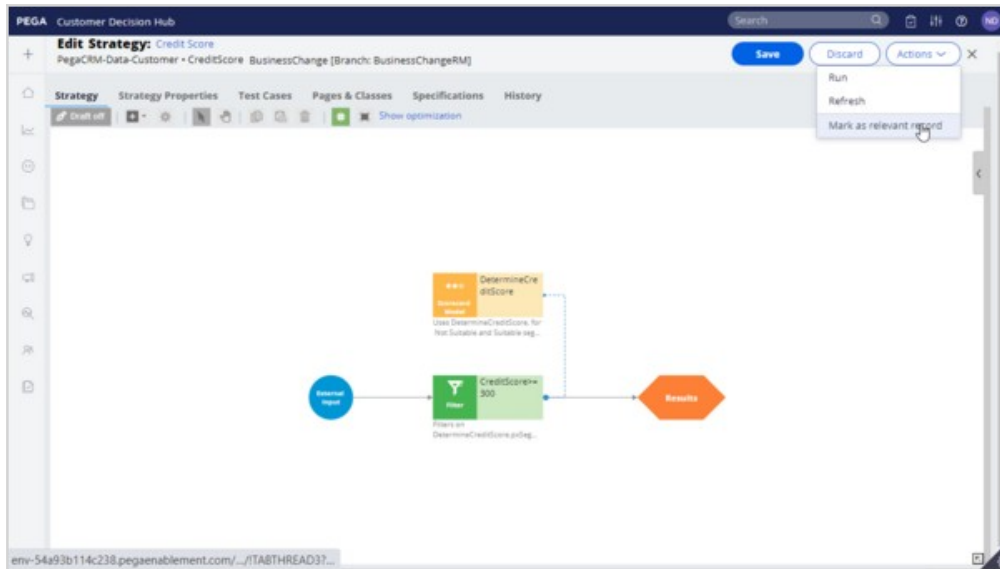
Once the change request is submitted, as an NBA designer, using revision management, you can access the change request assigned to you and the rules associated with it.

In the Revision Management landing page, the **My Work** section displays all the change requests that are assigned to the NBA designer. All the change requests created in 1:1 Operations Manager and in Customer Decision Hub are visible in the list of change requests. The **Scope of changes** section displays all the rules that are available to complete the change.

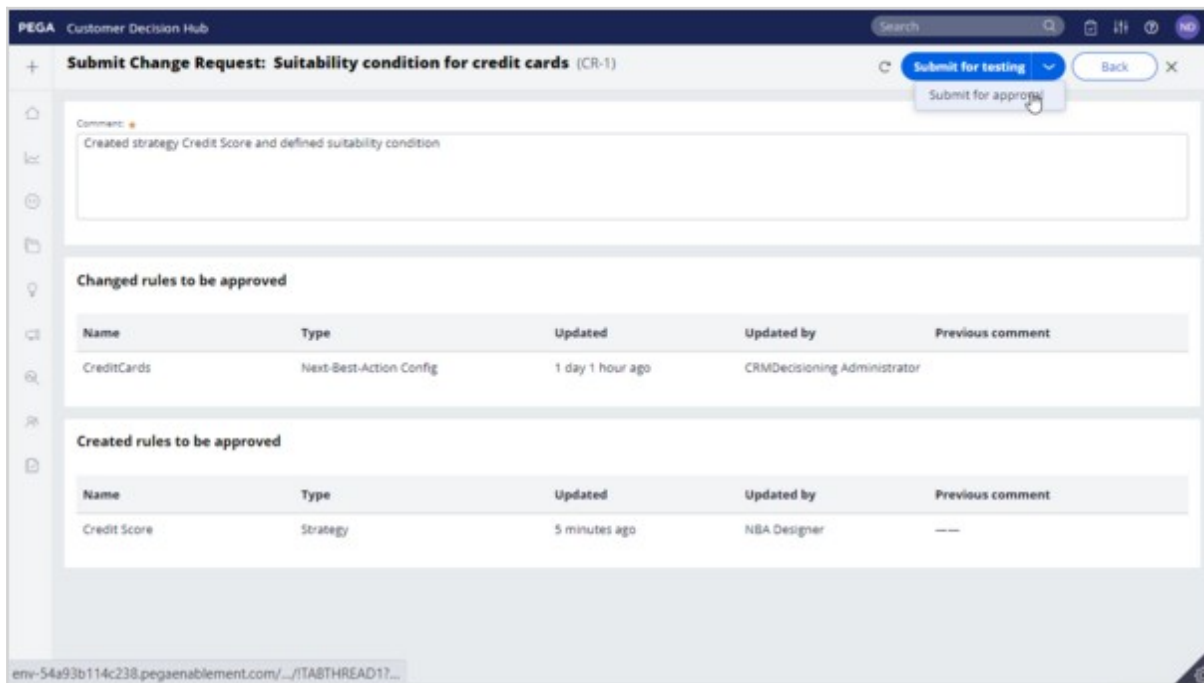
An NBA designer can send the change request back to the team lead if additional rules are required to complete the change request. The NBA designer can also create new rules, and those new rules are automatically added to the change request.

The NBA designer then modifies the engagement strategy that uses a scorecard to determine whether a customer is suitable to receive the credit card.

To make the decision strategy available in the Next-Best-Action Designer, the NBA designer marks the strategy as a relevant record, if not marked already and then submits the change request for approval to the team lead.



In a real-life implementation, at this stage, the NBA designer typically submits the change request for testing to ensure that the business requirement is properly implemented. For this demo, the change request is submitted directly to the team lead for approval.



On the Revision Management landing page, the team lead can access the change request, and then approve it. The status of the change request changes to *Resolved-Completed*.

Once the change request is complete, the changes are available to the revision manager, who can deploy the changes to production.

You have reached the end of the video.

What did it show you?

- How to create a new revision in 1:1 Operations Manager.
- How to create a change request by using revision management in the Customer Decision Hub portal.
- How a team lead can include one or more rules for editing in a change request.
- How an NBA designer can implement a change request in revision management.

Editing a group-level engagement policy

Pega 1:1 Operations Manager currently supports three types of change requests. Selecting the correct change request category initiates the proper change management process flow and ensures that the appropriate rules are copied to the revision, to provide a smooth user experience.

Transcript

This demo shows you how to initiate a business change in 1:1 Operations Manager and implement it using Revision management in Customer Decision Hub portal.

U+ Bank uses Pega Customer Decision Hub™ and currently cross-sells on the web by showing various credit card offers to its customers. The bank now wants to offer credit cards to only those customers with a credit score greater than or equal to 300. So, the bank wants to add a group-level suitability condition in the engagement policy to implement the business change.

Such a change is classified as a low-volume business change and is treated as an “**Other**” type of change request. The objective of this change request is to edit the group-level engagement policy condition. This change request type has similar stages to creating or updating action change requests.

In 1:1 Operations Manager, as a business user, you start implementing the business requirement by creating a change request to capture the business context of the required change. The change request category indicates the type of change, in this case **Other**.

The second step in the change management process flow is to prioritize the change request created by the business user. In this stage, as the team lead, you access the change request created by the business user from the **Change Requests** landing page and assign a rank to the change request.

Then you add the tasks required to implement the business change.

Every task has an objective selected by the team lead from a predefined list of objectives. Selecting the right objective is important because depending on the objective selected, the right rules are automatically made available to the NBA

Designer to make changes during the build stage. This means that in the background, these rules are automatically added to the revision.

In this case, since the business use case is to offer the actions only to those customers with a credit score greater than or equal to 300, you select the objective **Manage group-level Engagement policies** and then assign it to the NBA designer.

Once the change request is submitted, as an NBA designer you log in to the Customer Decision Hub portal to access the change requests assigned by the team lead.

When you open the change request, in the Scope of changes section, the rule required to edit the engagement policy and add the suitability condition is available.

To add the suitability condition, access Next Best Action Designer and then open the engagement policy for the specific business issue and group. In the **Suitability** section, enter the right engagement policy condition, in this case, *Credit Score higher or equal to 300*, and then save the engagement policy. Then submit the change request.

Once the changes are complete, the change request moves to the test phase and is routed to the NBA specialist for validation. The NBA specialist validates all the artifacts configured and then submits the change request to the team lead.

The team lead then reviews the artifacts and approves them to complete the change request and promotes it to the revision manager for deployment. The status of the change request changes to **RESOLVED-COMPLETED**.

Once the change requests are complete, the changes are made available to the revision manager, who can deploy the changes to production.

You have reached the end of the demo.

What did it show you?

- How to create a change request of **Other** type.
- How to add tasks to a change request and assign it to an NBA Designer.
- How an NBA designer implements a change request in the Customer Decision Hub portal.
- How a team lead can approve and promote the changes to deployment.

Implementing a fast release revision

Description

Learn how to efficiently implement and deploy critical and emergency business changes by using the CDH fast release functionality.

Learning objectives

- Explain how the fast release capability of Pega 1:1 Operations Manager works.
- Differentiate the change management process of a fast-track change request from a standard change request.

Fast-release revisions

You implement business changes in the business operations environment (BOE) by using revisions. Each revision consists of one or more change requests. After individually implementing these change requests, when they are ready for deployment, the Revision Manager deploys all the change requests together as a release. This process is a standard release.

However, you might need to implement some unplanned business changes that are of high priority. These unplanned changes are emergency changes to implement and deploy immediately. Such change requests are called fast-track change requests.

Fast-track change requests

Fast-track change requests effectively handle urgent business demands. A fast-track change request offers a means to efficiently implement critical changes without interfering with current revisions. Revision Management in Pega 1:1 Operations Manager addresses the fast-track changes by using fast release revisions.

Examples of fast-track change requests include:

1. Fixing a typo in the action that was deployed to production (updating an existing action).
2. Deactivating an action from production (updating an existing action).
3. Communicating with customers in an emergency (creating a new action).

With fast-release revisions, the business content team can implement and deploy critical business changes without having to wait for other change requests to be implemented and ready for deployment.

To create a fast-track change request, when you create a request, you mark the change request as a production fix. By completing this action, you add the change request to the fast-release revision, with the release type **CDH fast release**.

Update existing Actions — ✕

Name *

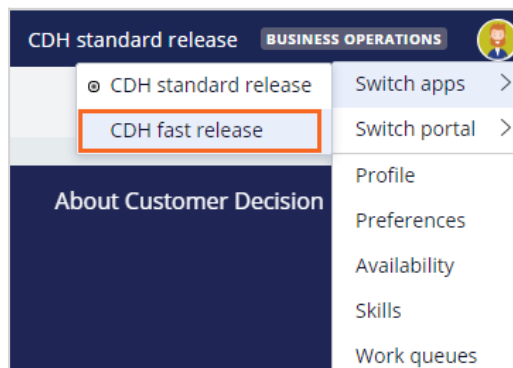
Completion date *

Are you making this change to fix a production defect? ?
 Yes No

Business issue ?

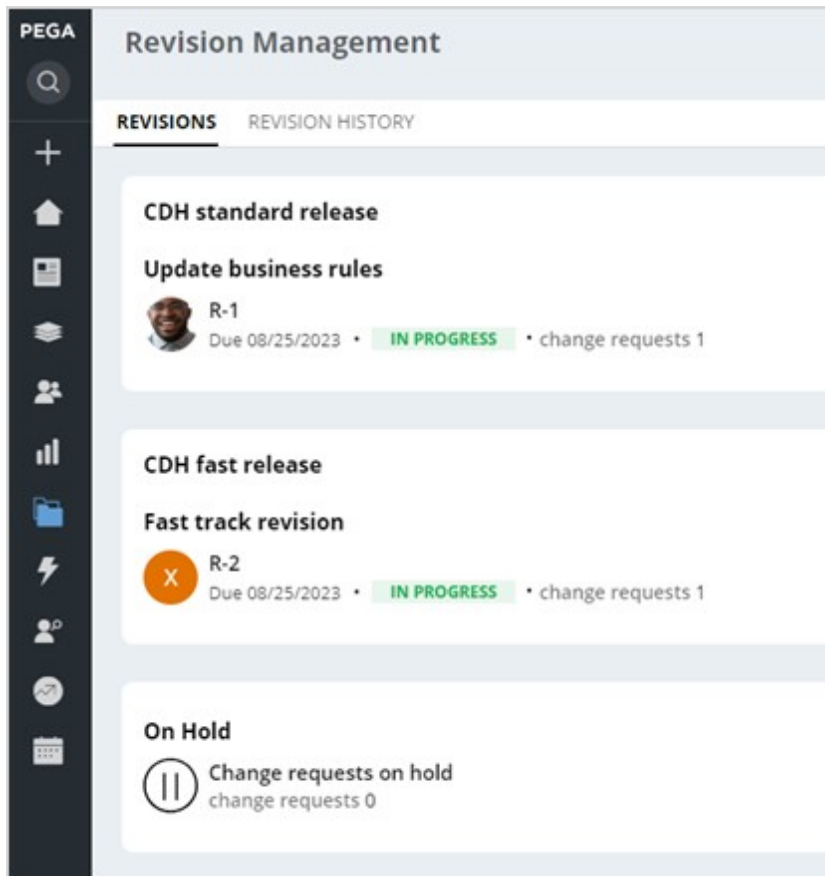
Group ?

The implementation of the fast-track change is similar to a standard business change. During the deployment process, the revision manager switches to the CDH fast release application to deploy the fast-release revision through the **fast release pipeline**.



Like a standard revision, a fast-release revision can also have multiple change requests. You can track the fast-track change requests in the **CDH fast release** section of Revision Management in Pega 1:1 Operations Manager.

The **CDH standard release** section displays the existing standard revision. You can access the list of change requests that are associated with a revision. The **On hold** section lists the change requests that are put on hold.



Note that there can be only one revision for a release, that is, one standard revision in a CDH standard release and one fast-release revision in a CDH fast release.

In summary, you implement and deploy business changes in the BOE in two types of revisions:

- Standard release revisions for regular business change requests
- Fast release revisions for emergency and high-priority change requests

Implementing outbound communication in the BOE

Description

Discover how to implement outbound customer engagement in a Business Operations Environment (BOE) and deploy necessary changes to production. Explore how you can create essential artifacts for outbound communication by using Pega 1:1 Operations Manager.

Learning objectives

- Create and deploy one-time Actions in a BOE.
- Create and deploy artifacts required for implementing outbound communication.

One-time actions

Effective communication is critical for a business, especially when it comes to delivering timely and urgent messages to customers. In exceptional circumstances, businesses may need to contact all, or a subset of their customers, outside of their regularly scheduled updates. For example, if U+ Bank temporarily closes one of its branches due to maintenance activity, it should send out an email communication to all customers, or those who are affected.

In such scenarios, you can send out the communications using one-time actions.

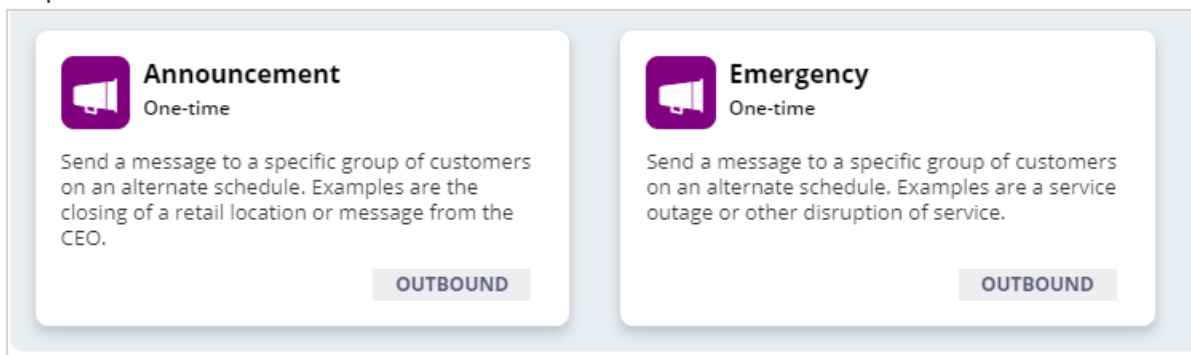
A one-time action is a batch communication that all or a subset of customers receive at a specific date and time, that is not coordinated with the always-on outbound schedules. These actions are created in a Business Operations environment to deliver one-time urgent or time-sensitive communications that do not repeat automatically.

When should you use One-time actions?

One-time actions are sent when the communication:

1. Must be sent to all or a group of customers.
2. Must be sent at once to all the customers and should not be repeated.
3. Communication is to be sent outside the regular outbound schedules.

To create a one-time action, you must choose the relevant tile while creating the change request.



Examples of one-time action change requests include:

- An emergency communication to all the customers from the CEO.
- Compliance notice notifying customers of changes in benefits and terms.

- Announcement of a branch closure.
- Communication of services outage due to maintenance activity in an area.

Unlike always-on actions, no engagement policies or constraints apply to one-time actions. One-time actions use a defined audience, and all the members of that audience receive the same message at the same time. These actions can be created and deployed to production through both standard release and fast-release revisions. However, as most one-time actions are used for immediate communication, it is recommended to use fast-release revision for deploying one-time actions.

The following figure shows which policies and constraints apply to different outbound interactions:

		Eligibility/ Applicability/ Suitability conditions	Arbitration	Outbound channel limits	Outbound action limits	Contact policies
Always-On actions	Promotional (Default actions)	✓	✓	✓	✓	✓
	Transactional actions	✓	✓	✗	✓	✓
	Mandatory actions	✓	✗	✗	✗	✓
One-time actions	One-time actions	✗	✗	✗	✗	✗

Creating a One-time action

One-time actions are created in 1:1 Operations Manager in a Business Operations environment, to deliver one-time, urgent, and time-sensitive communications that do not repeat automatically.

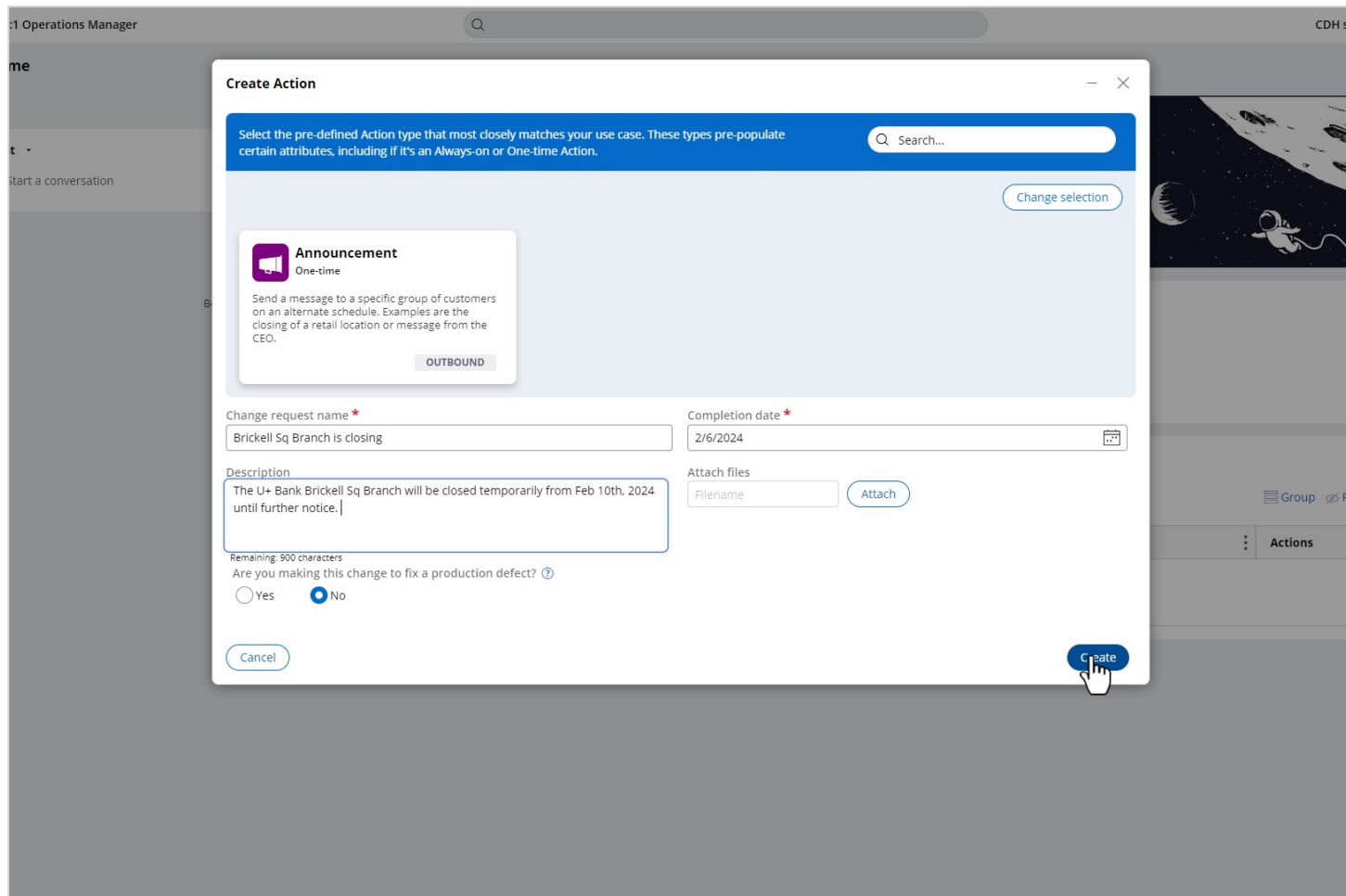
Transcript

This video demonstrates how to create a one-time action in 1:1 Operations Manager.

U+ Bank intends to temporarily close its operations at the Brickell Square branch because of planned maintenance work. The bank wants to communicate the message to all customers or those who are affected, through an email channel.

The process of creating a one-time action is the same as creating a new action in 1:1 Operations Manager.

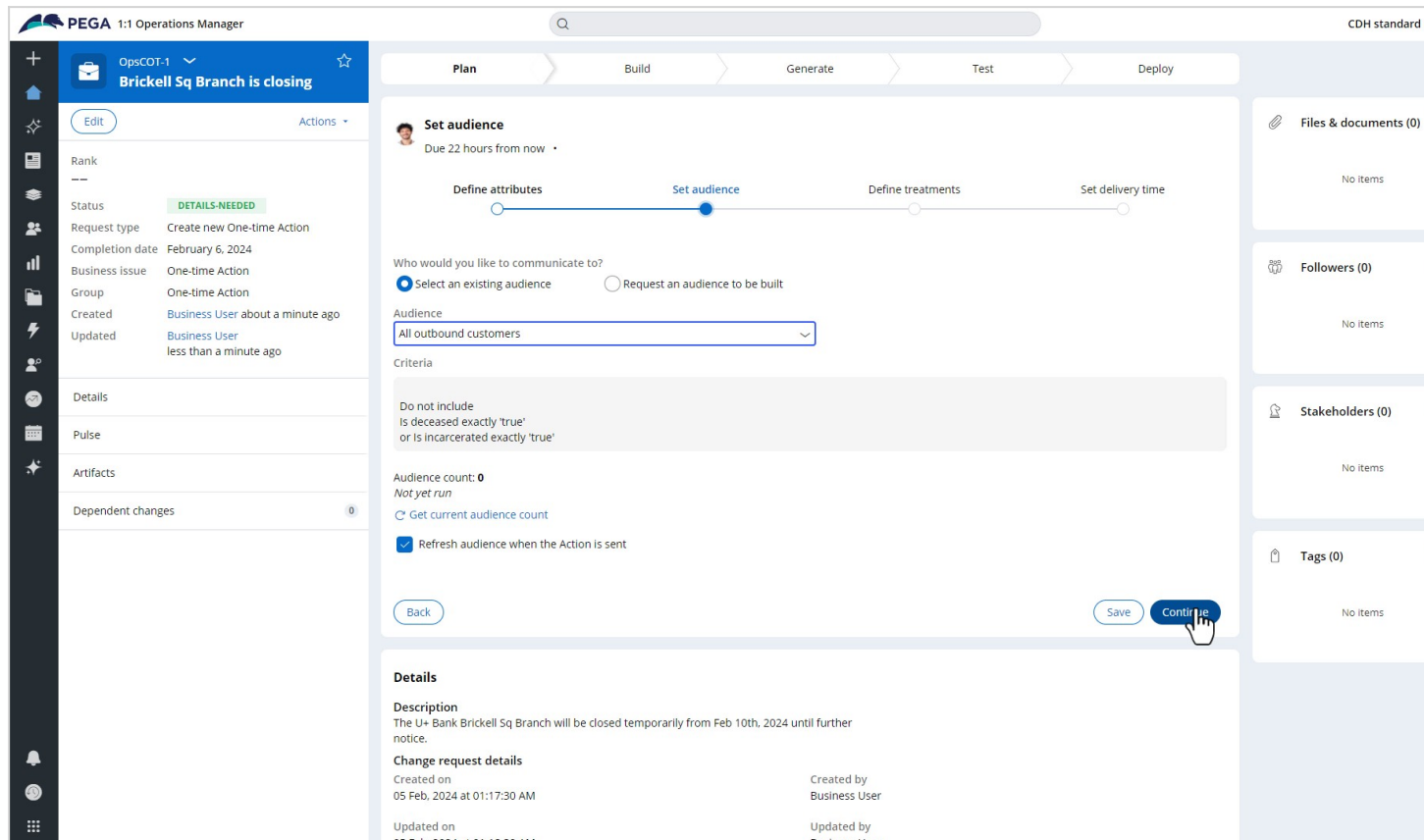
To begin, as a Business User, you create a new change request of the **Create > Action** type and select the appropriate tile. In this case, **Announcement**. Enter an appropriate name and description for the change request and specify the completion date.



In the **Plan** stage of the change request process, you define the attributes. In the **Define attributes** section, note that the Business issue and group default to One-time action. Enter a description and continue to set the audience.

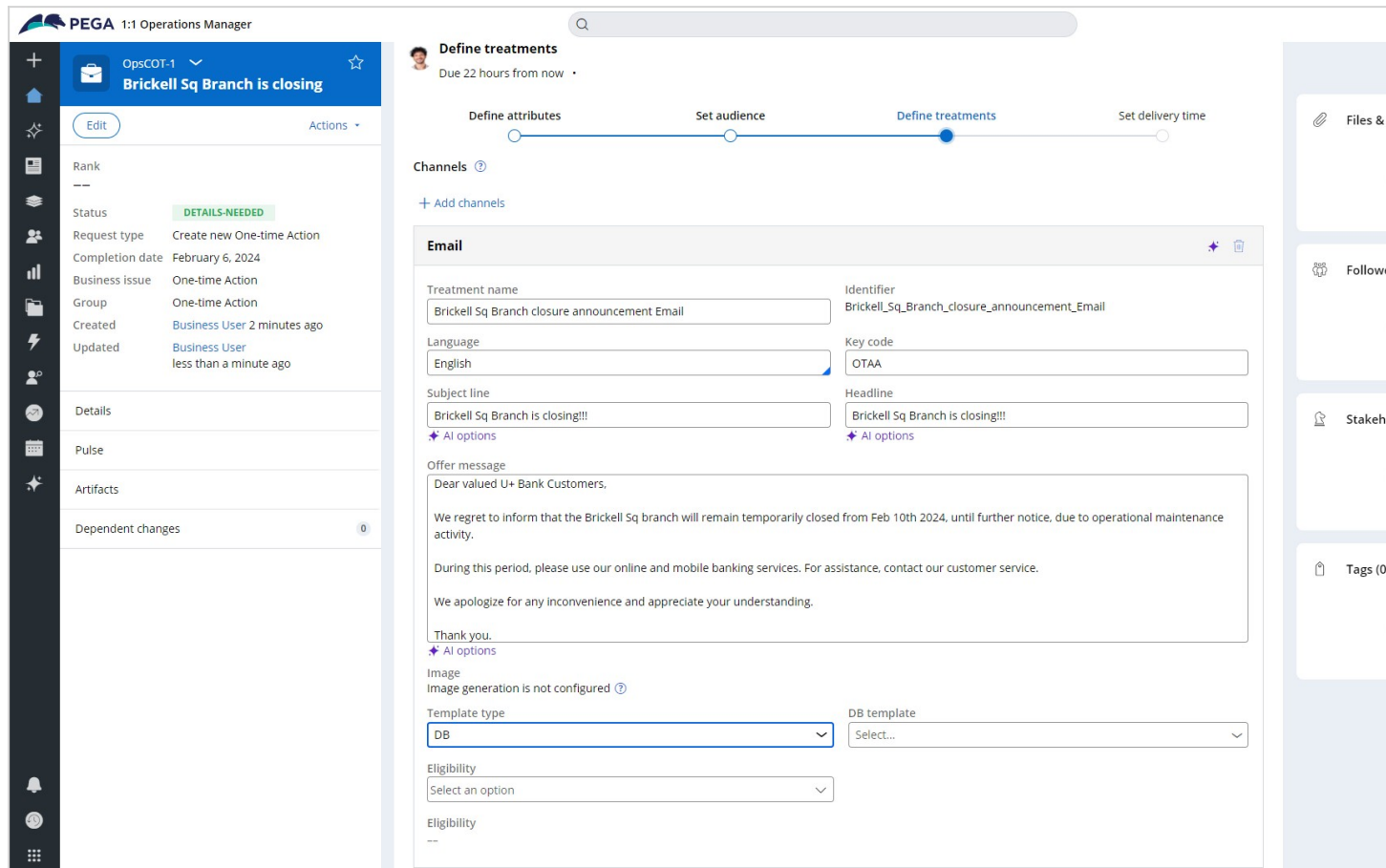
Audience refers to a list of potential target customers, which is implemented using the segment component. A segment is a reference to a database table containing a list of customer IDs that match specific criteria.

You can either select an existing audience or request a new audience to be built by NBA Designer. If you choose to request a new audience, the system creates a separate change request, called *Create an audience*. NBA Designer must complete this change request before the current change request moves to the Build stage. This ensures that the NBA Specialist can utilize the audience created by the NBA Designer while implementing the current change request.



Note that the existing audience is the list of the segments that are available in the overlay.

In the **Define treatments** task, enter the treatment name, email subject line, headline, and offer message, which forms the email body. You can use the Pega GenAI feature to generate the contents of the email. Additionally, you can specify the output template type as either DB or File.



Next, you set the delivery time and finish the task.

After you create the change request as a Business User, log in as a Team Lead to prioritize the change request.

The next step is to log in as an NBA Specialist and complete the action build tasks.

In the first step, finalize the action attributes. In the Set audience step, ensure that the right audience is selected. Then, define the treatments. Review the treatment details and make edits if necessary. As an NBA Specialist, you also need to configure the **Template type** field. Select the Email output template from the drop-down list.

PEGA 1:1 Operations Manager

OpsCOT-1
Brickell Sq Branch is closing

Plan Build Generate Test Deploy

Set delivery time
Due 22 hours from now

Define attributes Set audience Define treatments Set delivery time

When do you want this delivered? *

2/7/2024 1:31 AM

Email

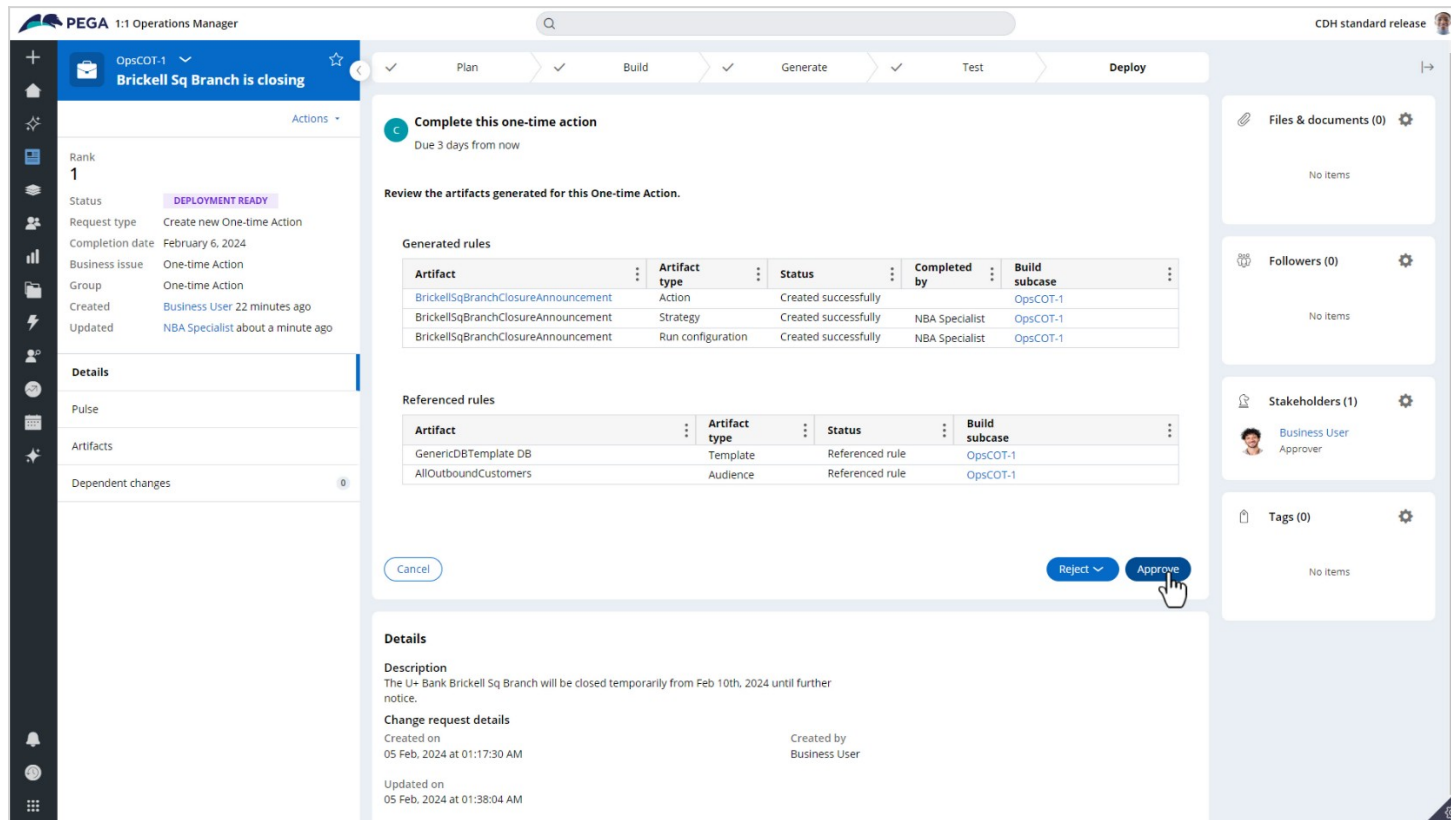
Eligibility	--	Identifier	Brickell_Sq_Branch_closure_announcement_Email
Treatment	Brickell Sq Branch closure announcement Email	Key code	OTAA
Language	English	DB template	--
Template type	DB	Offer message	Dear valued U+ Bank Customers.
Subject line	Brickell Sq Branch is closing!!!		We regret to inform that the Brickell Sq branch will remain temporarily closed from Feb 10th 2024, until further notice, due to operational maintenance activity.
Headline	Brickell Sq Branch is closing!!!		During this period, please use our online and mobile banking services. For assistance, contact our customer service.
			We apologize for any inconvenience and appreciate your understanding.
			Thank you.
Eligibility	--		

Back Save Finish

Review the time of delivery for the one-time action and finish the task. Then, complete the **Review all completed tasks in build stage** task.

As an NBA Specialist, access the **Test and validate artifacts** task to review and submit the artifacts generated for the One-time action.

Following the submission of updates by the NBA Specialist, log in again as a Team Lead to approve the updates and promote them for deployment.



The final step in the process is for the Revision Manager to deploy the changes to production.

After the one-time actions are deployed to production, the communication is sent out according to the delivery time defined during the creation of the action.

You can monitor the deployment and run status of the schedules in the Outbound Schedules landing page within 1:1 Operations Manager. The Outbound Schedules landing page provides an overview of all outbound schedules, including both always-on and one-time action schedules.

PEGA 1:1 Operations Manager

Outbound schedules

Primary schedule: All issues / All groups(Customer) NOT DEPLOYED - DRAFT

One-time action scheduled runs

Upcoming runs

Deployment status	Status	RunID	Action	Channels	Audience	Scheduled send date	Actual send date	Completed date	Initiated
DEPLOYED	SCHEDULED		Brickell Sq Branch closure announcement	Email	All outbound customers	2/7/24 1:31 AM			

Completed runs

Display past 30 days

Deployment status	Status	RunID	Action	Channels	Audience	Scheduled send date	Actual send date	Completed date	Initiated
No One-time Action runs									

You have reached the end of the demo. What did it show you?

1. How to create a one-time action in 1:1 Operations Manager.
2. How to define attributes for the one-time action, including setting the audience using segments.
3. How to monitor the deployment and schedule run status of one-time actions in 1:1 Operations Manager.