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.

Video

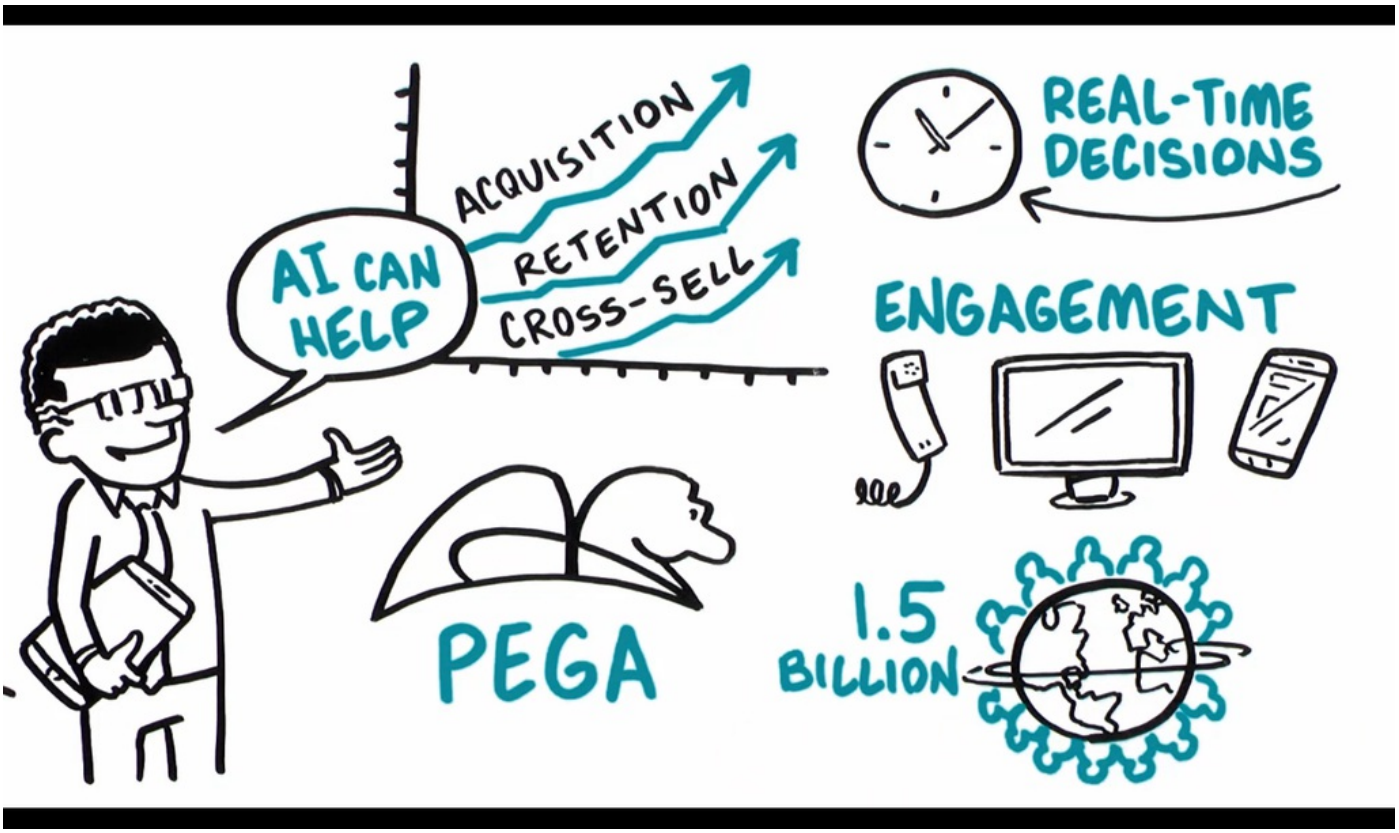
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

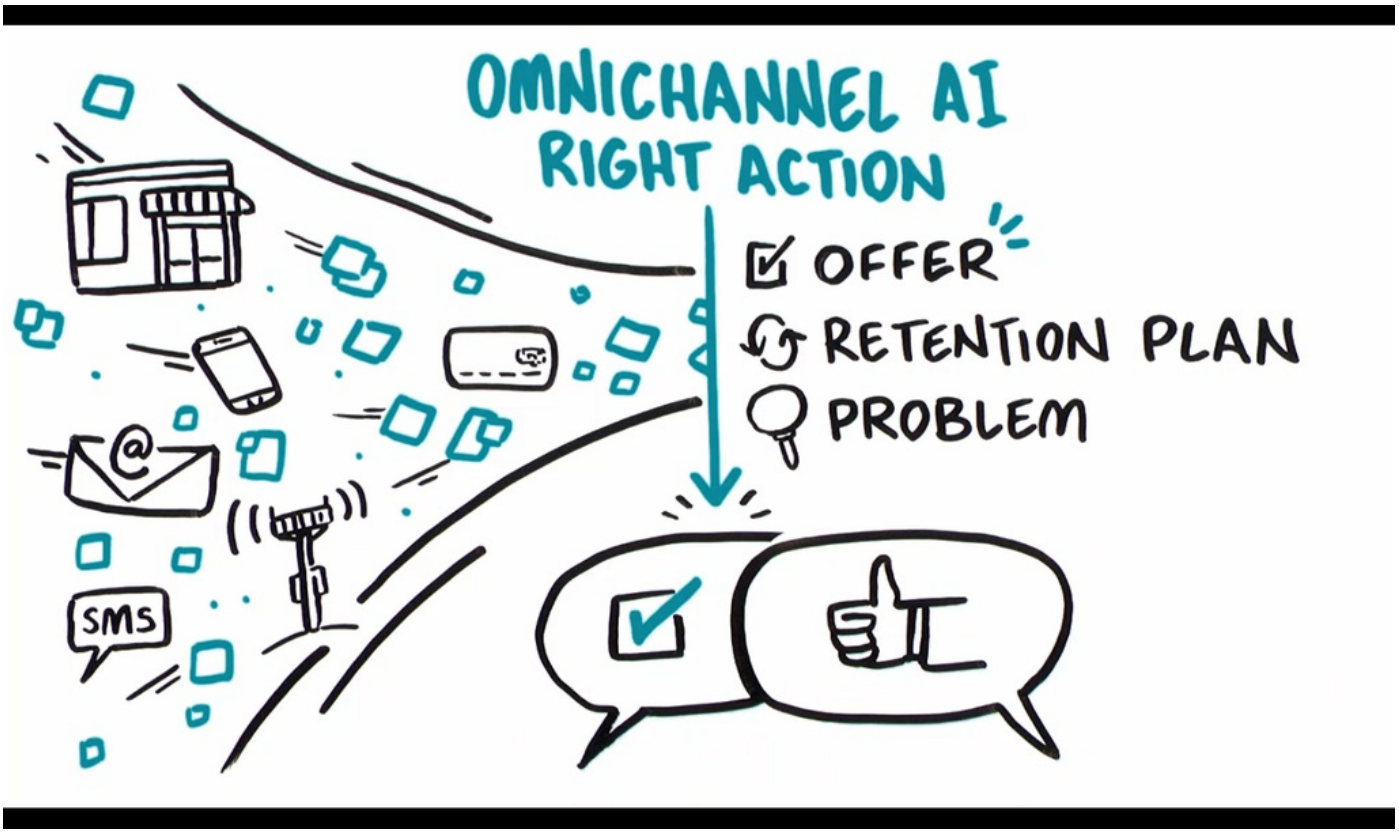
This is your customer. You want him to buy your products, use your services and have a great experience. And your competitors want the same thing. To compete, you have to take the right action at every customer touch, ensuring that each conversation delivers exactly the right message, offer and level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

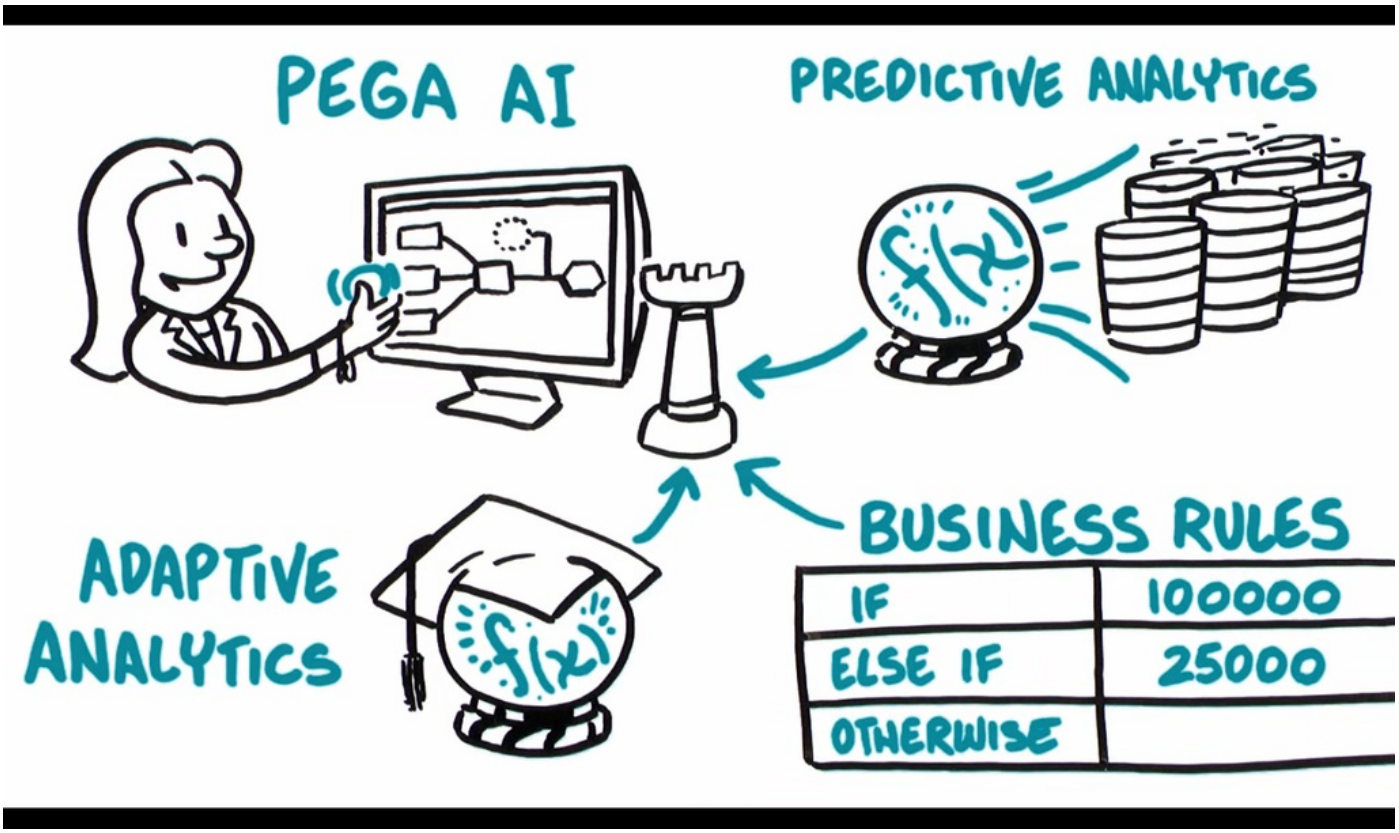


Pega’s omni-channel AI delivers the right action at every customer touch by crunching millions of data points in real-time. Make an offer, initiate a retention plan, predict a problem before it happens. Every decision generates the next-best-action for your customer, and your business.

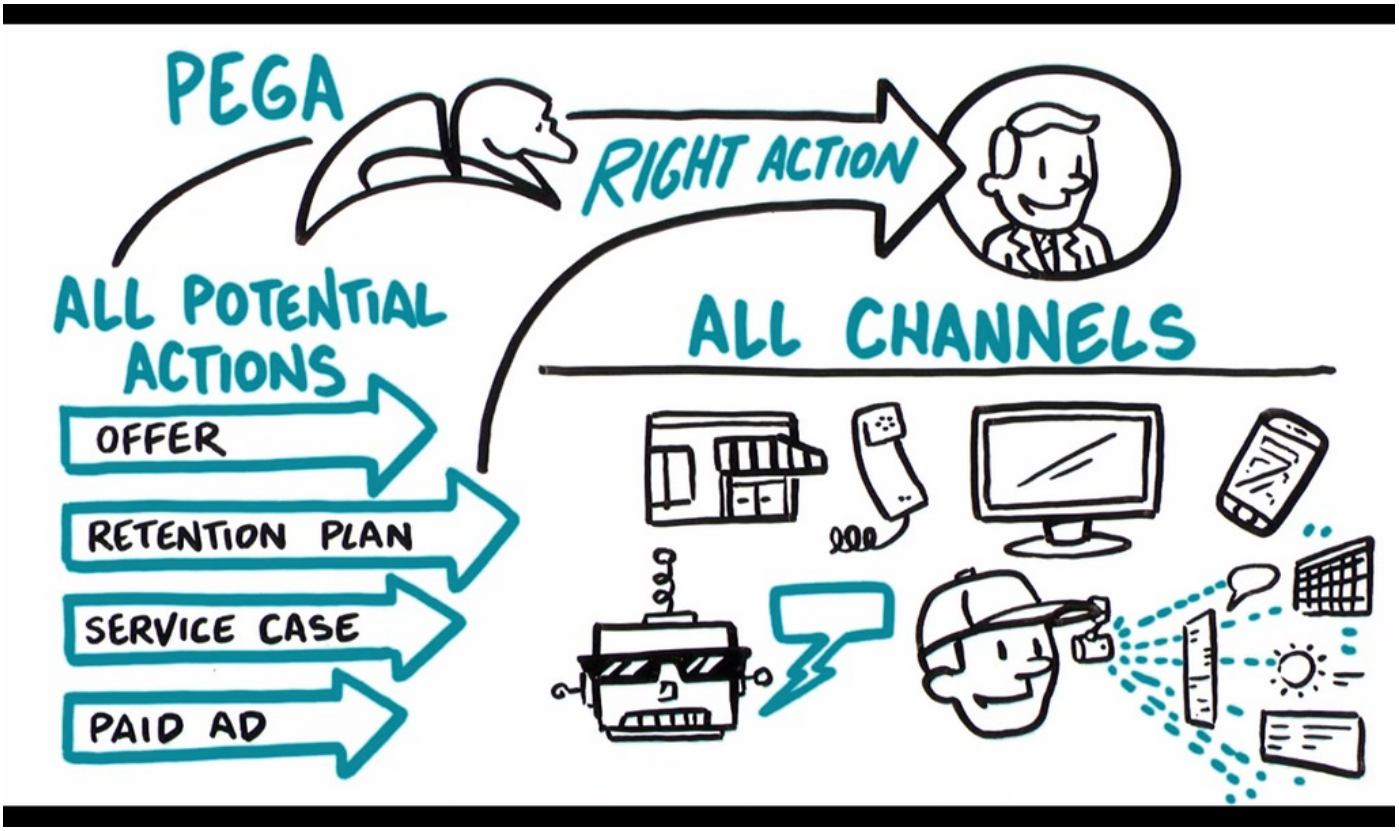


Pega’s AI is built for business people, not scientists or developers. They design visual decision strategies that combine predictive analytics, algorithms developed through mining large sets of data, adaptive analytics, machine-learning algorithms that improve with each interaction, and traditional business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

that allow users to prioritize and arbitrate between decisions.

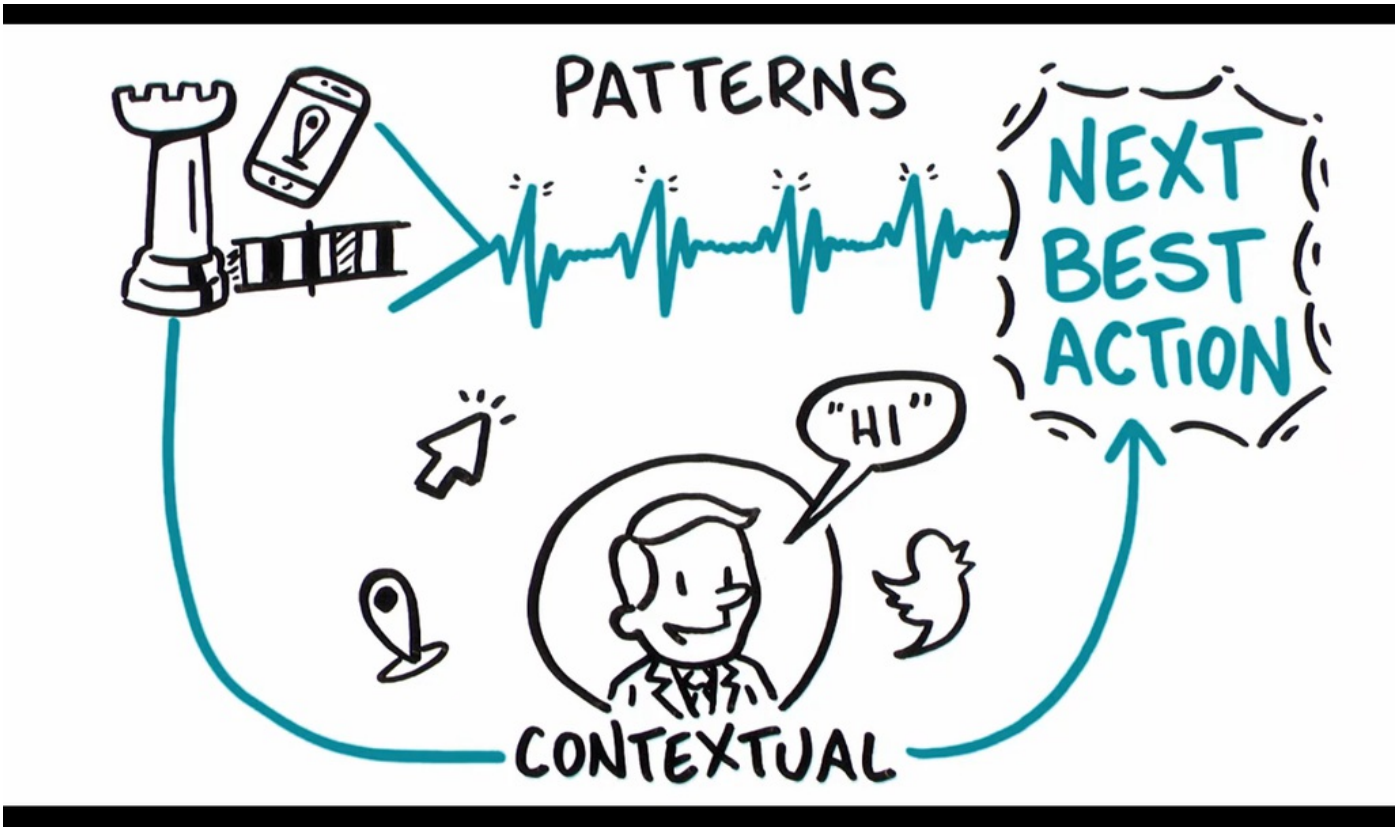


Pega uses the strategy to look across all the potential actions you may take with a customer, make an offer, initiate a retention plan, open a service case, place an ad, and ensure exactly the right action is taken at every interaction and it works across all channels to provide a consistent experience 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.

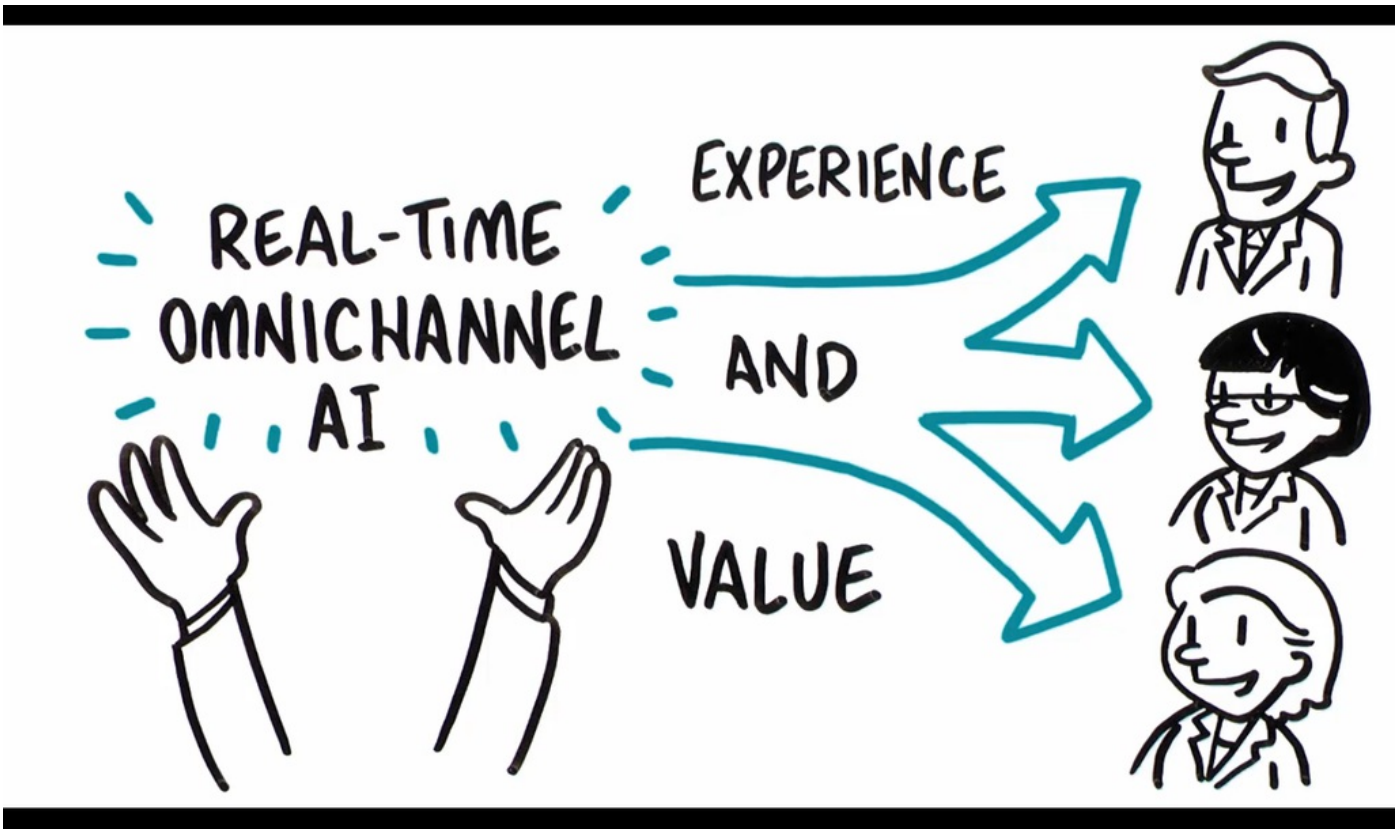


And Pega connects to streams like mobile locations or network events to detect patterns and drive the Next Best Action proactively. And strategies are completely contextual. Any change in the customer’s context —

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.



One-to-one Customer Engagement paradigm

Introduction

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 have to be able to perform the right action in the right channel at the right moment for each customer. We call this capability, “1-to-1 Customer Engagement”.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

In this video, learn about the 1-to-1 Customer Engagement paradigm and how the principles of Next-Best-Action are implemented using the Pega Customer Decision Hub™.

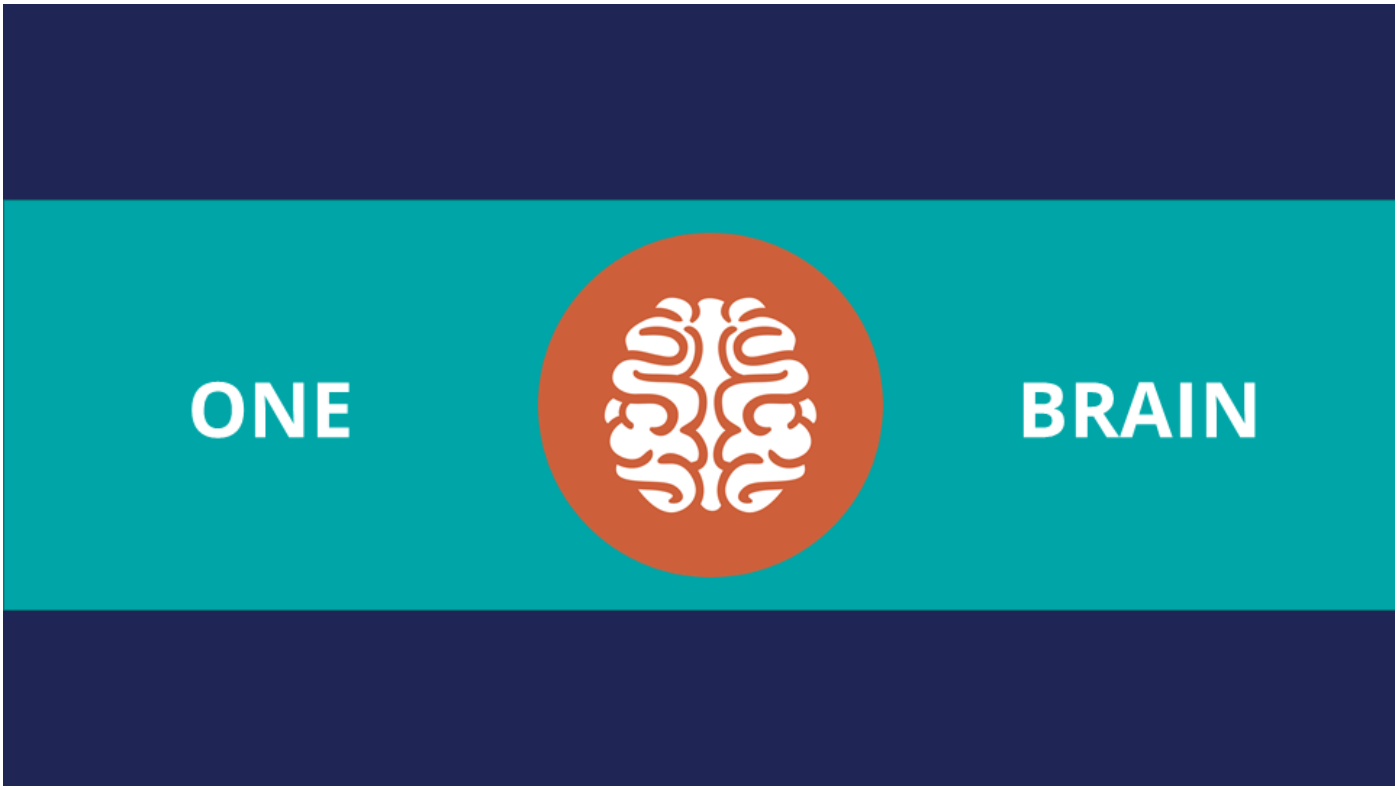
Customers are more empowered than ever before. As a result, they have very high expectations of the experiences they receive from their service providers. Their experiences must make sense within the context of their lives. This means they must be meaningful, consistent, and personalized across every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

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 have to be able to perform the right action in the right channel at the right moment for each customer.

We call this capability, “1-to-1 Customer Engagement”.

1-to-1 Customer Engagement enables companies to 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 1-to-1 Customer Engagement is an idea that’s simple to conceive, but very difficult to execute: one centralized brain.



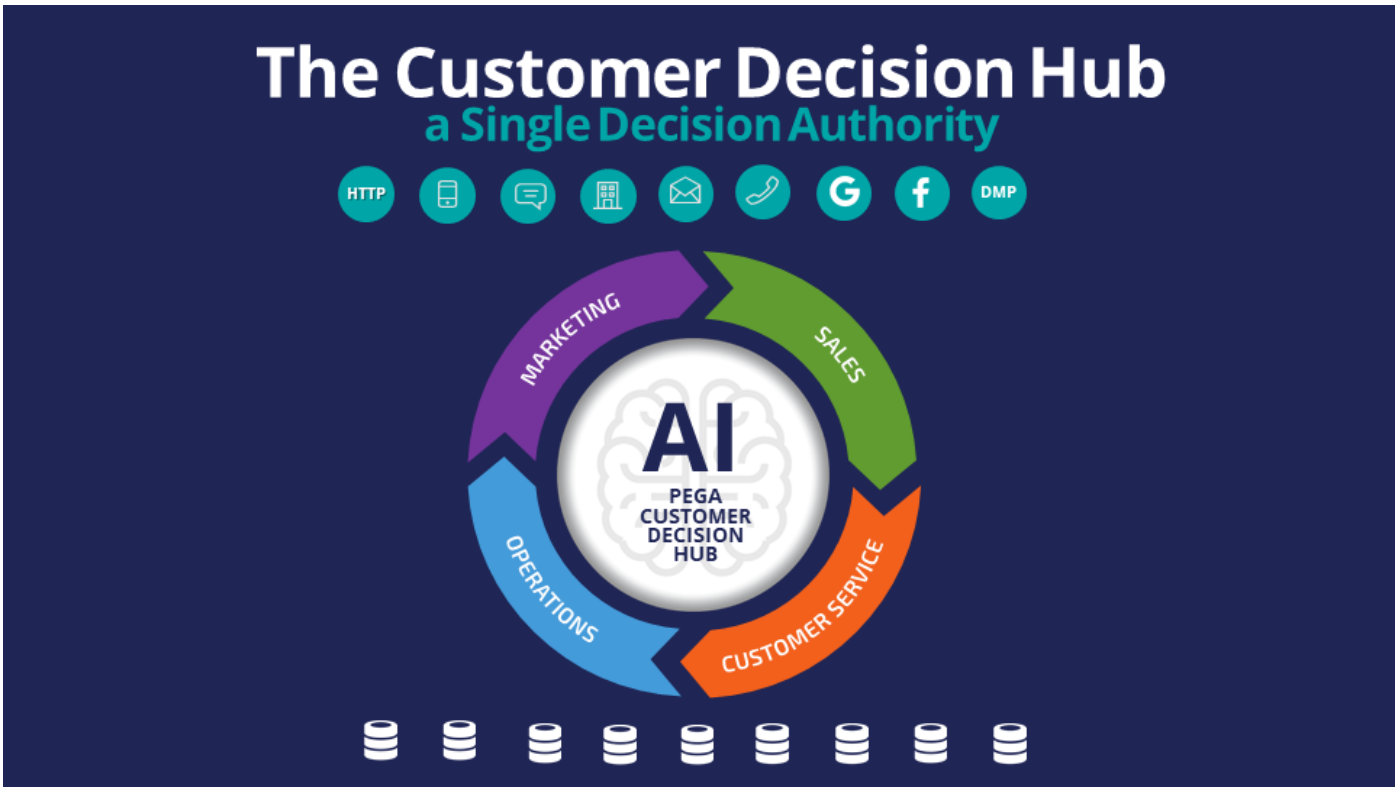
In other words, one piece of intelligence that acts as a single decision authority across your application

ecosystem.

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

In Pega Marketing™, this centralized brain is called the Pega Customer Decision Hub, and it leverages AI to enable 1-to-1 Customer Engagement.

In Pega Infinity™, the 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.



Data from every customer engagement across the enterprise is collected by the Brain and used to make predictions and decisions about every interaction in every channel.

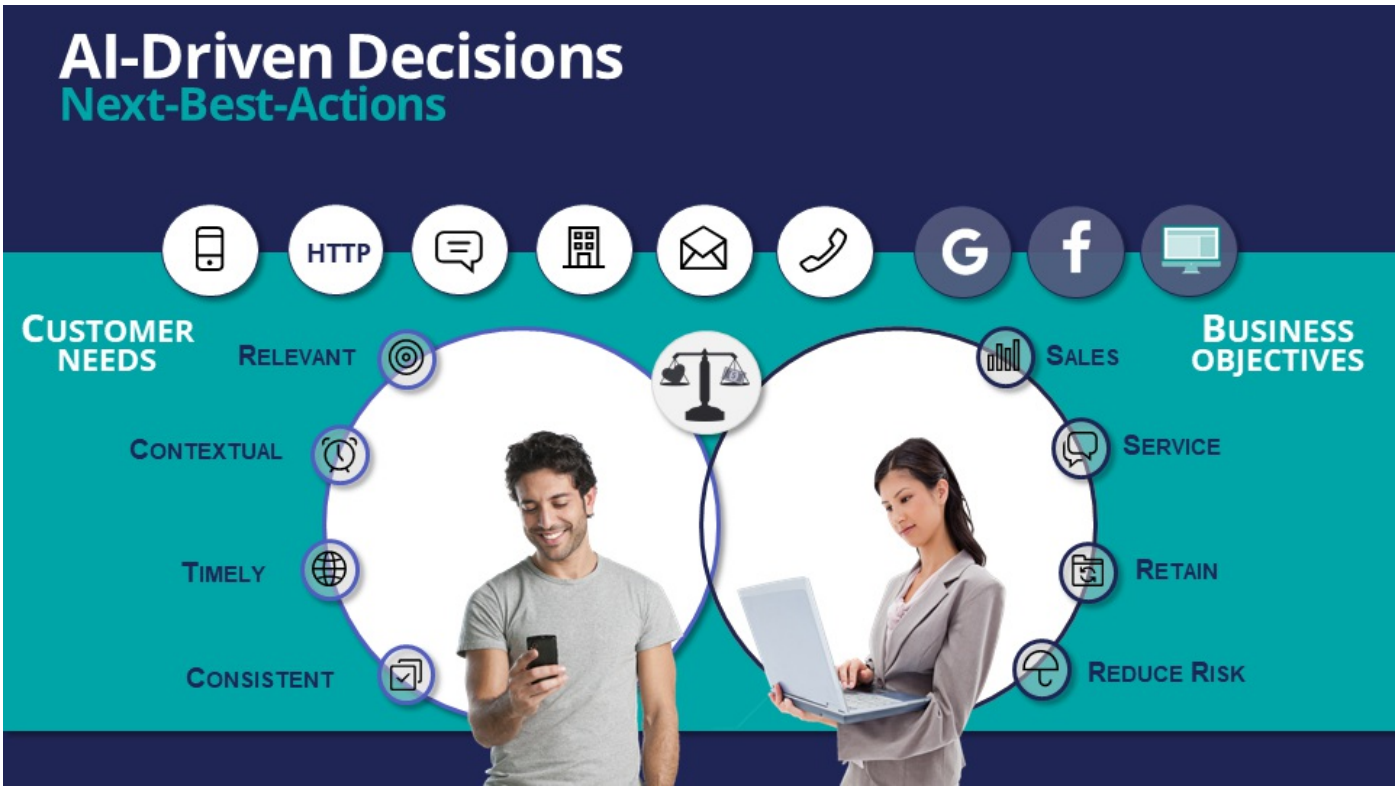
Continuous learning and decision-making are the foundation of a 1-to-1 Customer Engagement solution.

The 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.

In the milliseconds before interacting with a customer, the Customer Decision Hub processes thousands of predictive and adaptive models to determine customer needs, considering the customer's immediate context to ensure the Next-Best-Action is relevant, timely, and contextual. These models can be propensity, risk, or churn models.

Next, the decision strategy considers business rules and matches those with the customer's context and higher-level business goals.

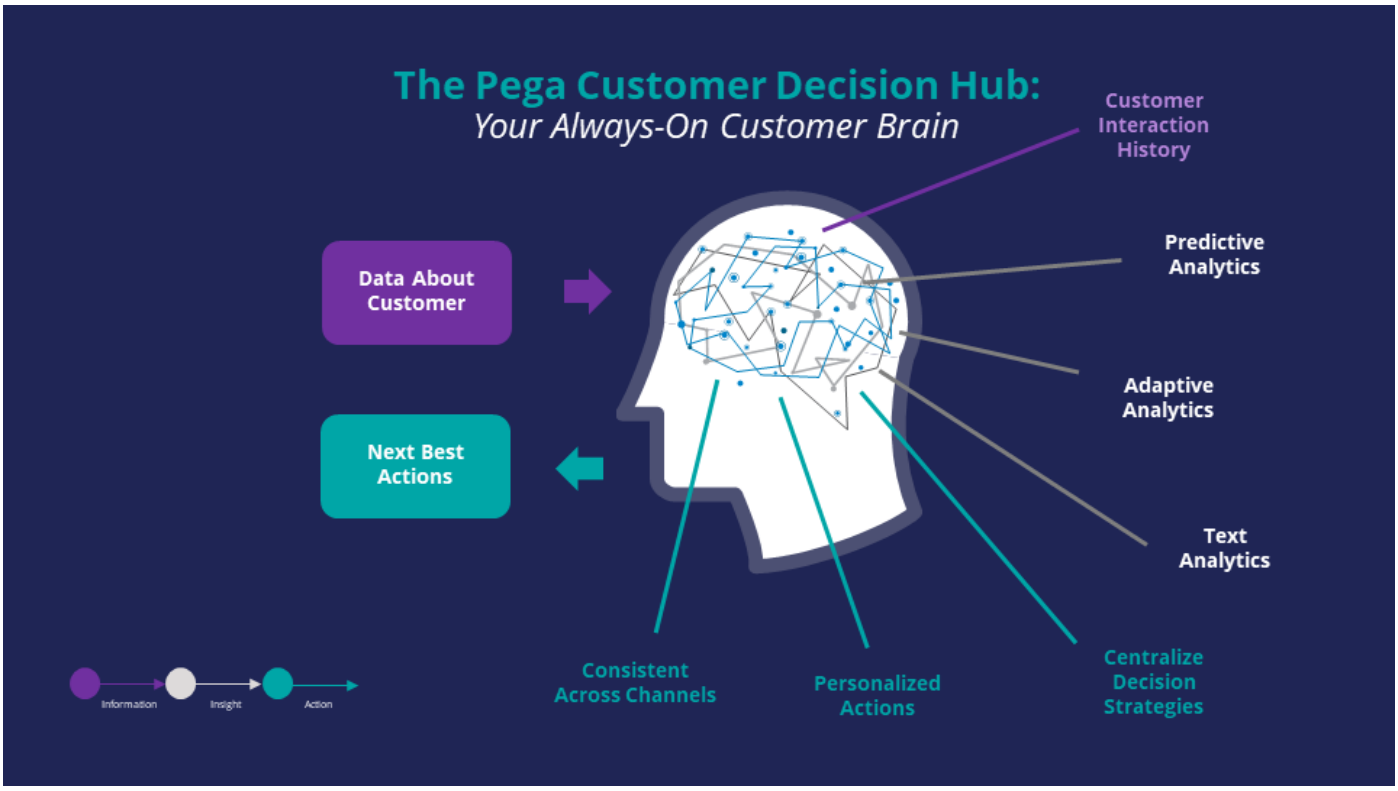


All of this information is used by the Next-Best-Action decision strategy to evaluate every potential action that could be taken with a particular customer in a given situation. The decision strategy then recommends the best way to interact with the customer to achieve the optimal result.

Using the Next-Best-Action approach, the Customer Decision Hub is able to identify the best moments for making a sale, providing a service, making a retention offer, or doing nothing at all (e.g. if nothing is relevant enough to warrant the customer's attention). Next-Best-Action is even able to select which offers are most likely to be accepted by the customer in a sales or retention situation. Next-Best-Action decisions are distributed, in real-time, to each of your real-time owned channels, such as web, mobile, and contact center. Through Pega Customer Decision Hub, Next-Best-Actions can also be distributed to real-time paid channels such as Google, YouTube, Facebook, LinkedIn and Instagram. Pega Marketing also integrates with non-real time outbound channels such as data management platforms (DMPs) and email.

Once the Next-Best-Actions are distributed and customer responses have been received by the Brain, the whole process begins again, and new Next-Best-Actions are distributed within milliseconds. Every outbound channel, including a data management platform, is dynamically updated with the Next-Best-Action to ensure consistency and an optimized customer experience no matter which channel the customer interacts with.

In summary, the Pega 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 leverages advanced AI techniques to make predictions.

And it uses decision strategies (which combine traditional business rules with predictive, adaptive and text analytics), to deliver consistent and personalized Next-Best-Actions across all channels.

One-to-one Customer Engagement paradigm -- Thu, 07/23/2020 - 05:55
To get the full experience of this content, please visit https://academy.pega.com

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

The 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 that need to be completed to define Next-Best-Actions.

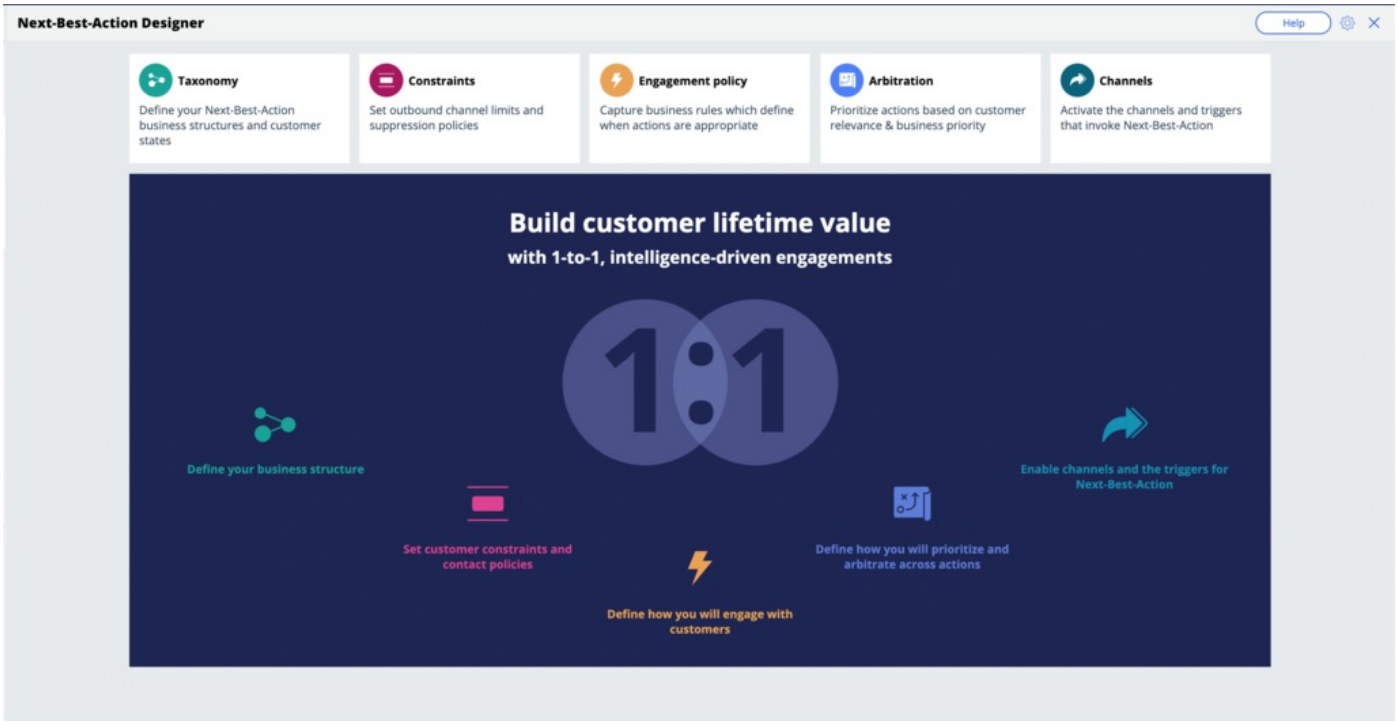
Use the **Taxonomy** component to define the business structure for your organization.

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.

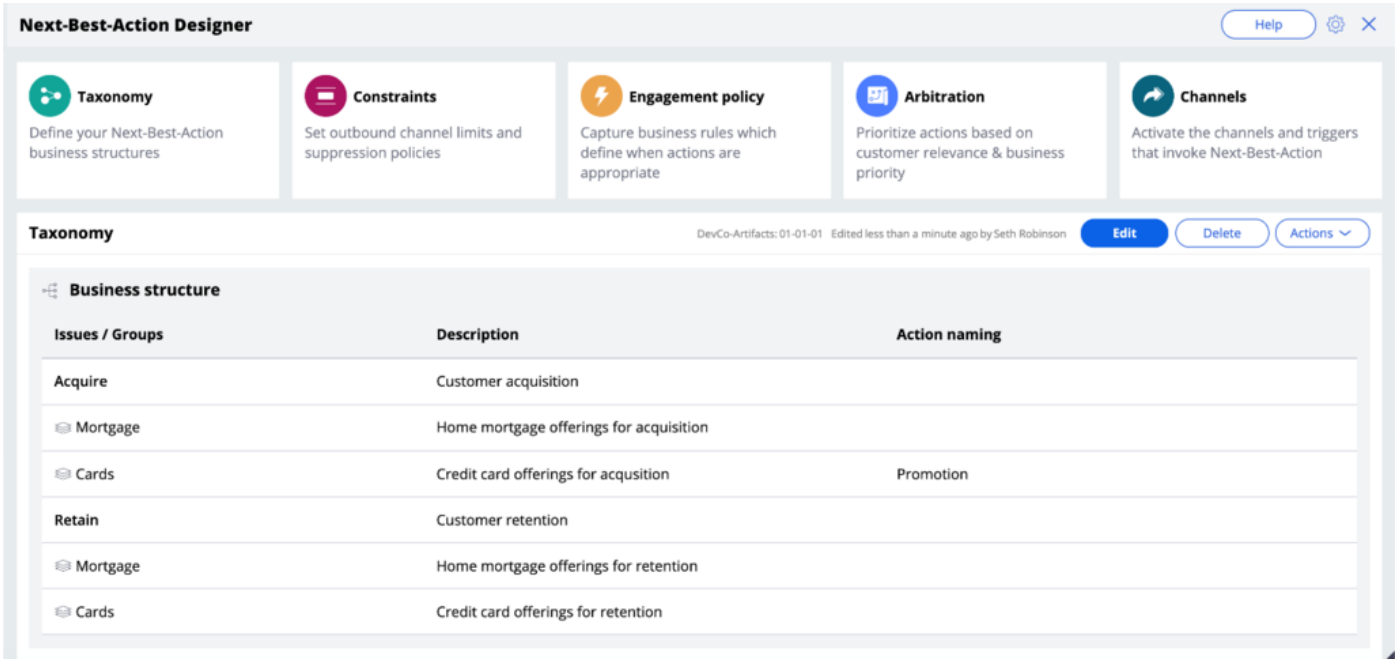
Use the **Arbitration** component to configure how actions are prioritized.

Use the **Channels** component to configure when and where Next-Best-Action is triggered.



The system uses these definitions to create an underlying Next-Best-Action Strategy framework. This framework leverages best practices to 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 Centralized Decision Hub, which determines the personalized set of Next-Best-Actions for each customer.

Use the **Taxonomy** component to define the hierarchy of Business Issues and Groups to which an action belongs.

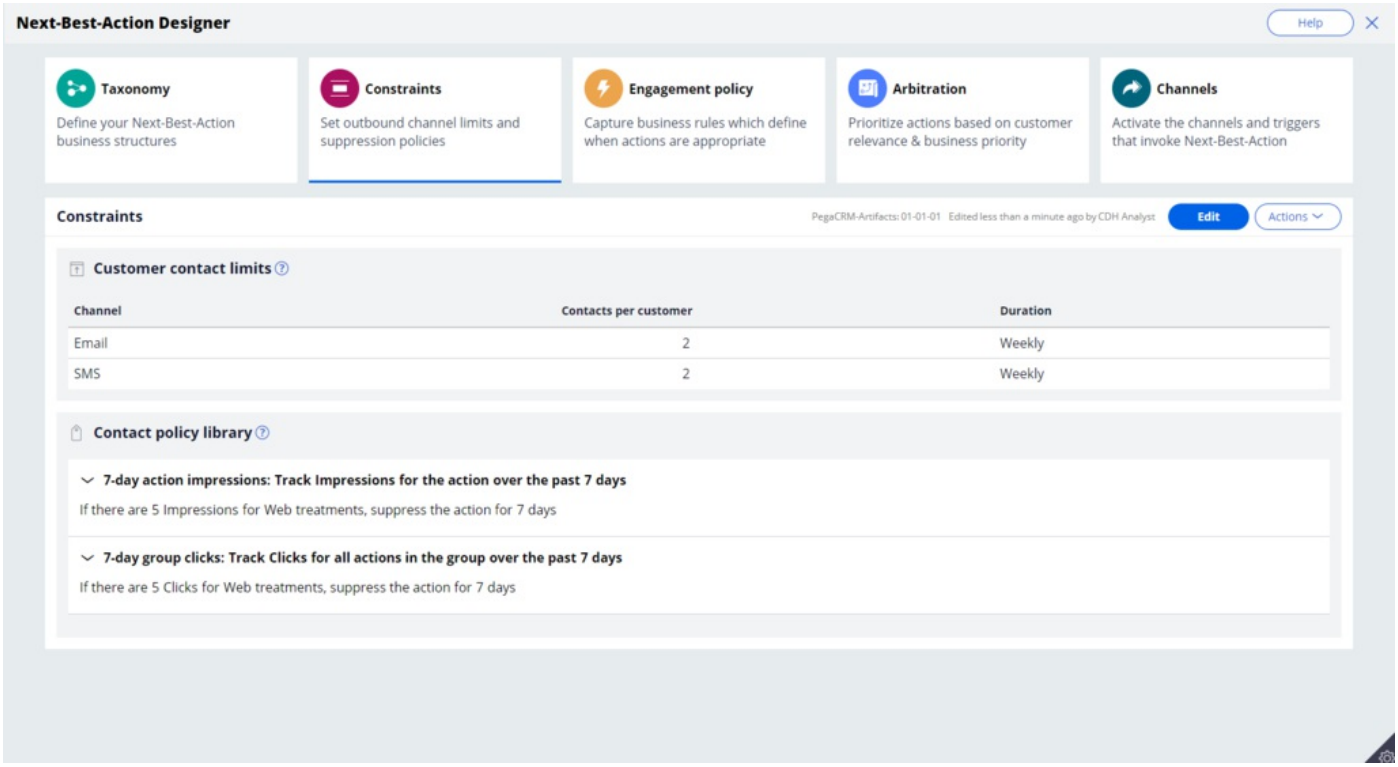


A Business Issue is the purpose behind the actions you offer to customers. For example, actions with the purpose of retaining existing customers should be grouped under the business Issue of Retention. Actions with the purpose of acquiring new customers belong to the business Issue of Acquisition.

Business Groups are used to organize customer actions into categories. For example, as part of the business Issue of Acquisition, you can create Groups for products like Credit Cards, Mortgages, or Personal Loans, with the intention of offering these to potential customers.

Use **Constraints** to specify outbound contact limits as well as to limit overexposure to a specific action or

group of actions.

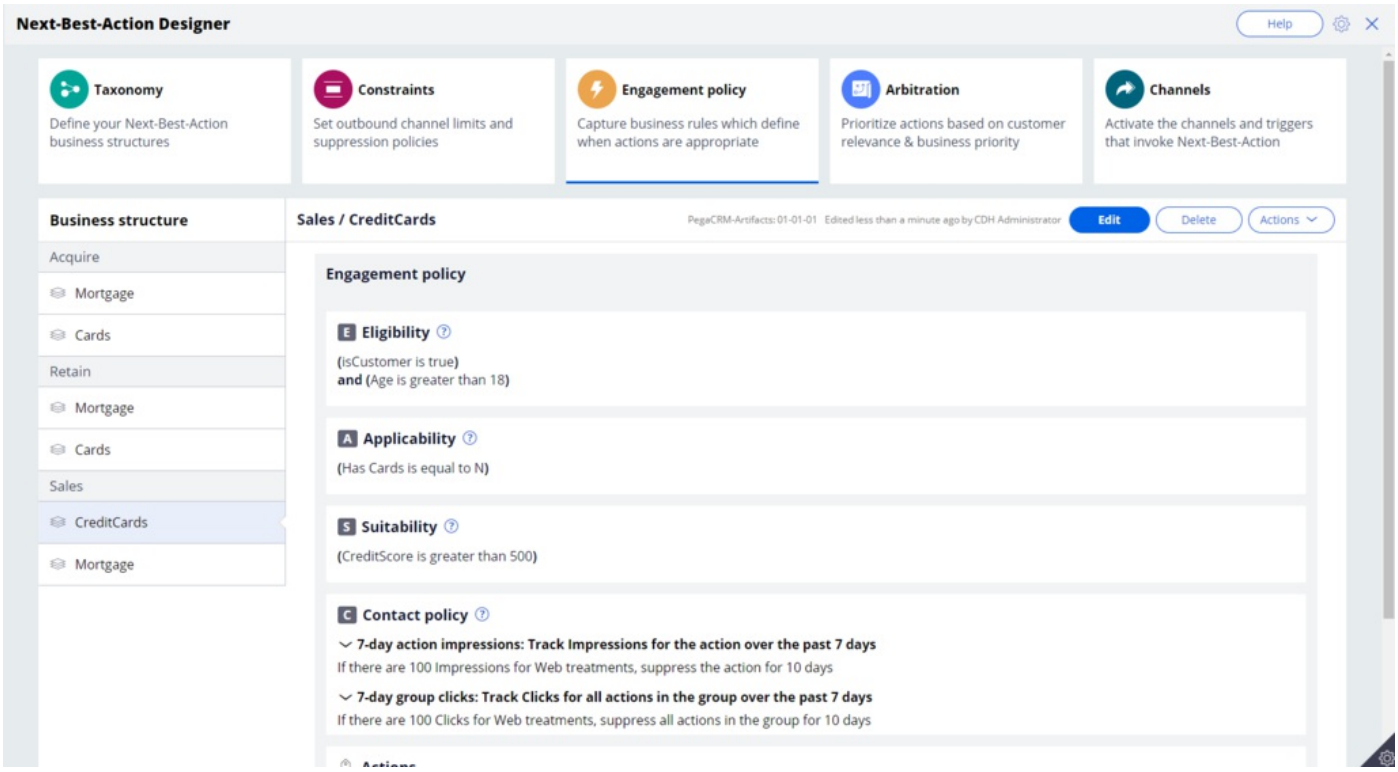


Customer contact limits allow you to limit the number of interactions that a customer can receive over a given period of time on a specific channel. For example, you can decide that you do not want your customers to receive more than two emails per week or two SMS messages per week.

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:

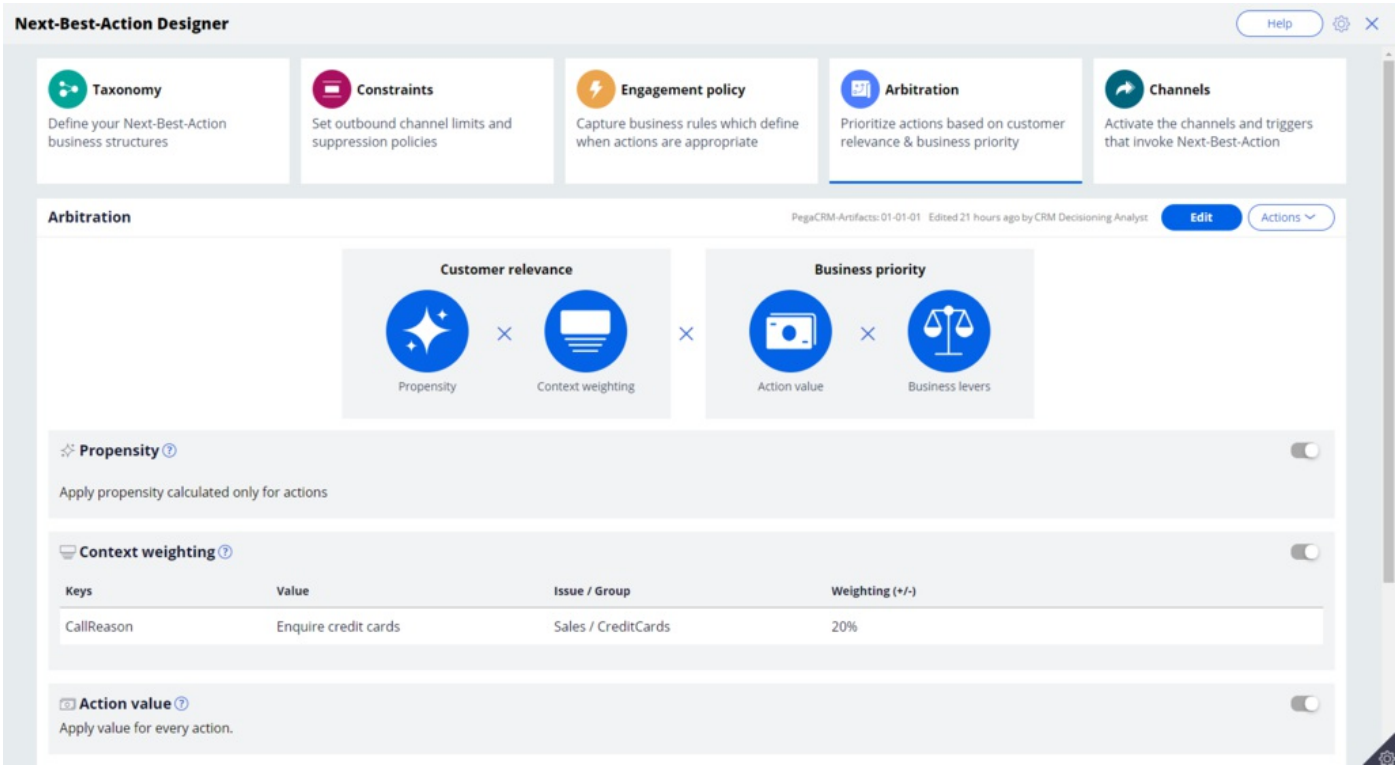
Eligibility determines whether or not a customer qualifies for an action or group of actions. For example, an action may only be available for customers over a specific age or living in a specific geographic location.

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 has been sent to customers. To activate Contact Policy rules created in the library on the Constraints tab, add them to the Engagement Policy tab.

Arbitration determines how the 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, Action value, and Business levers, each represented by numerical values. A simple formula is used to arrive at a prioritization value, which is used to select the top actions.

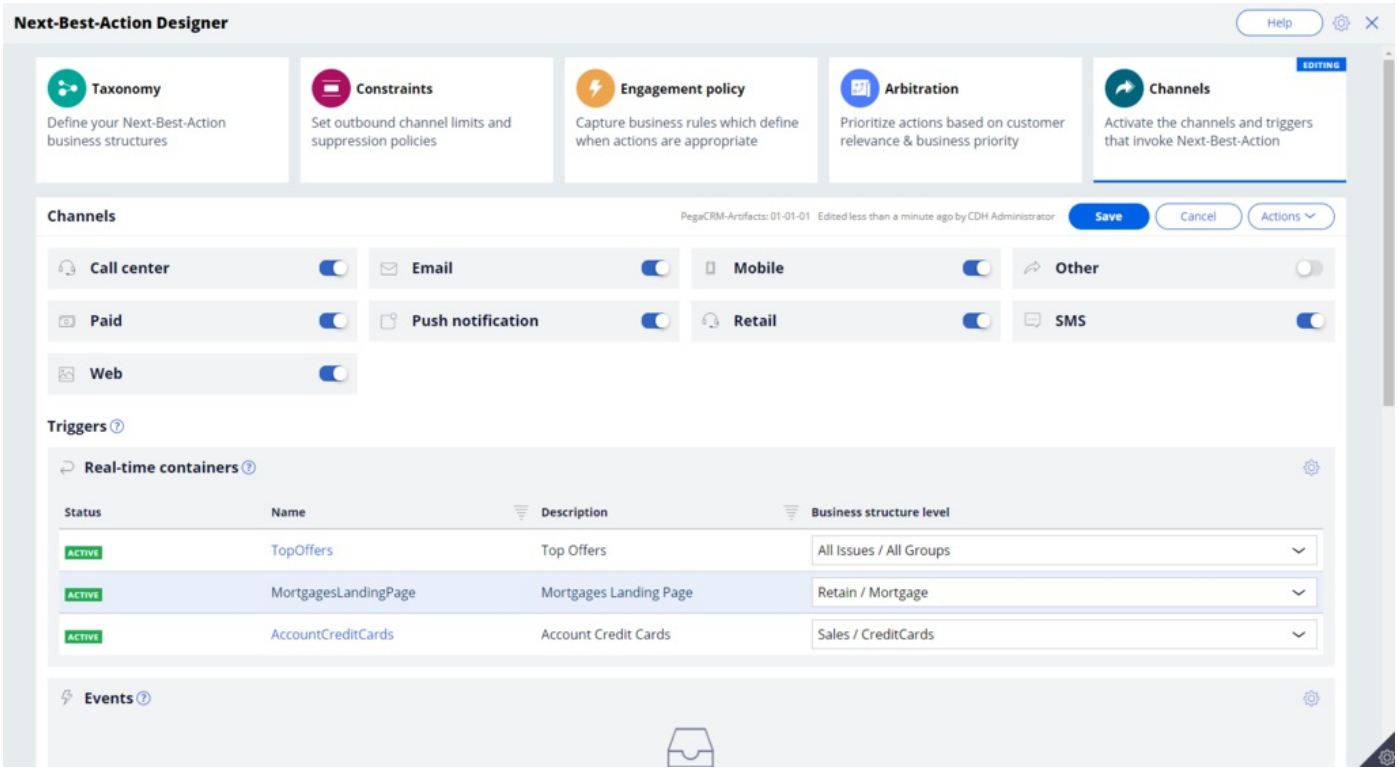
Propensity is the likelihood of a customer responding positively to an action. This is calculated by Artificial Intelligence (AI). For example, a click on an offer banner or an accept of an offer in the contact center are considered positive behaviors.

Real-time contextual data is an important part of making highly relevant recommendations**. Context weighting** allows you to assign 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 will increase, and the highest priority action will be to ensure that the relevant service is delivered to the customer.

Action 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.

Next-Best-Action Designer enables Next-Best-Actions to be delivered via inbound, outbound and paid channels.



These channels can be toggled on or off. If a channel is toggled off, the Next-Best-Actions will not be delivered to that channel.

An external real-time channel is any channel that presents actions selected by the Customer Decision Hub to a customer. These channels can include a website, or a call-center or 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 will be delivered back to the invoking channel. For example, when a real-time container called “Mortgages Landing Page” is configured, the website invokes this real-time container before loading the mortgage 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 rules that control which actions are offered to which customers
- Configuring how actions are prioritized
- Configuring when and where Next-Best-Action is triggered

Next-Best-Action Designer -- Thu, 07/23/2020 - 05:56
To get the full experience of this content, please visit https://academy.pega.com

Next-Best-Action in a contact center

Introduction

The Customer Decision Hub is an “always on” centralized decisioning “brain” that calculates a 1-to-1 business case for every Next-Best-Action recommendation. To create the business case, the Customer Decision Hub combines customer profile with previous interaction results, the current call context, business rules, and then applies predictive analytics.

Video

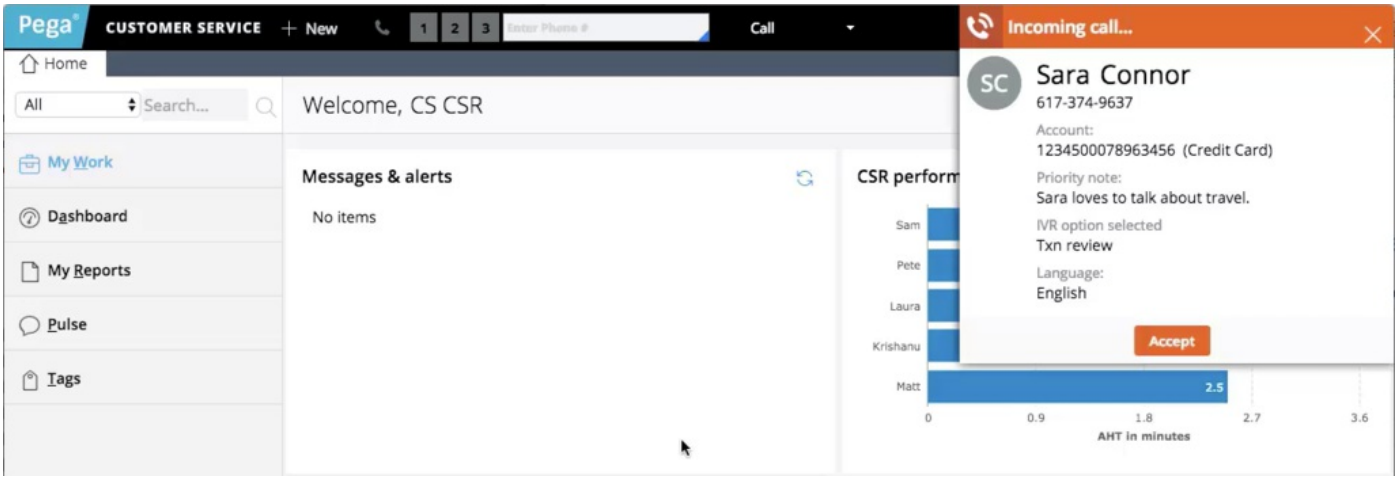
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

Next-Best-Action is used to ensure a customer service representative takes relevant actions at every step during a customer interaction.

In this scenario, U+ is a retail bank that uses Pega Customer Service in its contact center.

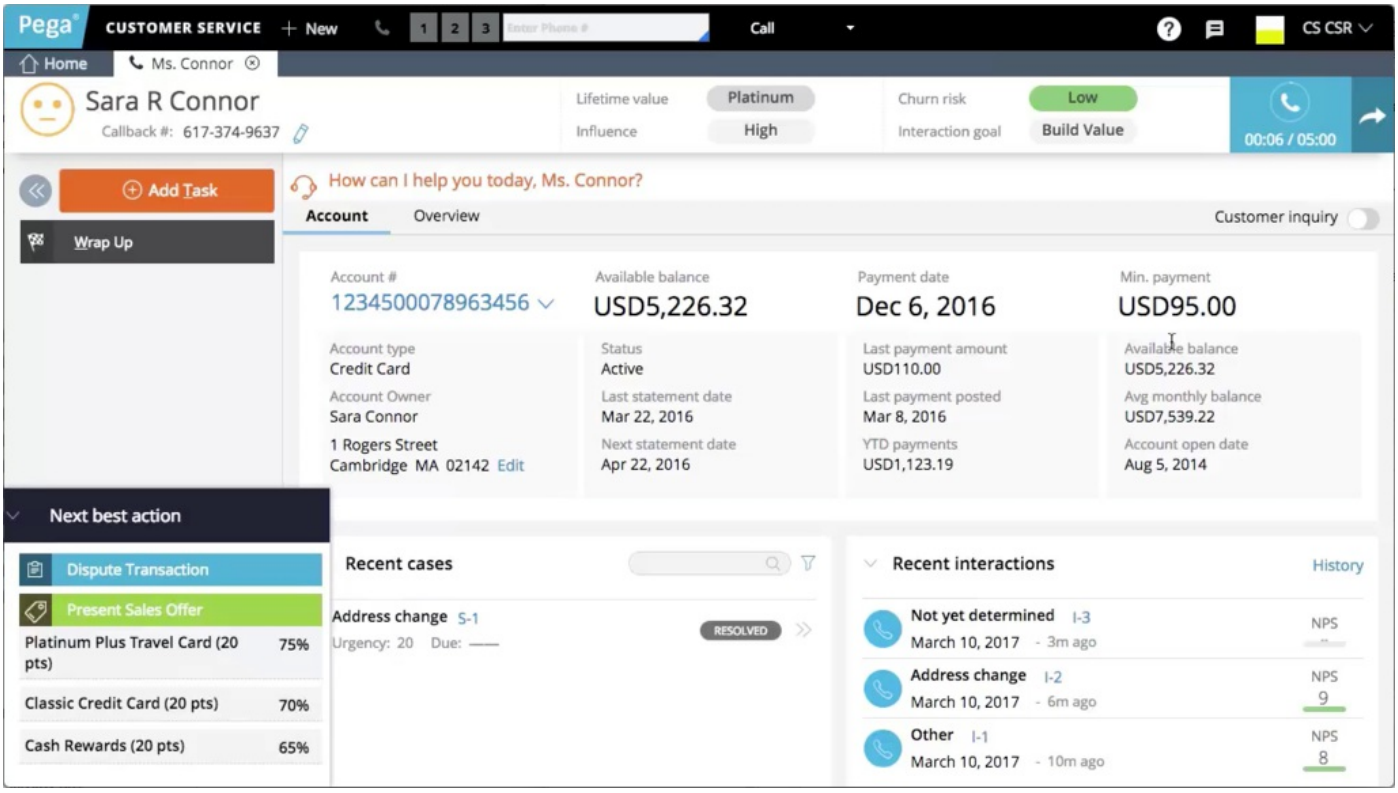
A call comes in to the U+ service center from Sara Connor, a U+ customer. The call is immediately routed to a service representative.



The call details from the IVR system indicate that Sara would like to discuss recent credit card transactions.

Once the service representative accepts the call, all relevant details about Sara are visible on screen.

Next-Best-Action then guides the representative to take the next step with Sara.

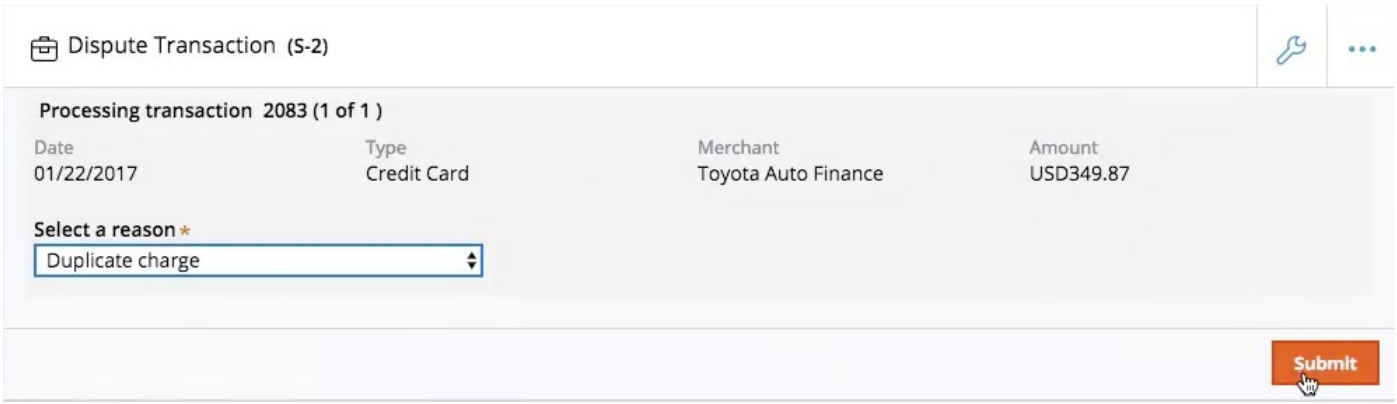


In the lower left corner of the screen you can see the Next-Best-Action that has been recommended to the agent by the Customer Decision Hub.

The Customer Decision Hub is an “always on” centralized decisioning “brain” that calculates a 1-to-1 business case for every Next-Best-Action recommendation. To create the business case, the Customer Decision Hub combines customer profile with previous interaction results, the current call context, business rules, and then applies predictive analytics.

The Customer Decision Hub re-evaluates 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 Sarah’s transaction dispute. So the service representative carries out the task. But the representative is always in control and can select other service actions as appropriate during the conversation.



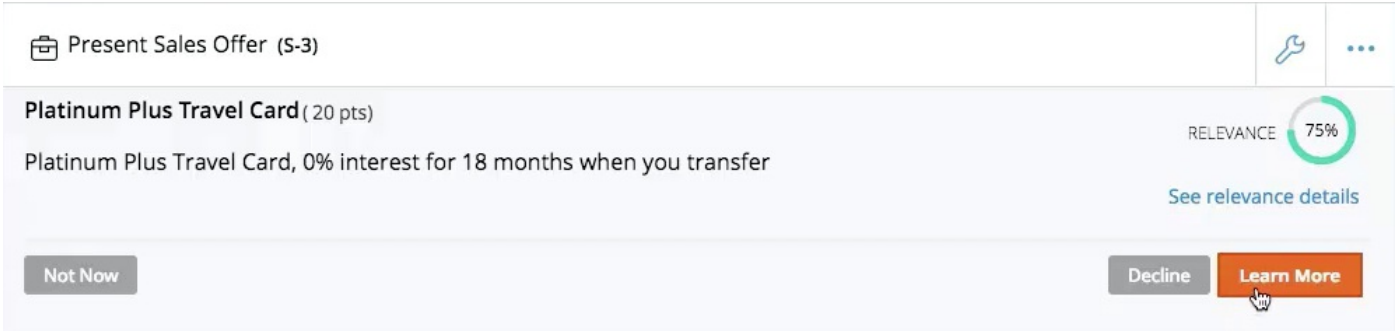
Once the service representative completes the task, the Next-Best-Action is refreshed to show the next recommended action, which is to present a sales offer.

The Customer Decision Hub has analyzed Sara's credit card usage patterns, which indicate that she is a frequent traveler.

As a frequent traveler she often pays a large amount in foreign currency transaction fees. So the highest recommendation is for a Platinum Plus Travel Card, which has no foreign currency transaction fees.

The percentages to the right are scores that are used to rank all relevant offers. The scores are calculated by balancing what the bank would like to promote with what Sara is likely to be interested in.

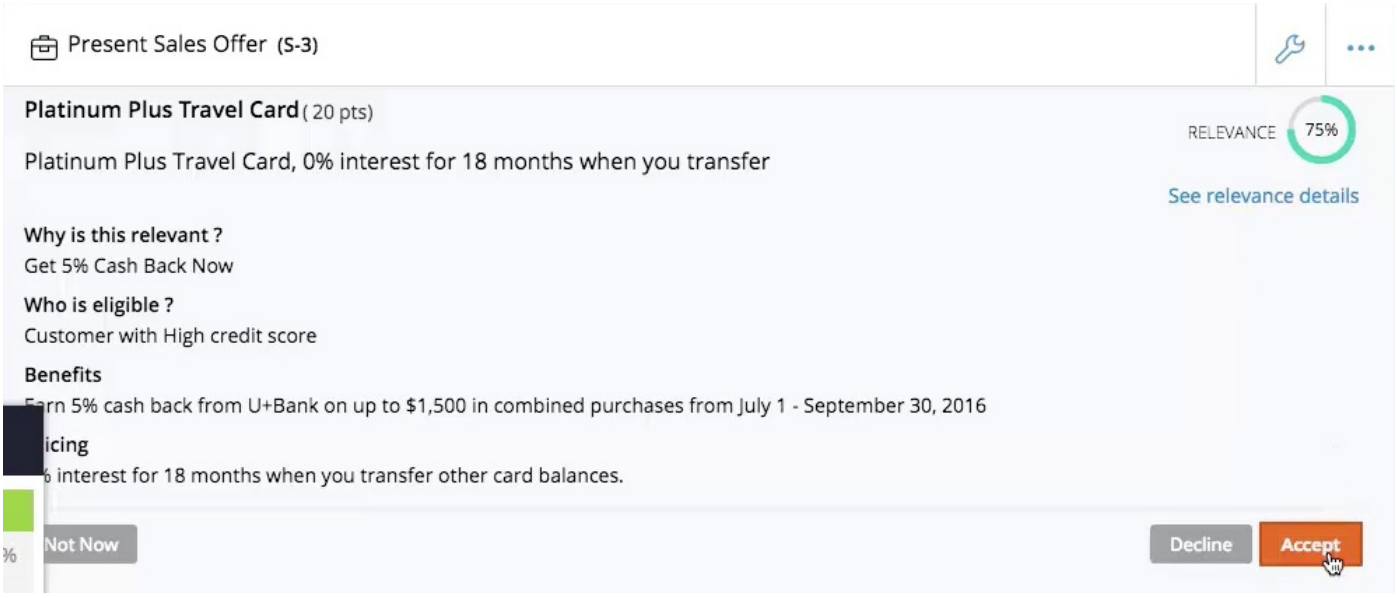
The representative can view more details about the recommended offer so he can further discuss its benefits with Sara.



In this case, Sara is interested in knowing more.

After learning about the benefits of the offer, Sara is indeed convinced that it's a good offer for her, and she decides to accept it.

When the representative clicks Accept, the customer response is recorded in the Interaction History.



This new information is used by the Customer Decision Hub to make the next recommendation.

Notice that the Next-Best-Action is now refreshed.

The Platinum Plus Travel Card and all other credit card offers have been removed from the list.

To summarize, the always on, centralized Customer Decision Hub combines customer and contextual information with business rules and analytics to recommend the Next-Best-Action in real-time.

The Next-Best-Action is re-evaluated when new information becomes available.

Next-Best-Action on digital

Introduction

Leverage a website as a marketing channel to improve one-to-one customer engagement, drive sales, and deliver Next-Best-Actions in real-time. Use the Pega Customer Decision Hub™ to recommend more relevant banner ads to customers when they visit their personal portal.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

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

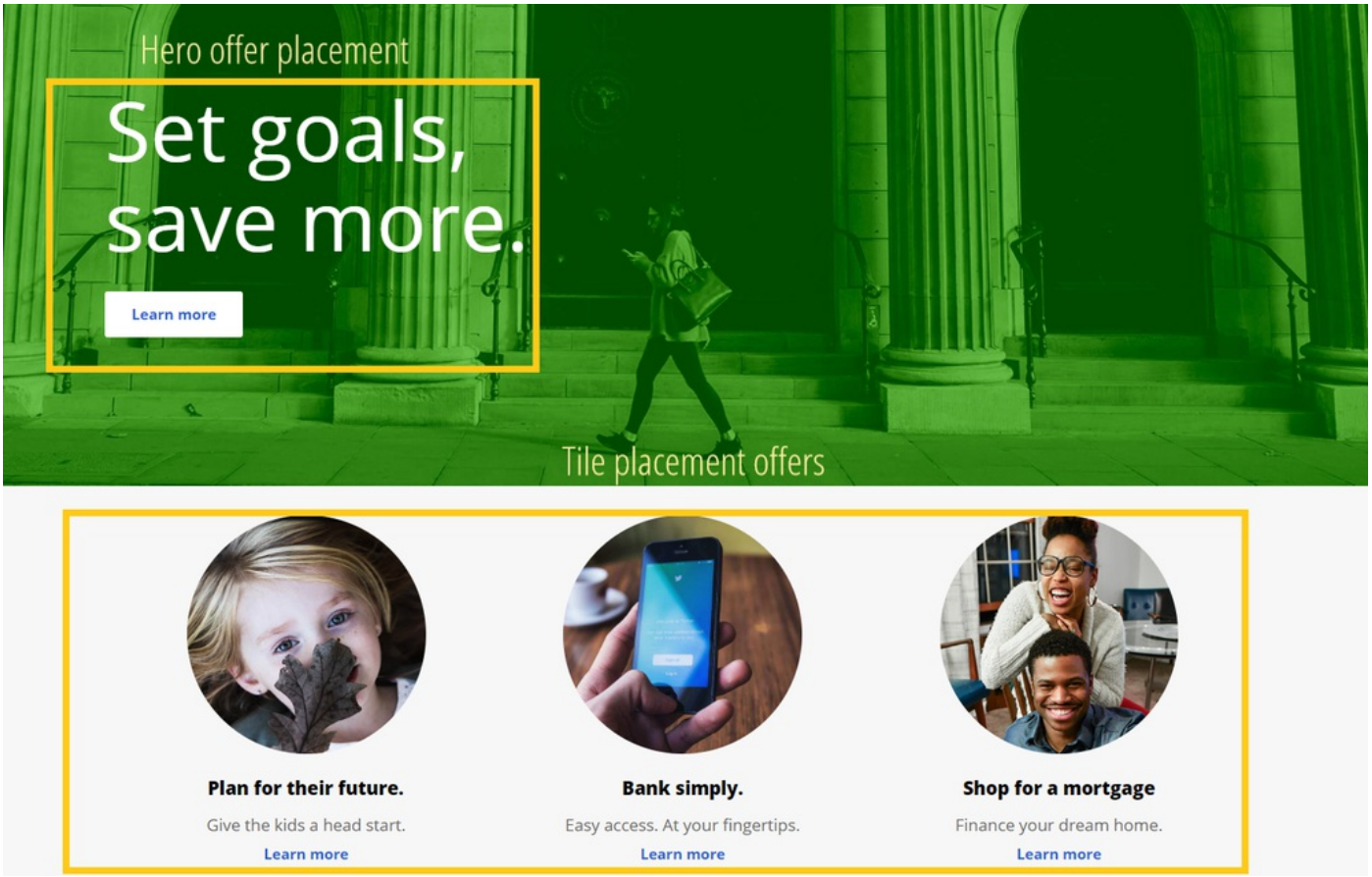
U+ is a retail bank. The bank would like to leverage its website as a marketing channel to improve 1-to-1 customer engagement, drive sales, and deliver Next-Best-Actions in real-time.

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.

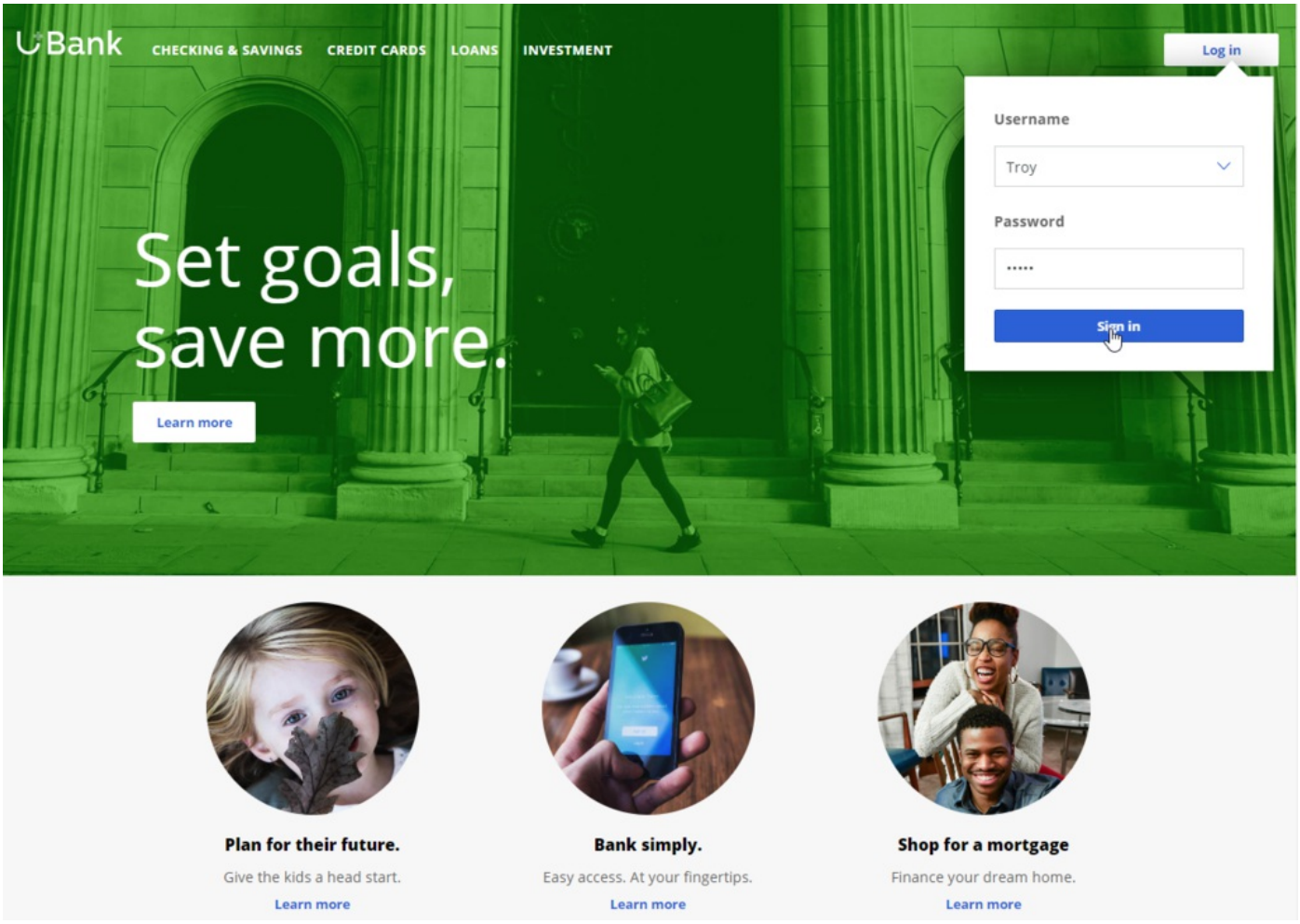
Banner ads are shown on various pages throughout the website.

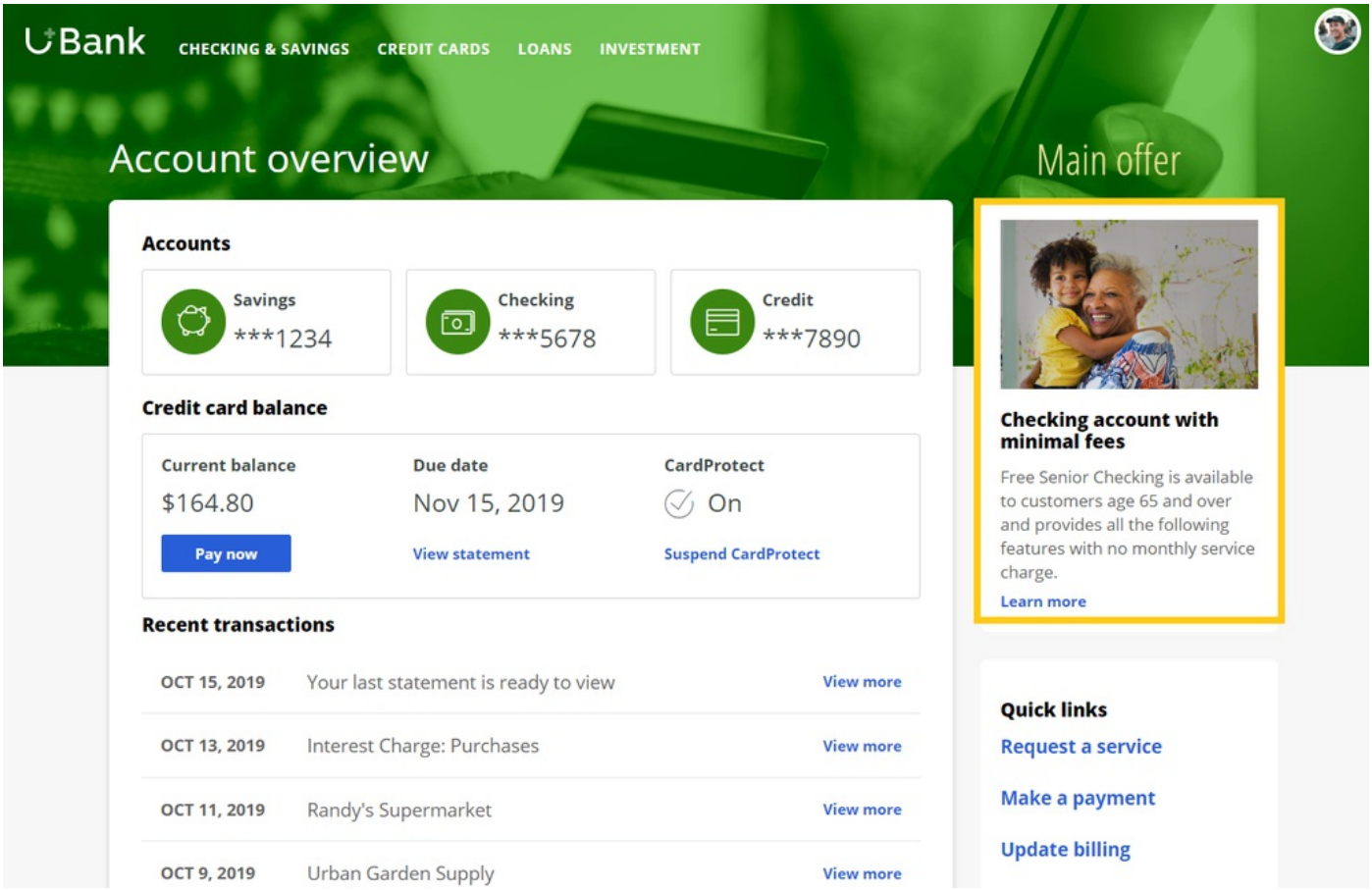
For example, on the home page, U+ can display a Hero banner at the top of the page, which is typically a larger image with bigger typeface.

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

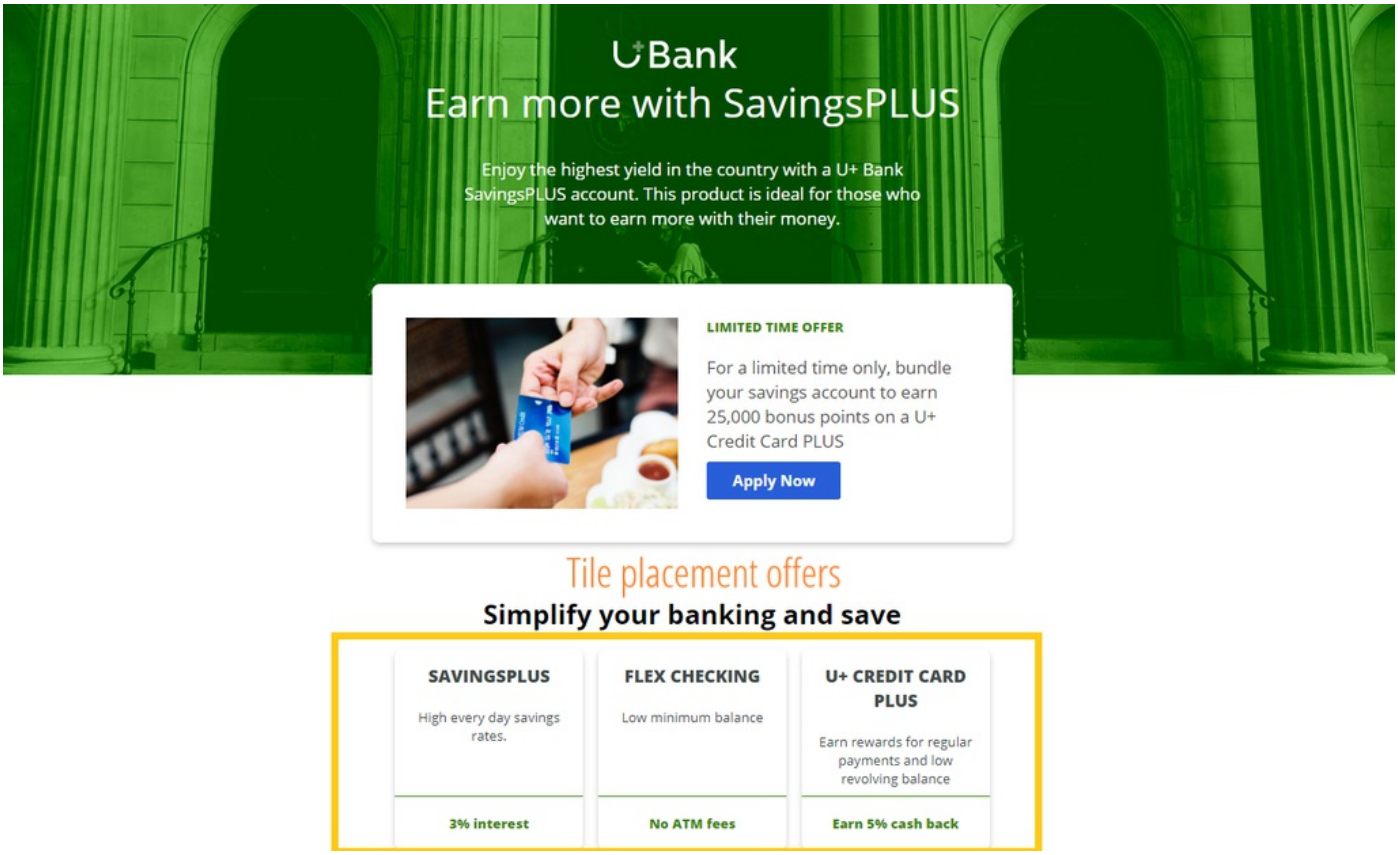


When customers log in to their personal portal, they also see a Tile banner on the Account overview page.





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.



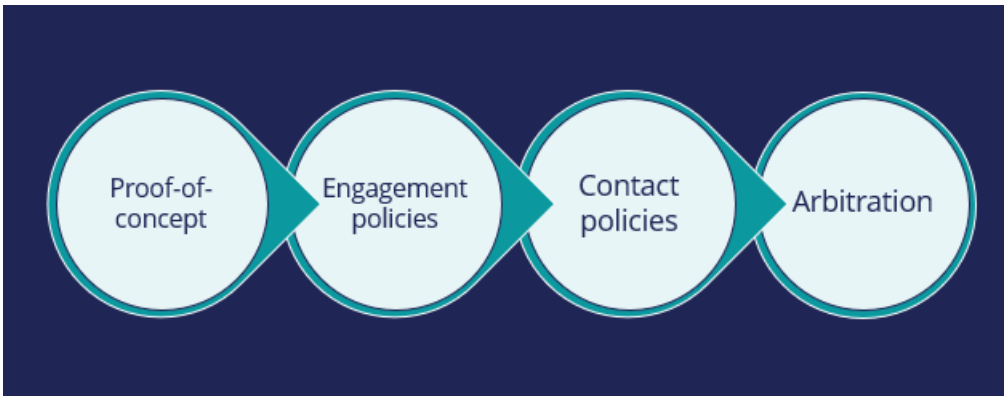
The bank would like to use these banners on the Account overview page, to display offers that are more relevant and likely to receive a positive response.

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, Action 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.

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

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

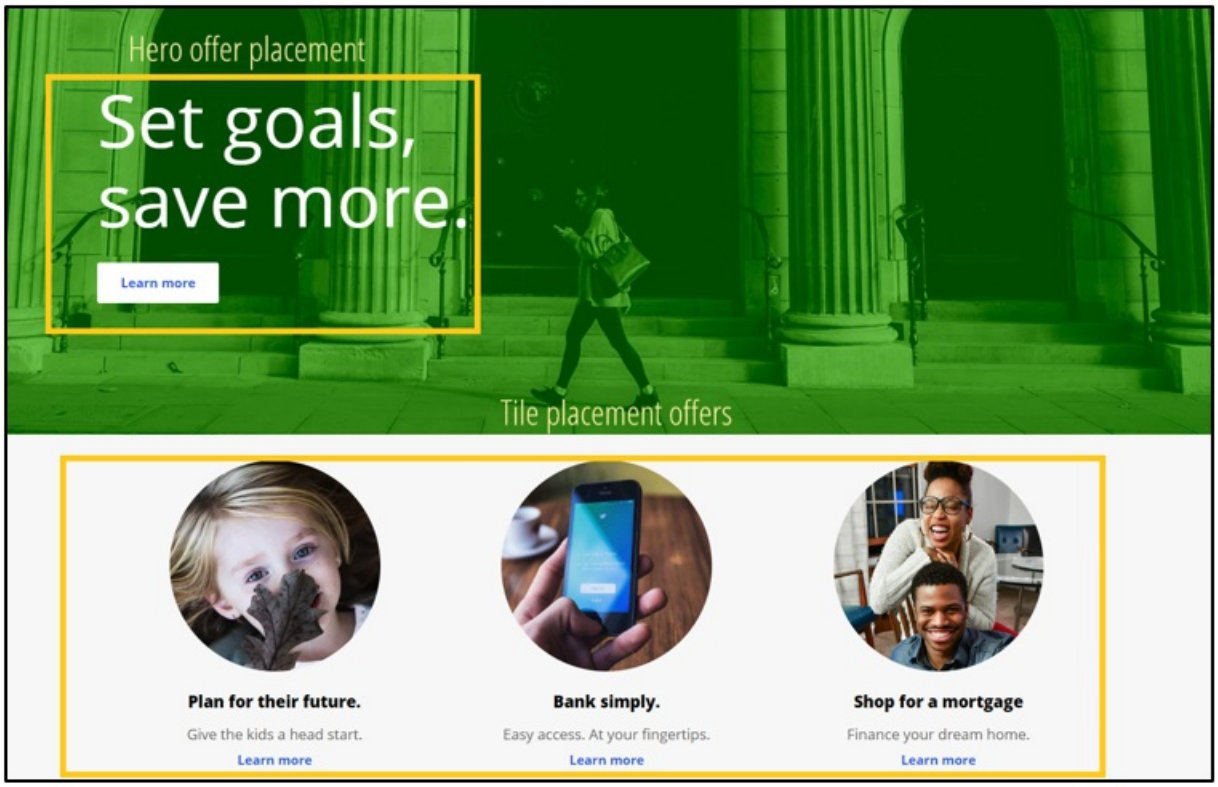
U+ is a retail bank. The bank would like to leverage its website as a marketing channel to improve 1-to-1 customer engagement, drive sales, and deliver Next-Best-Actions in real-time.

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.

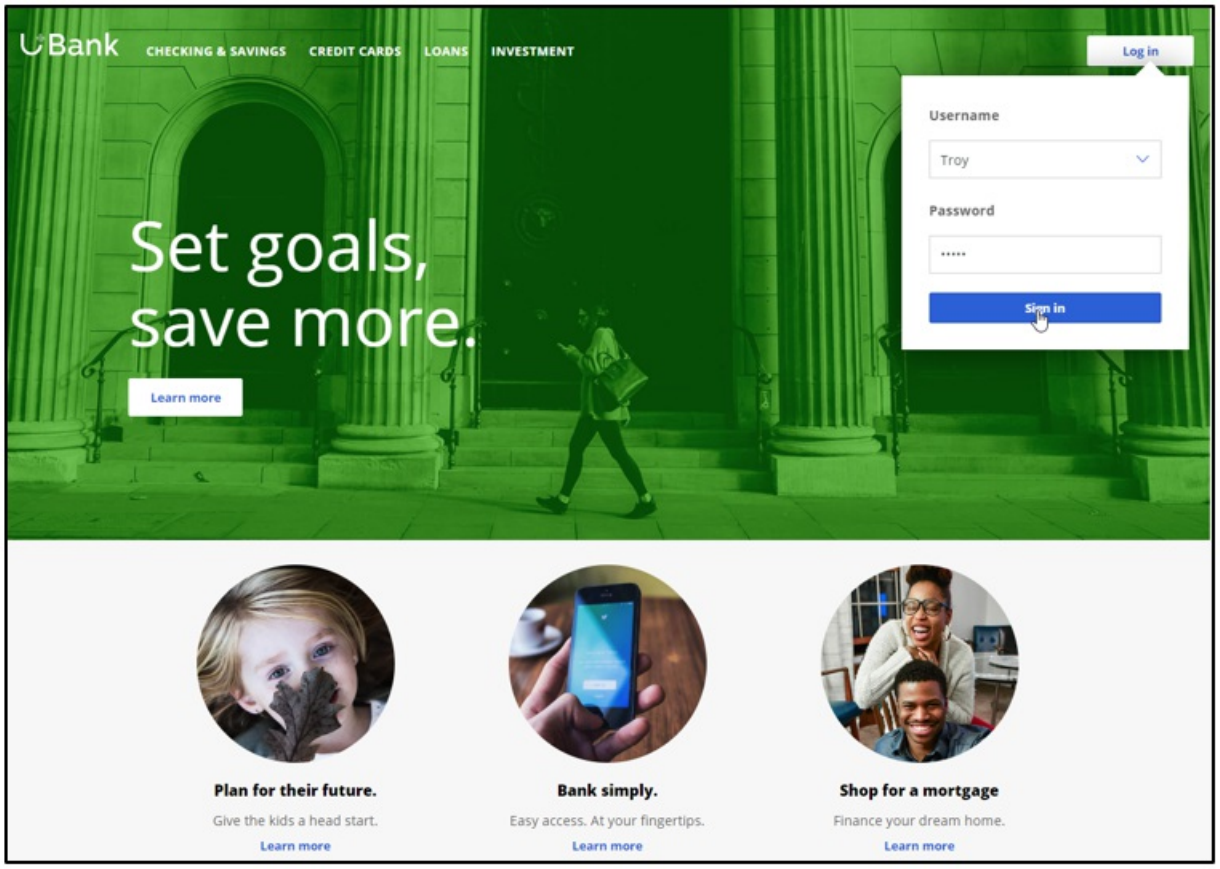
Banner ads are shown on various pages throughout the website.

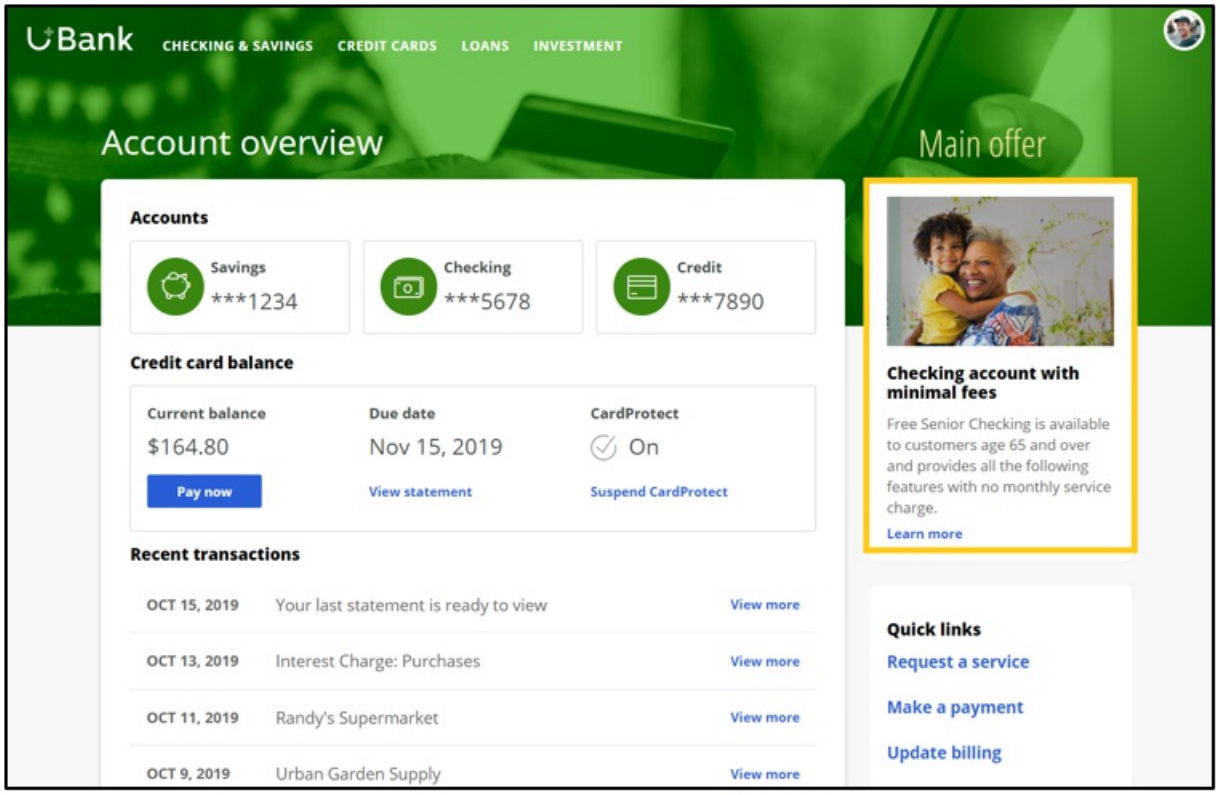
For example, on the home page, U+ can display a Hero banner at the top of the page, which is typically a larger image with bigger typeface.

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

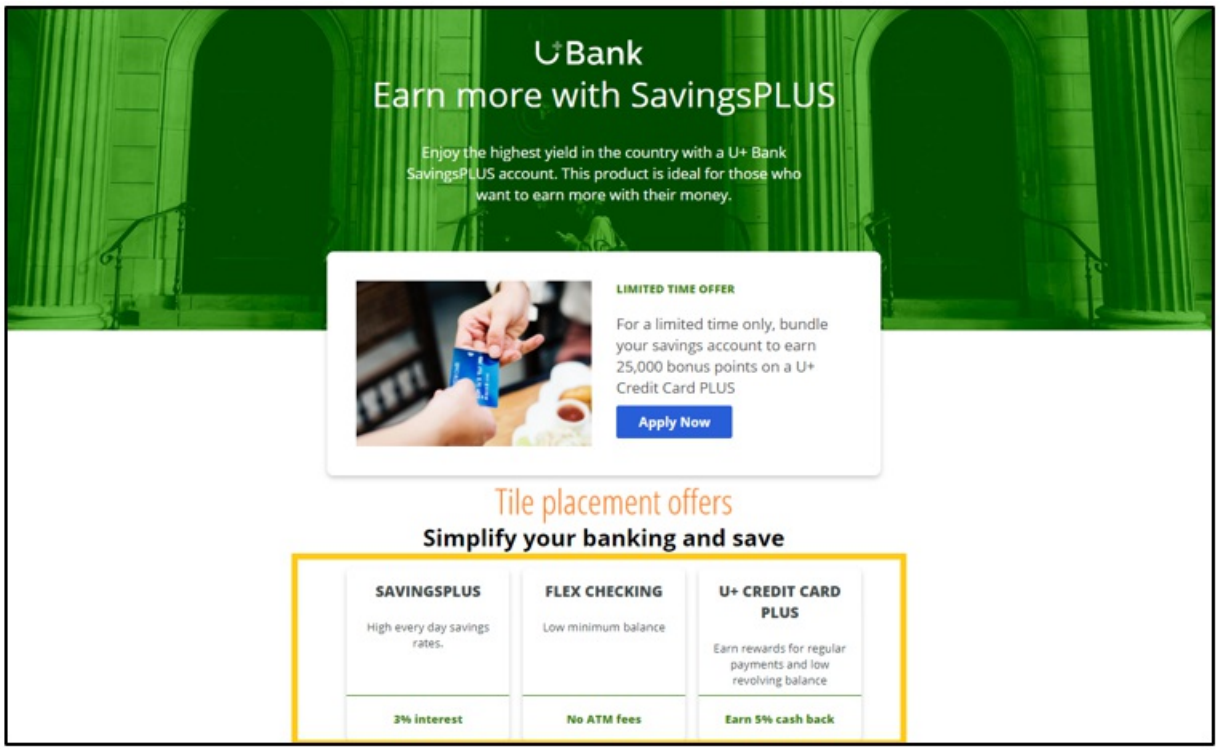


When customers log in to their personal portal, they also see a Tile banner on the ‘Account overview’ page.





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.



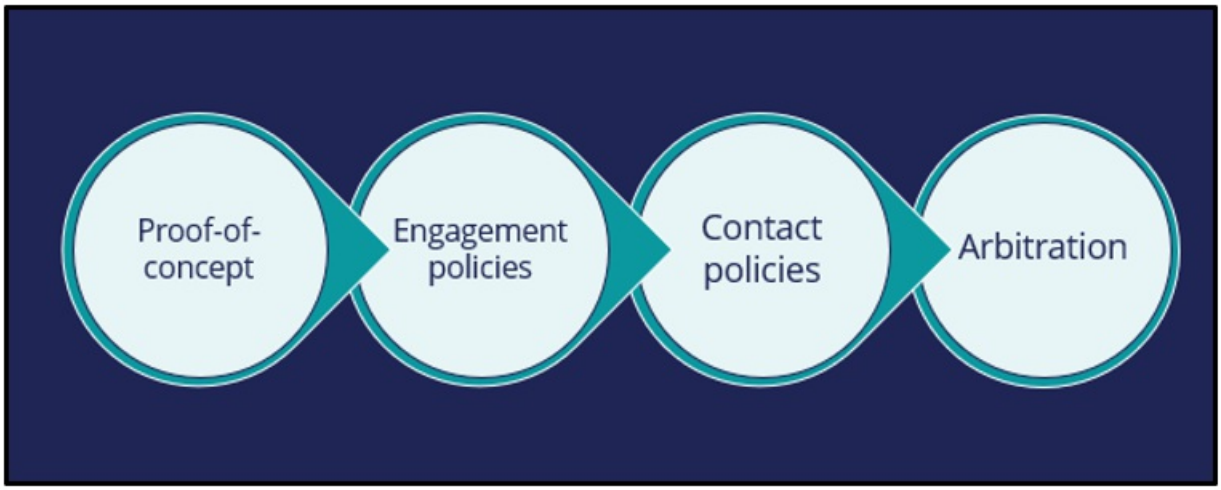
The bank would like to use these banners on the ‘Account overview’ page, to display offers that are more relevant and likely to receive a positive response.

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, Action 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.

Business use case: Cross-sell on the web -- Wed, 12/02/2020 - 00:33
To get the full experience of this content, please visit https://academy.pega.com

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

The 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 that need to be completed to define Next-Best-Actions.

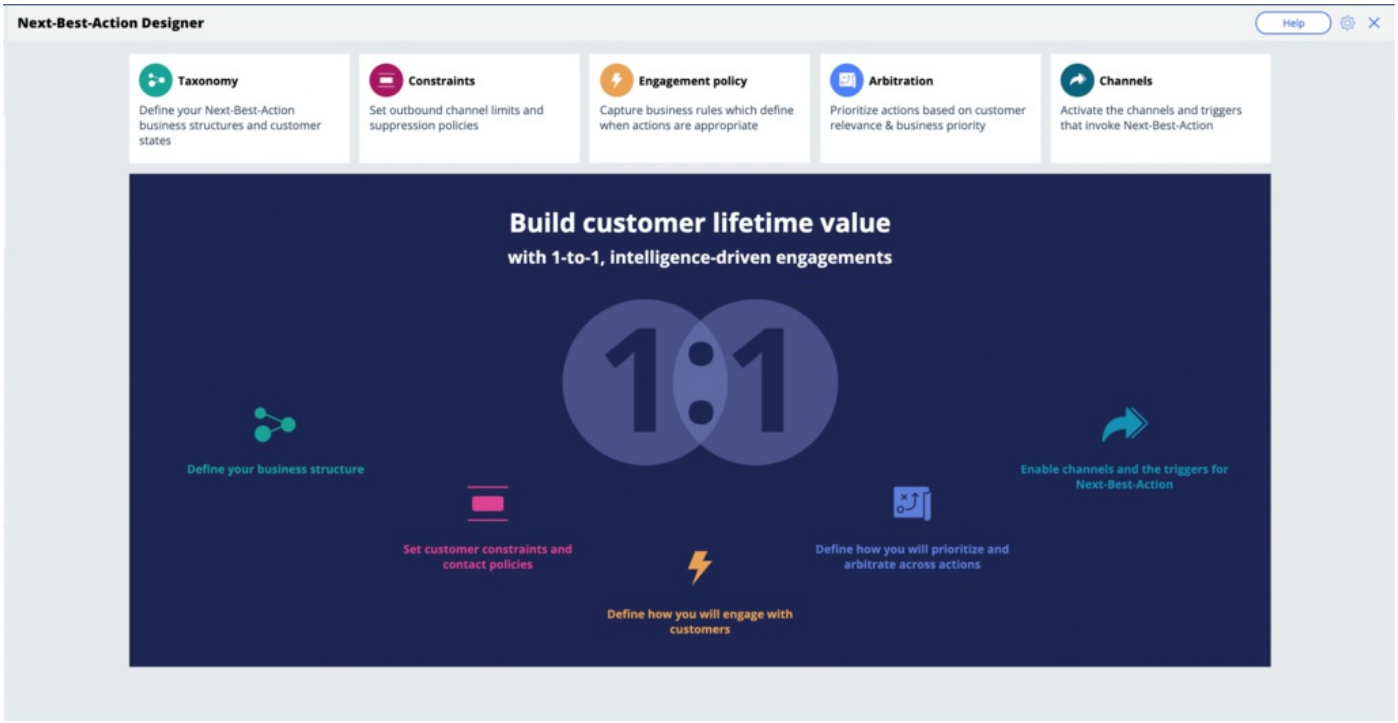
Use the **Taxonomy** component to define the business structure for your organization.

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.

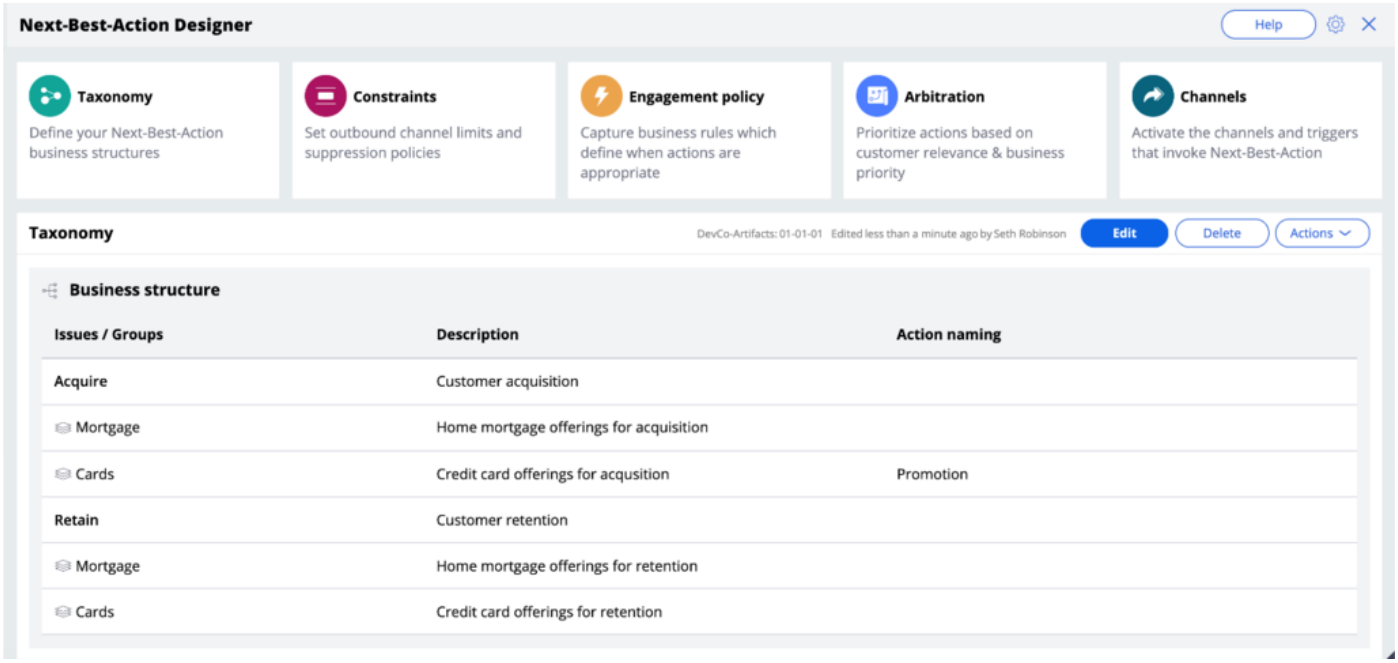
Use the **Arbitration** component to configure how actions are prioritized.

Use the **Channels** component to configure when and where Next-Best-Action is triggered.



The system uses these definitions to create an underlying Next-Best-Action Strategy framework. This framework leverages best practices to 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 Centralized Decision Hub, which determines the personalized set of Next-Best-Actions for each customer.

Use the **Taxonomy** component to define the hierarchy of Business Issues and Groups to which an action belongs.

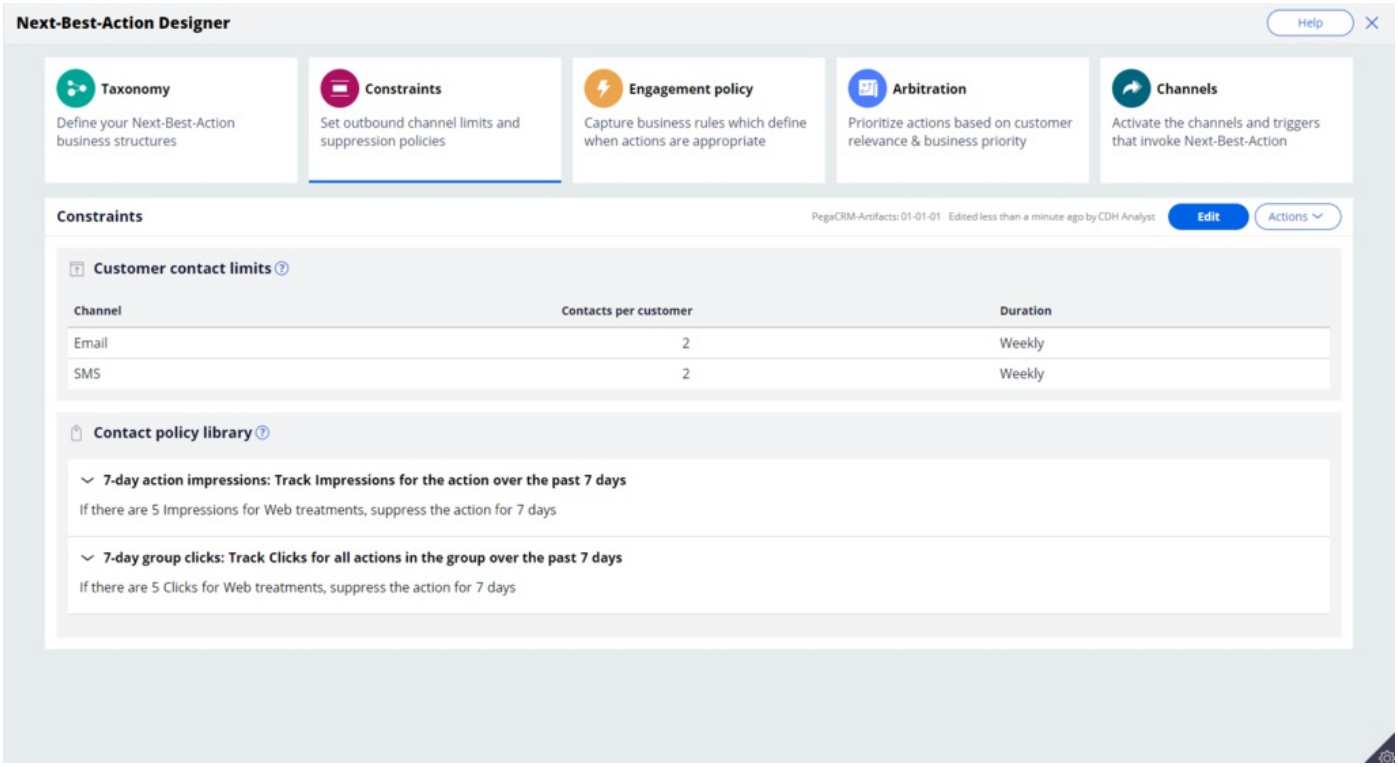


A Business Issue is the purpose behind the actions you offer to customers. For example, actions with the purpose of retaining existing customers should be grouped under the business Issue of Retention. Actions with the purpose of acquiring new customers belong to the business Issue of Acquisition.

Business Groups are used to organize customer actions into categories. For example, as part of the business Issue of Acquisition, you can create Groups for products like Credit Cards, Mortgages, or Personal Loans, with the intention of offering these to potential customers.

Use **Constraints** to specify outbound contact limits as well as to limit overexposure to a specific action or

group of actions.

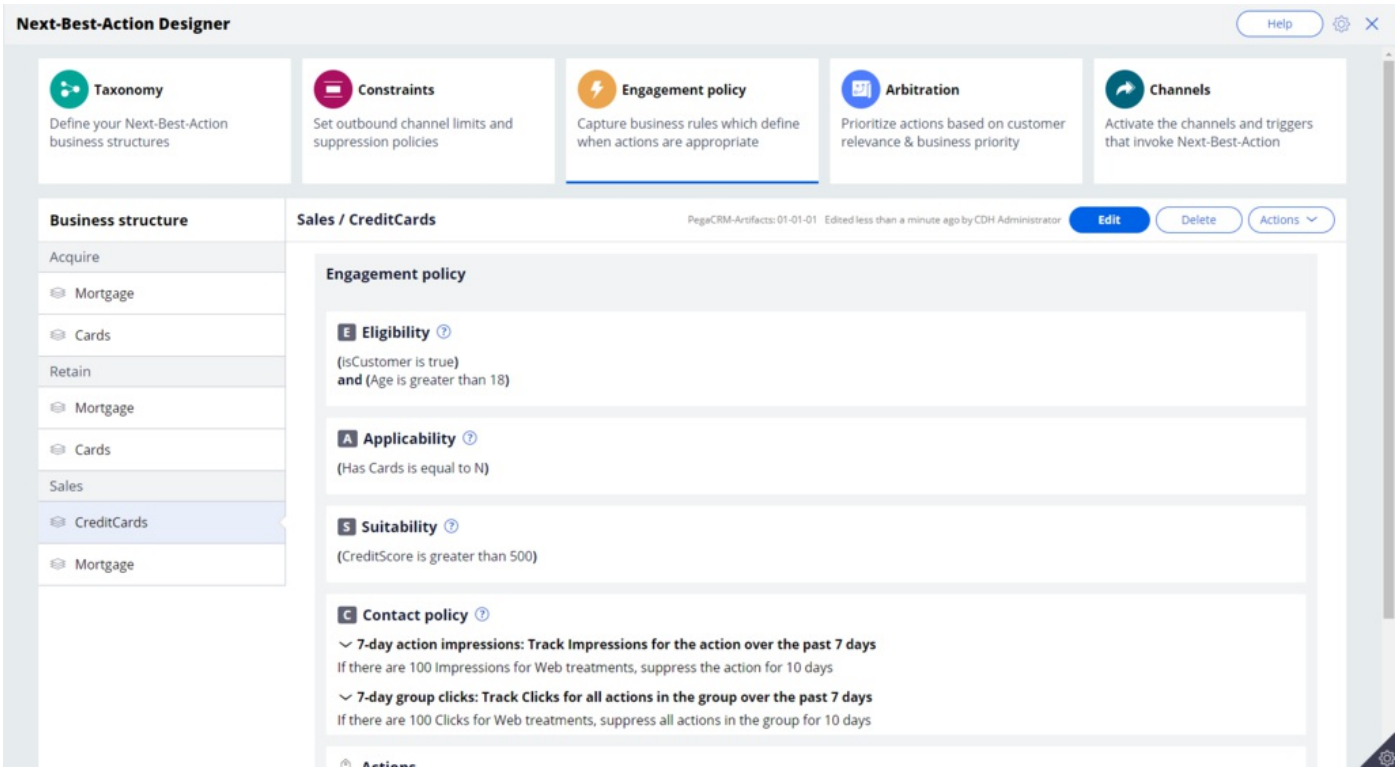


Customer contact limits allow you to limit the number of interactions that a customer can receive over a given period of time on a specific channel. For example, you can decide that you do not want your customers to receive more than two emails per week or two SMS messages per week.

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:

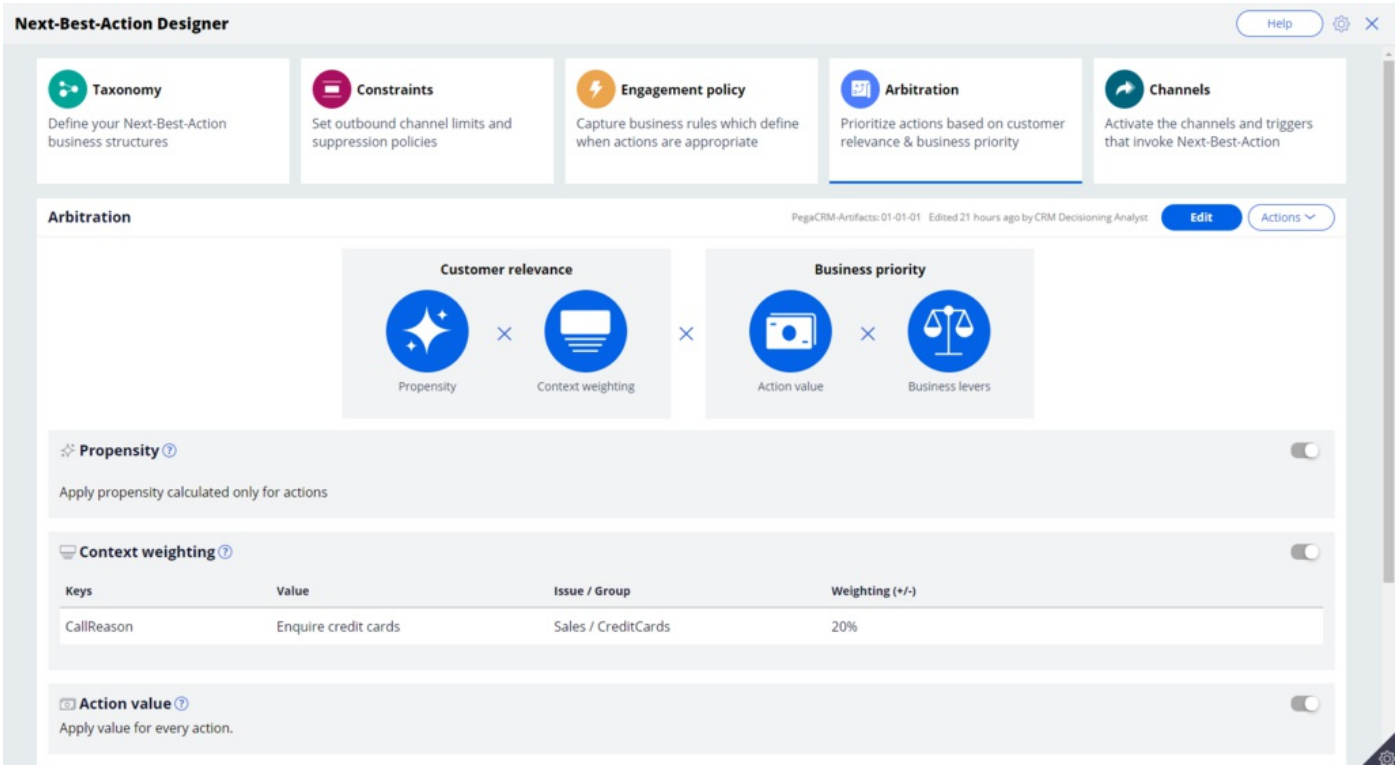
Eligibility determines whether or not a customer qualifies for an action or group of actions. For example, an action may only be available for customers over a specific age or living in a specific geographic location.

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 has been sent to customers. To activate Contact Policy rules created in the library on the Constraints tab, add them to the Engagement Policy tab.

Arbitration determines how the 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, Action value, and Business levers, each represented by numerical values. A simple formula is used to arrive at a prioritization value, which is used to select the top actions.

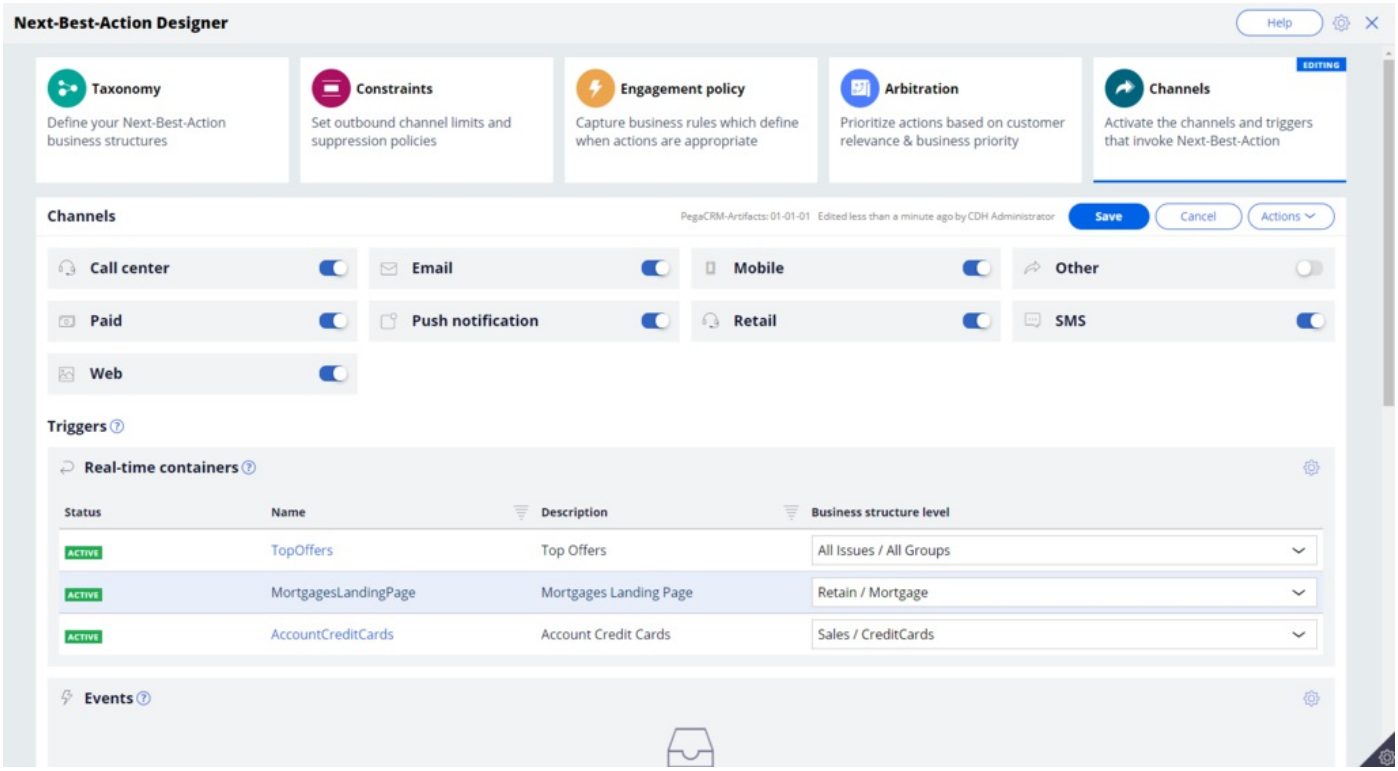
Propensity is the likelihood of a customer responding positively to an action. This is calculated by Artificial Intelligence (AI). For example, a click on an offer banner or an accept of an offer in the contact center are considered positive behaviors.

Real-time contextual data is an important part of making highly relevant recommendations**. Context weighting** allows you to assign 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 will increase, and the highest priority action will be to ensure that the relevant service is delivered to the customer.

Action 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.

Next-Best-Action Designer enables Next-Best-Actions to be delivered via inbound, outbound and paid channels.



These channels can be toggled on or off. If a channel is toggled off, the Next-Best-Actions will not be delivered to that channel.

An external real-time channel is any channel that presents actions selected by the Customer Decision Hub to a customer. These channels can include a website, or a call-center or 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 will be delivered back to the invoking channel. For example, when a real-time container called “Mortgages Landing Page” is configured, the website invokes this real-time container before loading the mortgage 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 rules that control which actions are offered to which customers
- Configuring how actions are prioritized
- Configuring when and where Next-Best-Action is triggered

Next-Best-Action Designer -- Thu, 07/23/2020 - 05:56
To get the full experience of this content, please visit https://academy.pega.com

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.

Video

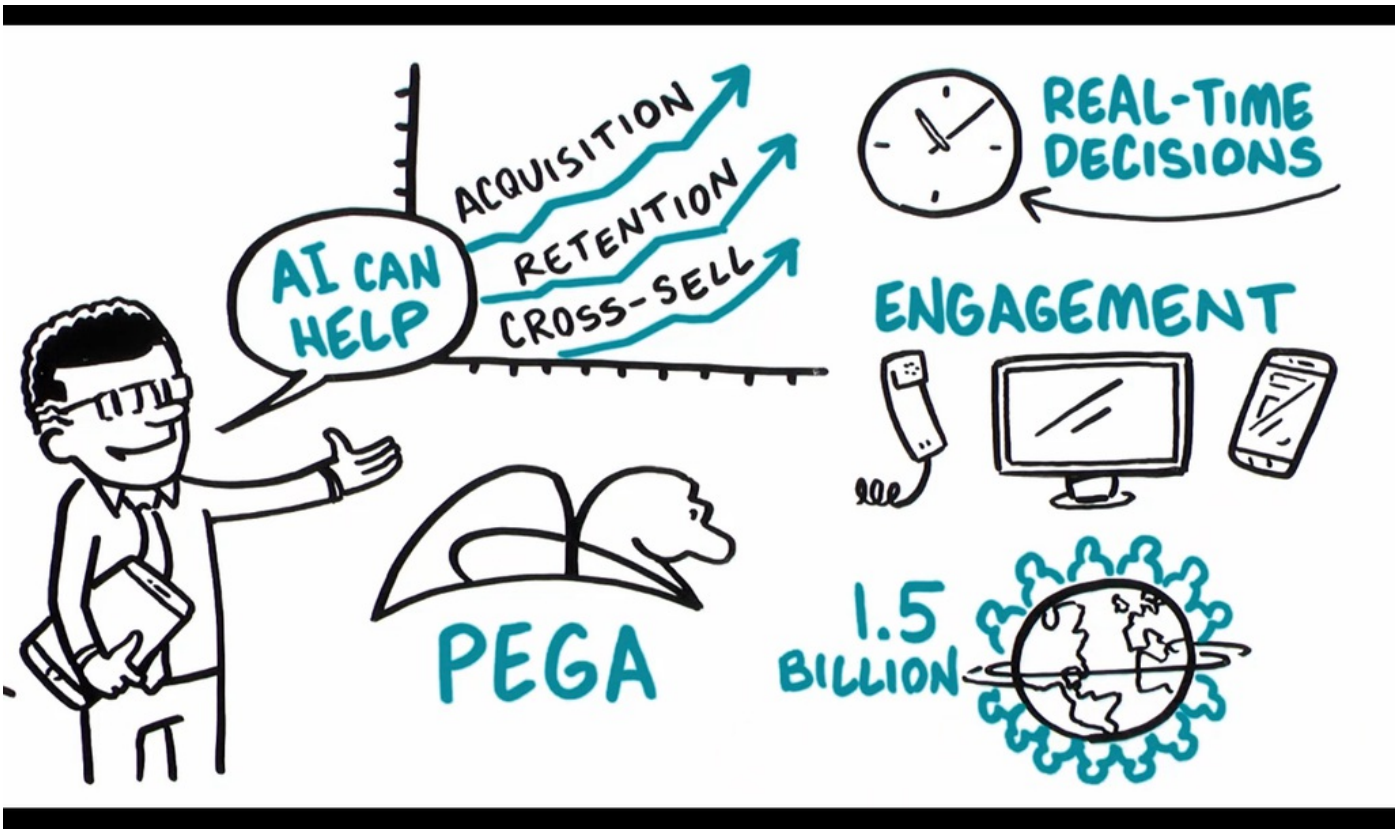
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

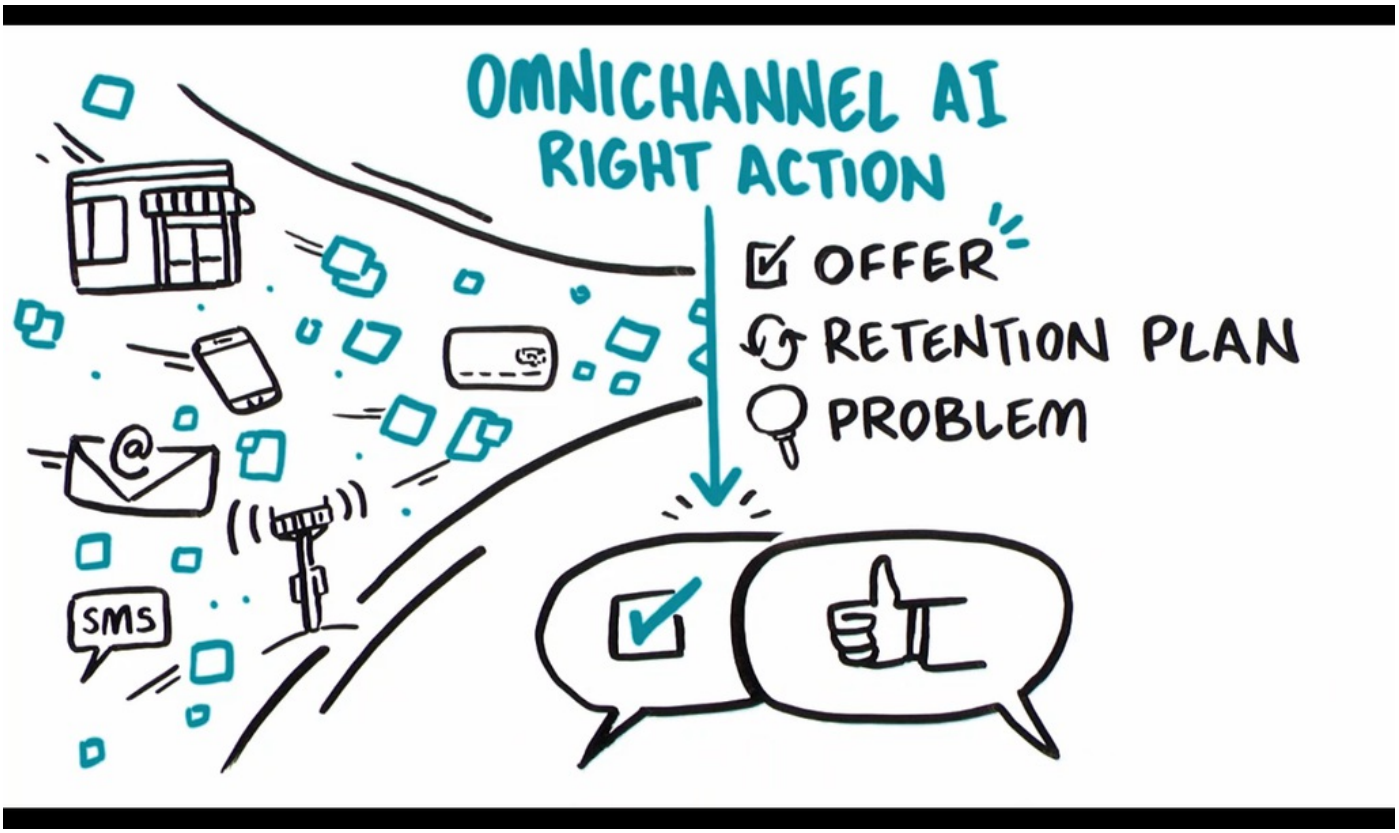
This is your customer. You want him to buy your products, use your services and have a great experience. And your competitors want the same thing. To compete, you have to take the right action at every customer touch, ensuring that each conversation delivers exactly the right message, offer and level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

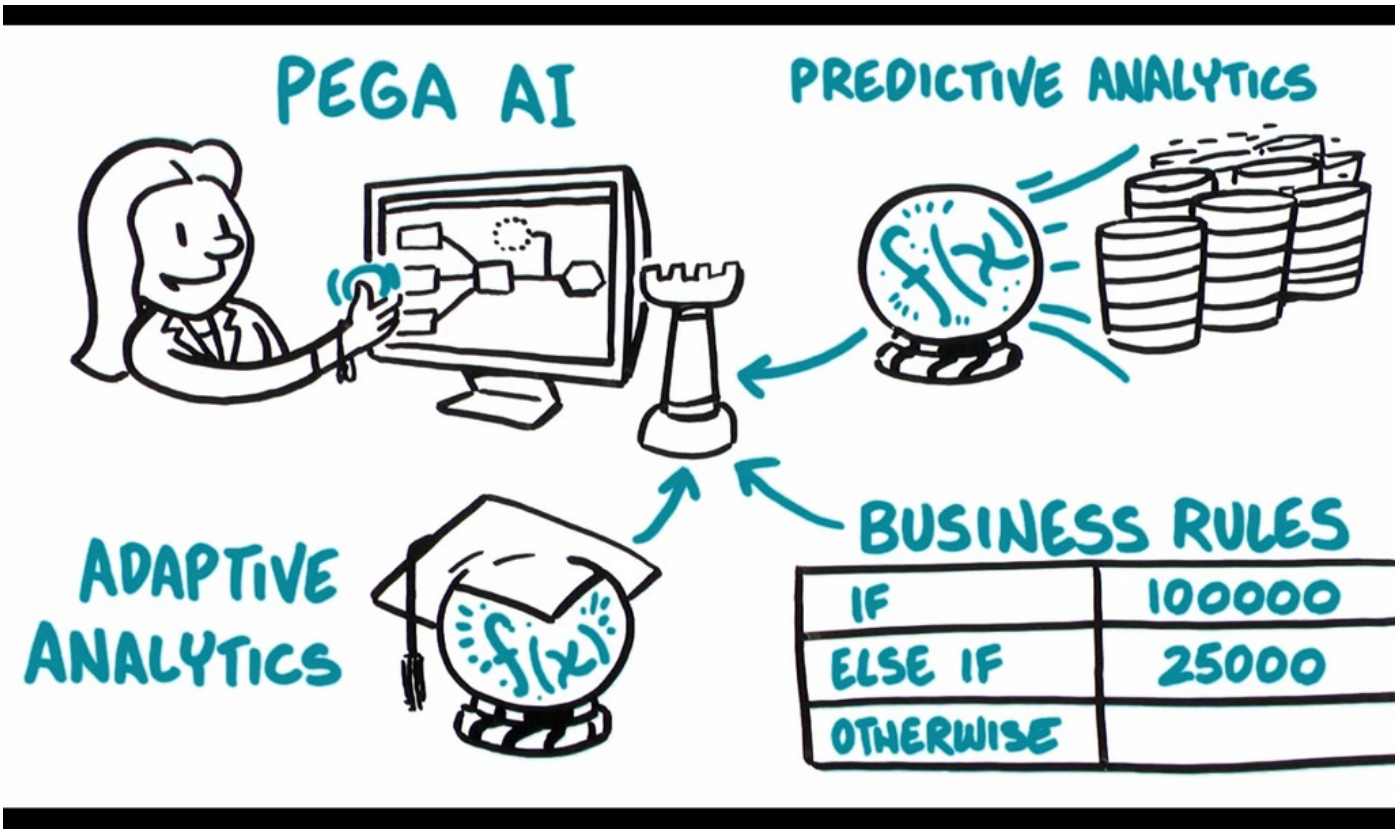


Pega’s omni-channel AI delivers the right action at every customer touch by crunching millions of data points in real-time. Make an offer, initiate a retention plan, predict a problem before it happens. Every decision generates the next-best-action for your customer, and your business.

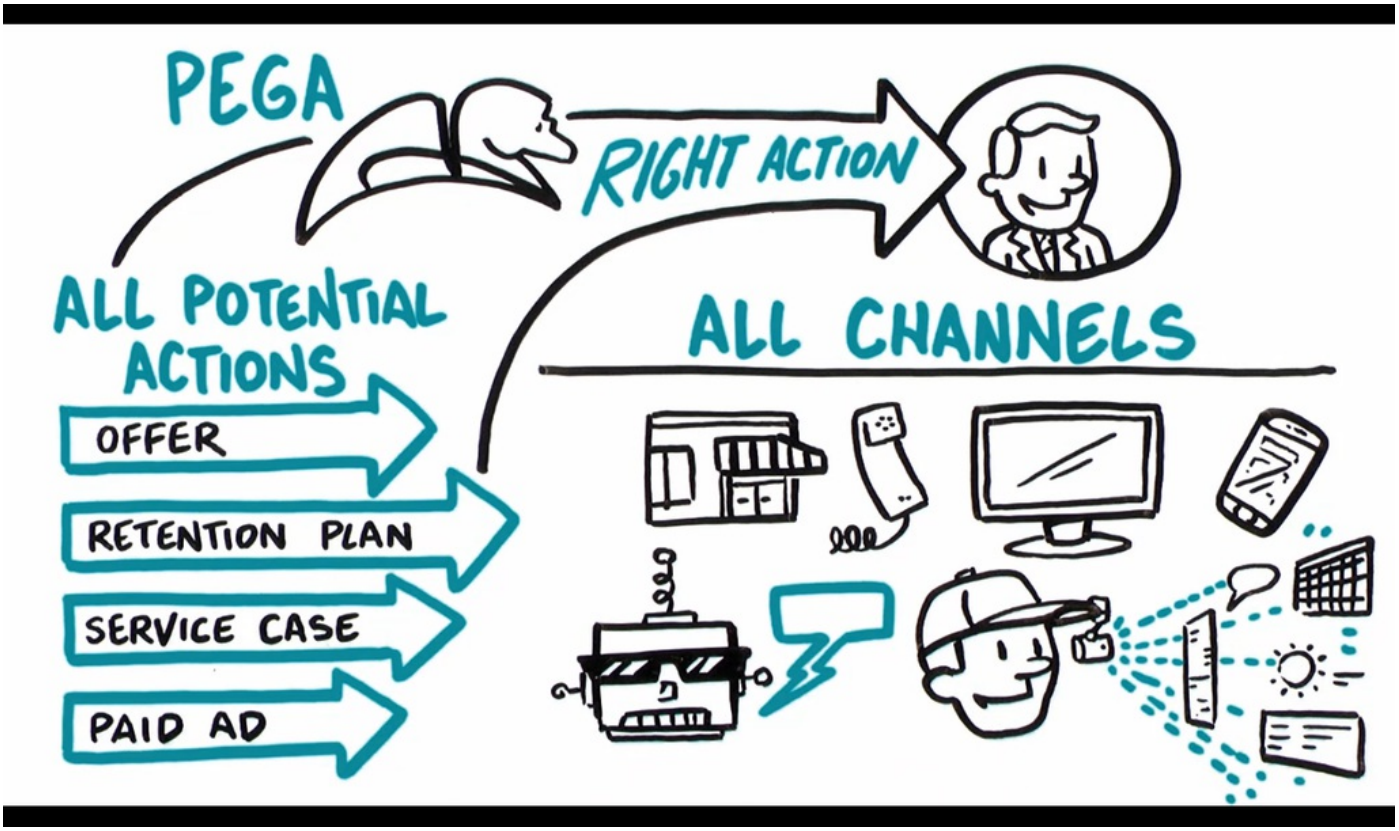


Pega’s AI is built for business people, not scientists or developers. They design visual decision strategies that combine predictive analytics, algorithms developed through mining large sets of data, adaptive analytics, machine-learning algorithms that improve with each interaction, and traditional business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

that allow users to prioritize and arbitrate between decisions.

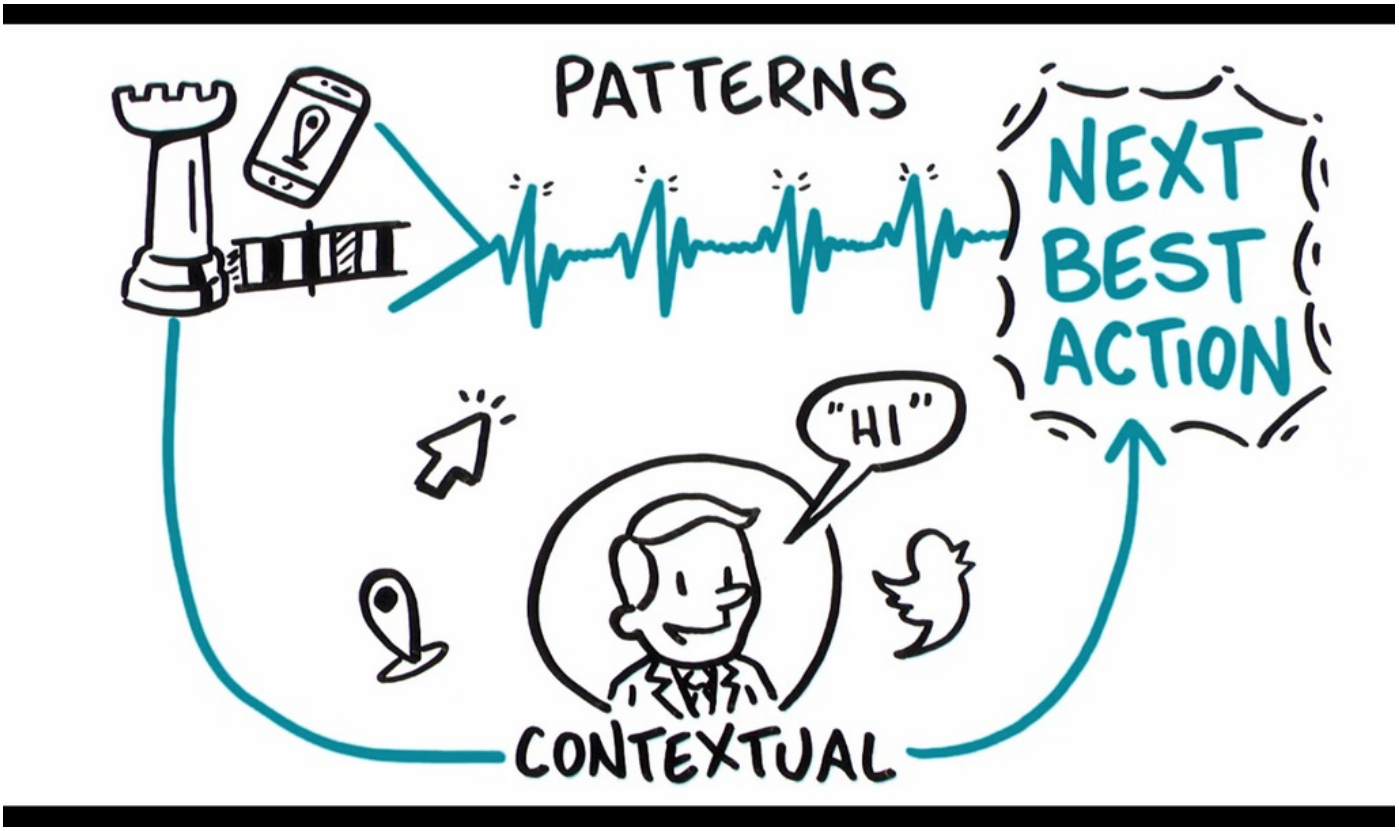


Pega uses the strategy to look across all the potential actions you may take with a customer, make an offer, initiate a retention plan, open a service case, place an ad, and ensure exactly the right action is taken at every interaction and it works across all channels to provide a consistent experience 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.

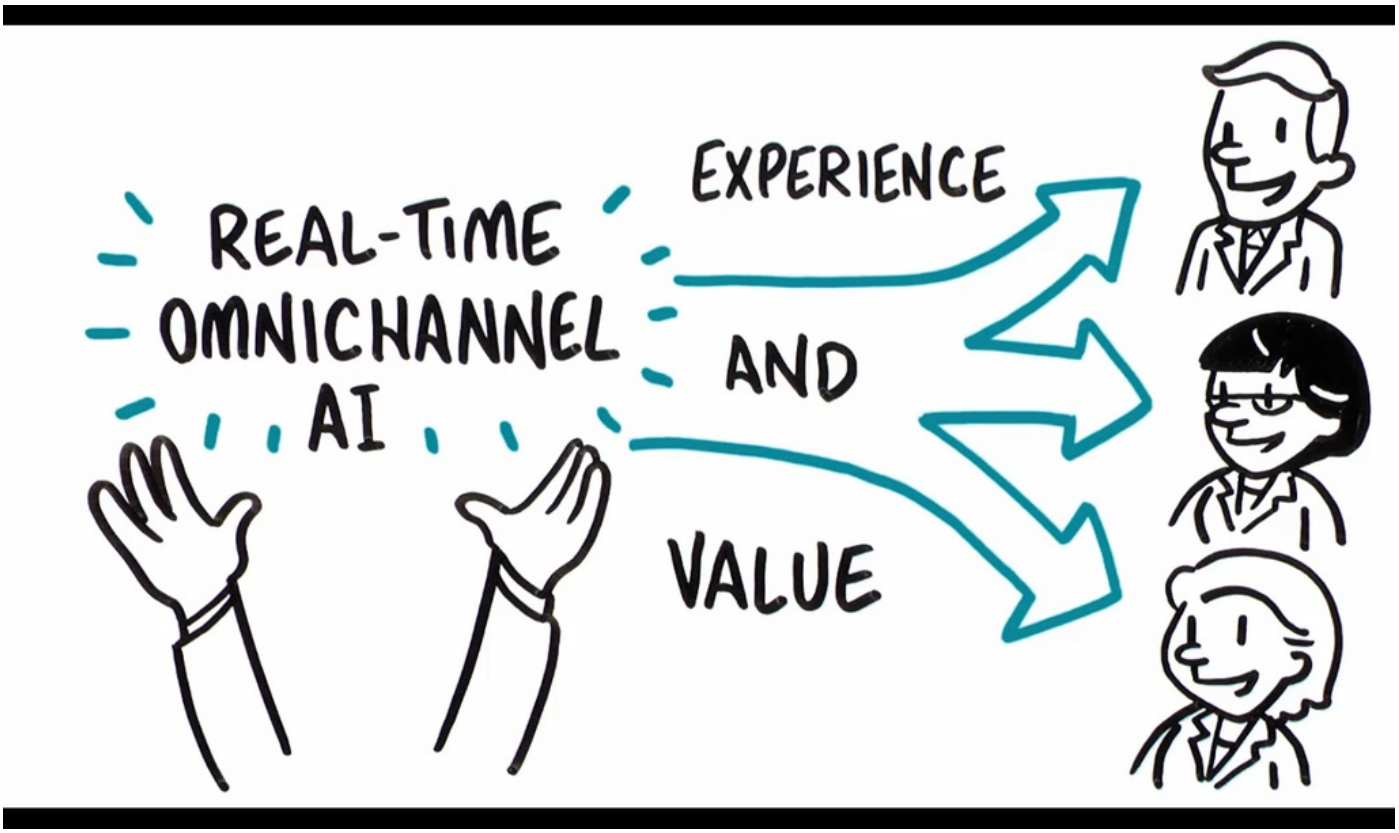


And Pega connects to streams like mobile locations or network events to detect patterns and drive the Next Best Action proactively. And strategies are completely contextual. Any change in the customer’s context —

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.



One-to-one Customer Engagement paradigm

Introduction

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 have to be able to perform the right action in the right channel at the right moment for each customer. We call this capability, “1-to-1 Customer Engagement”.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

In this video, learn about the 1-to-1 Customer Engagement paradigm and how the principles of Next-Best-Action are implemented using the Pega Customer Decision Hub™.

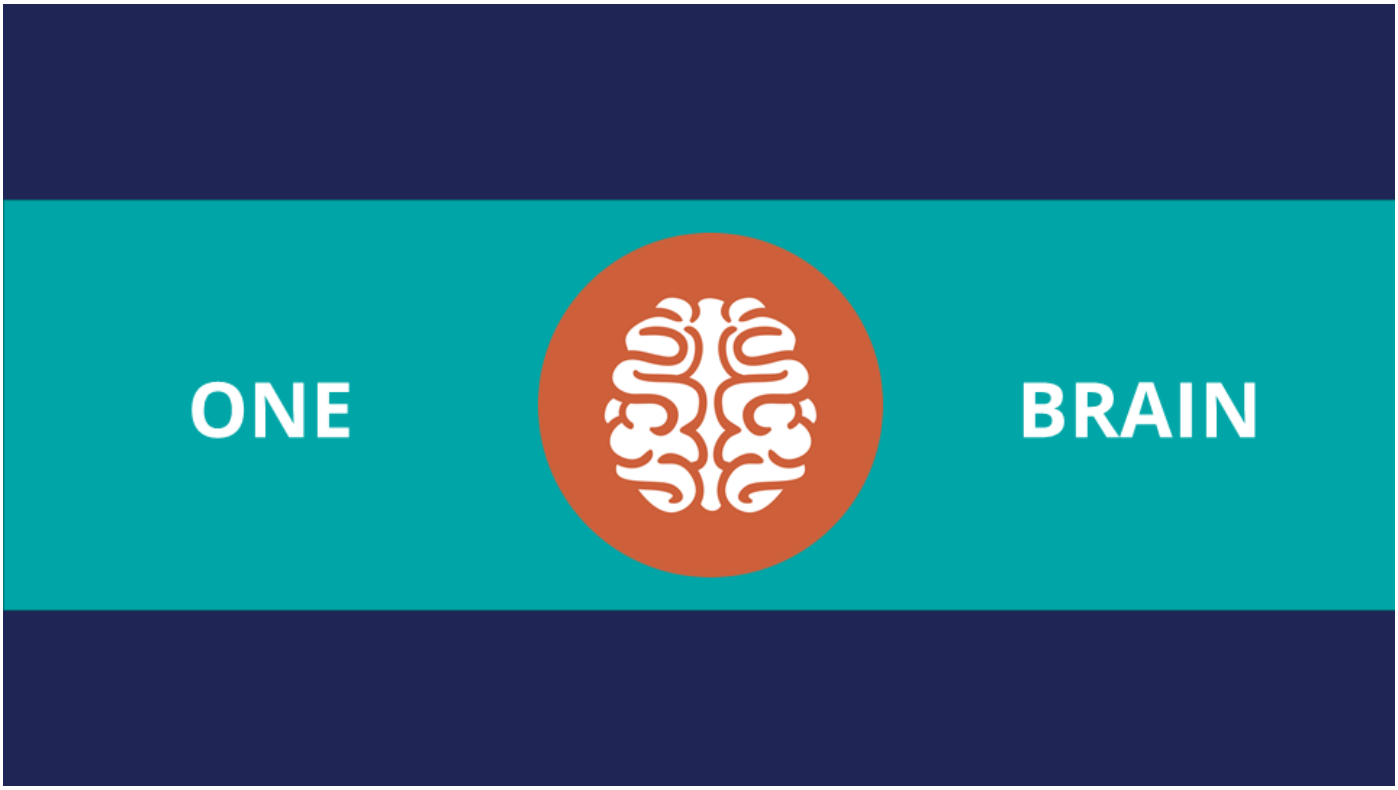
Customers are more empowered than ever before. As a result, they have very high expectations of the experiences they receive from their service providers. Their experiences must make sense within the context of their lives. This means they must be meaningful, consistent, and personalized across every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

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 have to be able to perform the right action in the right channel at the right moment for each customer.

We call this capability, “1-to-1 Customer Engagement”.

1-to-1 Customer Engagement enables companies to 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 1-to-1 Customer Engagement is an idea that’s simple to conceive, but very difficult to execute: one centralized brain.



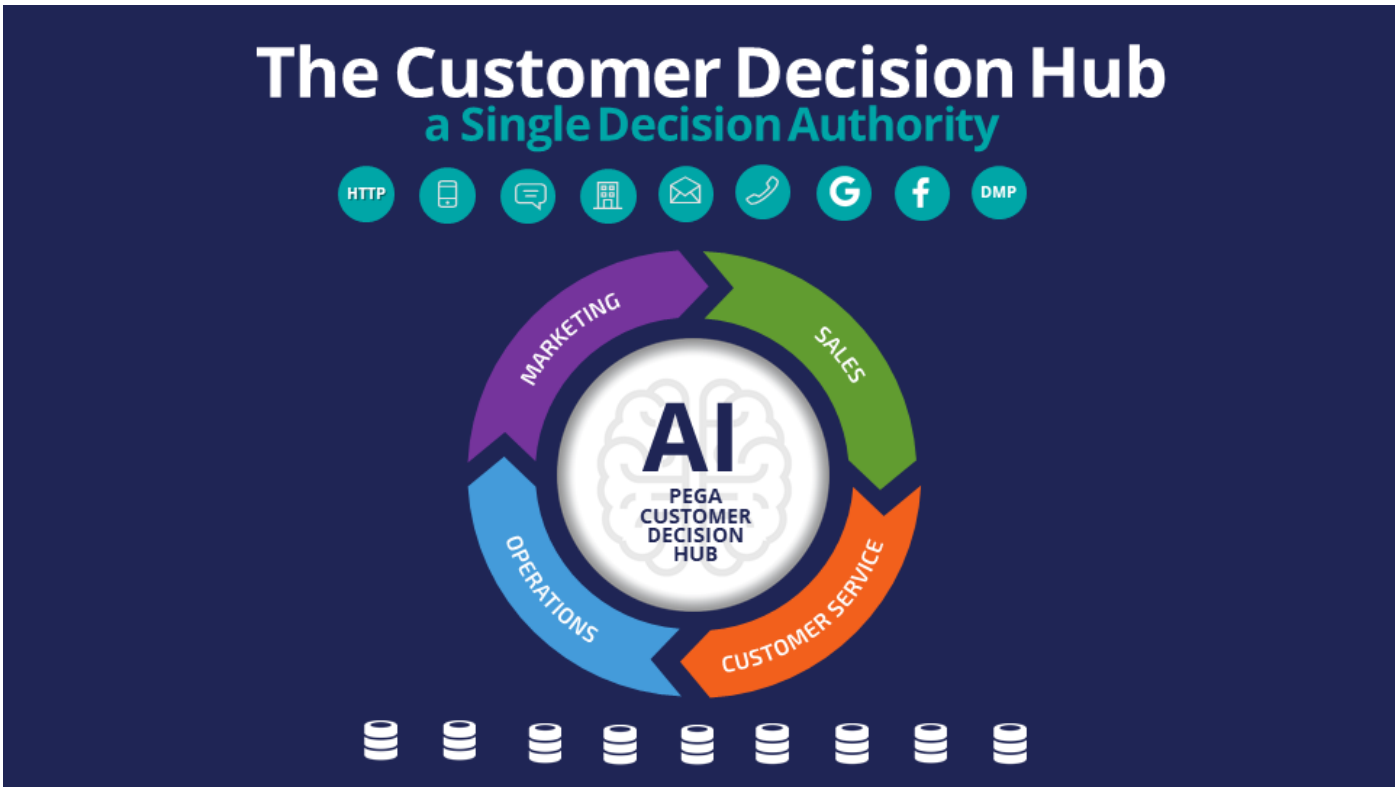
In other words, one piece of intelligence that acts as a single decision authority across your application

ecosystem.

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

In Pega Marketing™, this centralized brain is called the Pega Customer Decision Hub, and it leverages AI to enable 1-to-1 Customer Engagement.

In Pega Infinity™, the 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.



Data from every customer engagement across the enterprise is collected by the Brain and used to make predictions and decisions about every interaction in every channel.

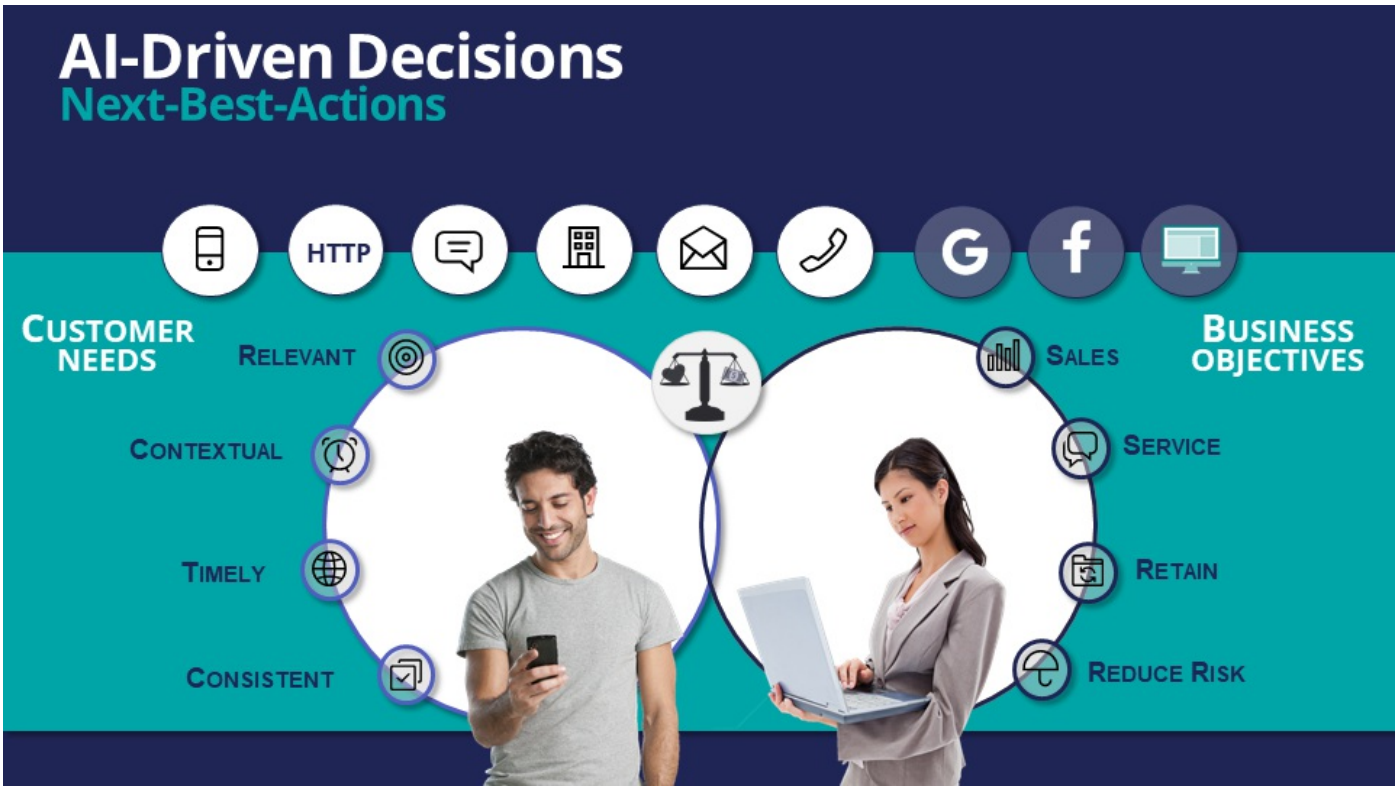
Continuous learning and decision-making are the foundation of a 1-to-1 Customer Engagement solution.

The 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.

In the milliseconds before interacting with a customer, the Customer Decision Hub processes thousands of predictive and adaptive models to determine customer needs, considering the customer's immediate context to ensure the Next-Best-Action is relevant, timely, and contextual. These models can be propensity, risk, or churn models.

Next, the decision strategy considers business rules and matches those with the customer's context and higher-level business goals.

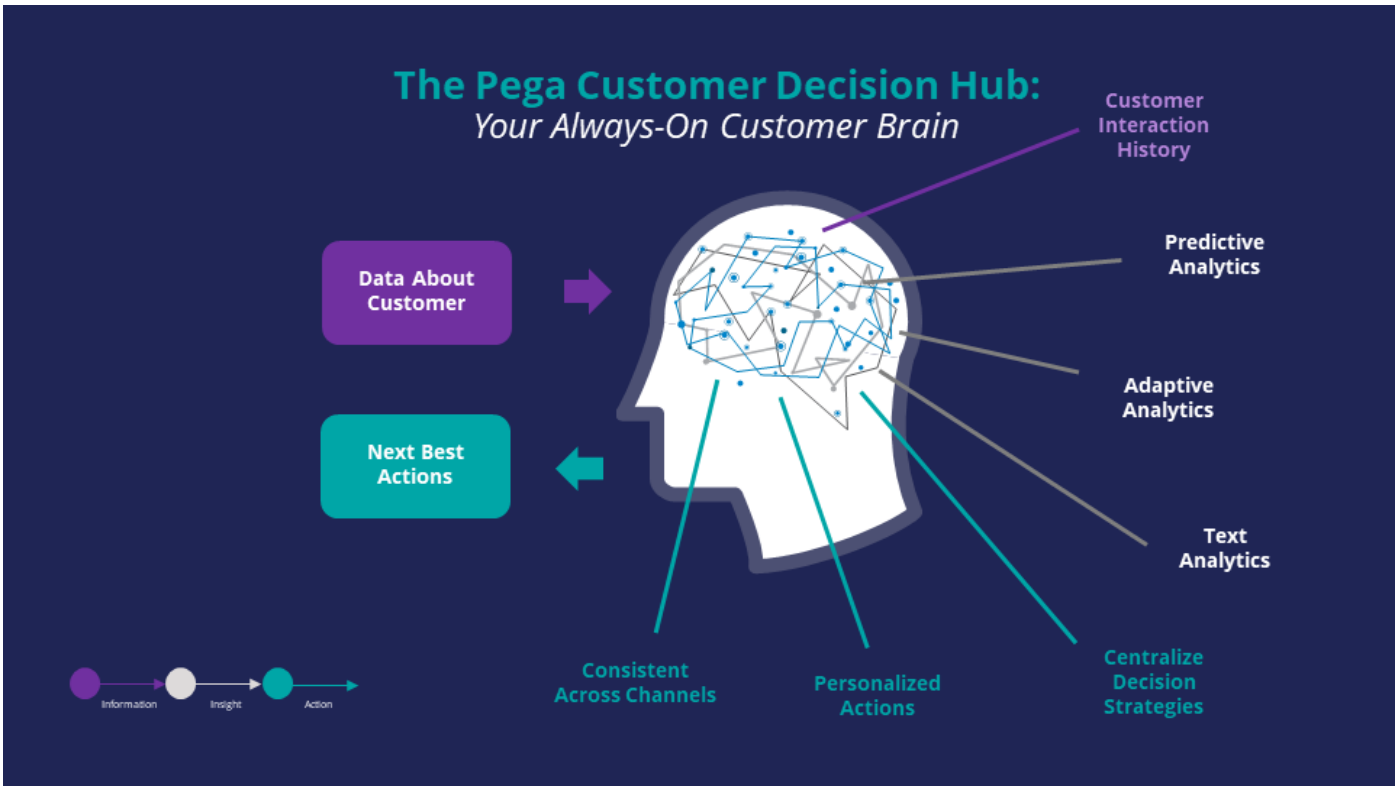


All of this information is used by the Next-Best-Action decision strategy to evaluate every potential action that could be taken with a particular customer in a given situation. The decision strategy then recommends the best way to interact with the customer to achieve the optimal result.

Using the Next-Best-Action approach, the Customer Decision Hub is able to identify the best moments for making a sale, providing a service, making a retention offer, or doing nothing at all (e.g. if nothing is relevant enough to warrant the customer's attention). Next-Best-Action is even able to select which offers are most likely to be accepted by the customer in a sales or retention situation. Next-Best-Action decisions are distributed, in real-time, to each of your real-time owned channels, such as web, mobile, and contact center. Through Pega Customer Decision Hub, Next-Best-Actions can also be distributed to real-time paid channels such as Google, YouTube, Facebook, LinkedIn and Instagram. Pega Marketing also integrates with non-real time outbound channels such as data management platforms (DMPs) and email.

Once the Next-Best-Actions are distributed and customer responses have been received by the Brain, the whole process begins again, and new Next-Best-Actions are distributed within milliseconds. Every outbound channel, including a data management platform, is dynamically updated with the Next-Best-Action to ensure consistency and an optimized customer experience no matter which channel the customer interacts with.

In summary, the Pega 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 leverages advanced AI techniques to make predictions.

And it uses decision strategies (which combine traditional business rules with predictive, adaptive and text analytics), to deliver consistent and personalized Next-Best-Actions across all channels.

One-to-one Customer Engagement paradigm -- Thu, 07/23/2020 - 05:55
To get the full experience of this content, please visit https://academy.pega.com

Action hierarchy

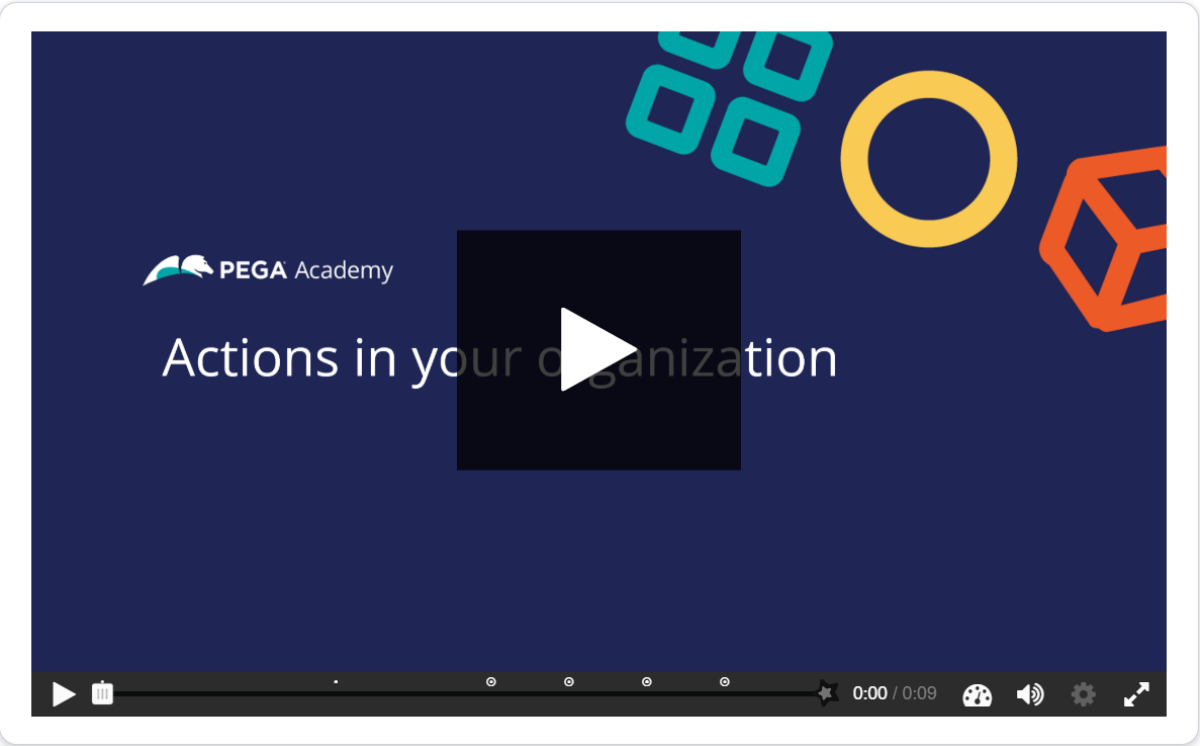
Customer action introduction

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.

Actions in your organization

Click the **Play** icon to learn about customer actions and how to organize them.

Full-screen mode is available in the lower right of the interaction.

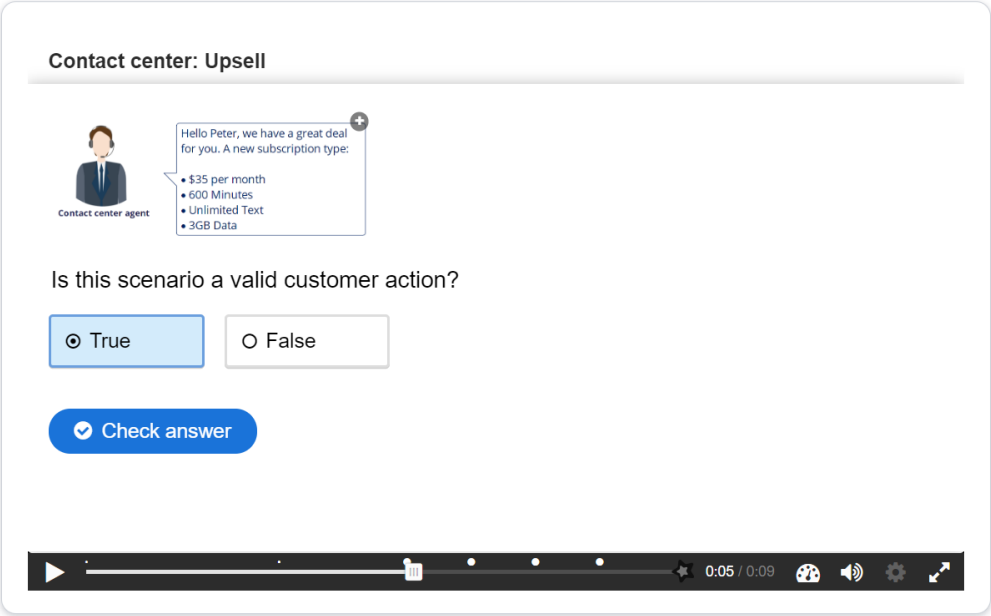


In Pega Customer Decision Hub, customer actions are offerings that list the details of your products. Next best action uses these properties to determine the priority of each offering for each customer and provide you with the next best action to take.

To learn more about the different type of customer actions, consider the following scenarios in various channels: contact center, mobile, and web.

Click **Continue** to see how actions are used in various scenarios.

Scenario 1: Contact center: up-sell



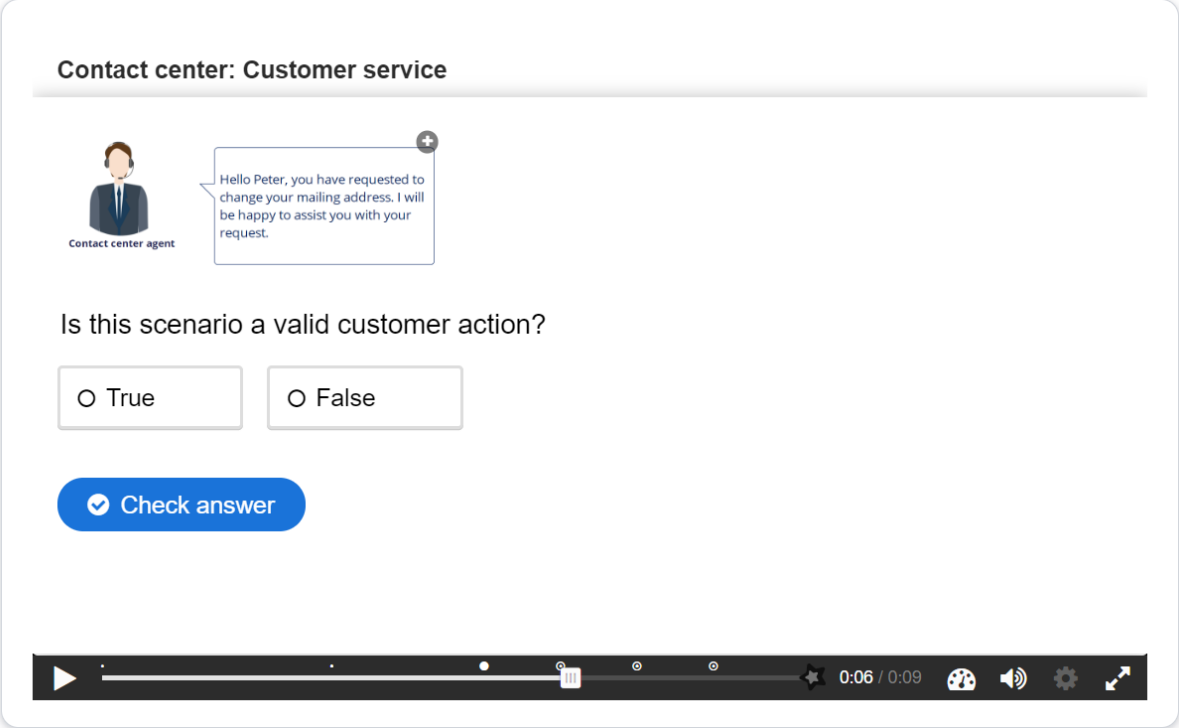
Is this scenario a valid customer action?

- True
- False

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

Incorrect feedback: **An upsell offer made in a contact center is a valid customer action.**

Scenario 2: Contact center: customer service



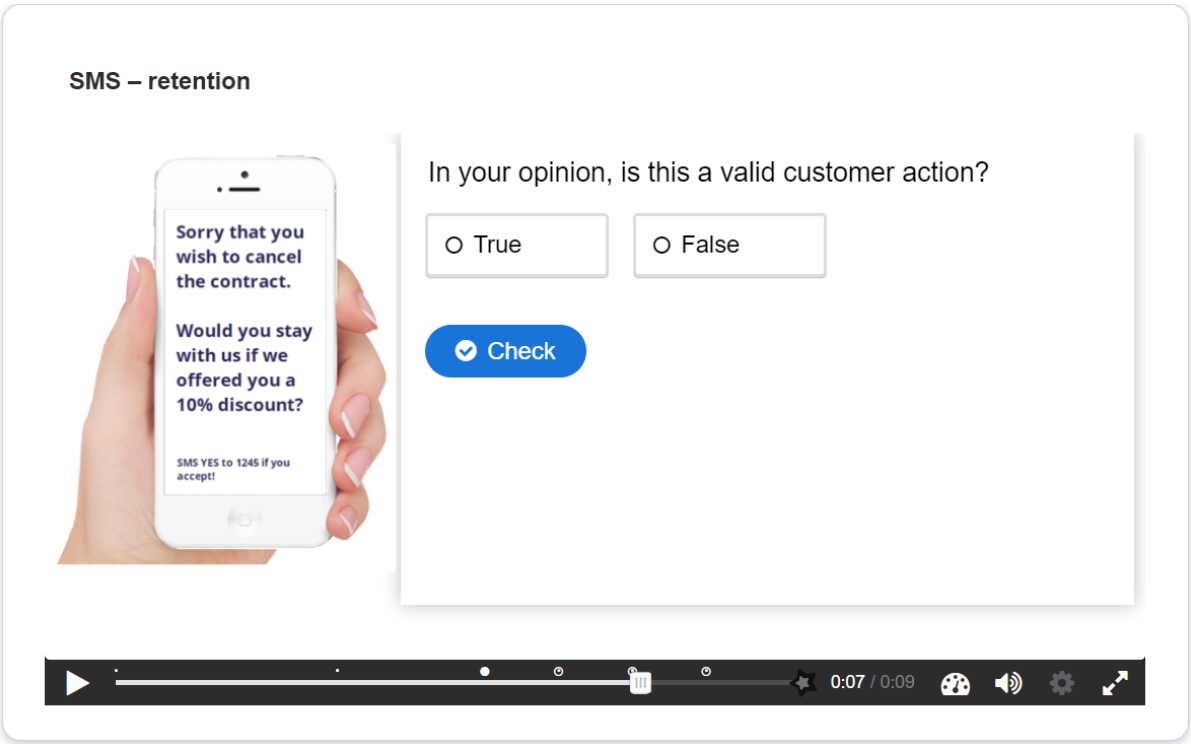
Is this scenario a valid customer action?

- True
- False

Correct feedback: **This scenario is a service message recommended by the Pega Customer Decision Hub and is a valid customer action.**

Incorrect feedback: **A service message recommended by the Pega Customer Decision Hub is a valid customer action.**

Scenario 3: SMS – retention



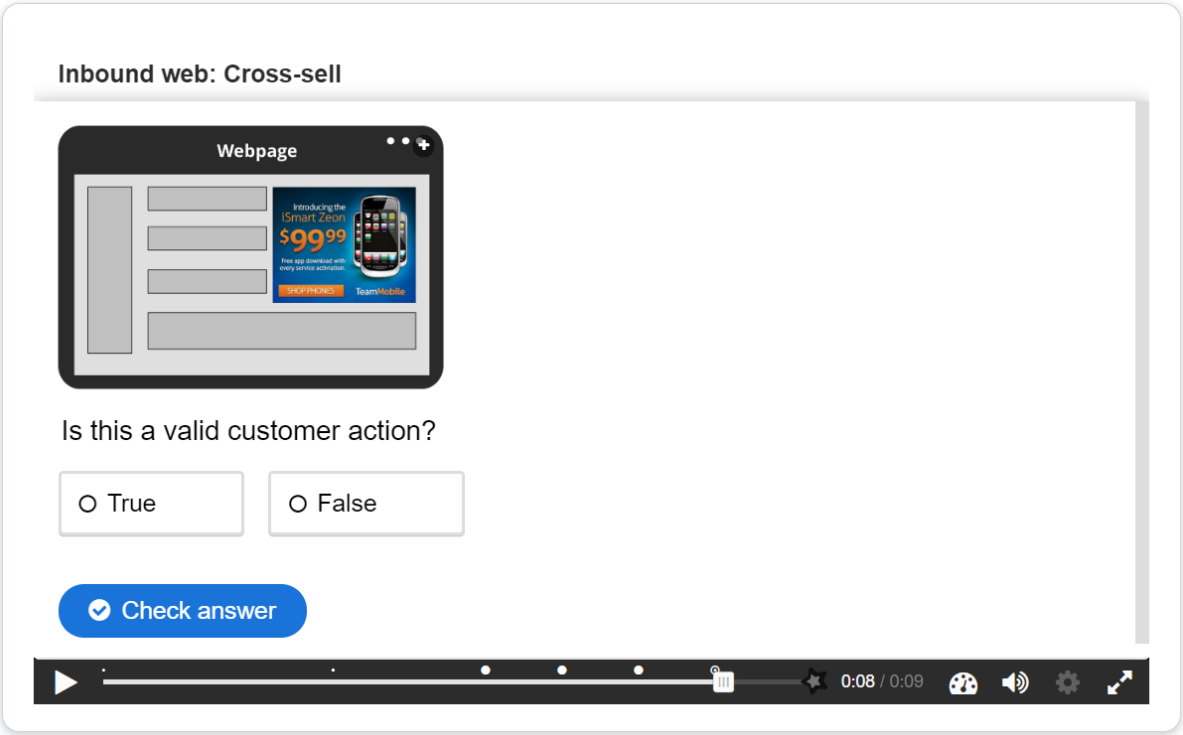
In your opinion, is this a valid customer action?

- True
- False

Correct feedback: **This scenario is a retention offer made on the mobile channel to retain a customer and is a valid customer action.**

Incorrect feedback: **A retention offer made on the mobile channel to retain a customer is a valid customer action.**

Scenario 4: Inbound-web: cross-sell



Is this a valid customer action?

- True
- False

Correct feedback: **This scenario is a banner advertisement on a website and is a valid customer action.**

Incorrect feedback: **A banner advertisement on a website is a valid customer action.**

Customer action properties

Every customer action has properties that define its characteristics.

In the following image, click the **+** icons to identify the valid properties of customer actions.

Image is a valid property and represents the image that is used for the banner.

Price is a numeric property that represents the price of the customer action.

Benefit is a valid property that represents the value proposition to the customer.

Short title is a valid property that represents a short description of the customer action.

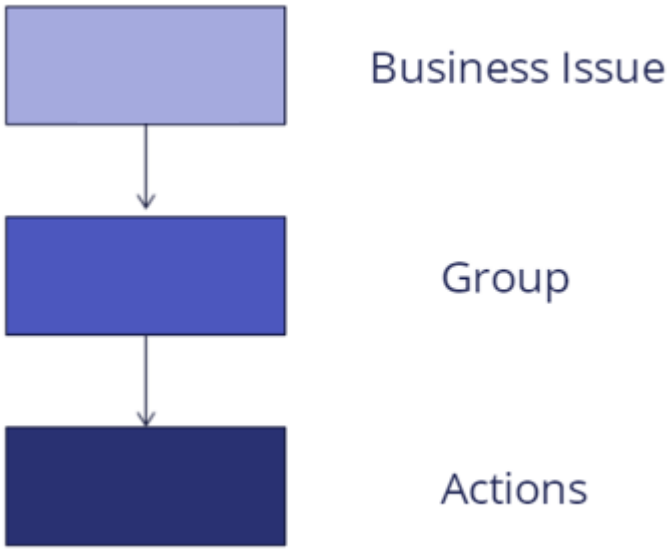
Call to action is not a valid property. This button is a call to action for people who want to purchase the product.

Company name is not necessarily a valid property, as this text represents the company name.

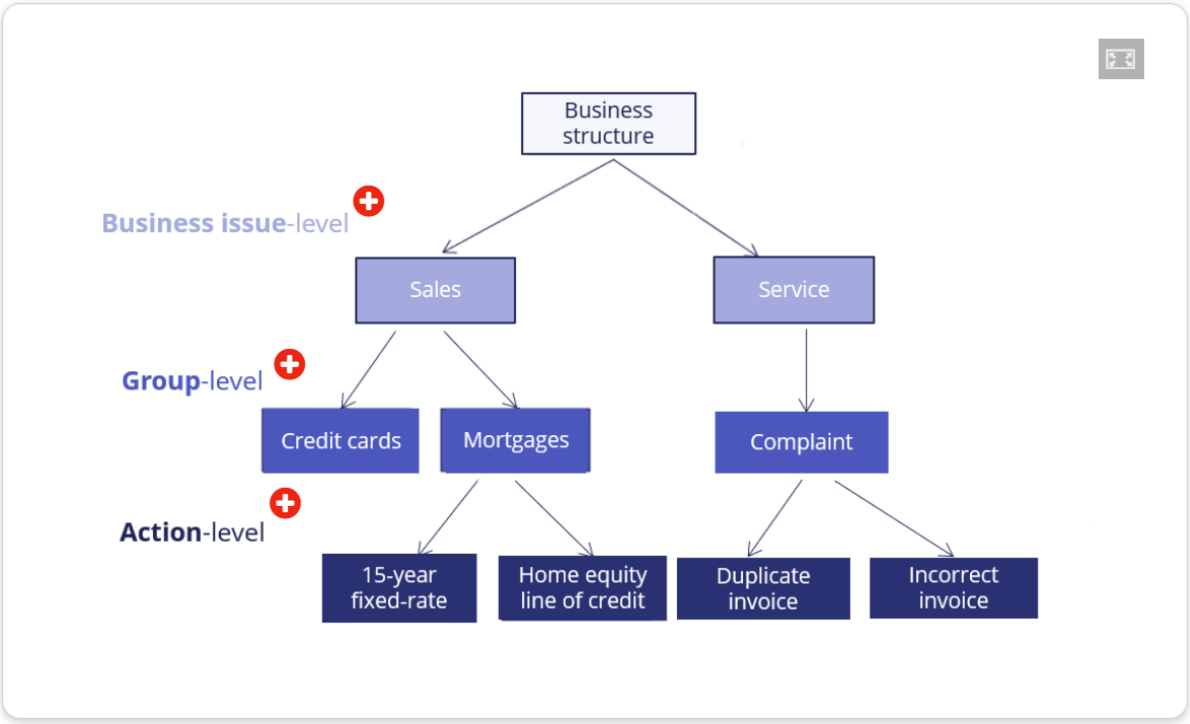
Business hierarchy

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.



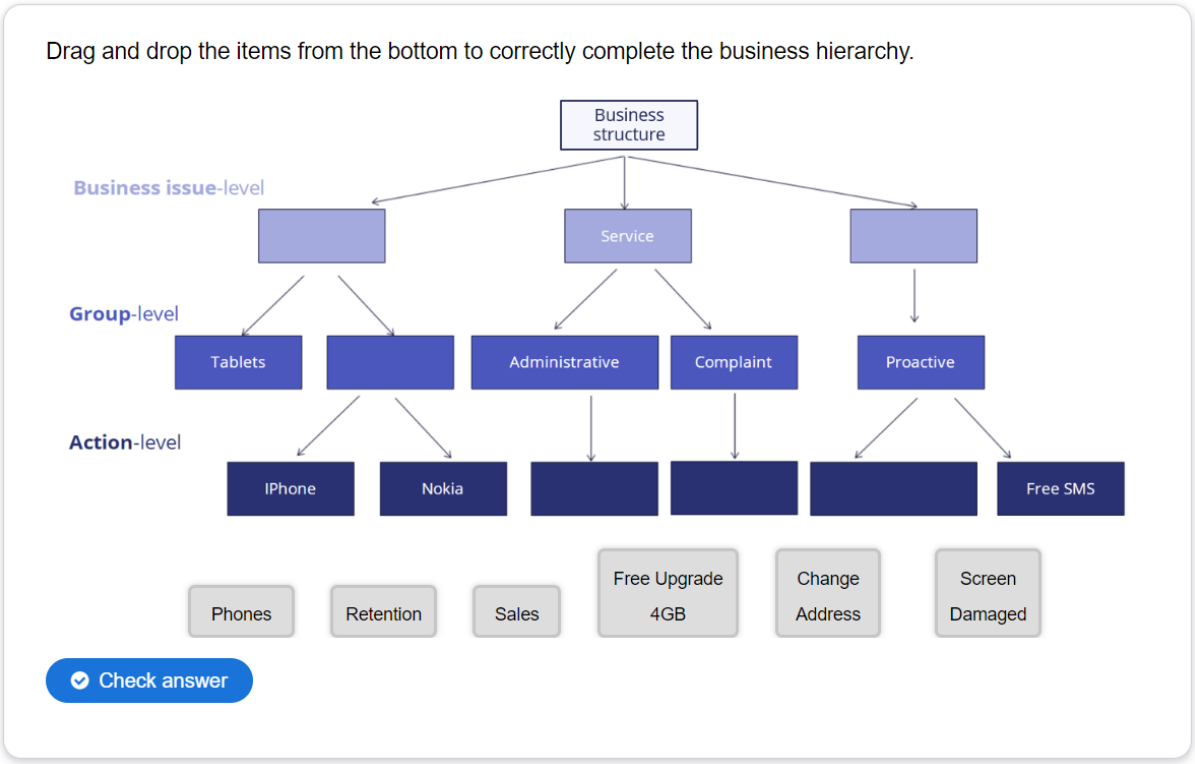
In the following image, click the **+** icons to learn more about the three-level hierarchy. Then, check your knowledge with the following interaction.



Business issue: In Pega Customer Decision Hub, a **business issue** represents the business area for which a customer action is applicable (for example, **Sales**, **Retention**, and **Service**).

Group: In Pega Customer Decision Hub, a **group** organizes actions into logical buckets (for example, **Credit Cards** or **Mortgages**).

Actions: In Pega Customer Decision Hub, an offering is referred to as an **action** (for example, **Reward card** or **30-year fixed rate)**. Each action that you prepare for your customers is associated with an issue and a group.



Drag and drop the items from the bottom to correctly complete the business hierarchy.

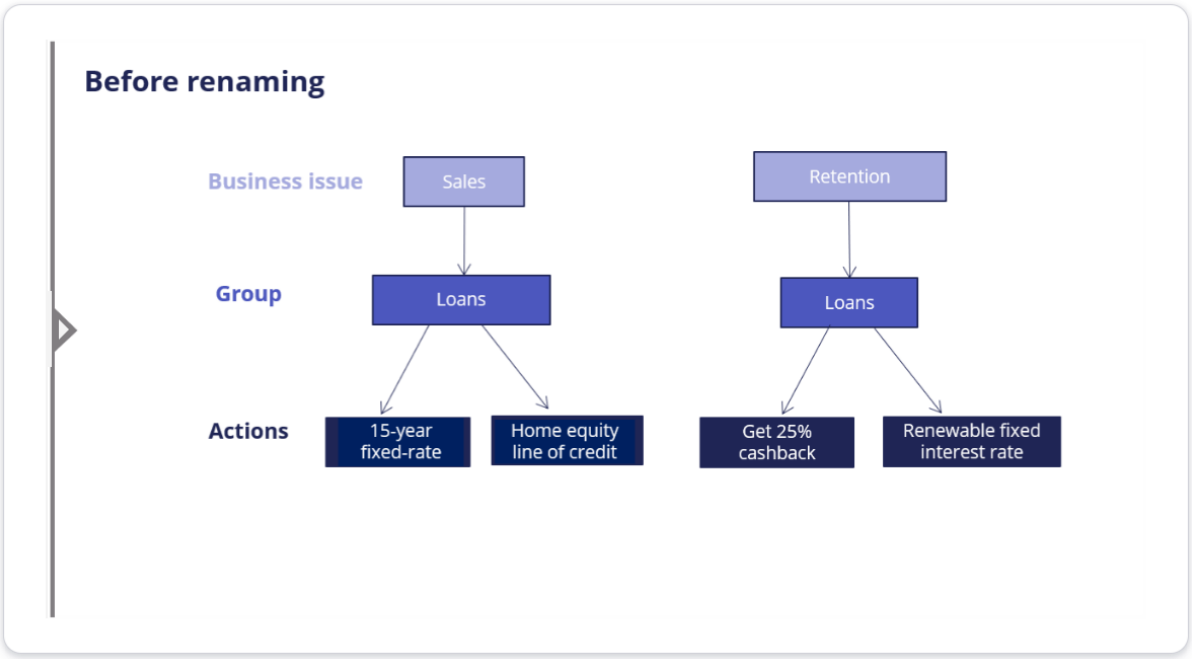
Correct feedback: **That's right! You selected the correct response.**

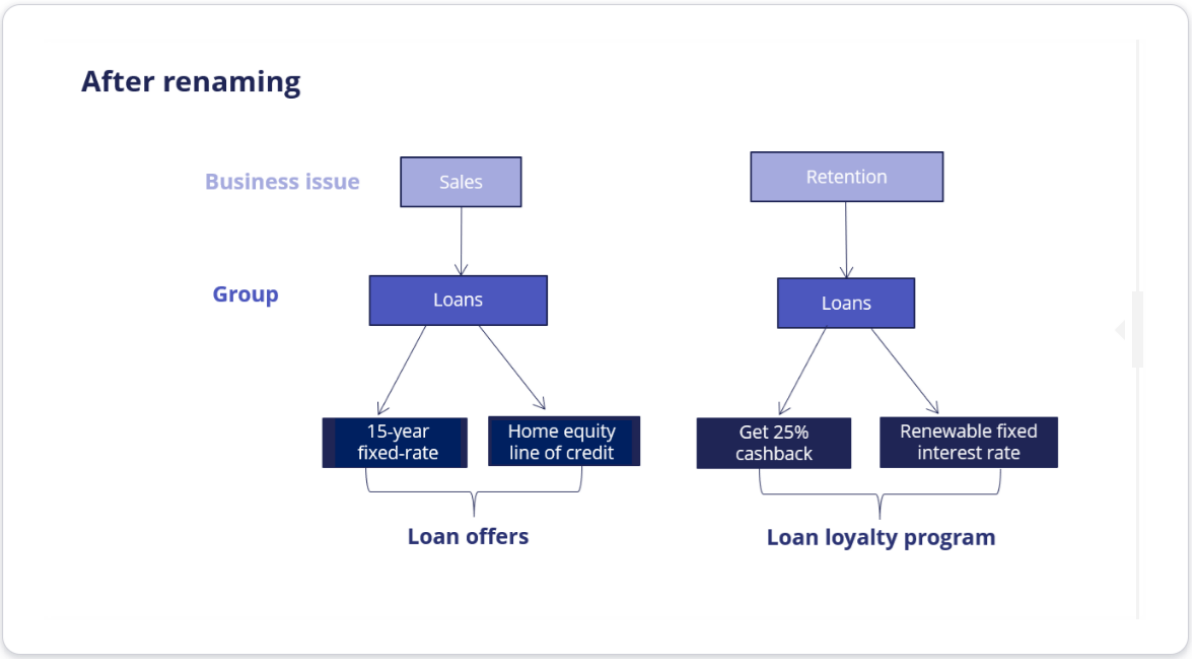
Incorrect feedback: **You did not select the correct response. Please try again.**

Customer actions and renaming

Pega Customer Decision Hub uses the default term actions to refer to messages delivered during customer interaction. If needed, you can change the term actions to a more appropriate term. For example, actions under a Loans group (Sales issue) can be changed to Loan offers whereas actions under a Loans group (Retention issue) can be changed to Loan loyalty program.

In the center of the following image, slide the vertical line to view a sample hierarchy with renamed actions. Then, check your knowledge with the following interaction.





Set 1: A customer action is any banner ad, retention offer, or service message.

Customer actions are properties that define their characteristics.

Set 2: A customer action can be presented on any customer interaction channel such as Web, Mobile, Contact center, or Social.

A customer action can be presented only on Web.

Set 3: A customer action is organized into a hierarchical structure. The default hierarchy is: Business issue -> Group -> Action.

A customer action is organized into a hierarchical structure. The default hierarchy is: Business issue -> Action -> Group.

Choose the correct statement.

Correct feedback: **Well done!**

Incorrect feedback: **We recommend you go through the material once more.**

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.

Video

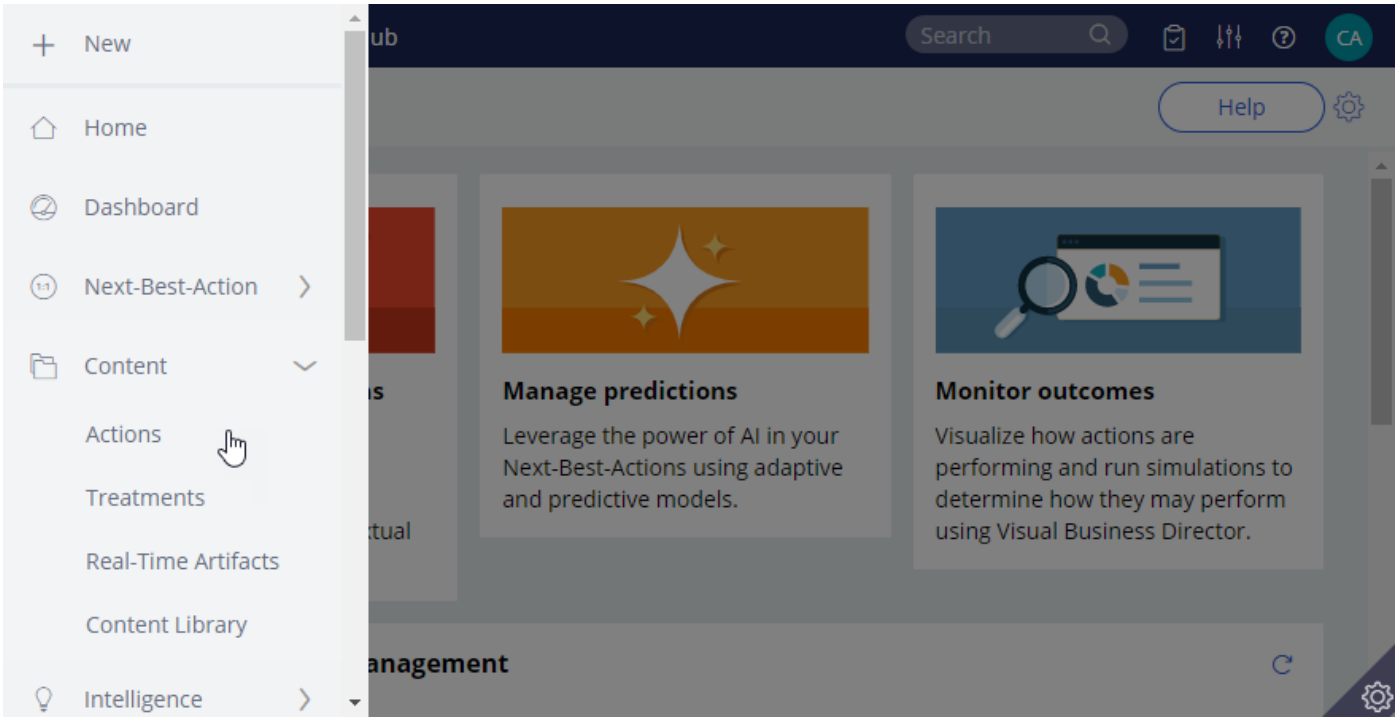
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

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

U+, a retail bank, recently introduced several new credit card offers that they would like to display to their customers. 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.

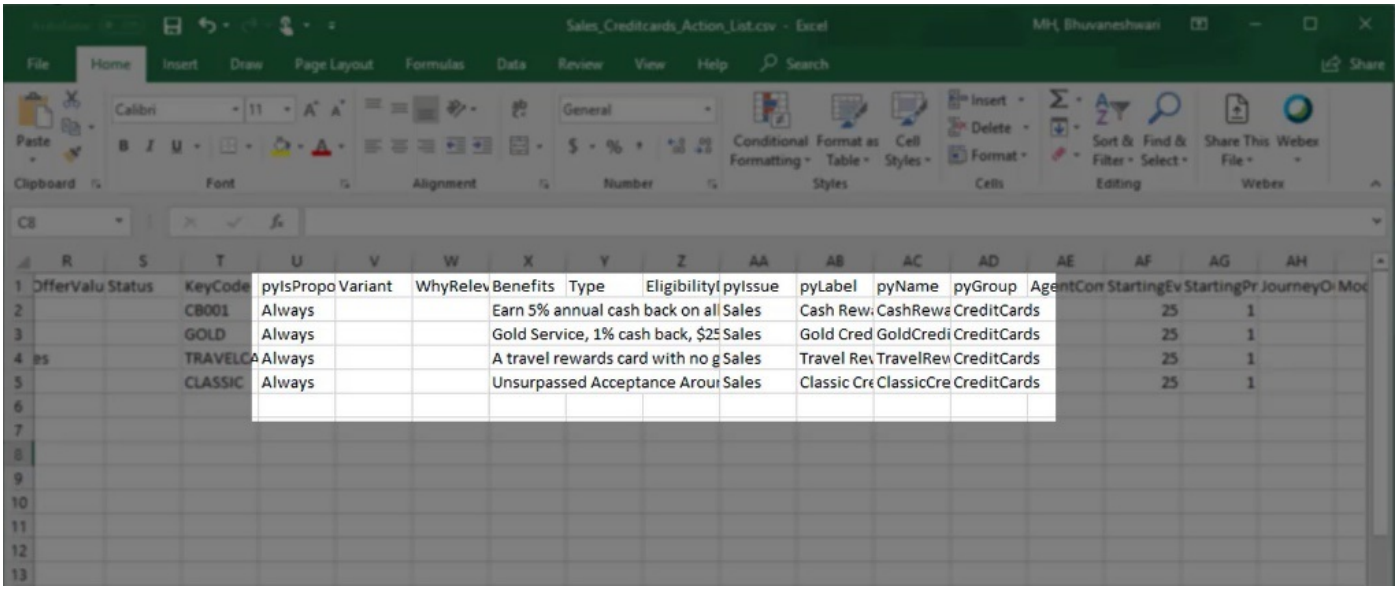
The attributes of an action distinguish it from other actions. Attributes are used by the Customer Decision Hub to select the right action for a customer. Fill in the attributes relevant to this action. For example, provide a **Description** that explains its purpose and **Benefits** that describe how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



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.

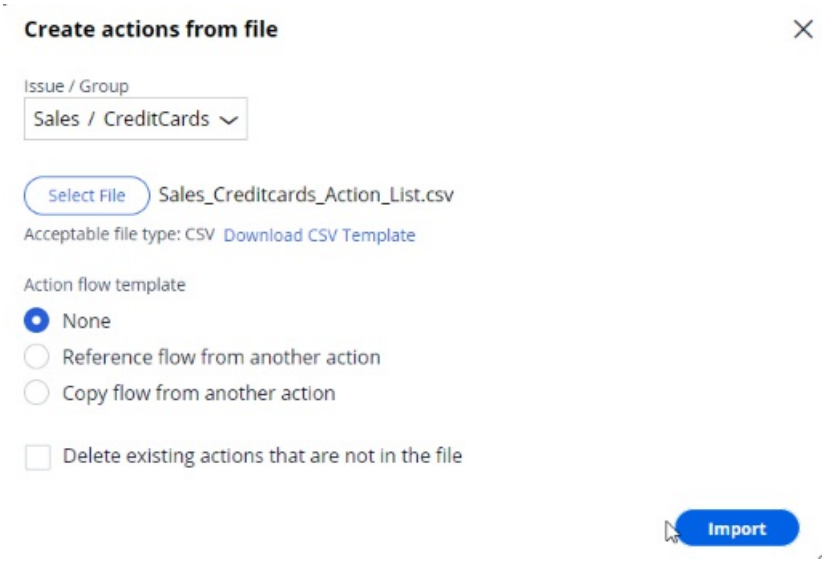
pyLabel corresponds to a short description of the Action.

pyName corresponds to the Action name.

pyGroup is the Group to which the Action belongs.

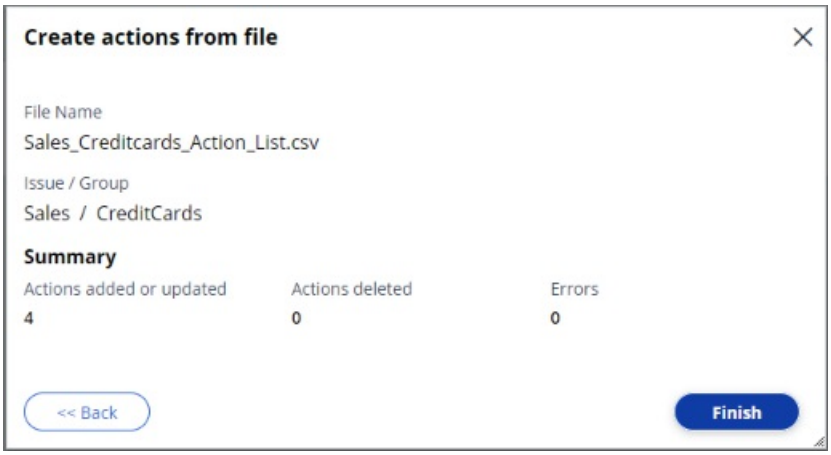
Save and close the CSV file.

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.



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

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



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.

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.

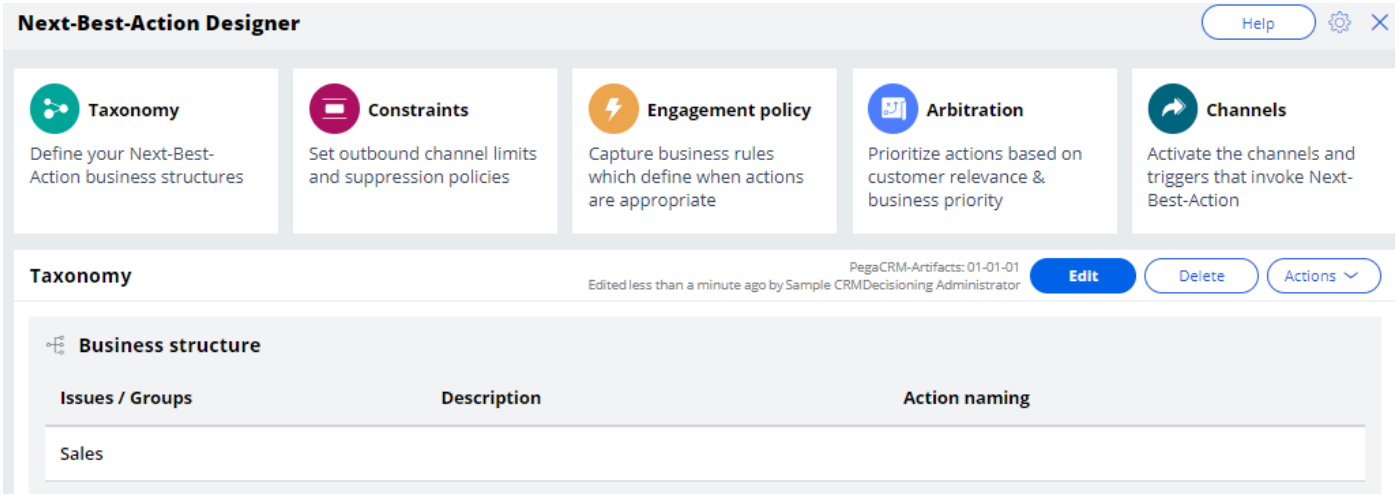
Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

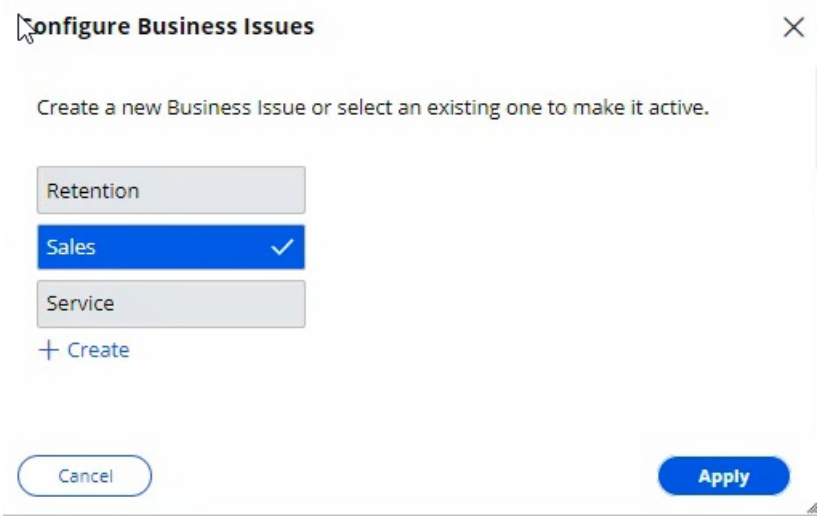
This demo will show you how to set up your business structure in Next-Best-Action Designer.

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.

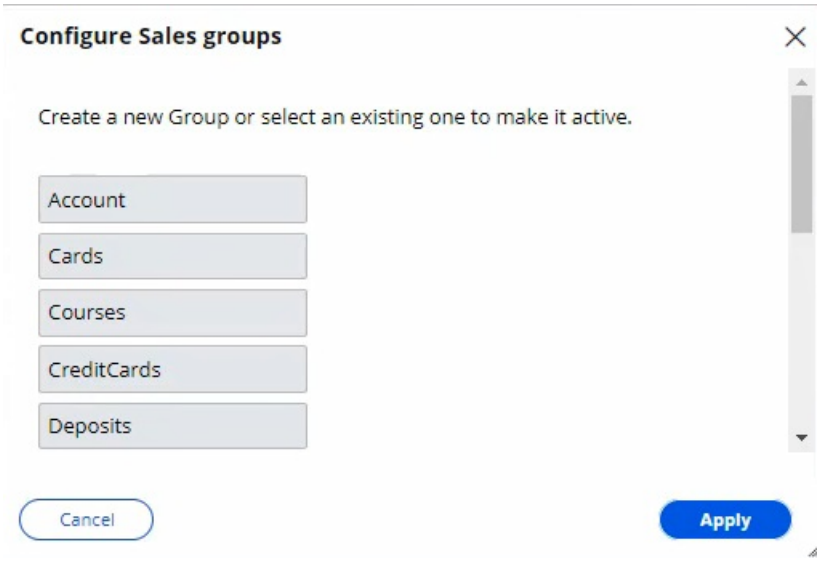


In this case, you will set up U+ bank's business structure based on their various areas of business focus. Currently, there is only one Issue in the business structure.

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.

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.

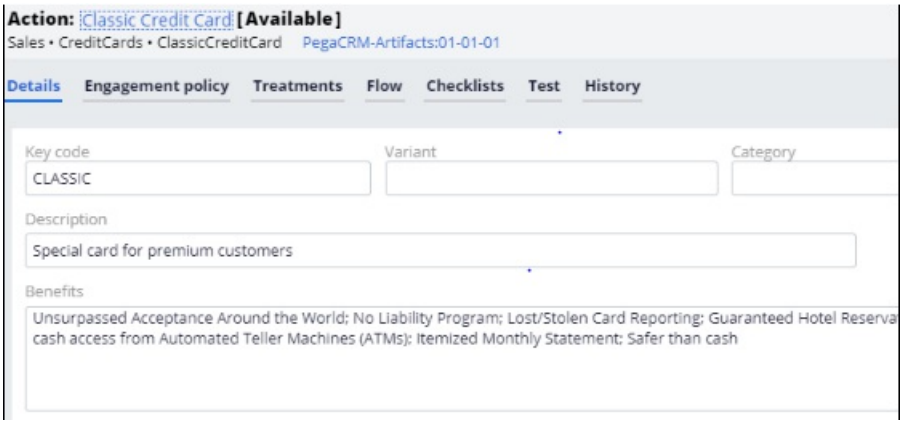
Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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 **Sales** Issue and **CreditCards** Group. Open an Action to view the name. Notice that the Sales/CreditCards Group uses the default terminology ‘**Action**’.

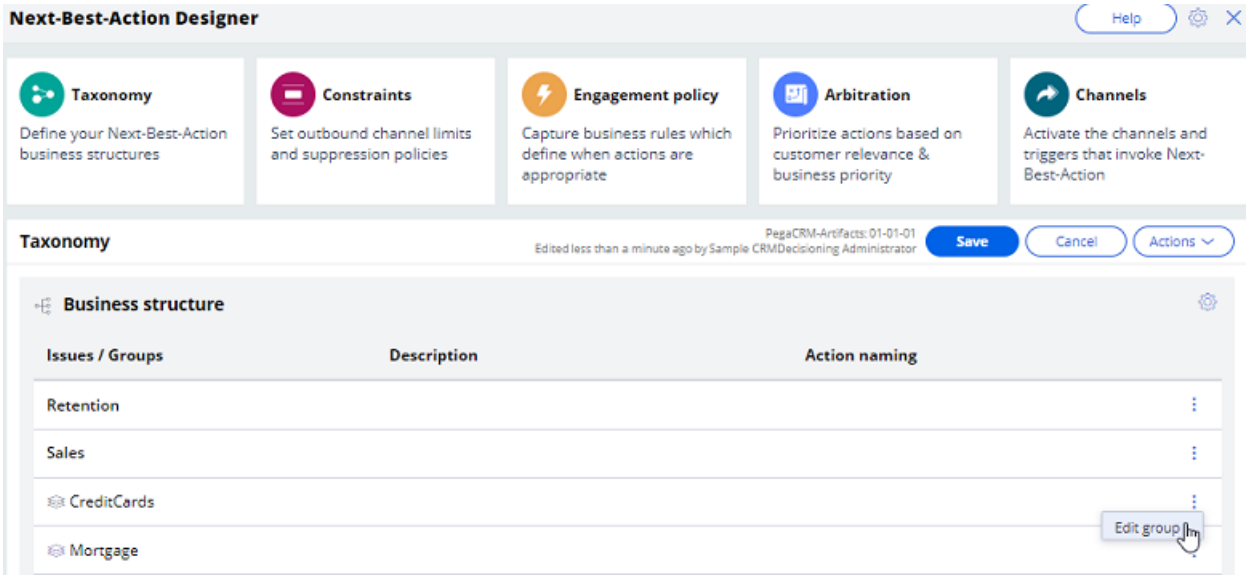


U+ bank wants to rename Actions under the **Sales** Issue and **CreditCards** Group to better suit their business purpose. The Actions under the **CreditCards** Group are always promotional credit card offers, therefore the bank wants to change the naming convention from ‘Action’ to ‘Offer’.

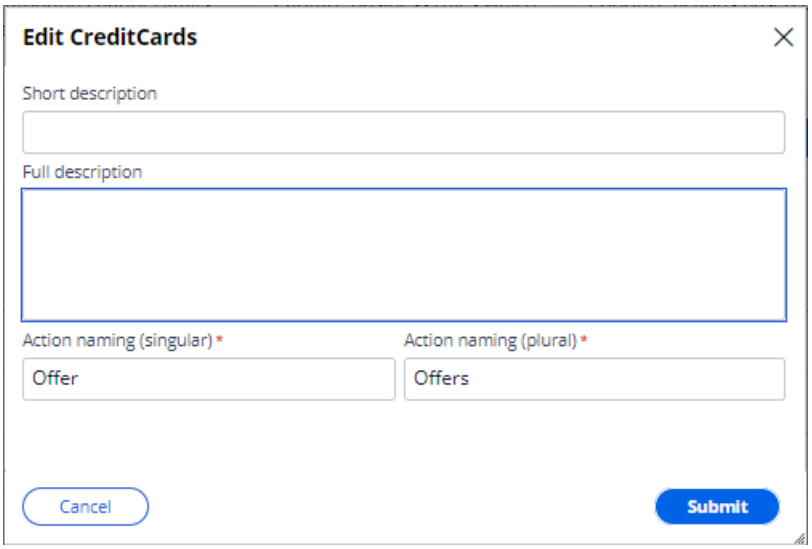
You can change the default naming convention of Actions in Next-Best-Action Designer.

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.

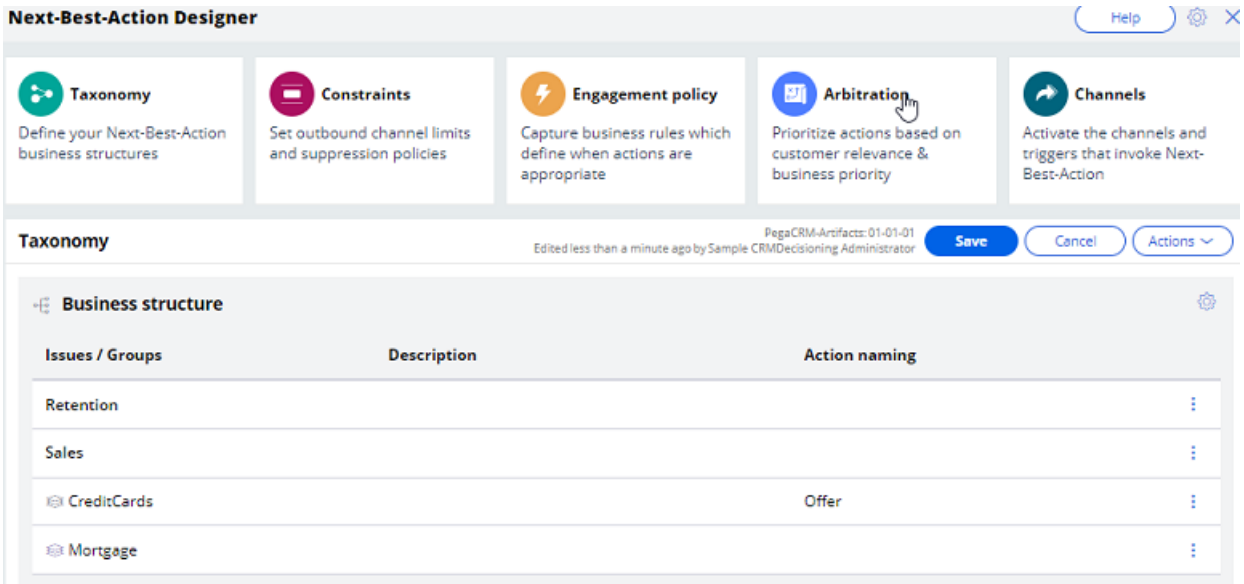
Edit the hierarchy to modify the Group attributes.



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 **Sales** Issue and **CreditCards** Group to view the effect of renaming.



From now on, Actions under Sales/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.

Creating 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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

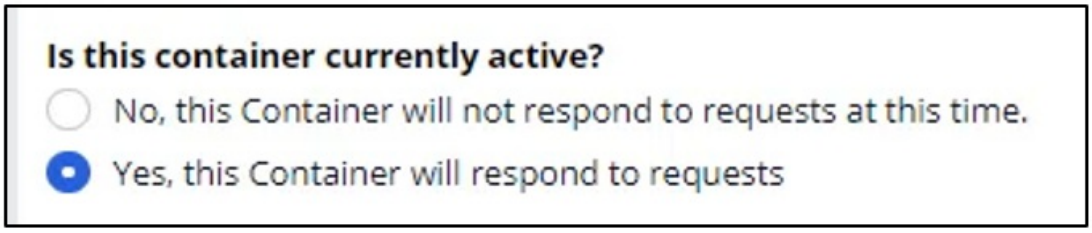
An external channel uses the real-time container's identifier when it makes a request to the Pega Customer Decision Hub.

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.

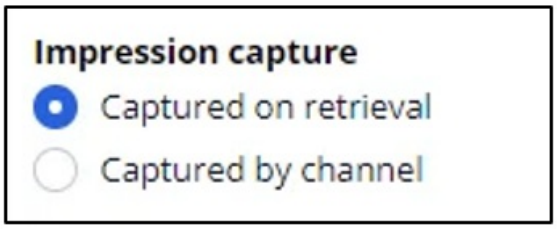
Now, open the real-time container.

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



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.

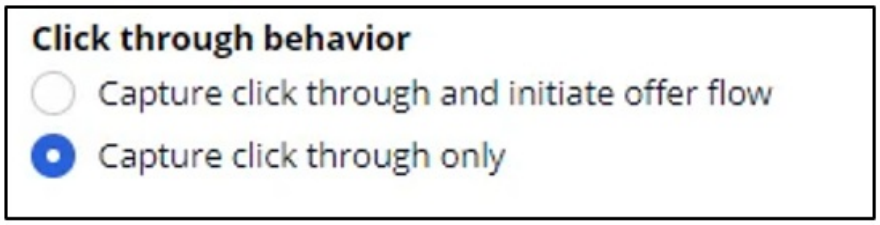


‘Captured on retrieval’ means that the Customer Decision Hub records an impression in the Interaction History immediately after sending the action details to the external channel.

‘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 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.



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.

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.

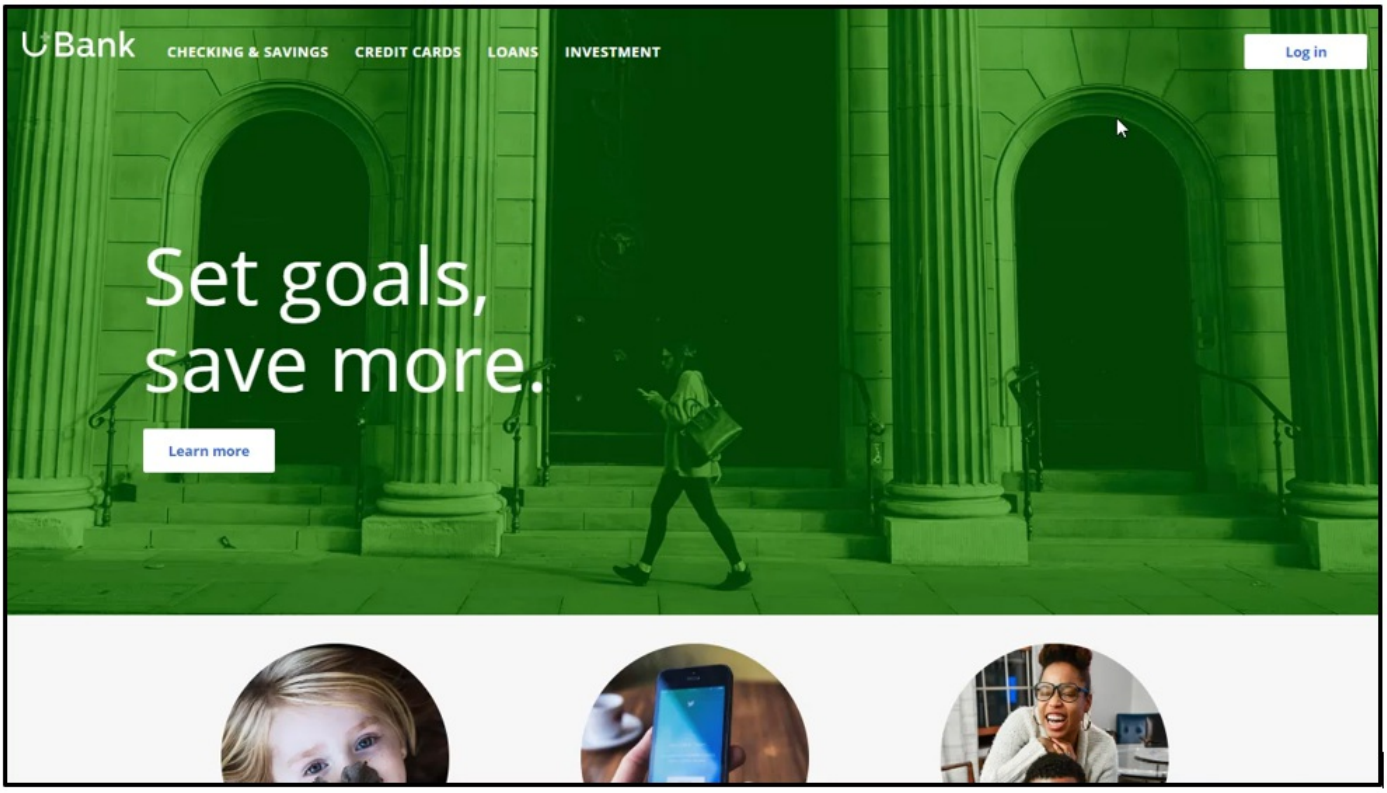
Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

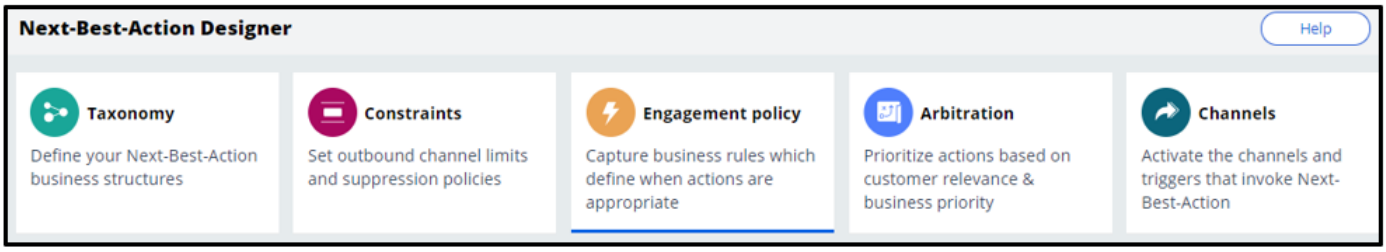
For example, if customer Troy logs in to his Accounts page, the ‘Cash back’ offer is displayed.

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

This page shows the offer details.

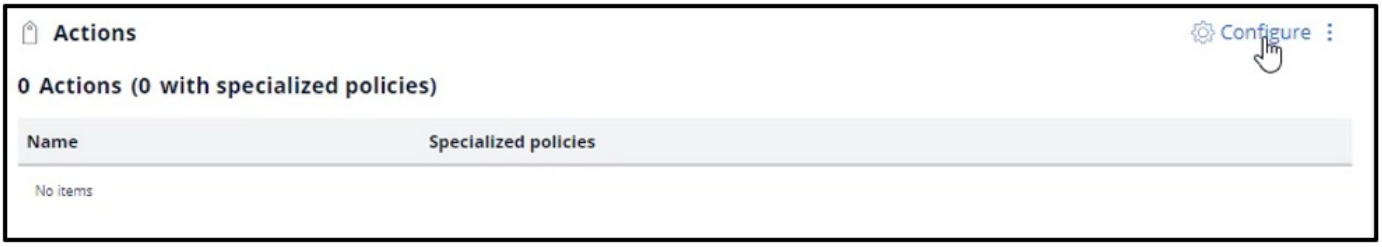
To implement this business requirement, navigate to Next-Best-Action designer.

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.

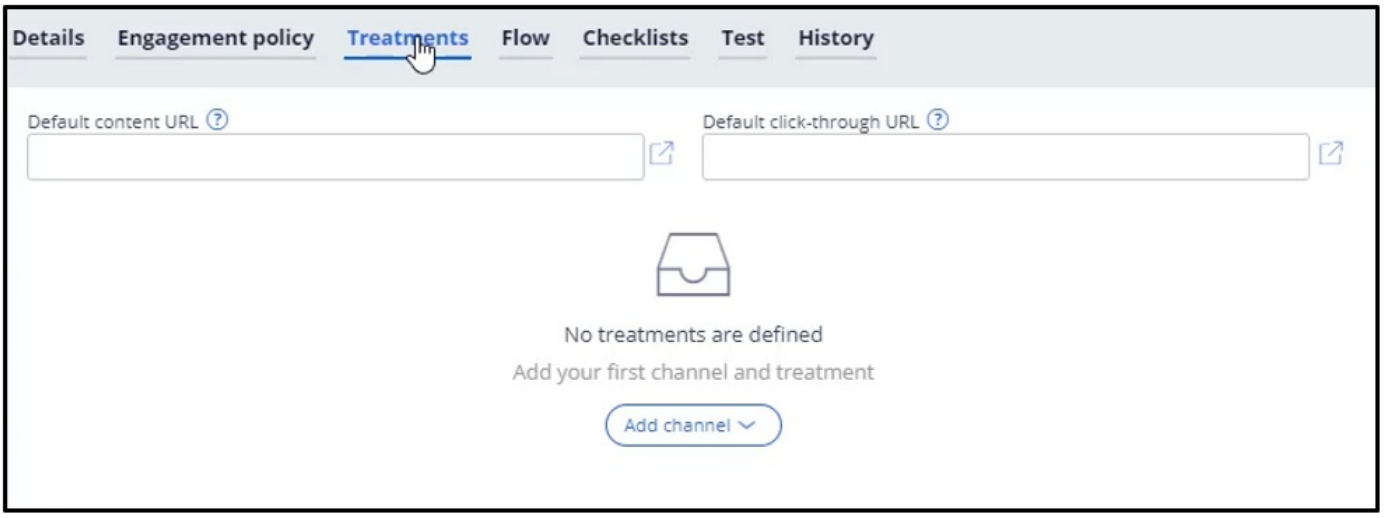


A consultant has already created a few actions under Sales/CreditCards. However, in this scenario, the bank wants to show only the ‘Cash back’ card.

Now, open on the ‘Cash back’ offer.

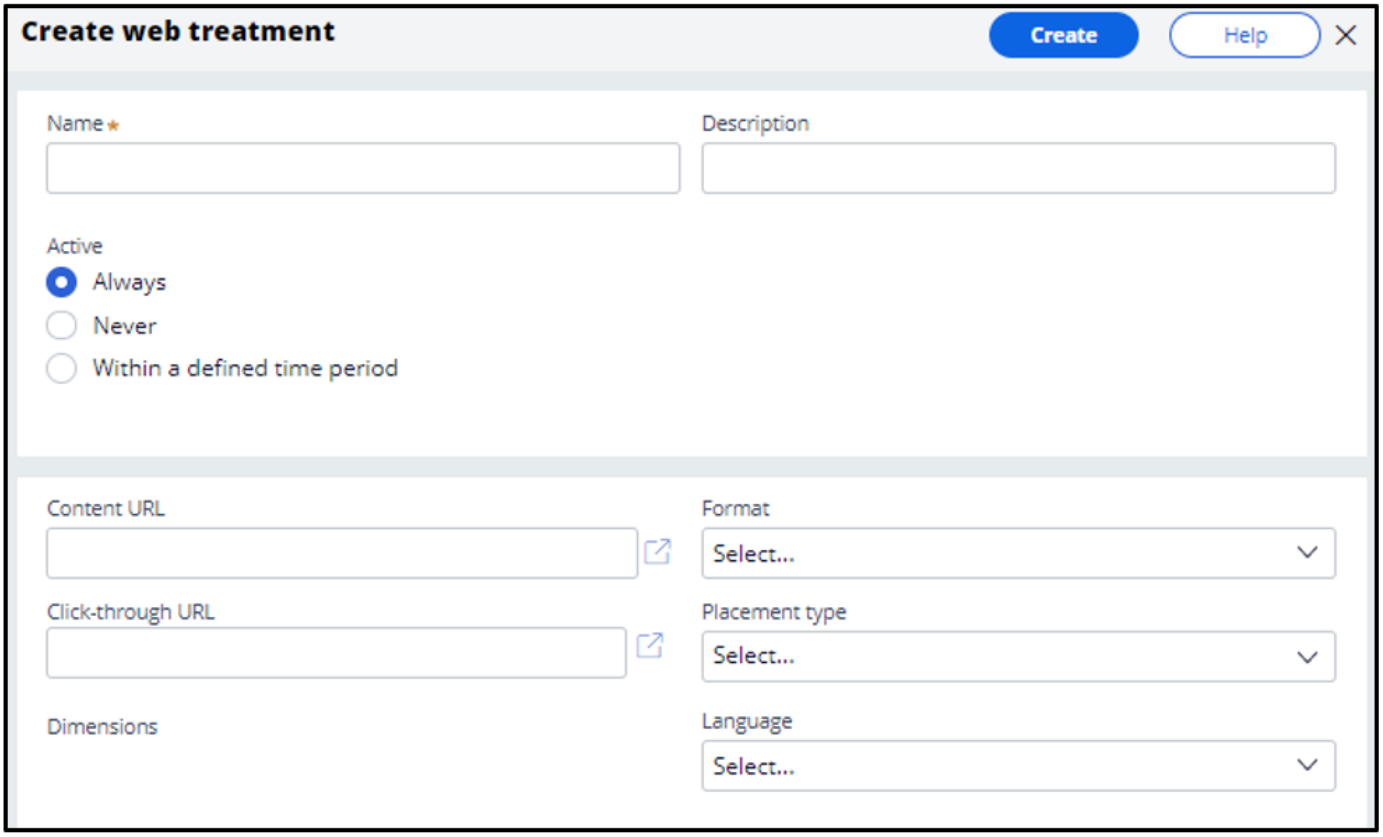
To display an offer, you need to add a treatment that is specific to the channel.

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.



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.

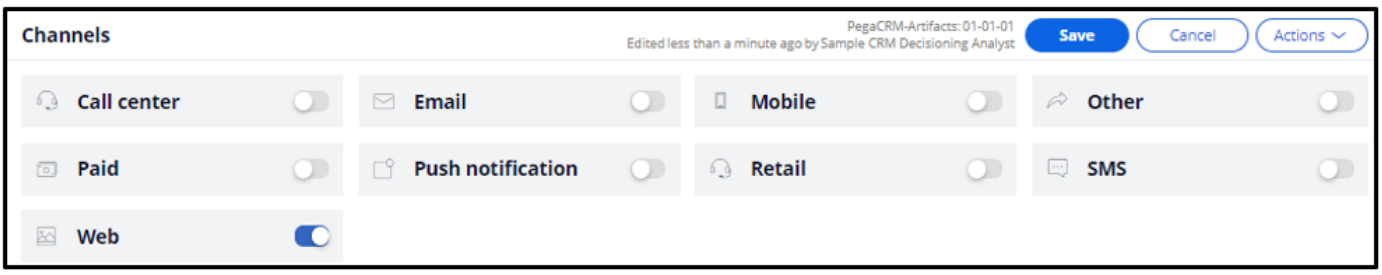
You can specify the language of the treatment if required. Specifying the treatment language allows Pega Customer Decision Hub to consider the customer’s preferred language when selecting the treatment.

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.

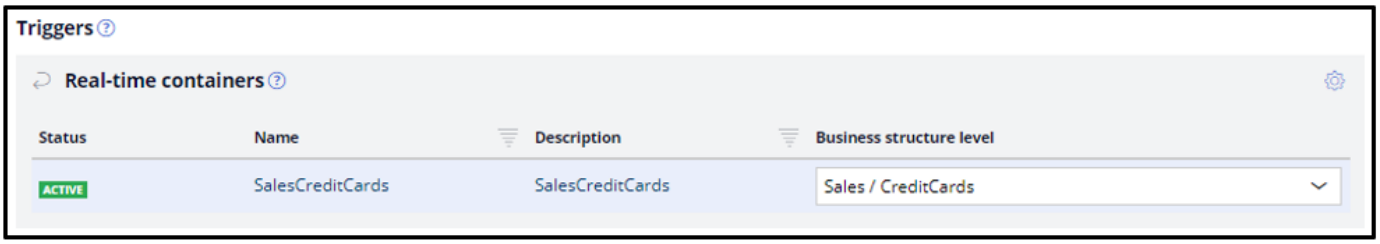


Now, configure the real-time container that the U+ website will use to display the offer banner on the account page.

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.



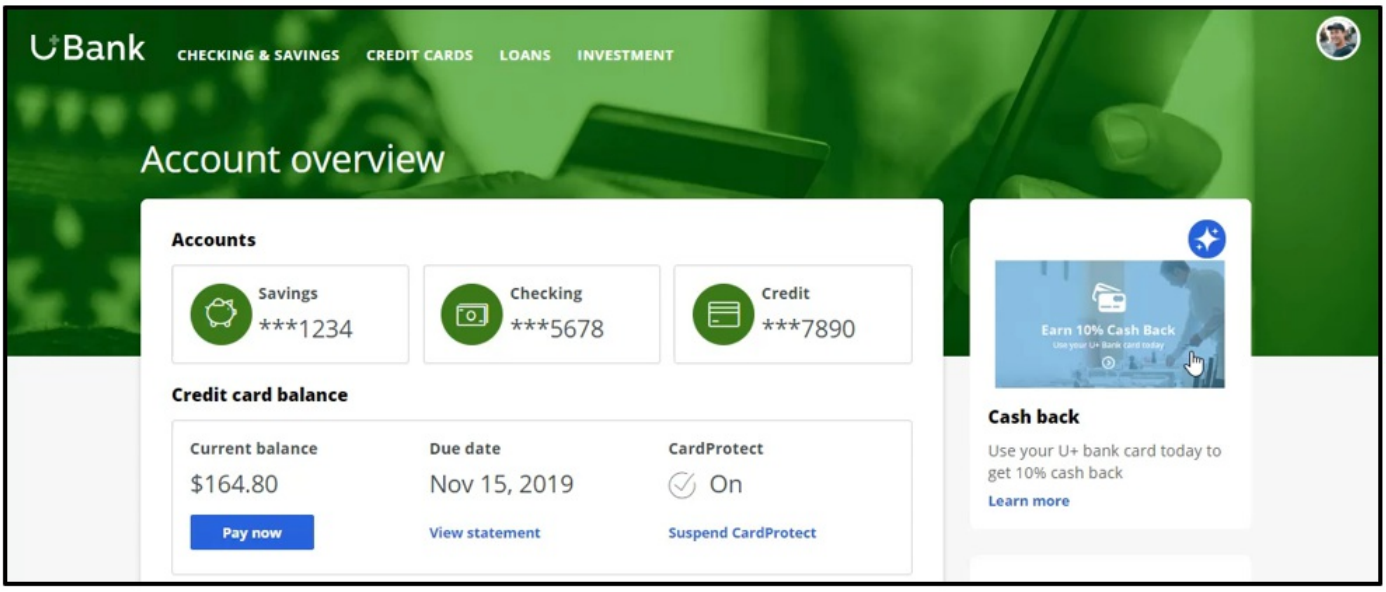
With that, all the necessary configuration for this scenario is complete. Save the changes for the configuration to take effect.

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.

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.

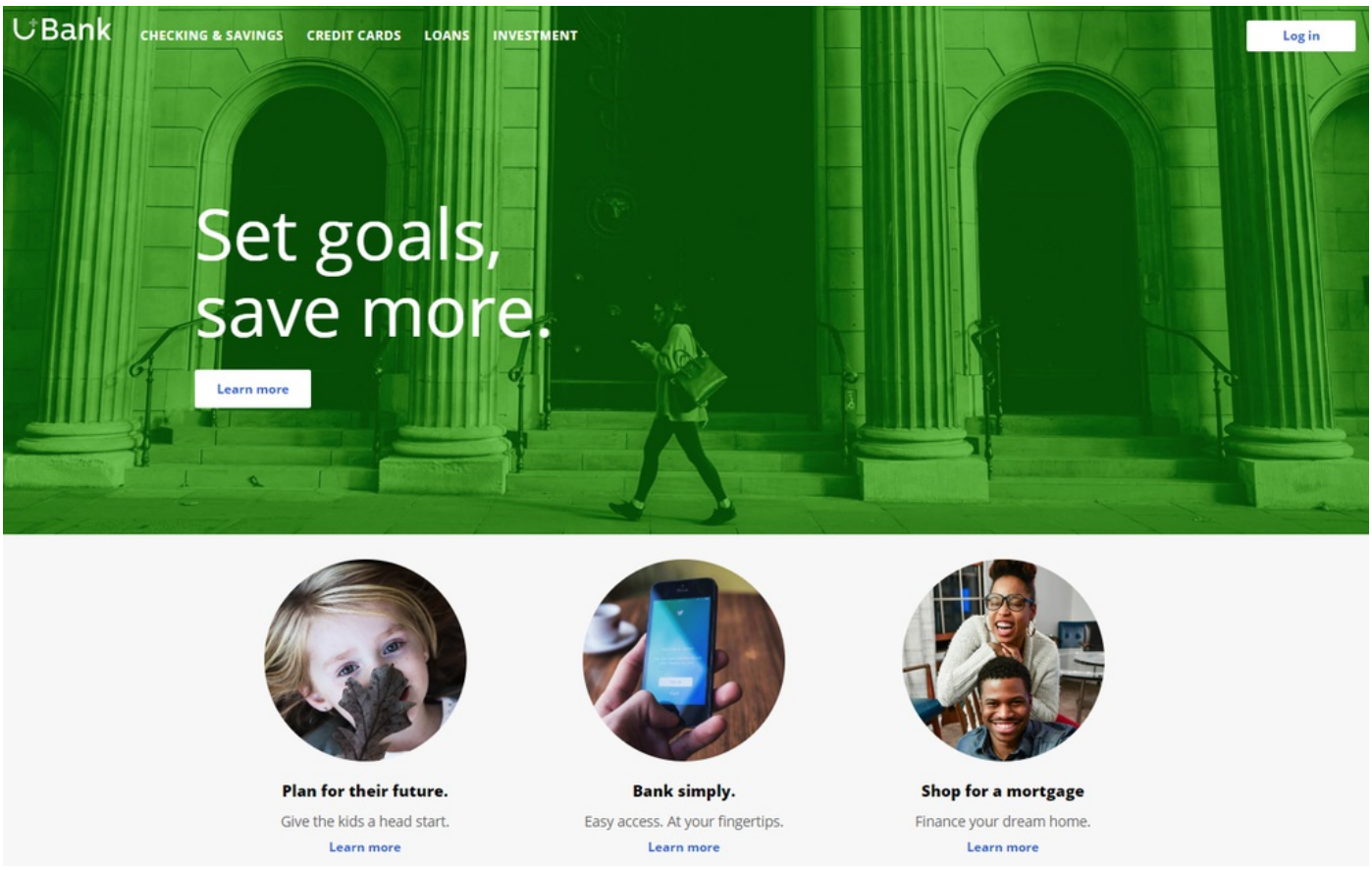
Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

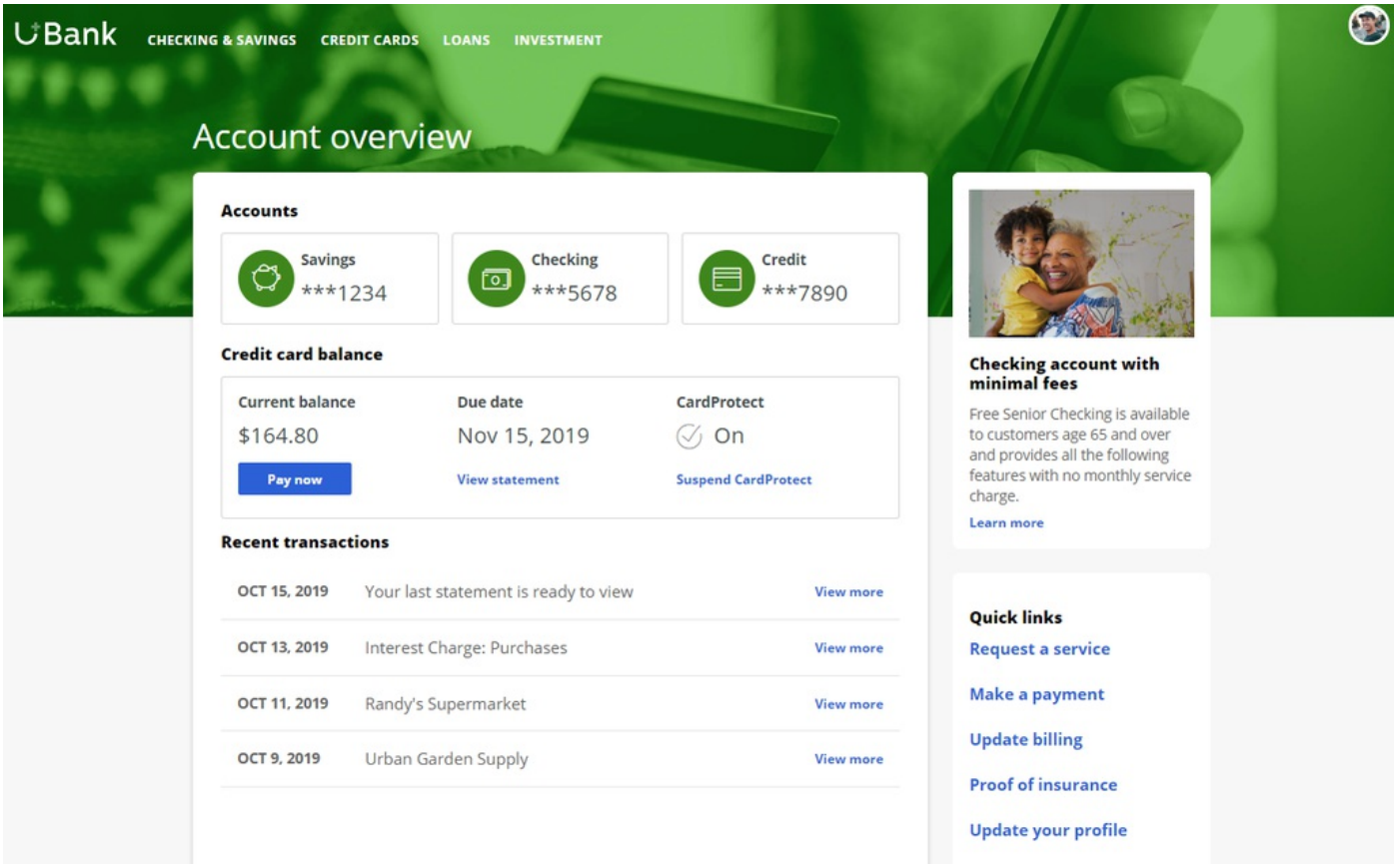
Transcript

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.

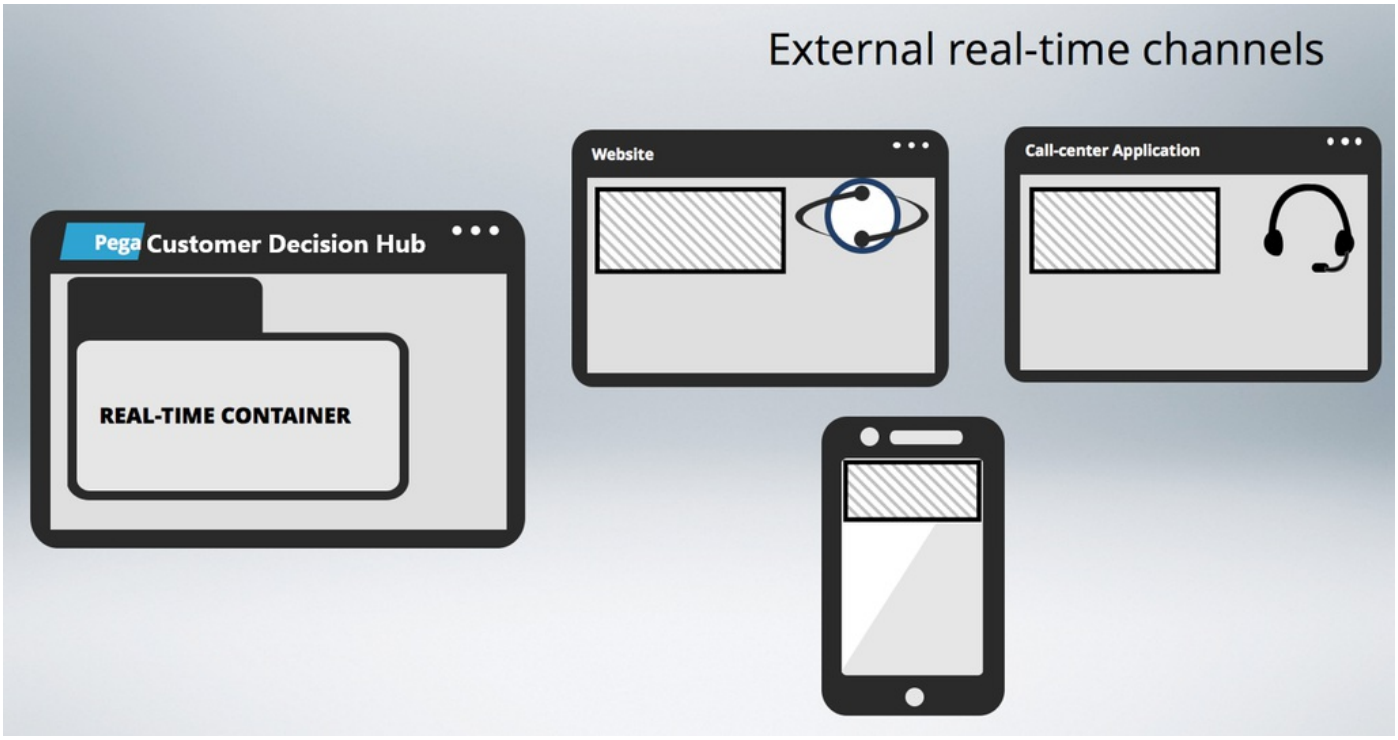


The marketing department wants to leverage the Customer Decision Hub's Next-Best-Action capability to display the right offer for each customer.

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.

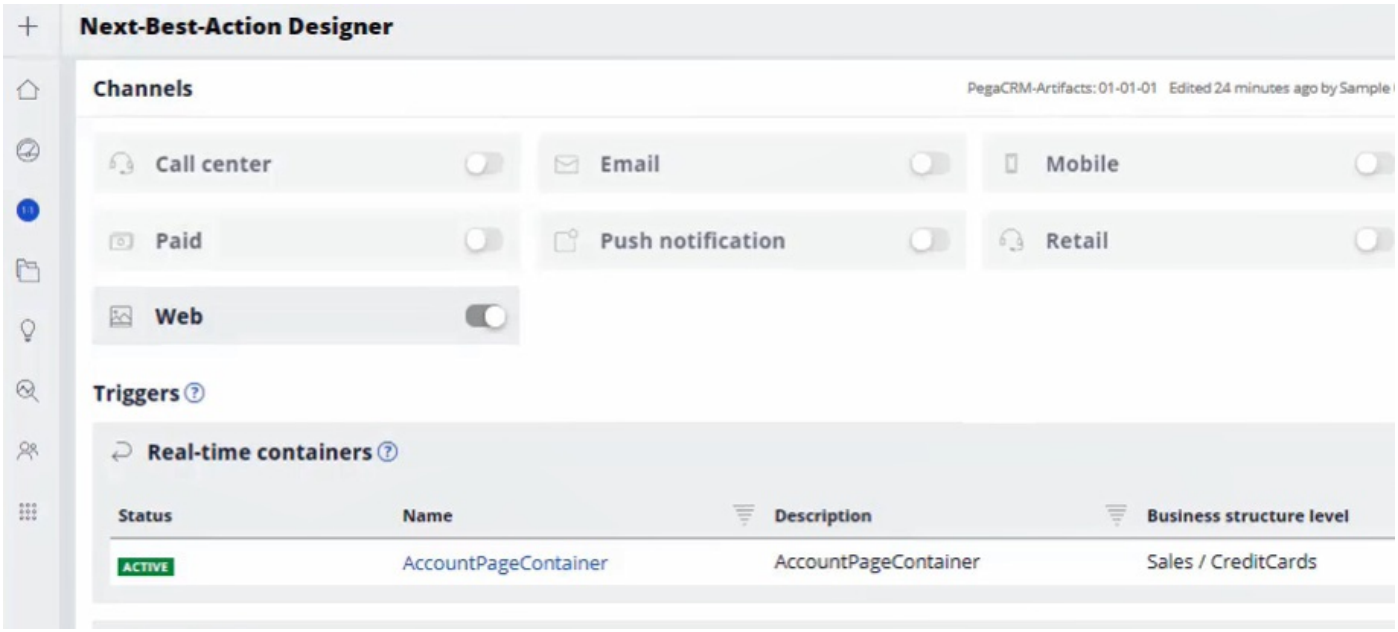
An external real-time channel is any channel that presents actions selected by the Customer Decision Hub to a user or customer. For example, a website, a call-center application, or a mobile application.



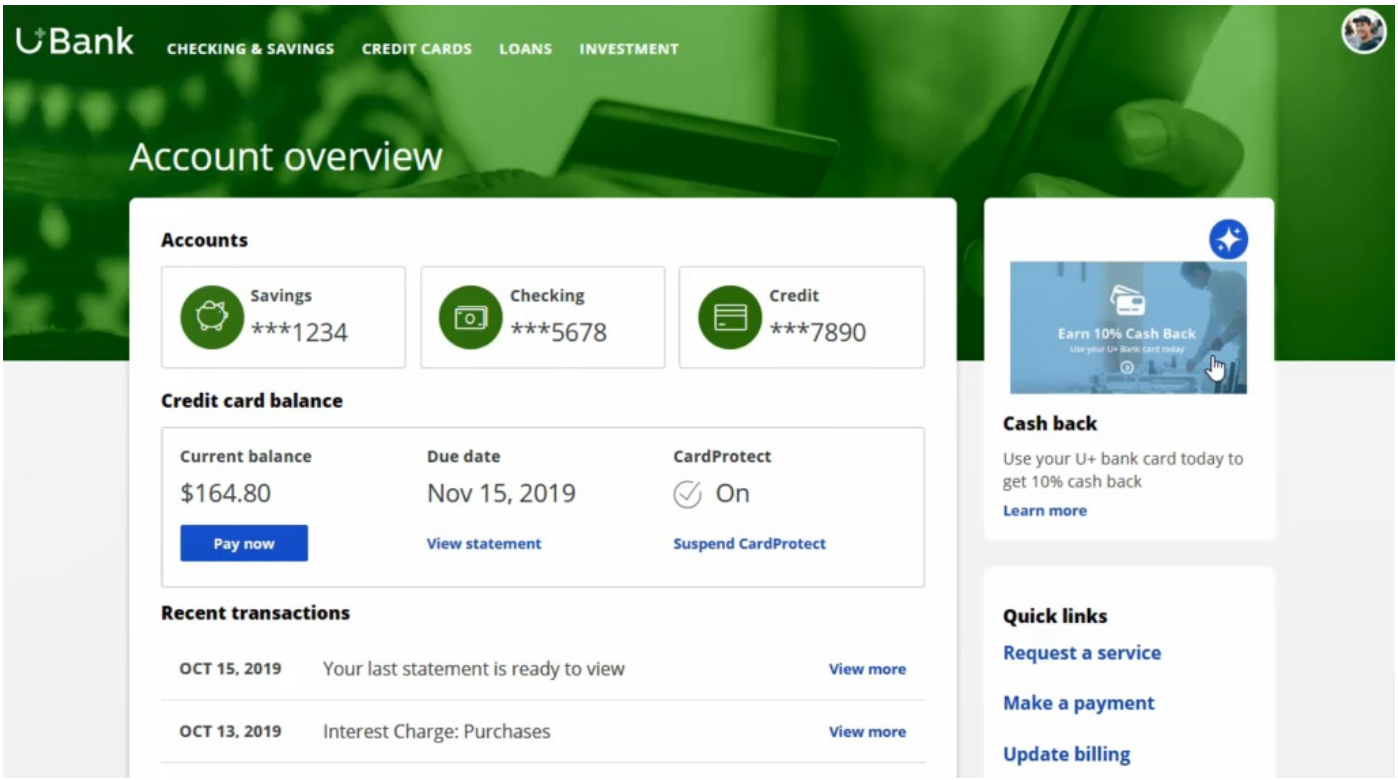
Here's how the website invokes the real-time container to present credit card offers on the account page.

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.



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.

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

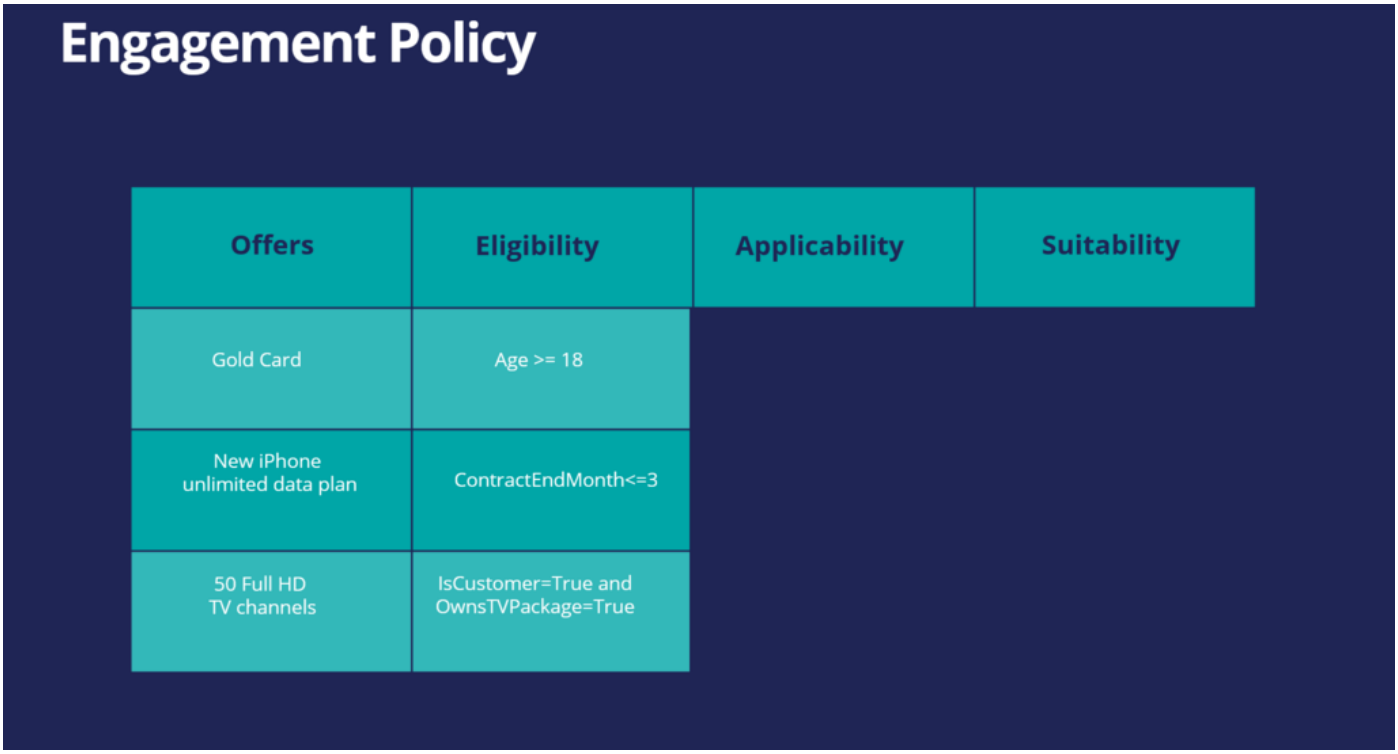
Consider the following examples: a retail bank is promoting a Gold Credit Card; a telco is offering a new iPhone upgrade with an unlimited data plan; and a communications and media company is promoting a new bundle of HD channels.

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.

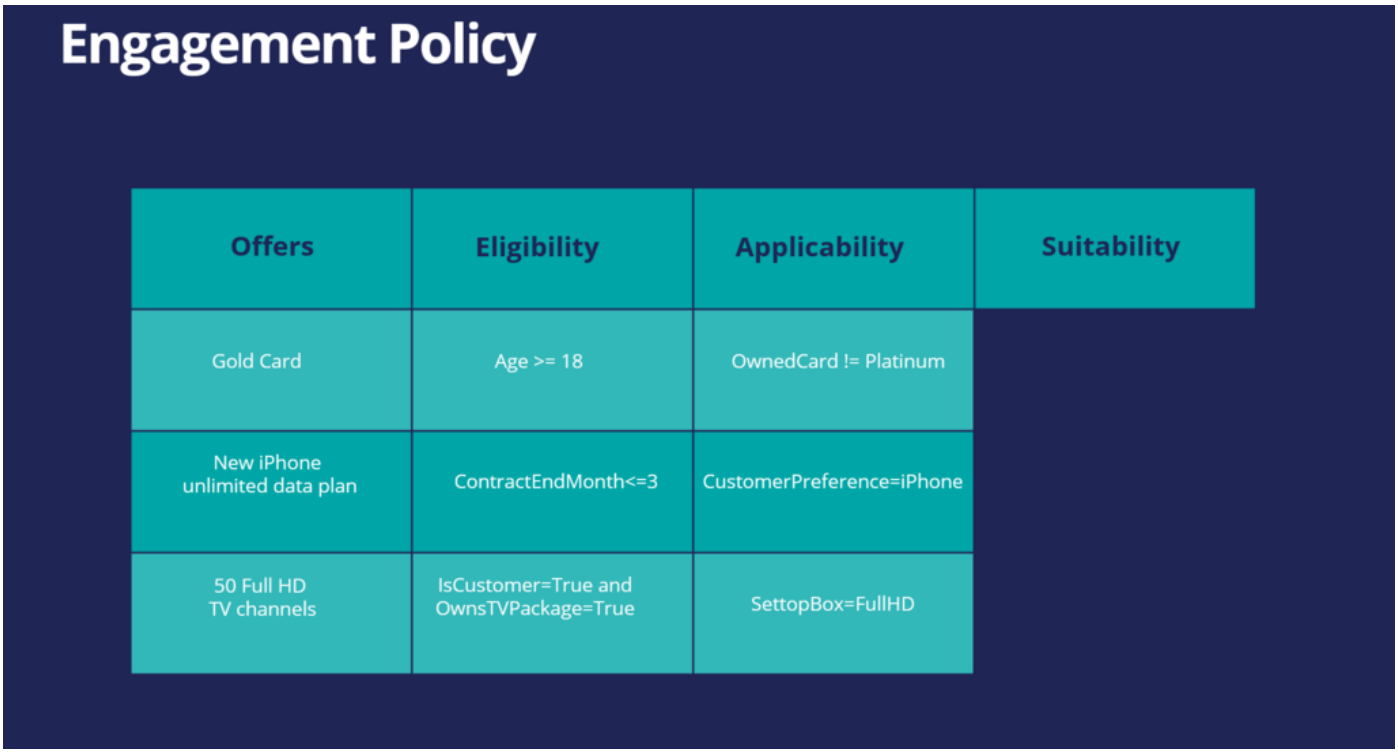
For the TV channels offer, customers must already own a TV subscription. This offer is not available for customers who have only mobile or landline subscriptions.



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.

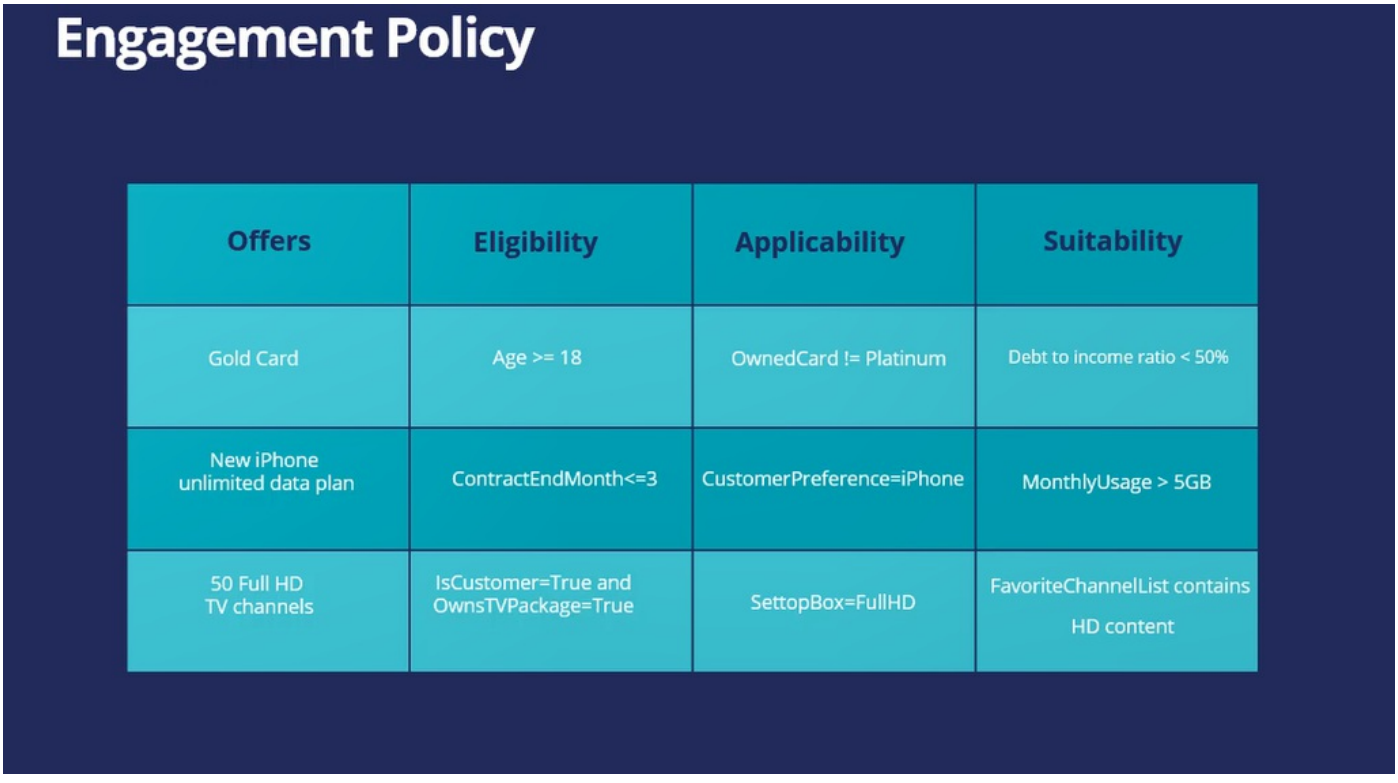
Similarly, with the iPhone upgrade offer, if a customer explicitly expressed in the last survey that they weren’t interested in an iPhone, this action is not applicable to them. For the TV channels offer, the business does not want to advertise HD channels to a customer who has recently bought a set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



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.

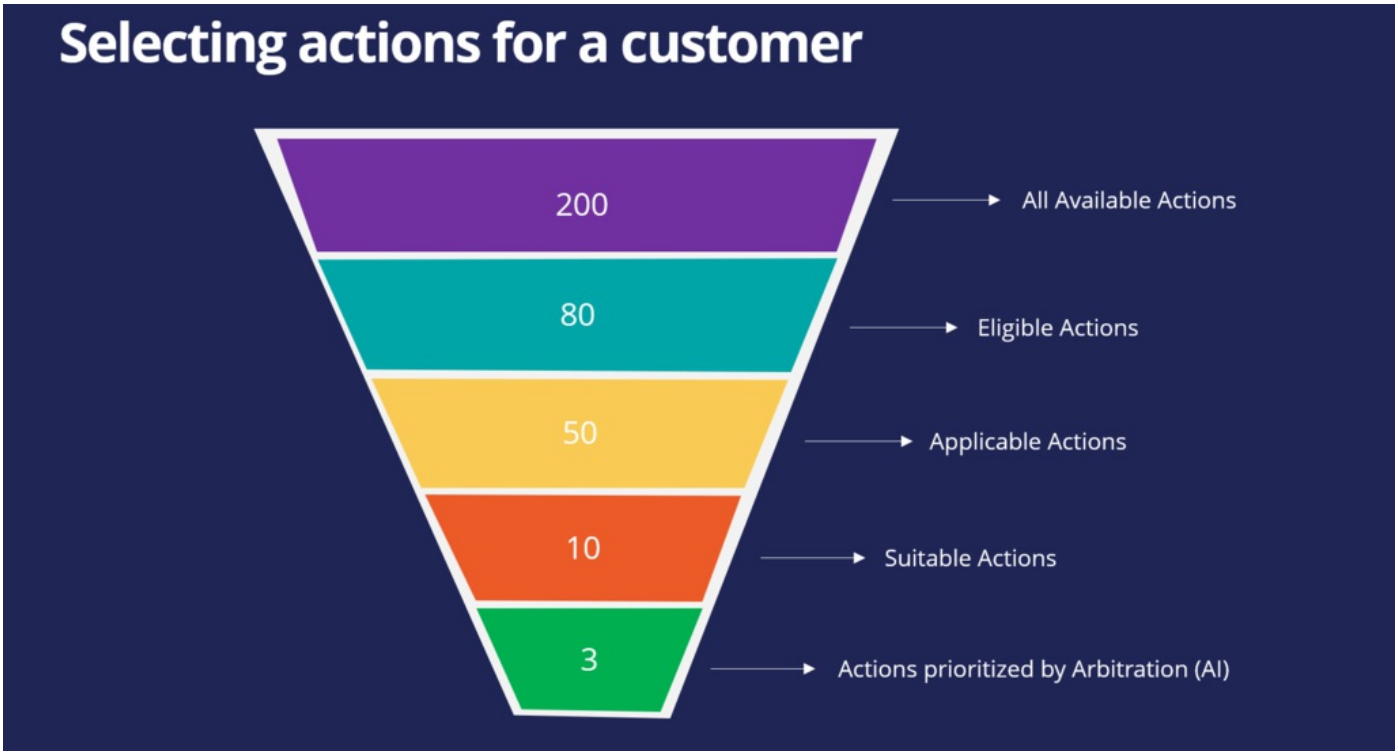


U+, a retail bank, has configured its engagement policy to suit its own business objectives as well as the

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.



Defining eligibility, applicability, and suitability rules

Introduction

Engagement policies are defined in Next-Best-Action Designer. Engagement policies specify the conditions under which an action or group of actions is available for a customer. These policies should be defined in the following categories: eligibility, applicability, and suitability, which represent the true nature of the associated conditions.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This demo will show you how to define engagement policy conditions such as Eligibility, Applicability, and Suitability using Next-Best-Action Designer.

U+, a retail bank, has introduced two new credit cards and would like 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 will be available for new customers who do not yet have a credit card.

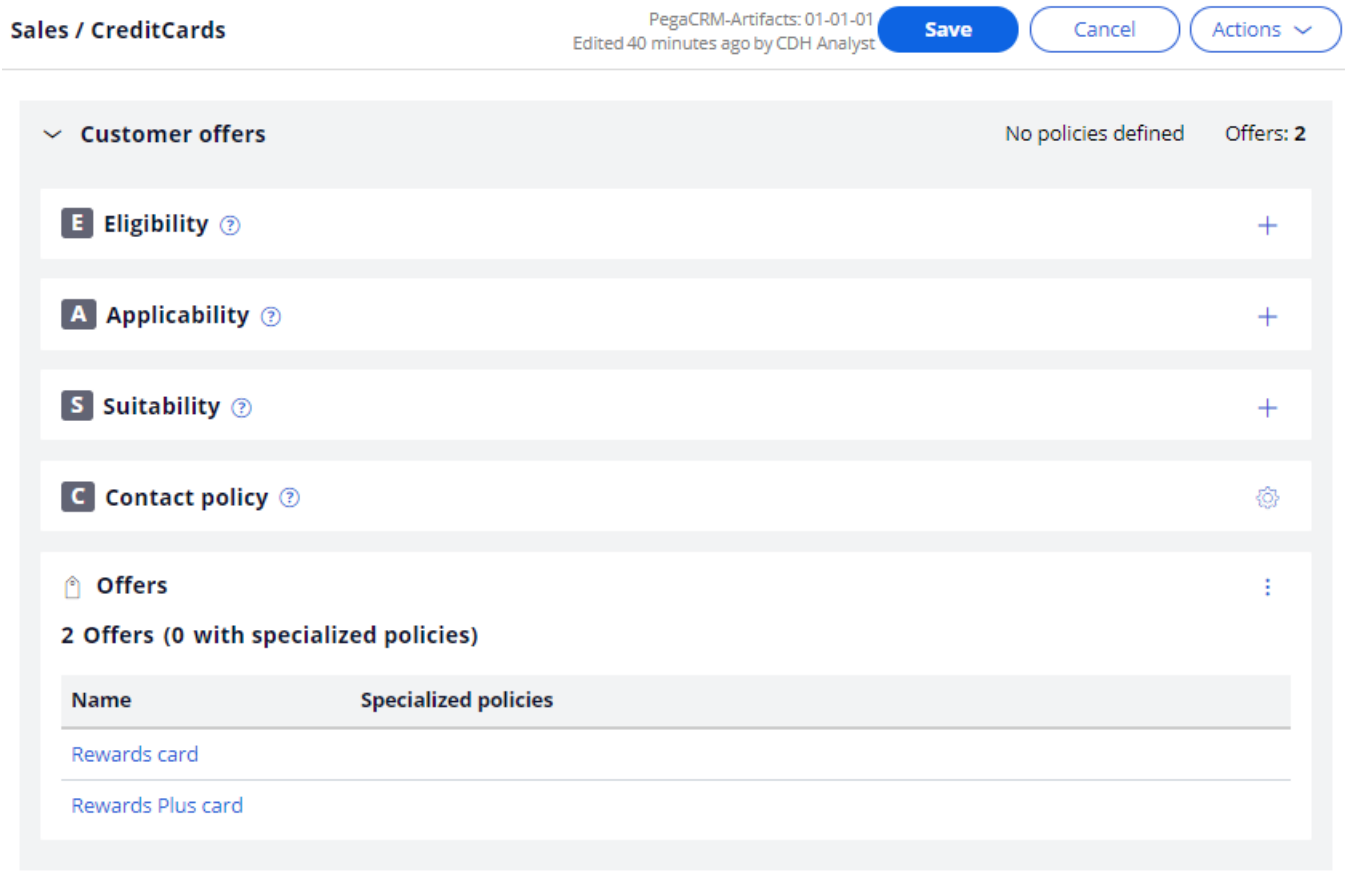
Last, the business understands that not all credit cards are suitable for everyone.

Due to 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.

This is 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.

Now, you will set up the engagement policies to reflect U+ bank's requirements. The engagement policies can be defined for a specific group within an issue and/or for individual actions.

Edit to configure the engagement policies. Notice that the two actions are listed under this group.

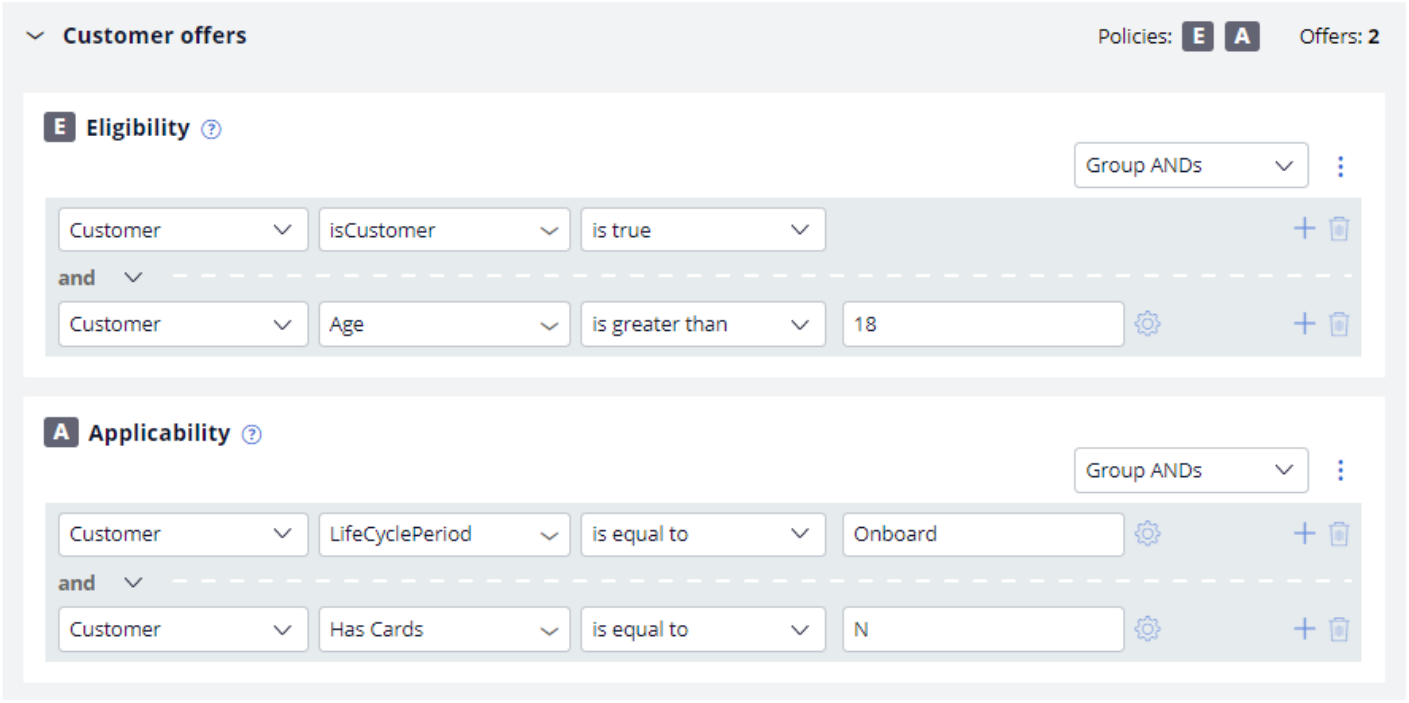


As a best practice, engagement policies should be defined in the following categories: Eligibility, Applicability and Suitability, which represent the true nature of the associated conditions.

First, define the Eligibility condition to ensure that only current customers are considered for the action, and that the customers are at least 18 years old.

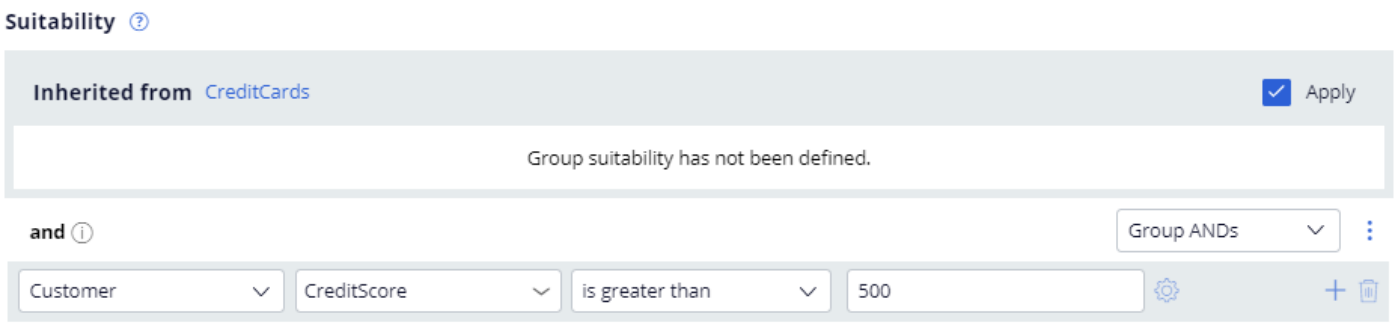
Then, define the Applicability condition so that only new customers, who currently do not have a credit card, qualify for the actions. These criteria are also being defined at the group level.

The "new customer" check is done using the **LifeCyclePeriod** property. Add a condition to check if the customer already has a credit card.

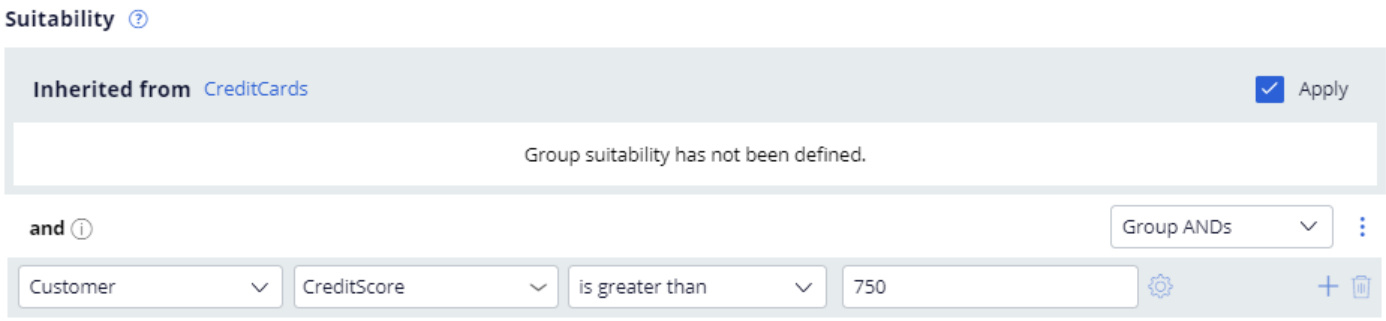


Lastly, business wants a different Suitability condition for individual cards depending on the card's limit. Since the conditions are specific to each credit card, the Suitability condition for each must be defined at the Action level.

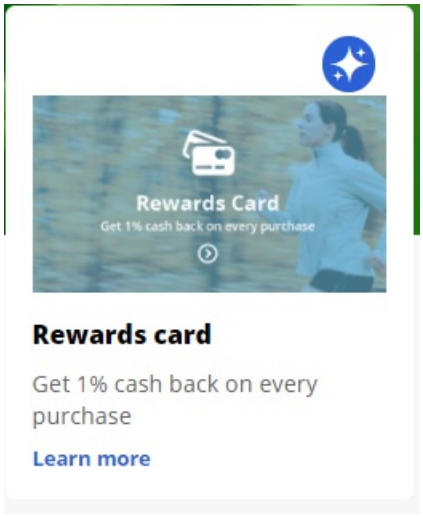
Open the Rewards Card Action. This card comes with a certain credit limit. Therefore, U+ believes this card is only suitable for customers with a credit score higher than 500, even though customers may satisfy the Eligibility and Applicability conditions. Thus, define the Suitability condition accordingly.



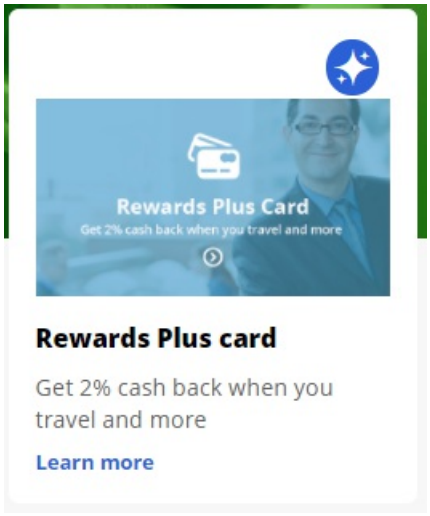
Next, open the Rewards Plus Card action. This card has an even higher credit limit than the Rewards card. Thus, this card is suitable for customers with credit score higher than 750. To complete the configuration, save the changes.



Log in as Troy, a 25-year-old Chef, who became a customer of U+ bank 3 months ago, and his credit score is 600. Notice that the Rewards Card offer is displayed, as Troy satisfies all Eligibility and Applicability conditions. Troy will only be offered this card because his credit score is higher than 500 but lower than 750.



Now, login as Barbara, a 40-year-old Engineer, who became a customer of U+ bank a month ago, and her credit score is 800. Notice that the Rewards Plus Card offer is displayed, as Barbara also satisfies all Eligibility and Applicability conditions. Barbara on the other hand will be offered both cards, as her credit score is higher than 750.



Defining eligibility, applicability, and suitability rules -- Tue, 10/27/2020 - 01:43
To get the full experience of this content, please visit https://academy.pega.com

Testing next-best-action configuration with persona testing

Introduction

You can design persona-based tests to verify that the Next-Best-Action Strategy gives the expected results. You can design these personas according to your requirements and the strategy that you are testing to ensure you do not introduce any regressions as you define your engagement policies and arbitration.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

U+, a retail bank, wants to verify that customers are receiving the correct offers. In other words, the bank wants to ensure that the strategy is sending offers to customers in line with their business requirements.

Personas are a representation of various customer profiles that you use to test the results of the next-best-action strategy framework. Persona-based tests use customer personas with specific characteristics to evaluate next-best-action strategy results.

The bank decides to use Troy, Barbara, and John as personas to test the configurations. Troy is a 25-year-old chef who became a customer of U+ bank 3 months ago, and his credit score is 600. Barbara is a 40-year-old engineer who became a customer of U+ bank a month ago, and her credit score is 800. John is a 45-year-old IT employee who became a customer of U+ bank 15 days ago, and his credit score is 600.

Persona-based tests are used to verify that the next-best-action strategy gives the expected results. On the Next-Best-Action Designer Engagement Policy tab, you can create test cases for any group.

You can create test cases for a specific group, or for all groups. Running test cases for all groups implies that you are running the next-best-action strategy for all issues and groups in your business structure. In this case, as the bank wants to target the credit cards group, let's 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, Troy is supposed to receive the Standard card and Rewards card actions.

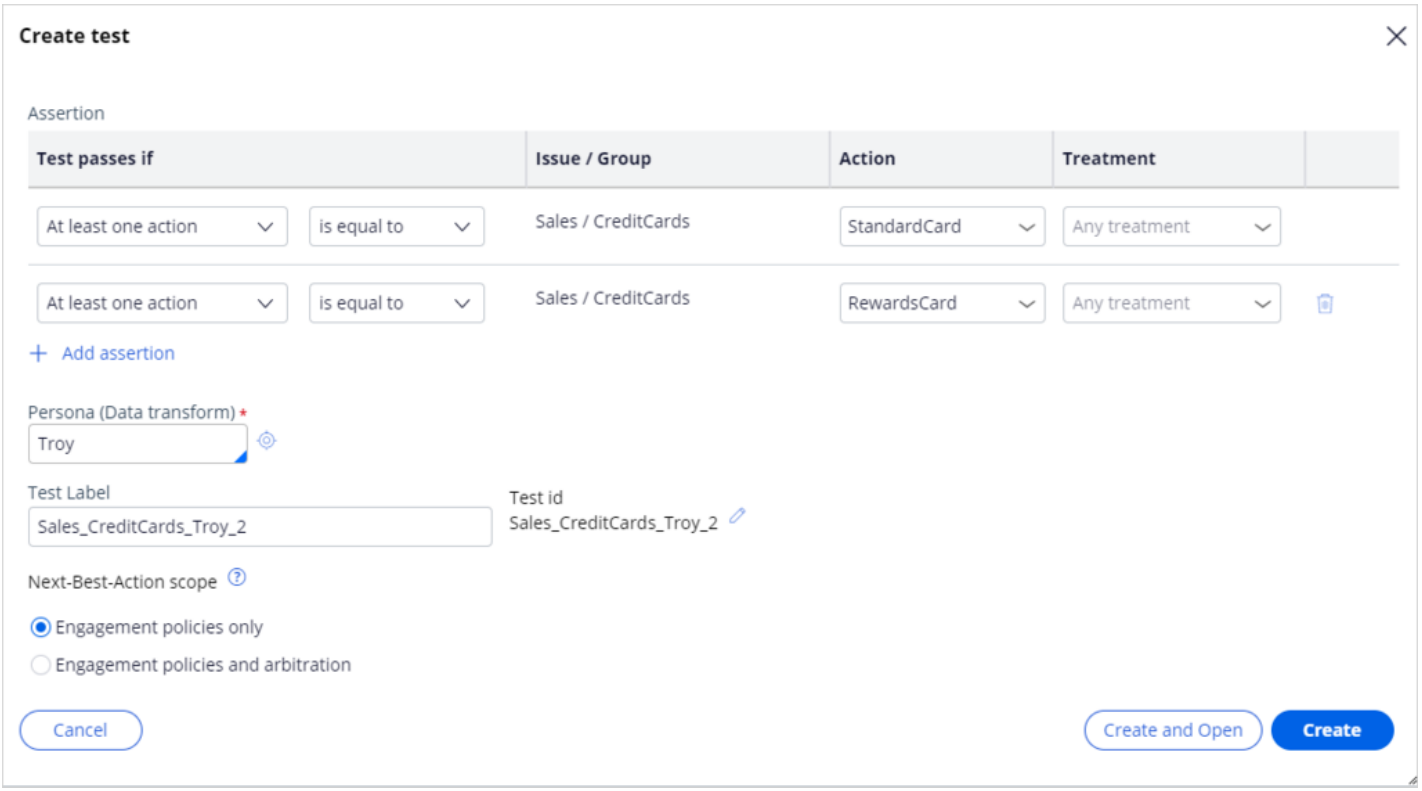
In the Persona (Data transform) field, select the persona against which you want to test the strategy. Select Troy.

In the next-best-action scope section, select whether the test should only check engagement policy configuration, or include additional elements such as constraints and arbitration.

Selecting “Engagement policies only” will validate that your policy conditions are providing the desired results. This will ensure that eligibility, applicability and suitability are tested.

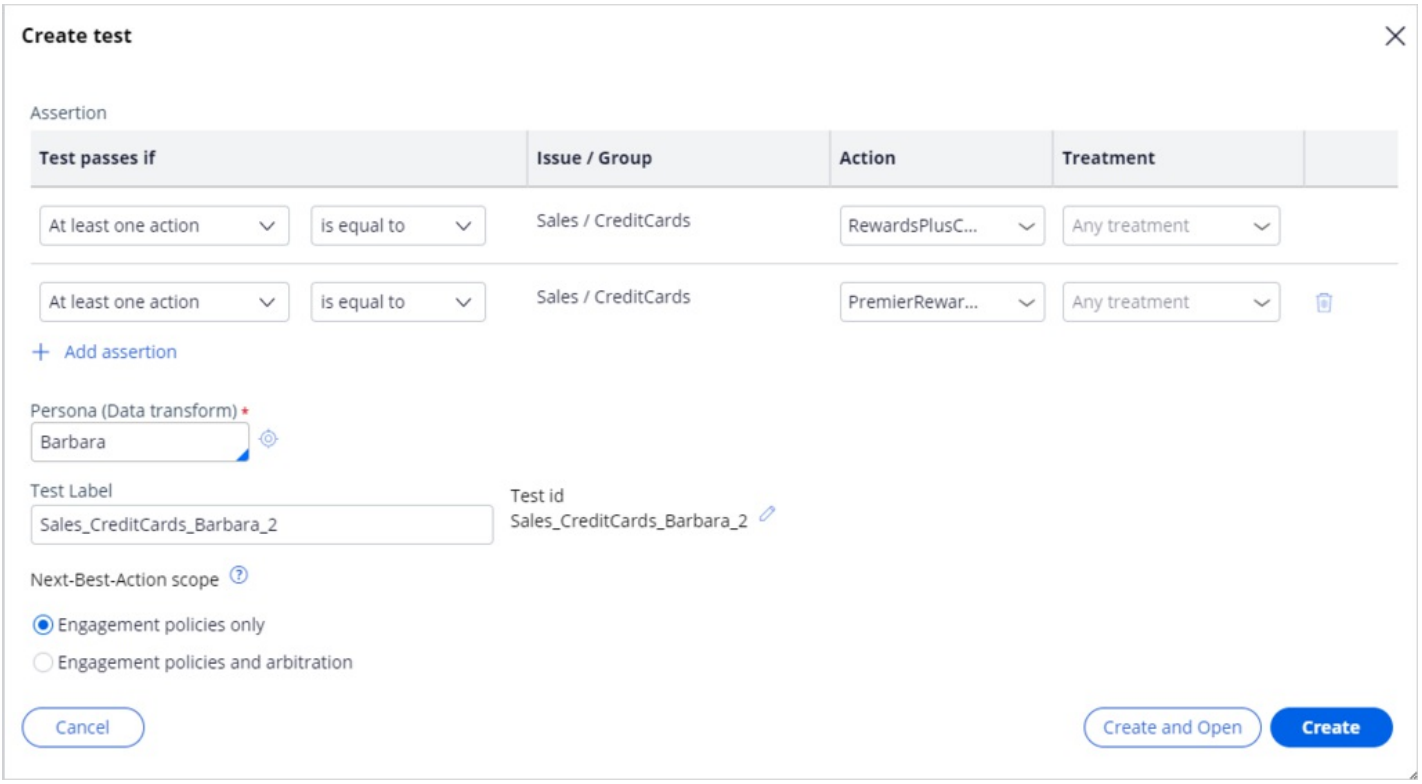
Selecting "Engagement policies and arbitration" will validate the effectiveness of your policies when arbitrating across all actions. The test will consider arbitration, adaptive analytics, treatment and channel processing, and constraints.

In this case, select "Engagement policies only".

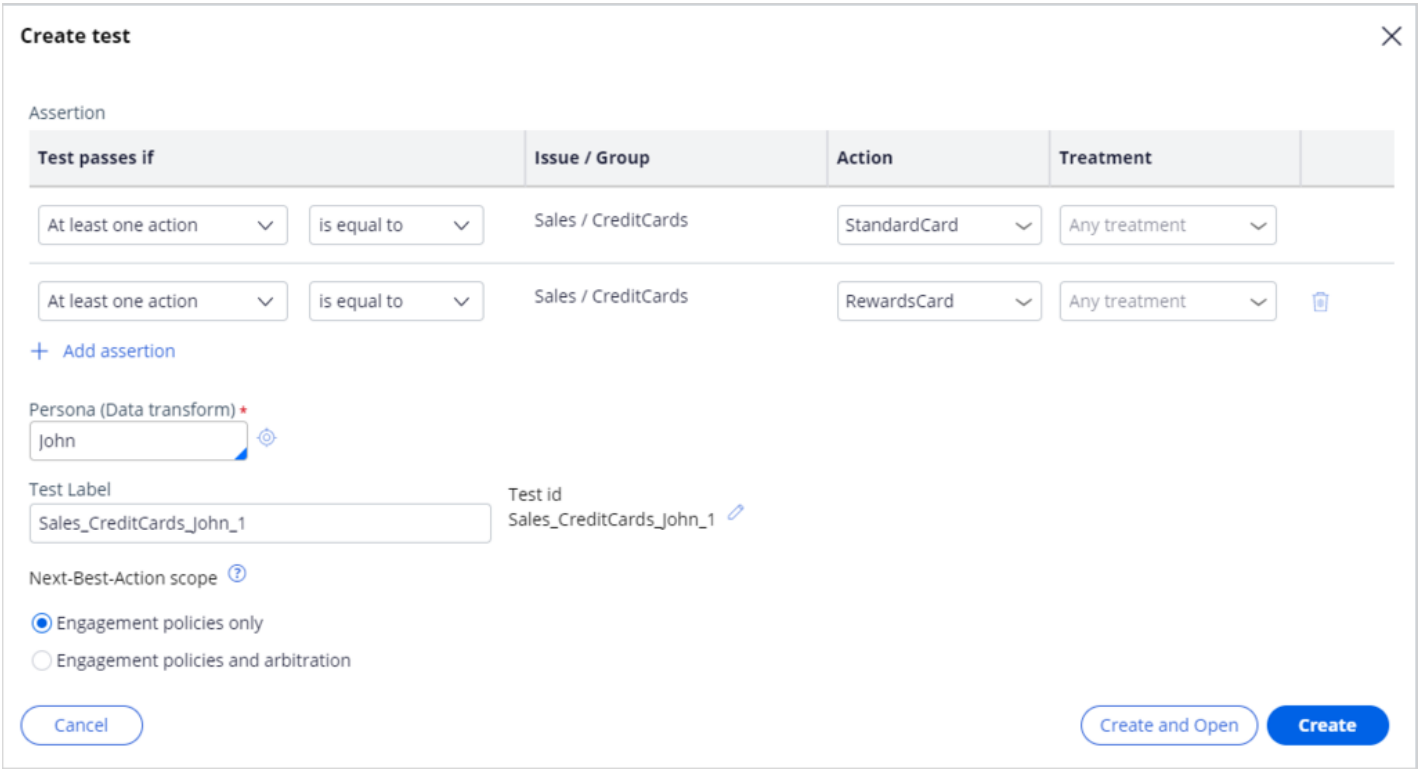


Once configured, select the test and click the Run selected tests icon.

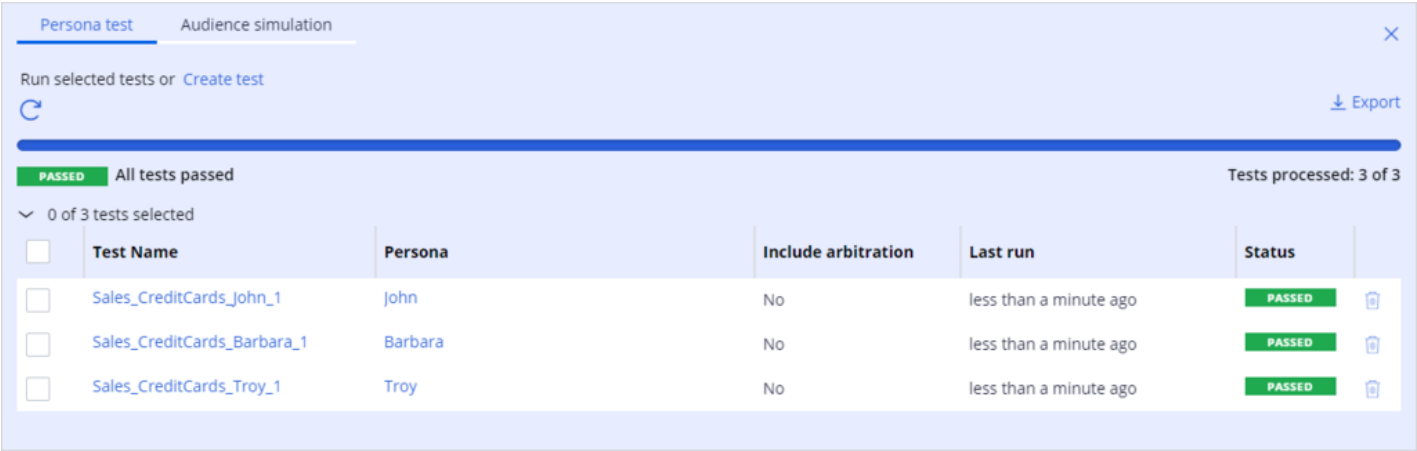
Barbara is a persona who is eligible for the Rewards plus card and the Premier rewards card. Let's create a test case for this persona.



John is a persona who is eligible for the Standard card and the Rewards card. Let's create a test case for this persona.



Run the test to confirm he passes the test.



The bank has now decided to offer the Standard card to customers who have an average balance greater than $2000.

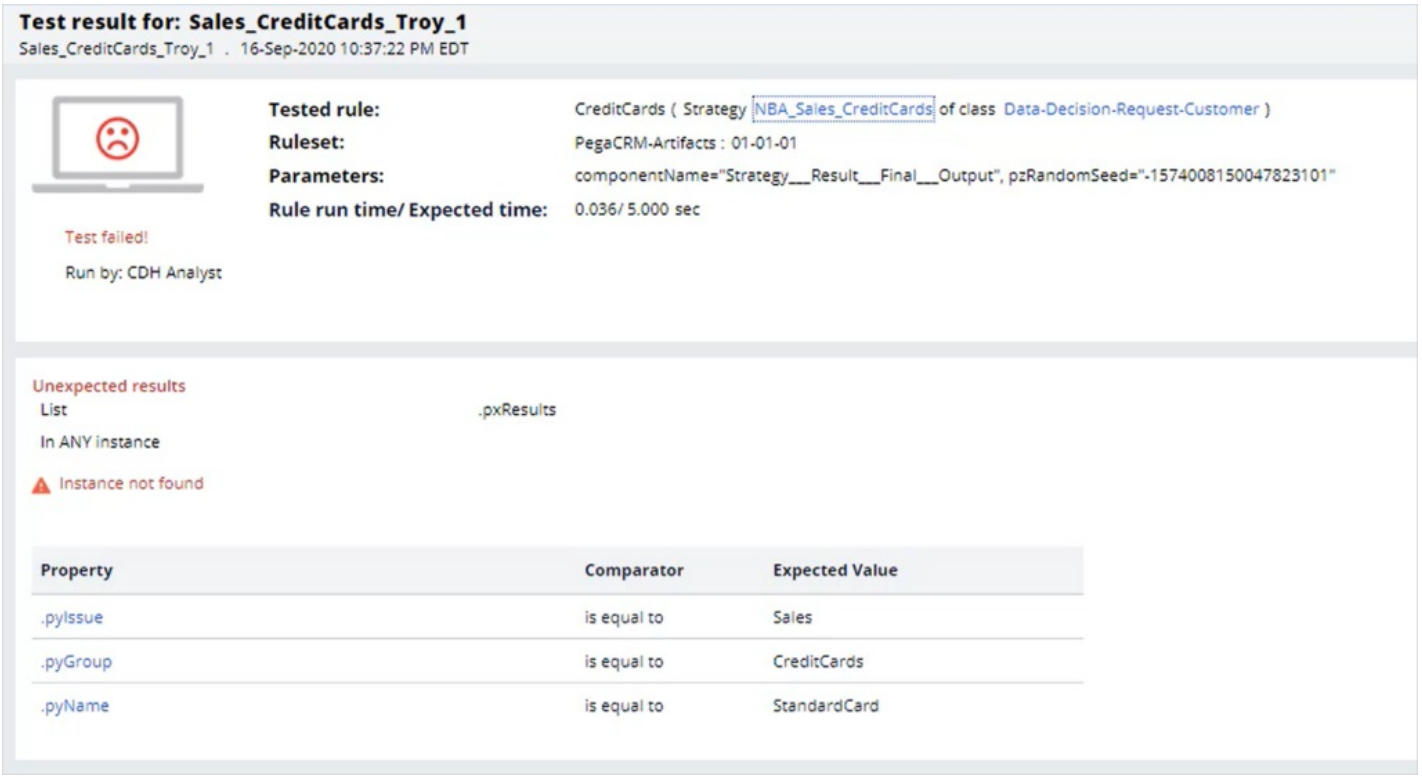
So, let's see how this engagement policy change affects these test results. To define an engagement policy, click Standard card.

Define an Applicability condition so that only customers who have a minimum average balance of $2000 qualify for the actions.

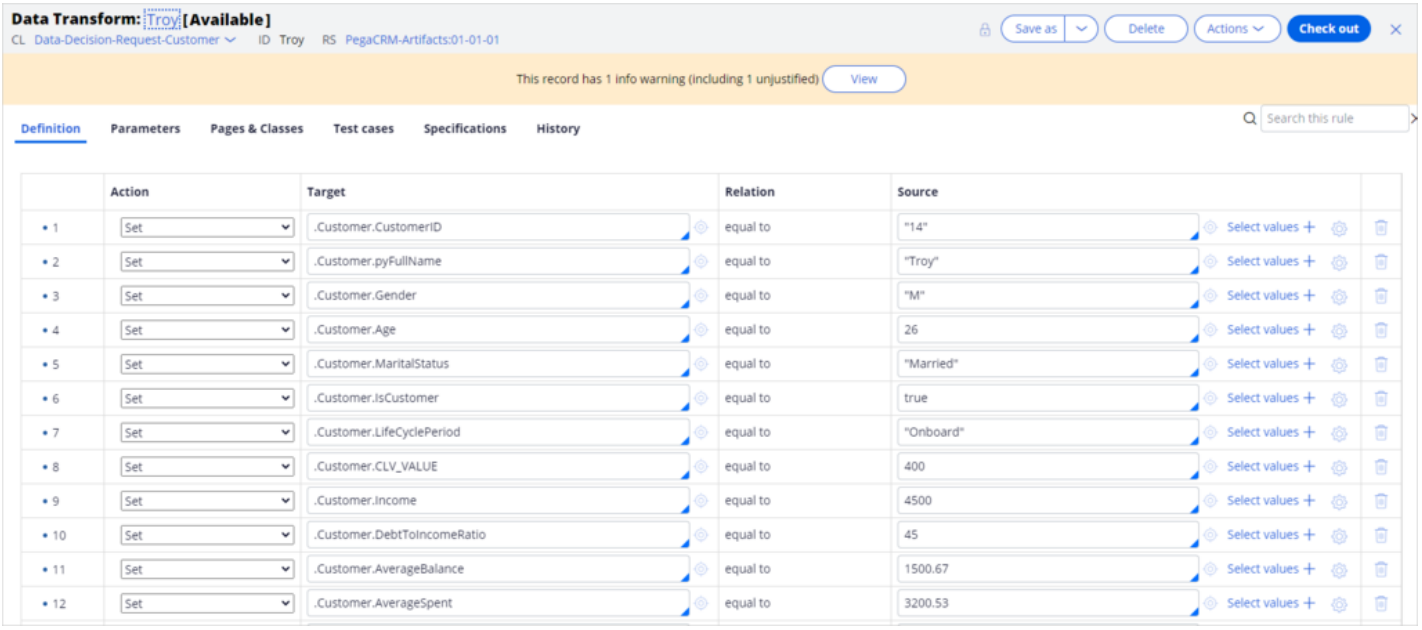
Run all the tests to see how this engagement policy configuration affects the test cases.

Observe that Troy fails the test. This means he does not satisfy all the existing engagement policies. Analyze the results to see why the test did not give the expected outcome.

Per the report, he is not eligible for the Standard card and, thus, this test failed.



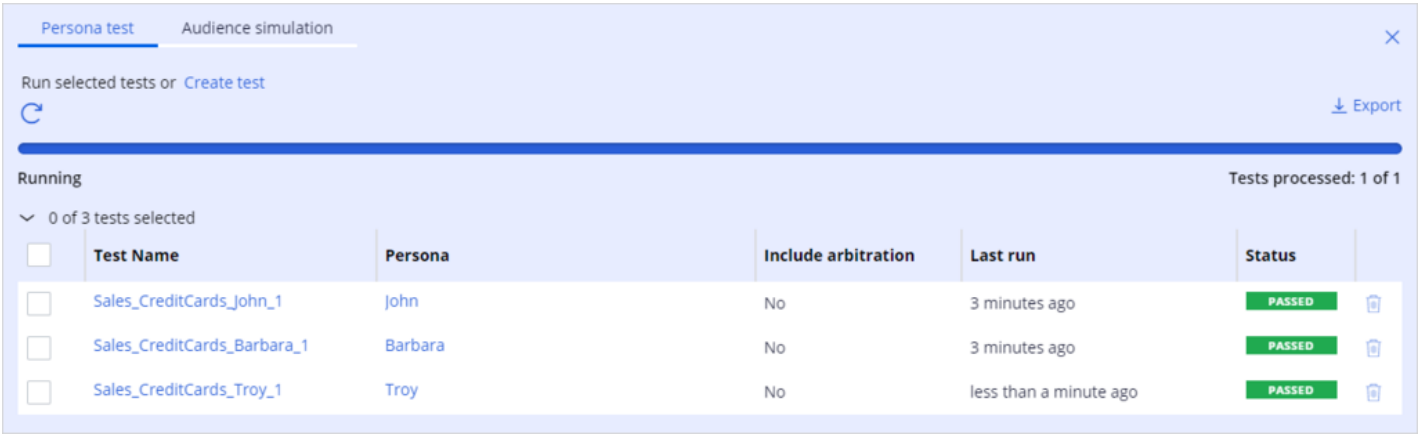
Troy's average balance is 1500, which is less than the defined applicability condition. Whereas, John's average balance is 3000 and Barbara's average balance is 3500, which are more than the defined applicability condition.



The Troy persona failed the test, but John and Barbara passed it.

Let's edit the test for Troy's persona to delete the Standard card assertion and run the test

So, you can design all these personas according to the business requirements and test them every time an engagement policy has changed.



You can also export the results into an excel sheet for further analysis.

This demo has concluded. 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.

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.
- Do not show an ad for six months if a customer clicks on the add 3 times over a period of 30 days.

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.

Video

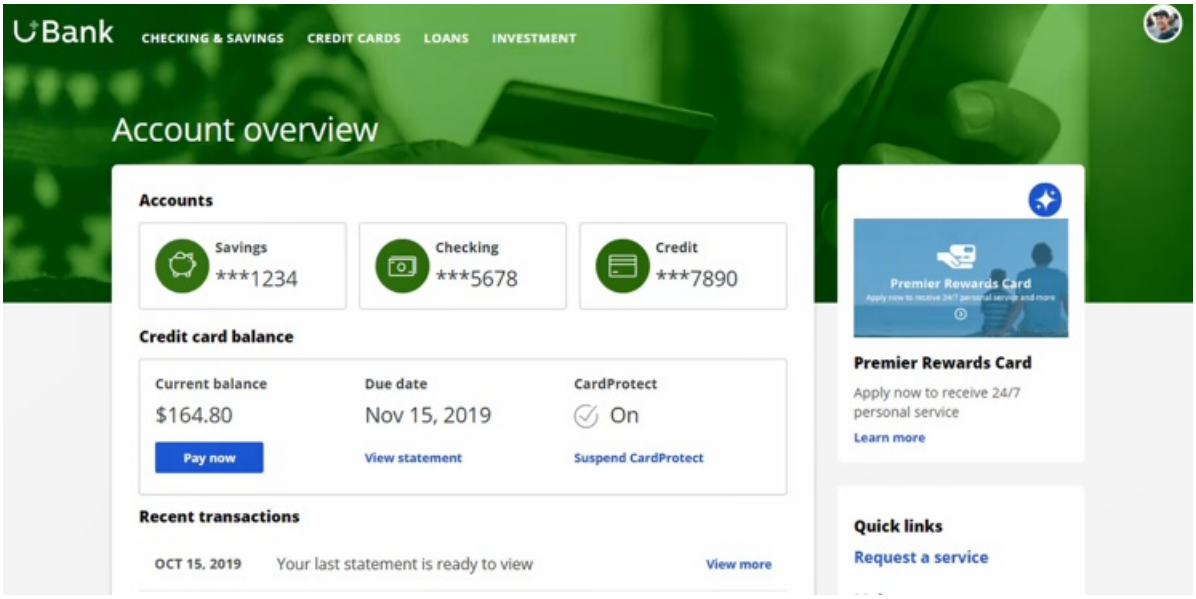
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.



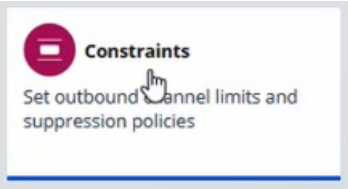
For a limited time period, the bank wants to automatically suppress offers that are shown or clicked too often.

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.



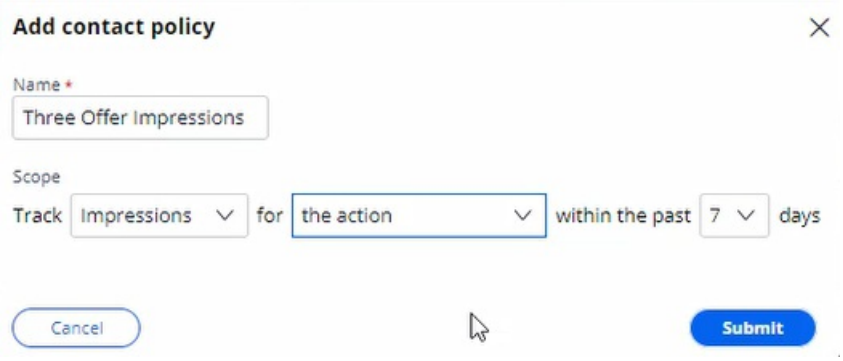
For the first requirement, showing a credit card a maximum of three times, configure a contact policy to track Impressions at the Action level.

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.



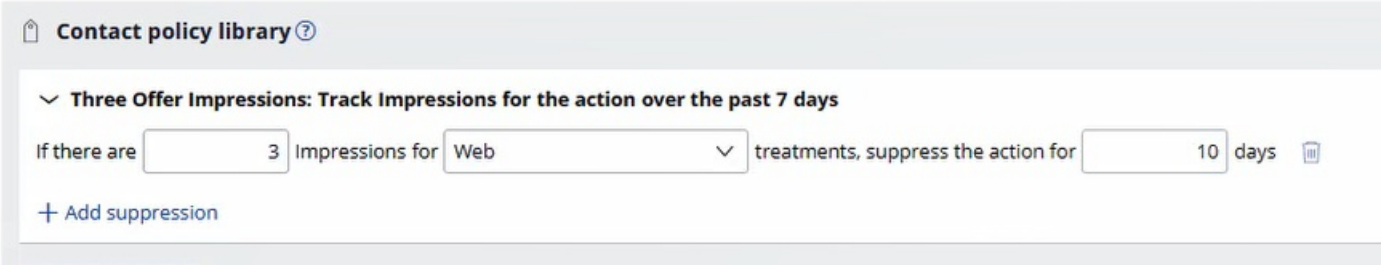
The newly configured contact policy has been added.

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.

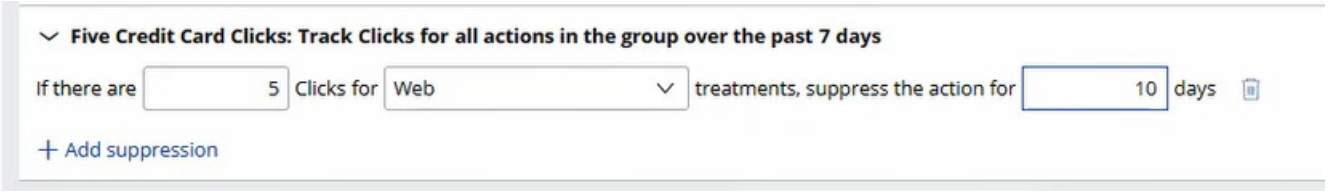


The next business requirement is to suppress the entire group of actions if there are five clicks for web treatments.

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 action 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.

Defining action suppression rules -- Thu, 07/23/2020 - 06:05
To get the full experience of this content, please visit https://academy.pega.com

Action arbitration

Introduction

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. Arbitration aims to balance customer relevance with business priorities by weighing numerical values for the following factors: propensity, context weighting, action value, and business levers. Learn to create a simple formula for arriving at a prioritization value, which is used to select the top actions.

Video

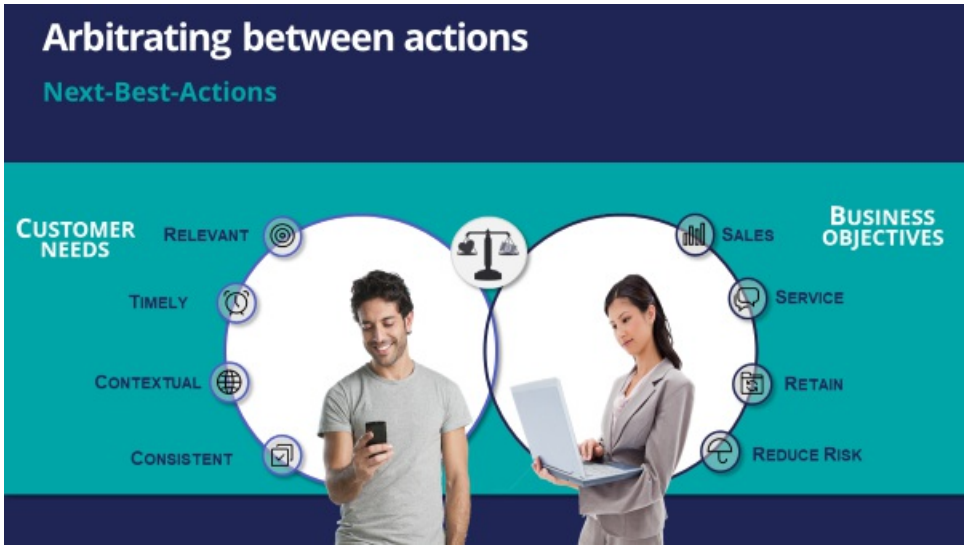
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This video explains the concept of action arbitration.

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-Action.

Every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business manager.

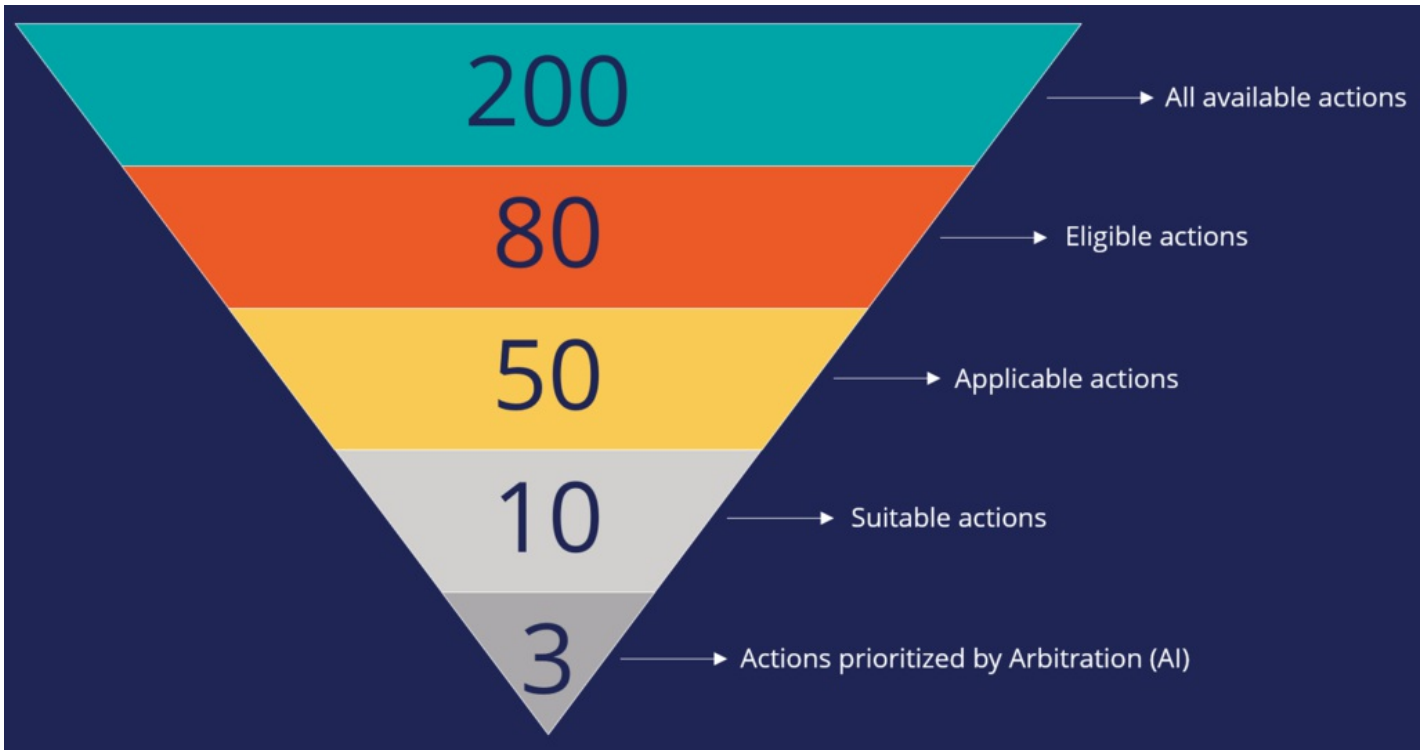


U+ Bank, a retail bank, has several actions for its customers and has configured engagement policies to suit both customer needs and business objectives.

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

final set of available actions.

These actions move through one final stage before being presented to customers: the arbitration stage. Arbitration is used to prioritize and choose the best actions based on what is relevant for the customer right now.



Arbitration aims to balance customer relevance with business priorities. The factors weighed are**Propensity**, **Context Weighting**, **Action Value**, and **Business Levers**, each represented by numerical values. A simple formula is used to arrive at a prioritization value, which is used to select the top actions. The number of top actions selected depends on the channel of interaction. For example, the top three actions, plus two tiles and one hero treatment, can be selected for display on a bank's website.

Propensity is the likelihood of a customer responding positively to an action; this is calculated by AI. For example, the higher the likelihood of a customer accepting an offer, the higher the Propensity value for that offer.

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. The priority of an action is increased by a specified value when the context is detected.

Action 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, an unlimited data plan is more profitable than a limited data plan. So, in a situation where a customer is eligible for both plans, the unlimited plan has higher priority.

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.

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.

Consider an example where three actions are selected for arbitration. At the moment, only the Propensity is used for prioritization.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
	Sales	Credit cards	Gold Card	0.5	1	1	1	0.5
	Retention	Proactive	10% discount	0.55	1	1	1	0.55
	Service	Administrative	Address change	0.4	1	1	1	0.4

Action arbitration with propensity before prioritization

The result of the arbitration is that the top action is the one with the highest Propensity.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Retention	Proactive	10% discount	0.55	1	1	1	0.55
2	Sales	Credit cards	Gold Card	0.5	1	1	1	0.5
3	Service	Administrative	Address change	0.4	1	1	1	0.4

Action arbitration with propensity after prioritization

Examine what happens when Context Weighting together with Propensity are considered for arbitration. For example, if the intent of a customer calling customer service is to change their address, the Context Weight of a Service action increases.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Retention	Proactive	10% discount	0.55	1	1	1	
2	Sales	Credit cards	Gold Card	0.5	1	1	1	
3	Service	Administrative	Address change	0.4	2	1	1	

Action arbitration with context weight before prioritization

As a result, the Arbitration caters to the current need of the customer and presents a Service action as the top action for the

customer. Thus, the Arbitration caters to the current need of the customer and presents a Service action as the top action for the customer.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Service	Administrative	Address change	0.4	2	1	1	0.8
2	Retention	Proactive	10% discount	0.55	1	1	1	0.55
3	Sales	Credit cards	Gold Card	0.5	1	1	1	0.5

Action arbitration with context weight after prioritization

Consider another scenario in which a customer is eligible for two credit cards and two other actions. Now, consider that the Action Value is also used in arbitration when prioritizing. In this case, the Platinum Card is assigned a higher value by the business than the Gold Card.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Credit cards	Gold Card	0.6	1	1	1	
2	Sales	Credit cards	Platinum Card	0.55	1	2	1	
3	Retention	Proactive	10% discount	0.2	1	1	1	
4	Service	Administrative	Address change	0.1	1	1	1	

Action arbitration with action value before prioritization

Thus, the arbitration selects the Platinum Card as the top action.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Credit cards	Platinum Card	0.55	1	2	1	1.1
2	Sales	Credit cards	Gold Card	0.6	1	1	1	0.6
3	Retention	Proactive	10% discount	0.2	1	1	1	0.2
4	Service	Administrative	Address change	0.1	1	1	1	0.1

Action arbitration with action value after prioritization

Finally, consider an example in which all four parameters are used for arbitration. In this case, U+ Bank wants to promote two new checking account offers under the Sales issue. The bank sets a higher Business Lever value for the Checking Accounts actions.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Credit cards	Gold Card	0.6	1	1	1	
2	Sales	Credit cards	Platinum Card	0.55	1	1	1	
3	Sales	Checking Accounts	Premium Checking	0.55	1	1	2	
4	Sales	Checking Accounts	Student Checking	0.5	1	1	2	
5	Retention	Proactive	10% discount	0.2	1	1	1	
6	Service	Administrative	Address change	0.1	1	1	1	

Action arbitration with business levers before prioritization

Although the Propensity of the Checking Accounts actions is low, they are selected as the top actions due to their high Lever values.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Checking Accounts	Premium Checking	0.55	1	1	2	1.1
2	Sales	Checking Accounts	Student Checking	0.5	1	1	2	1
3	Sales	Credit cards	Gold Card	0.6	1	1	1	0.6
4	Sales	Credit cards	Platinum Card	0.55	1	1	1	0.55
5	Retention	Proactive	10% discount	0.2	1	1	1	0.2
6	Service	Administrative	Address change	0.1	1	1	1	0.1

Action arbitration with business levers after prioritization

Action prioritization with AI

Introduction

Explore how AI-based arbitration works and how AI predicts customer behavior. Arbitration aims to balance customer relevance with business priorities. To select the top actions, a formula is used to arrive at a prioritization value. The formula uses the propensity value, which is calculated using AI. Propensity is the predicted likelihood of positive behavior, such as the likelihood of a customer accepting an offer.

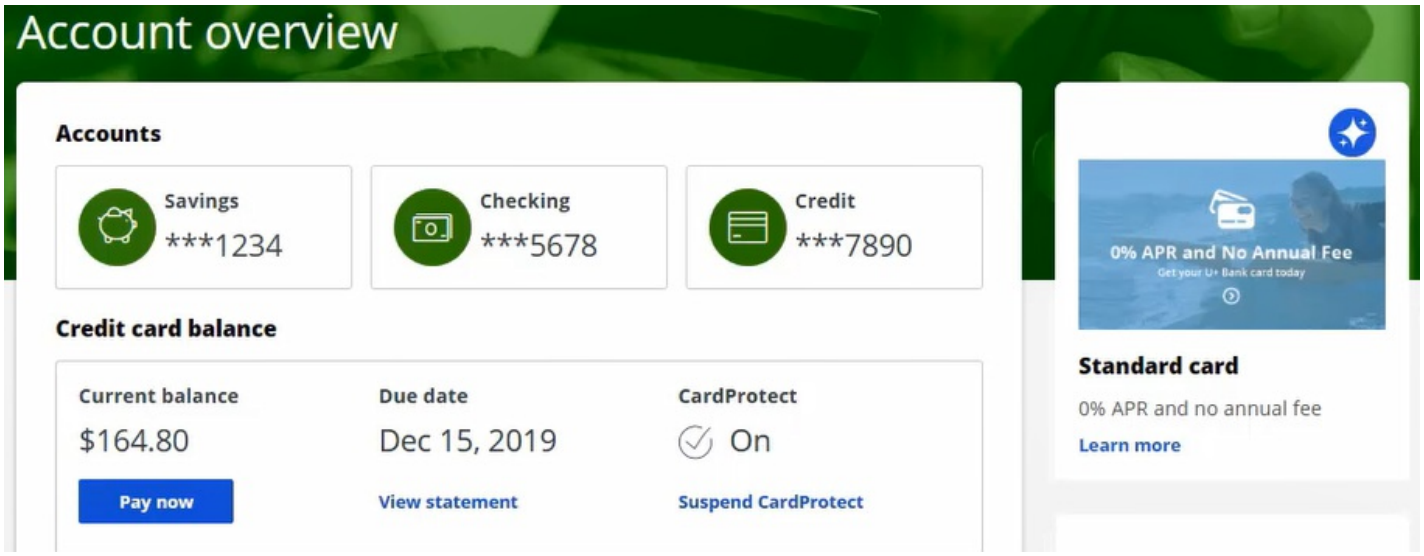
Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

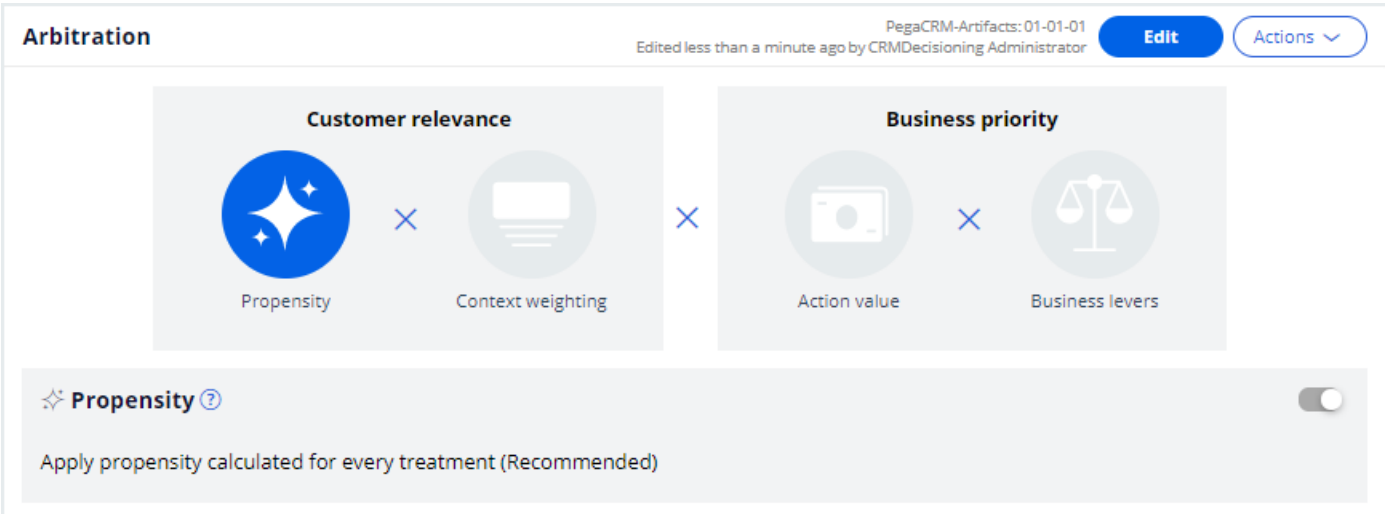
This demo will explore how AI-based arbitration works and explain how AI predicts customer behavior.

U+, a retail bank, uses the Pega Customer Decision Hub™, to display marketing offers to customers on its website. The bank would like to display more relevant offers to customers based on their behavior.
Troy, a customer, qualifies for two credit card offers. When he logs into the bank's website, he sees the top offer for him, the **Standard Card.**



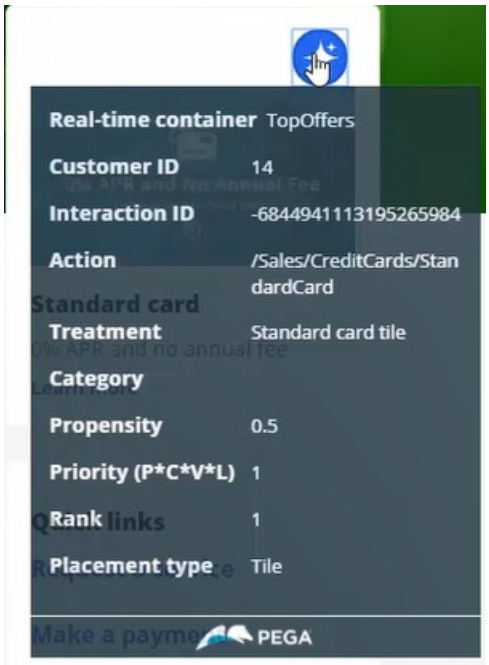
These are the Arbitration settings defined in Pega Customer Decision Hub's Next-Best-Action Designer. Arbitration aims to balance customer relevance with business priorities. To achieve this balance, Propensity (P), Context weighting (C), Action value (V), and Business levers (L) are represented by numerical values and plugged into a simple formula, P * C * V * L. This formula is used to arrive at a prioritization value, which is used to select the top actions.

Notice that only Propensity is currently enabled. Propensity is the predicted likelihood of positive behavior, such as the likelihood of a customer accepting an offer. The value of Propensity is calculated using AI.

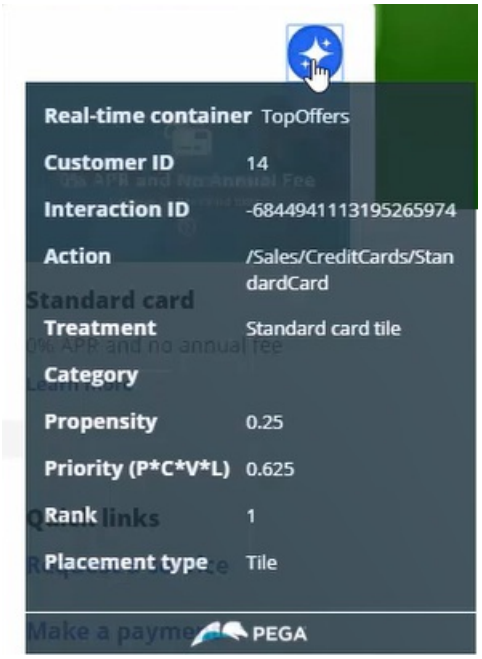


Note the Propensity and Priority values of the **Standard Card**. The Propensity for every action starts at 0.5 or 50%, the same as the flip of a coin. This is because in the beginning, the AI has no past customer behavior on which to base its predictions. Propensity is one of the factors used to arbitrate between relevant offers and select the top offer for a customer.

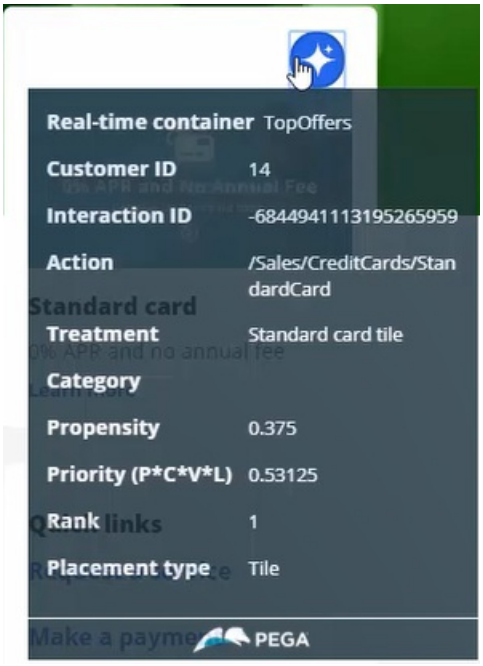
Notice that although only Propensity is enabled for arbitration, the value of Priority, which is currently based on Propensity only, does not match the Propensity value. This is because the Priority calculation doesn't use the raw Propensity value directly. Instead it uses the value resulting from a built-in Propensity smoothing mechanism. The Propensity smoothing mechanism is used to jump-start the process of AI learning. It helps to equalize the sudden changes in Propensity values calculated by AI during the initial phase of its learning, when it has yet to gather enough customer behavior data to make accurate predictions.



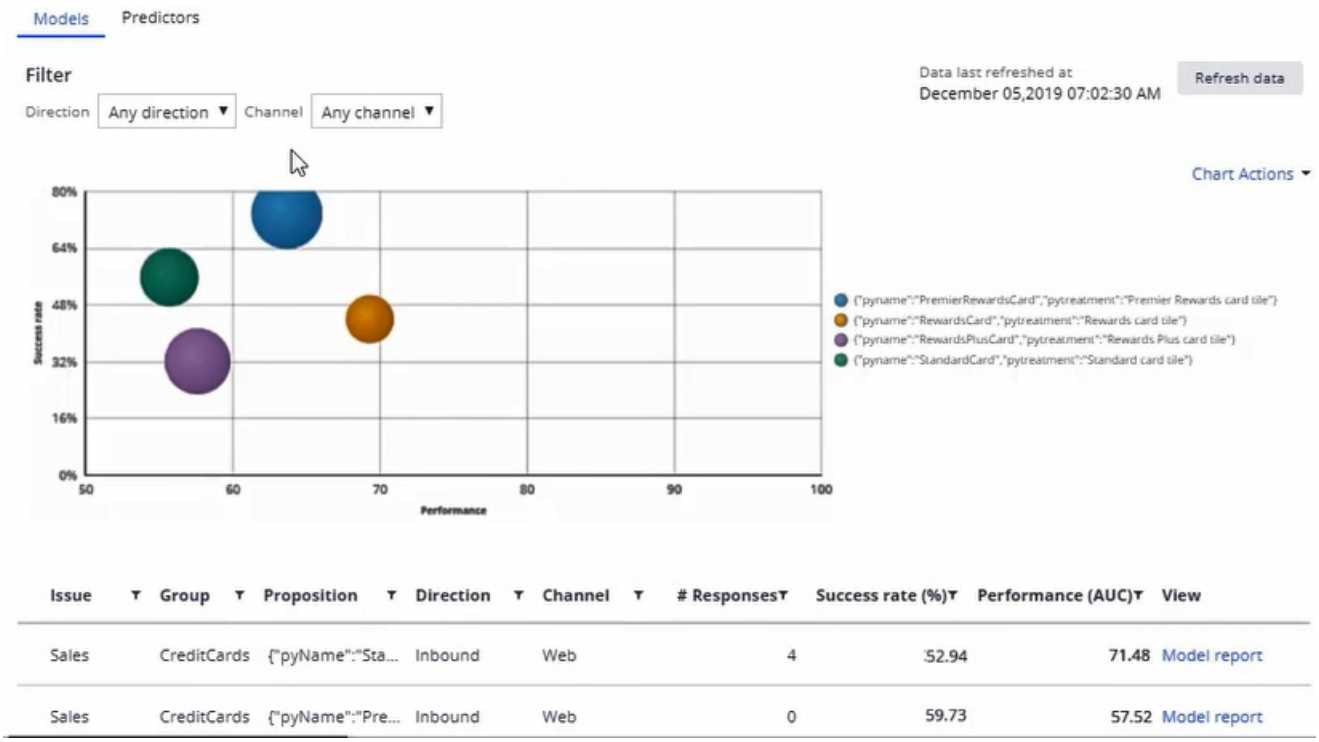
If Troy doesn't click on the current offer this time, a different offer will be shown the next time he visits the website. The next offer Troy is eligible for, the Rewards Card, is then selected for display. If Troy ignores this card as well, by not clicking on it, then the next time he logs in, the Standard Card offer will be displayed again.Why this behavior? First, Troy only qualifies for these two credit card offers. Second, the AI model behind these offers is configured to treat an Impression as a negative behavior. In other words, when a customer is presented with an offer but doesn't click on it, the AI records this as a negative behavior. As a result, the Propensity, and therefore the Priority, of the not-clicked-on offer decreases. Notice that the Propensity value of the **Standard Card** offer dropped from 0.5 to 0.25.



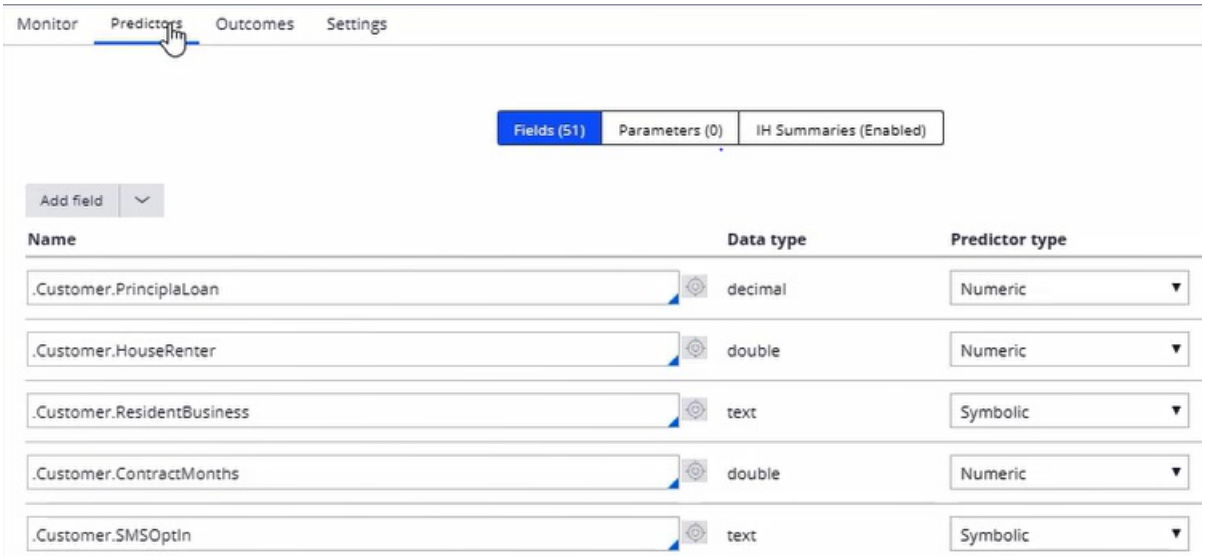
Now, if Troy clicks on the 'Learn more' link for the**Standard Card** offer, a positive response is recorded, and thus the Propensity value of the **Standard Card** increases.



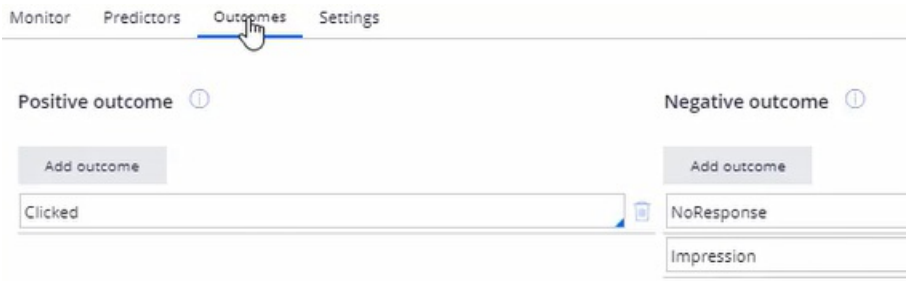
The Customer Decision Hub is configured to calculate the Propensity for each Treatment. To understand how this works, let's examine the AI behind a Treatment. This pop-up window provides a summary of the AI behind this Treatment. In the Pega Customer Decision Hub, the AI that determines the Propensity for positive behavior towards an action or Treatment is called an adaptive model. From here, you can navigate to the adaptive model itself.



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.



This adaptive model considers an Impression, when a marketing offer is displayed on a website, a negative behavior. It considers a Click a positive behavior.



Therefore, when a customer sees an offer message but doesn't click on it, the model records a negative behavior.

The monitoring tab provides an overview of the model's performance. The business can use this information to assess the contribution of the model's predictions with respect to the success of the actions.

The model report provides more insight into the AI model itself. This AI model is automatically generated by the system, and it adapts its prediction algorithm in real-time, based on incoming customer responses. The report shows more information about the predictors, such as how they are grouped and details a data scientist can use to analyze the current health of the model and diagnose any potential problems.

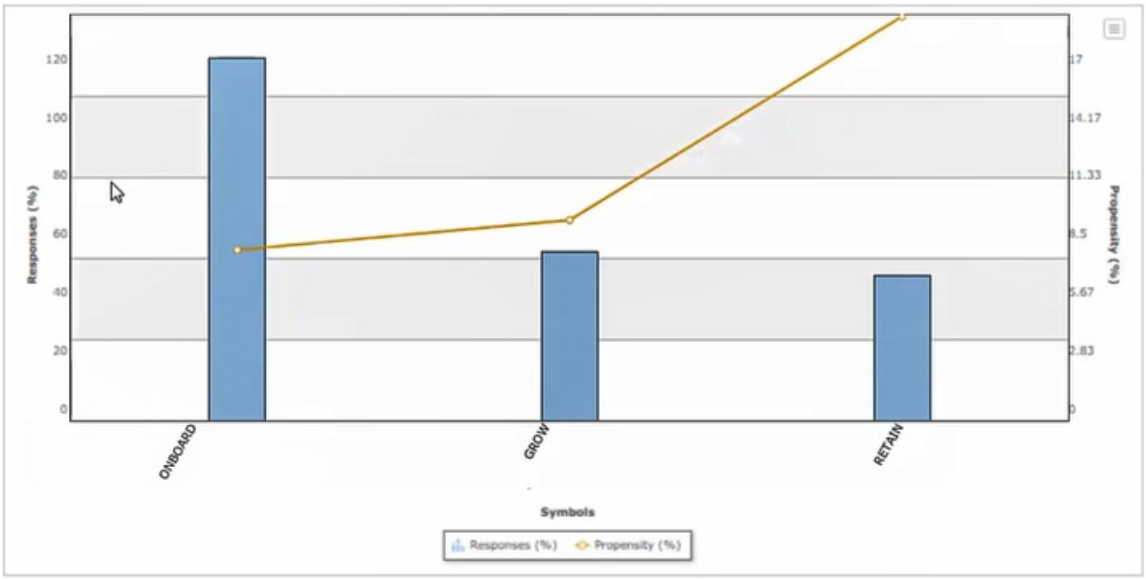
Predictors | Score distribution | Trend

Correlated predictors are grouped and the best performing predictors become active in the model.

Predictors	Status	Type	Performance (AUC)	Range/Symbols(#)	Bins(#)
IH.Web.Inbound.Impression.pyHistoricalOutcomeCount	Active	Numeric	90.00	[10.0; 18.0]	3
IH.Web.Inbound.Impression.pxLastOutcomeTime.DaysSince	Active	Numeric	60.00	[0.0; 0.0]	2
Customer.LifeCyclePeriod	Active	Symbolic	60.00	1.00	1
Customer.AverageCallsOut	Inactive	Numeric	50.00	[0.0; 0.0]	2
Customer.CLV_VALUE	Inactive	Numeric	50.00	[400.0; 400.0]	1

In the Predictor report, you can examine the performance of individual predictors. Let's examine the LifeCyclePeriod predictor.

This a predictor of type Symbolic. The individual Predictor report shows that a customer whose lifecycle stage is RETAIN is most likely to accept the Standard Card action in the web channel.



The behavior of one customer can influence the Propensity calculation for other customers with a similar profile. For example, when Robert, a customer with a profile similar to Troy, logs in, he is shown the same offer as Troy. The same AI model is behind the Treatments for both customers, so Robert's action will influence Troy's Propensity score.

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.

Video

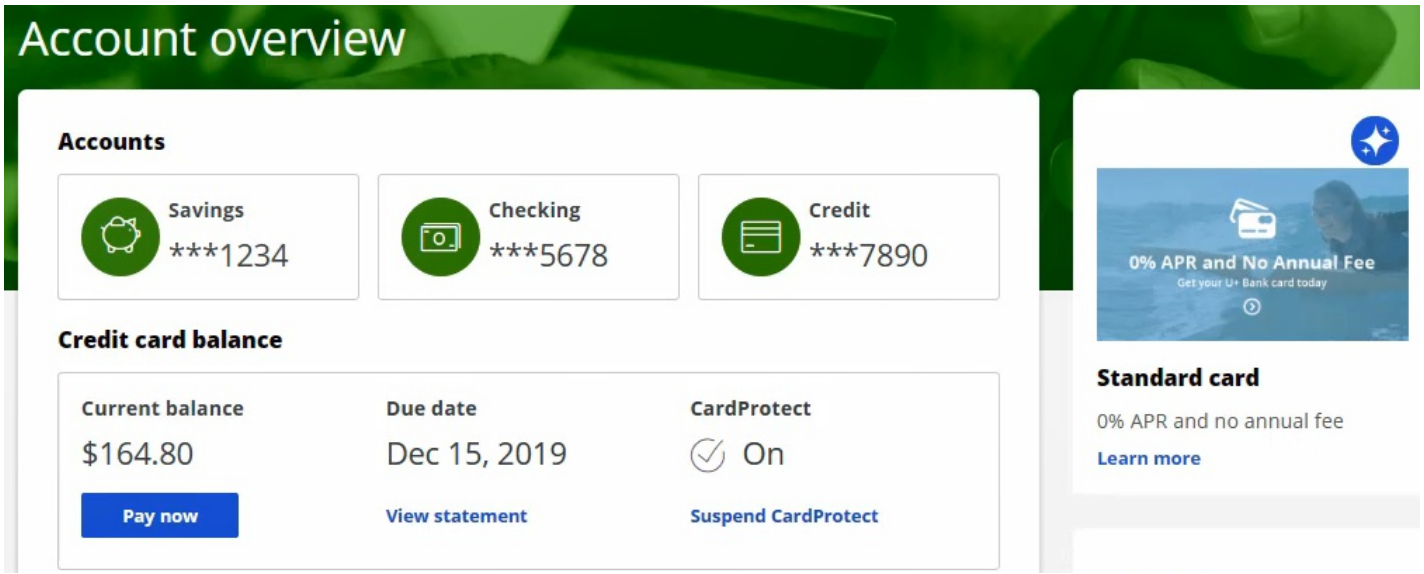
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

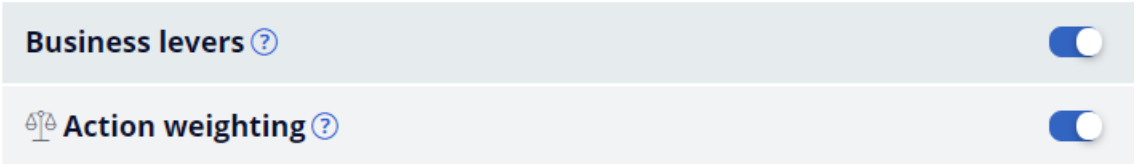
U+, a retail bank, noticed that one of the offers, the Rewards card offer, was not presented frequently enough due to its low propensity because customers ignored it during the initial launch.

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.

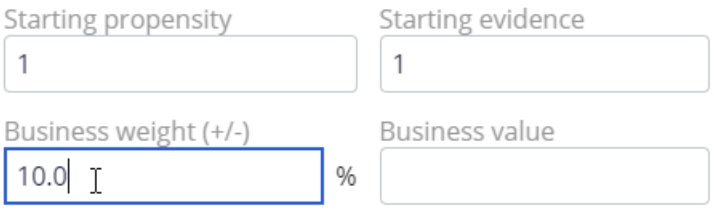


Now, due to an internal ad-hoc priority, the bank wants to boost the chances of the Rewards Card being selected as the top offer. That is, the bank would like to present more relevant offers to customers based on not only their behavior but also on business priorities.

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.

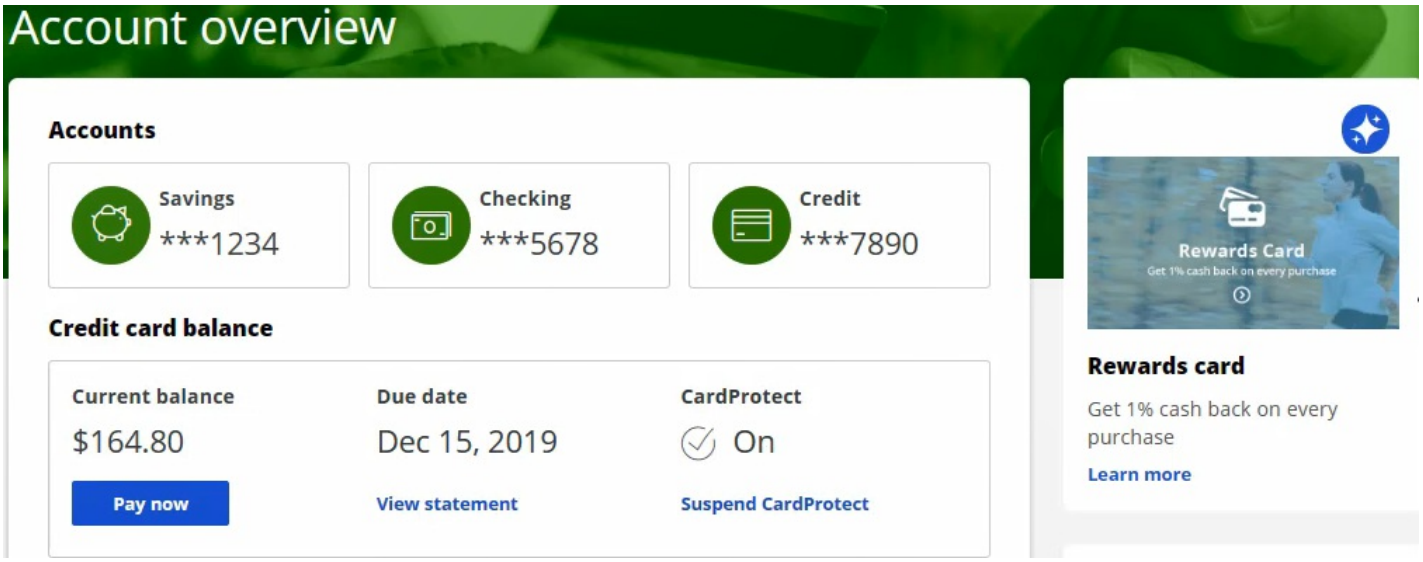


In this case, the bank wants to boost the Rewards Card. So, open the Rewards Card offer. Edit the offer to set a business weight, a value in percentage, that is required to boost the offer. In this case, U+ wants to increase the changes of this action selected by 10%.



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.



Business use case: Cross-sell on the web extended

Introduction

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

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

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.

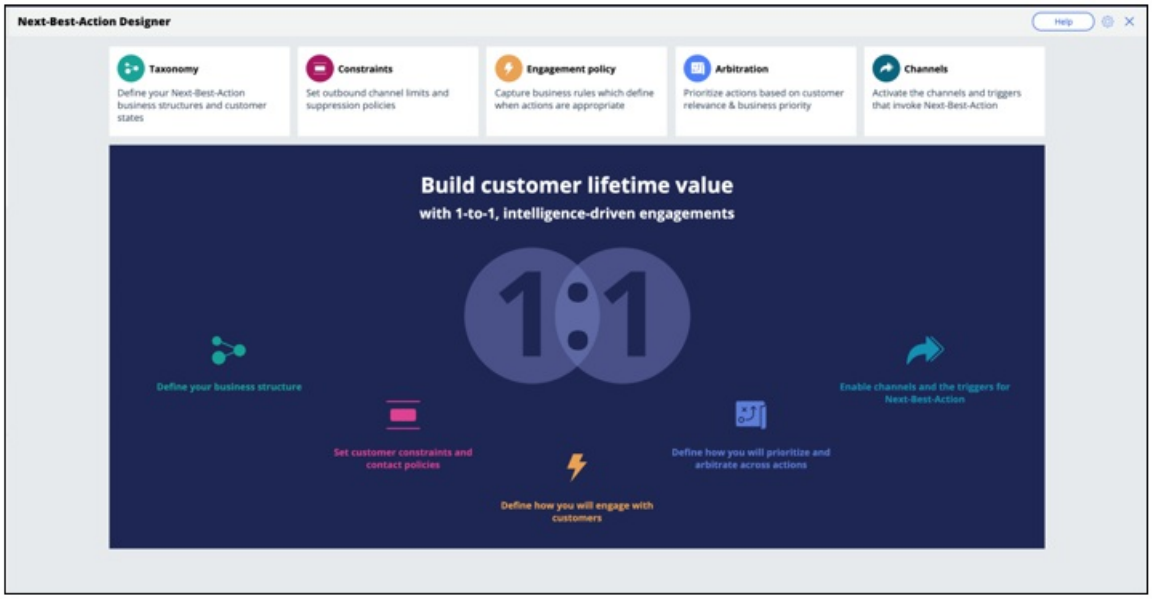
The bank is using the Pega Customer Decision Hub™ to recommend more relevant banner ads to its customers when they visit their personal portal. In the start-up phase, U+ successfully implemented all their use cases using Next-Best-Action Designer.

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.

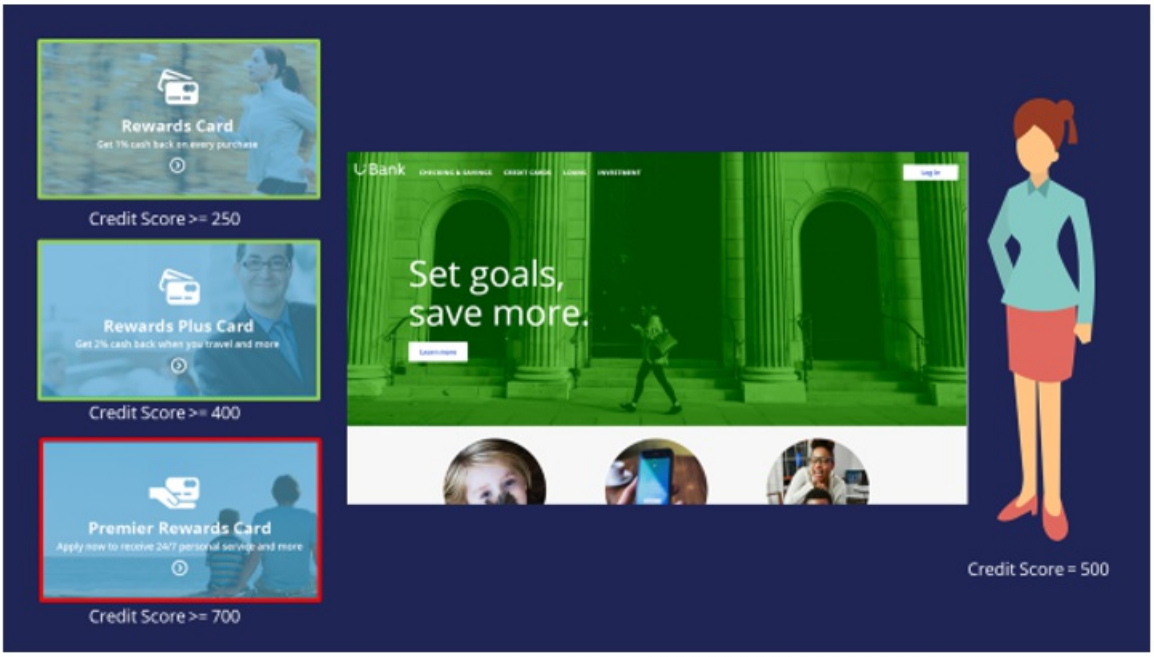
These decision strategies are a combination of the business rules and AI models that form the core of the Pega Customer Decision Hub, which determines the personalized set of Next-Best-Actions for each customer.



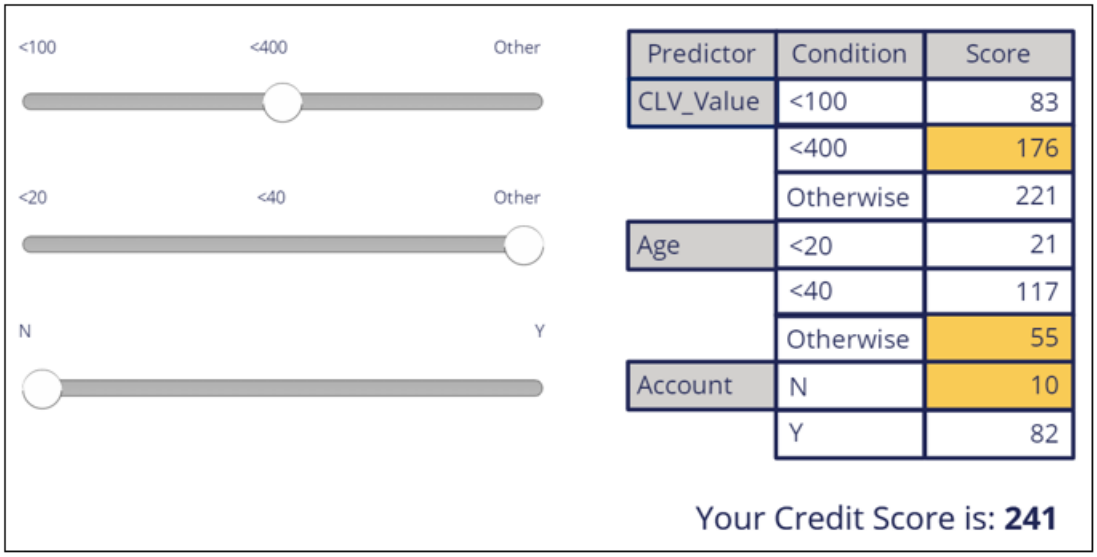
U+ Bank wants to implement additional use cases to meet new business requirements. These use cases require U+ bank to extend Next-Best-Action Designer capabilities.

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 has introduced some strict regulations for which they need new eligibility rules. U+ does not want to offer credit cards to customers whom they classify as 'high risk'. Customers are divided into risk segments from AAA to CCC based on their outstanding loan amount and credit score. In the beginning, only customers in the risk segments BBB and CCC will be eligible for credit cards.

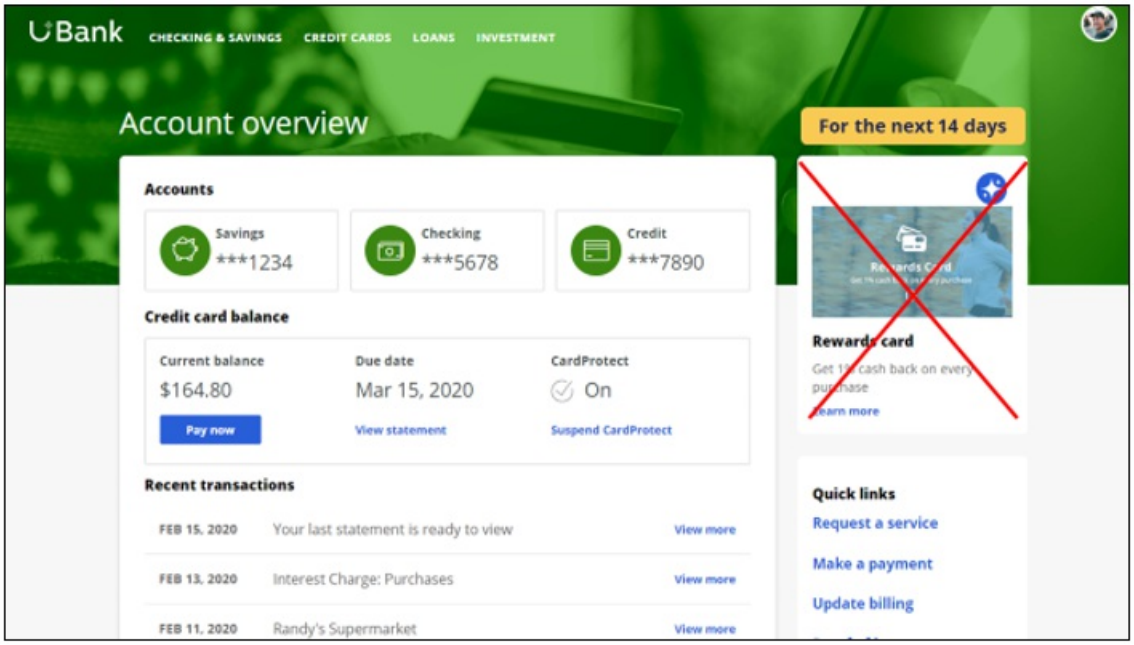
Risk Segmentation

Condition	Risk segment
If Outstanding loan amount >= $50000	AAA
If Outstanding loan amount is between $10000 and $25000 AND Credit score is between 600 and 800	BBB
If Outstanding loan amount is less than $10000 AND Credit score is between 100 and 200	BBB
If Outstanding loan amount is less than $10000 AND Credit score is more than 200	CCC
If Outstanding loan amount AND Credit score falls in any other range	AAA

To implement this, you use a special Eligibility condition in Next-Best-Action Designer. The Eligibility condition leverages a decision strategy that uses the scorecard to determine the customer's credit score, which it passes as an argument to a decision table. The decision table then determines which risk segment the customer is in.

The next use case is about adding more time periods for tracking customer behavior.

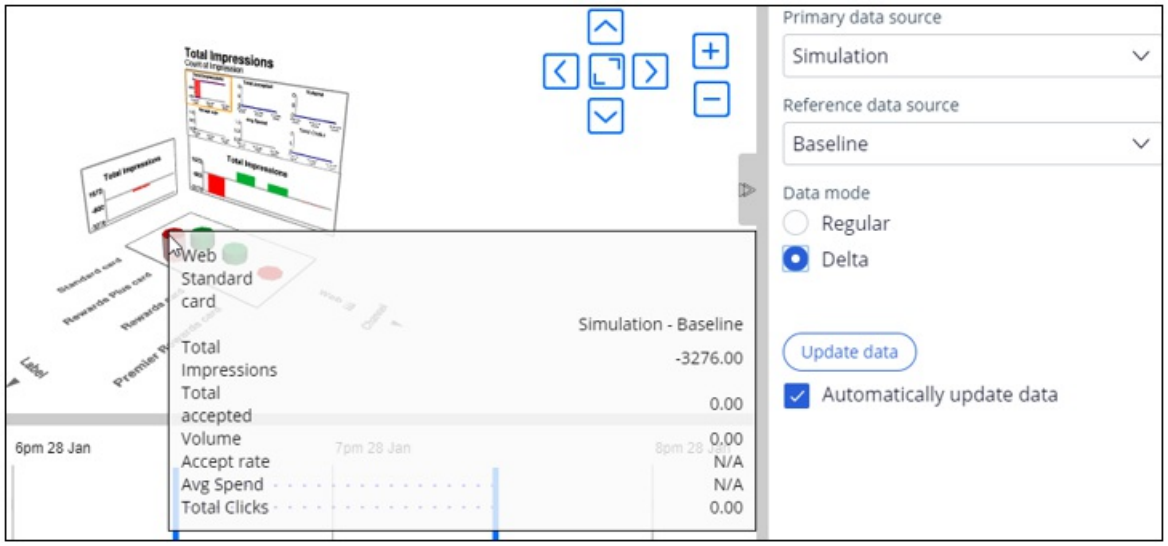
In this use case, the bank does not want to show the same offer to customers who have clicked on it three times in the last 14 days. By default, the Pega Customer Decision Hub tracks customer responses for a period of 7 or 30 days.



This use case requires an additional tracking time period of 14 days.

To add a new tracking time period you have to extend the decision strategies that are used to implement contact policies. This requires creating a new Interaction History Summary rule which tracks customer responses for 14 days.

Finally, while the bank is implementing these various changes, they want to understand what the impact will be on their Next-Best-Actions. Therefore, U+ Bank would like to run some simulations to test the effects of the changes. The results of the simulations can be analyzed using Visual Business Director.



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.

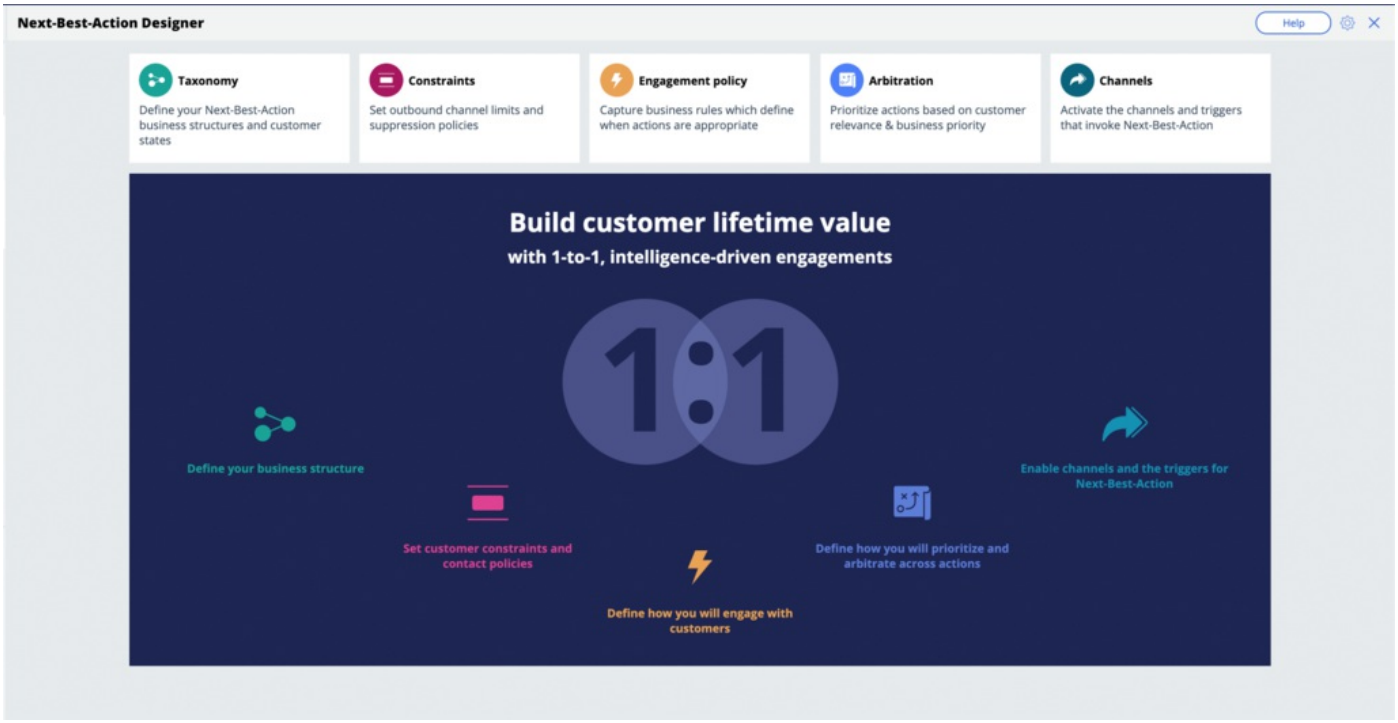
Business use case: Cross-sell on the web extended -- Fri, 07/24/2020 - 06:17
To get the full experience of this content, please visit https://academy.pega.com

Decision strategies

Next-Best-Action Designer

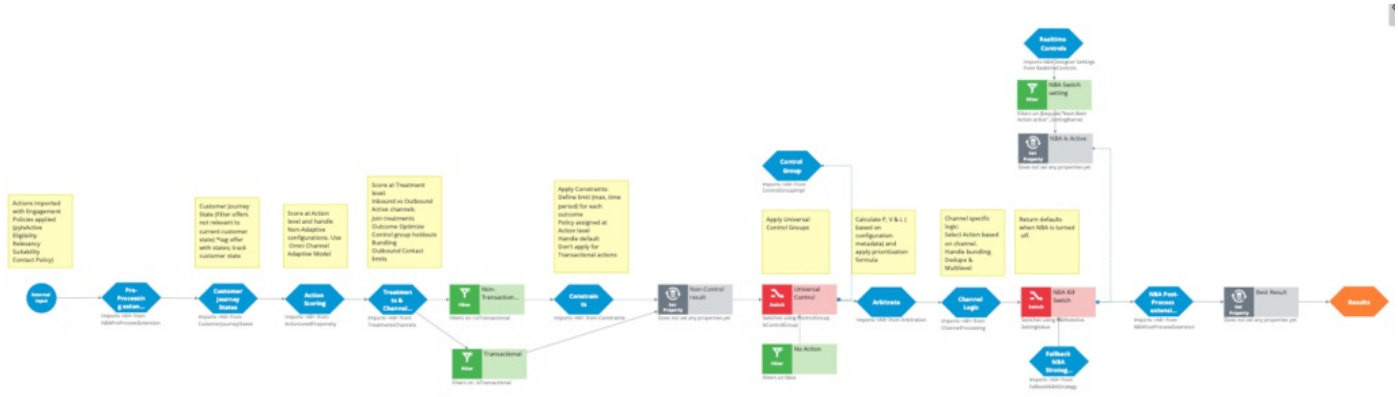
Next-Best-Action Designer guides you through the creation of a Next-Best-Action strategy for your business. Its intuitive interface, proven best practices and sophisticated underlying decisioning technology enable you to automatically deliver personalized customer experiences across inbound, outbound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

The Next-Best-Action Designer user interface allows you to easily define, manage and monitor Next-Best-Actions.



Next-Best-Action strategy framework

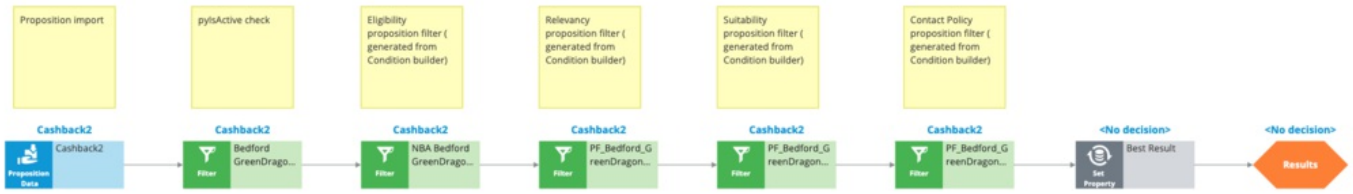
As you use the Next-Best-Action Designer user interface to define strategy criteria, the system uses these criteria to create the Next-Best-Action strategy framework. This framework leverages best practices to generate Next-Best-Action decision strategies at the enterprise level. These decision strategies are a combination of the business rules and AI models that form the core of the Pega Centralized Decision Hub, which determines the personalized set of Next-Best-Actions for each customer.



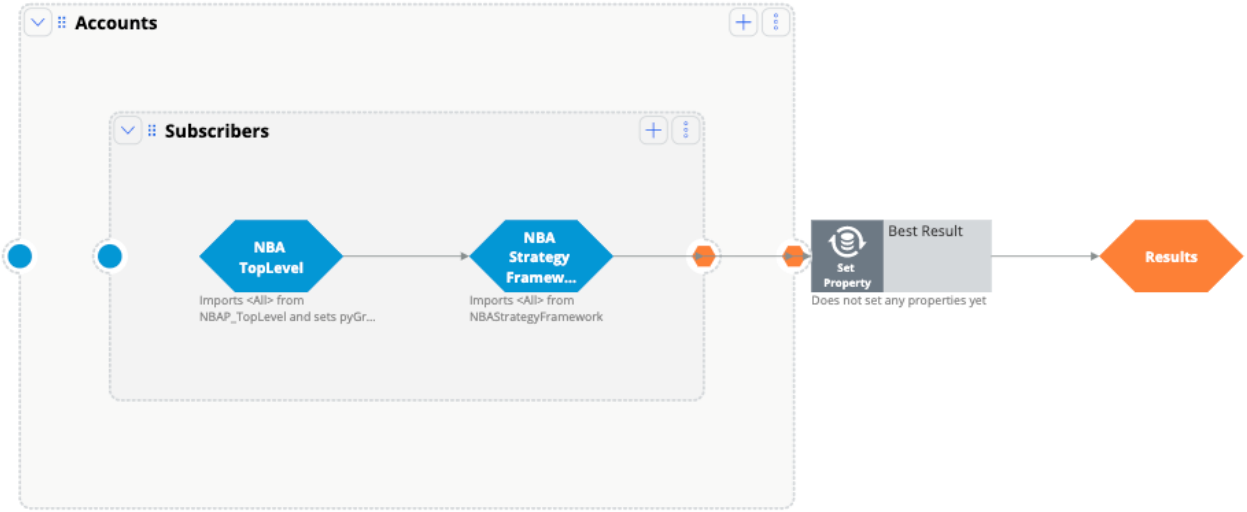
If you want to modify the strategy later, you can do that from Next-Best-Action Designer’s simple and transparent interface.

The strategy framework is applied to all relevant Actions and Treatments after you define a Trigger in the Next-Best-Action Designer **Channels** tab.

Each Trigger generates a strategy that first imports the Actions from the appropriate level of the business structure and then applies the Eligibility, Relevance and Suitability rules.



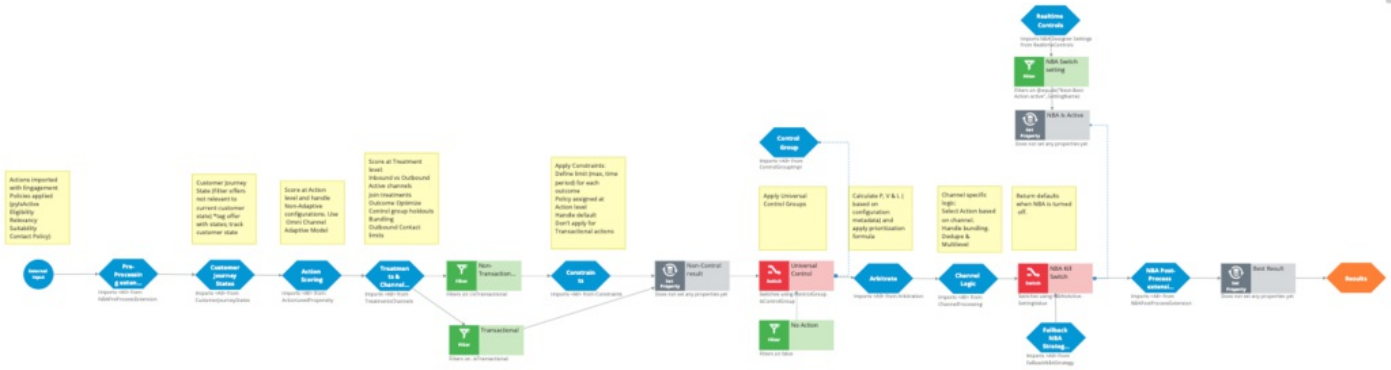
The strategy then passes these results to the strategy framework for processing.



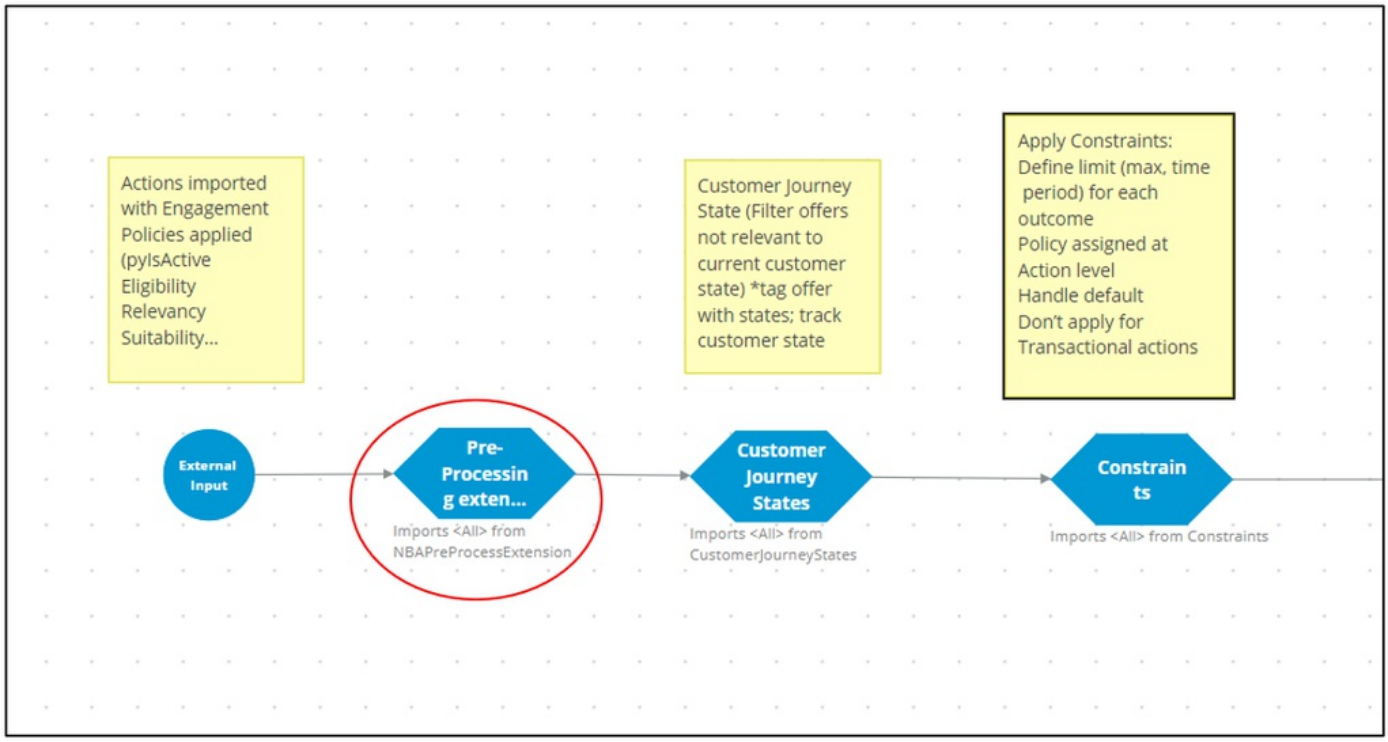
Strategy framework extension points

There are several extension points within the framework. An extension point is an empty rule or activity that is intended to be overridden to meet the specific needs of the application. When building an implementation of the current framework, the decision strategy designers must override the empty activity with a functioning interface to their customer master file.

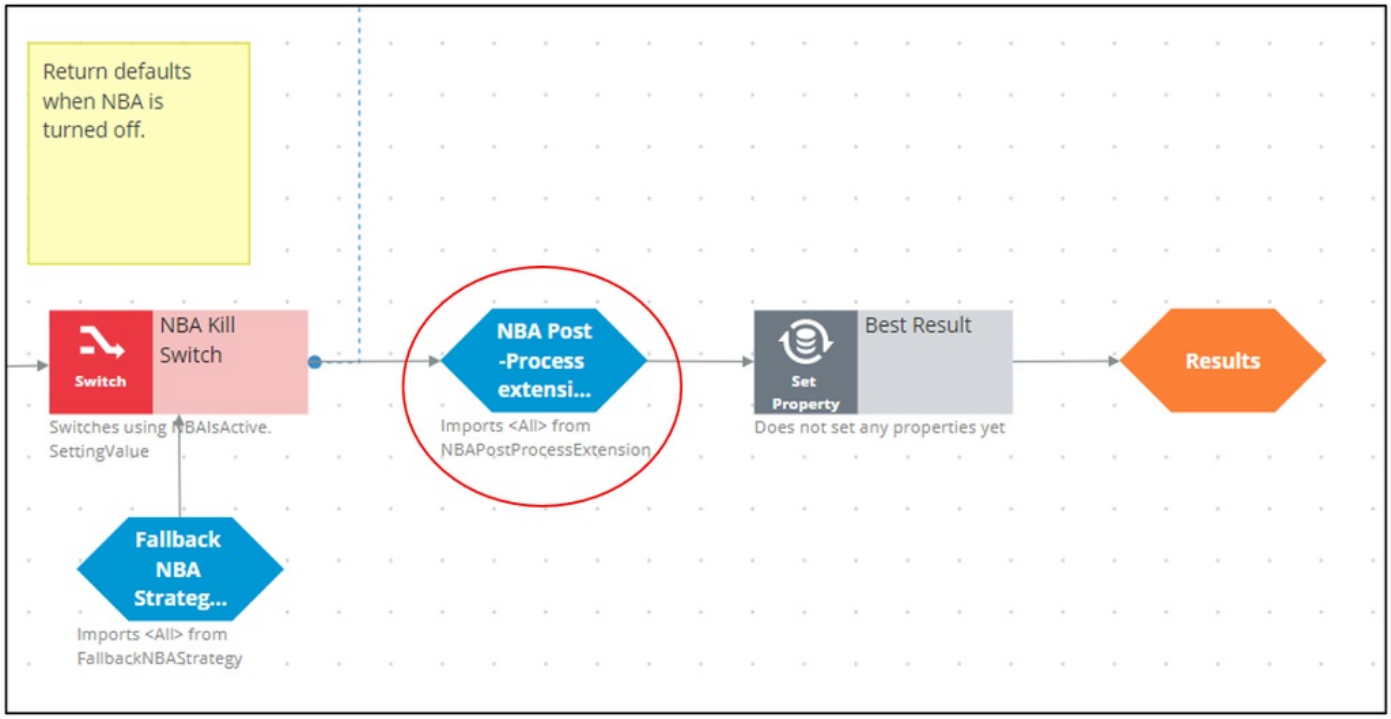
This is the NBA framework strategy when applied to each of the Actions.



The first component within the strategy framework is an extension point for any Action pre-processing you might need to perform.



The last functional component within the strategy framework is another extension point for any post-processing that must be performed.



Similarly, there are many other extension points such as the outbound limits extension points and business value extension points.

To ensure upgradeability, avoid overriding any part of the framework that is not a designated extension point.

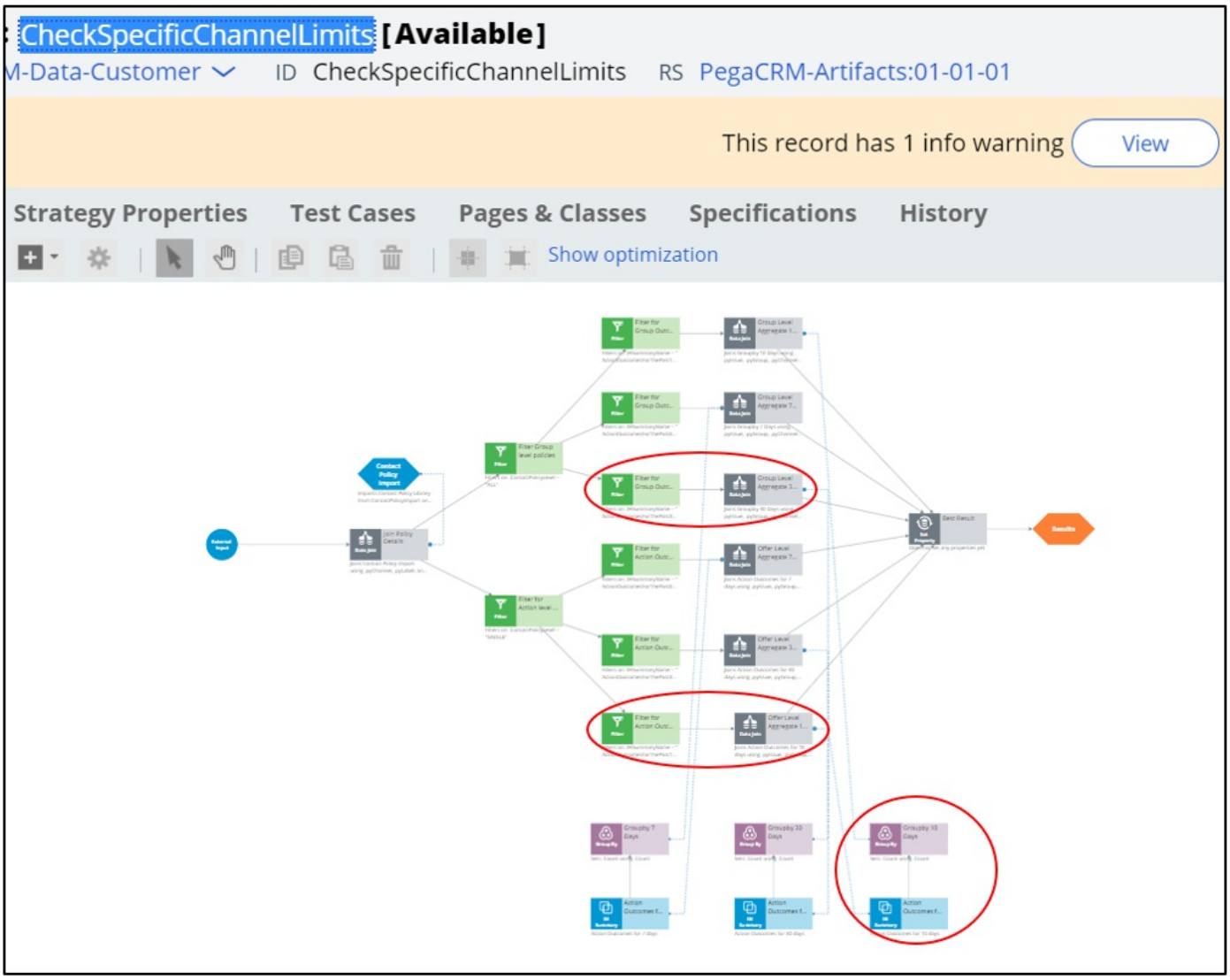
Also, the generated framework has some extension points where you can create strategies.

For example, while configuring values for Arbitration, you can specify a business value for an Action, or you can use a strategy the calculate the value. This can be done by adding a strategy to the existing framework.

Similarly, in defining the engagement rules, you can use a new strategy as a definition instead of an existing condition. Strategy designers can create such strategies from scratch using the decision strategy canvas.

Or, while defining the suppression rules, you can add a strategy to define new suppression rule limits instead of the existing 7 or 30 days.

For example, in the screenshot below, the CheckSpecificChannelLimits rule has been extended to have a 15-day limit:



In conclusion, the NBA Designer provides a guided and intuitive UI to bootstrap your application development with proven best practices. NBA designer generates the underlying strategies for you, which can be extended using existing values in the designated extension points or by building decision strategies from scratch, depending on the business requirement.

Decision strategies -- Fri, 07/24/2020 - 06:19
To get the full experience of this content, please visit https://academy.pega.com

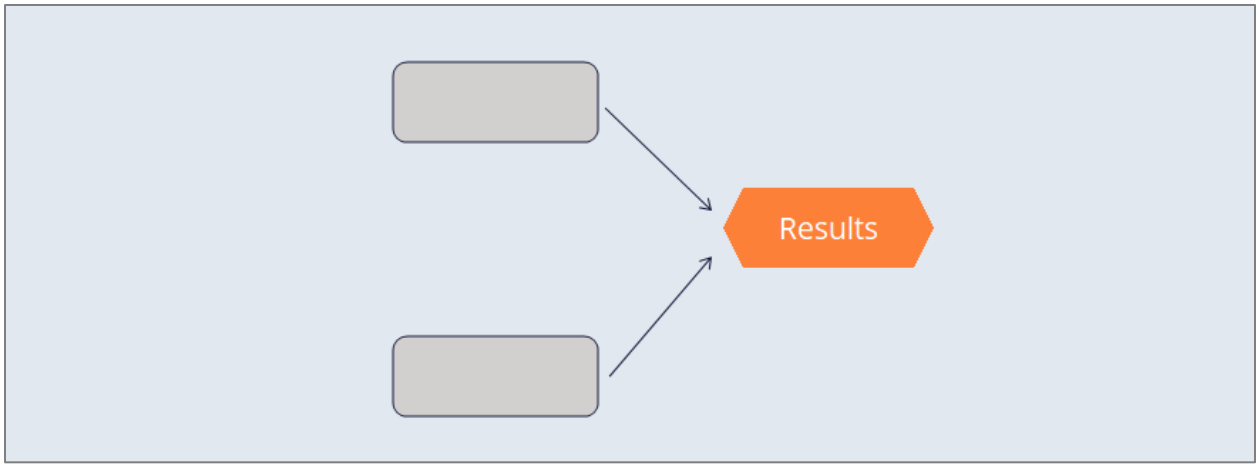
Introduction

Decision strategies drive the next best action and comprise a unit of reasoning represented by decision components. You use the Proposition Data component to import actions into a strategy canvas. The sequence of the components in the canvas determines which action is selected for a customer.

Click the **Play** button to learn more about decision strategies.

Screen1: U+ business scenario

U+, a telecom organization, wants to promote two new phones in the contact center: iPhone and Galaxy. Click the + icons to learn more about the elements of a decision strategy that is created for this requirement.



Decision Components: A decision strategy is comprised of building blocks called decision components. You can add and connect components to implement the business requirements.

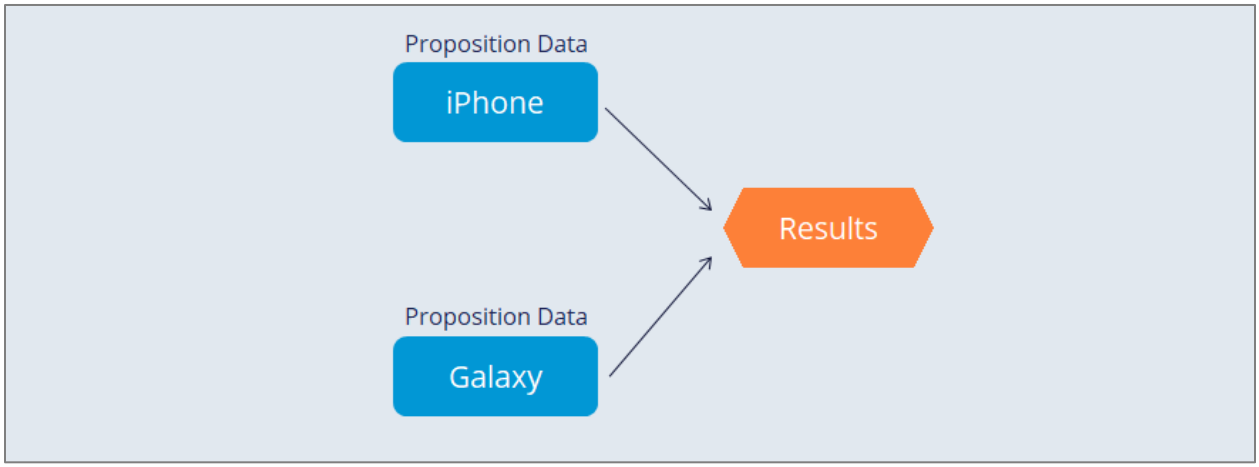
Arrows: An important element of the strategy canvas is the arrow. An arrow connects two decision components. A solid line means the data is copied from one component to another.

Strategy Canvas: In Pega, business users visually design decision strategies on what is known as a strategy canvas.

Screen2: Proposition Data component

The Proposition Data decision component imports the properties of an action. The result of this component is a flat list of all the properties.

Click the + icons on the proposition components to examine the components' results.



iPhone: This Proposition Data component outputs the Price and the Cost properties of the iPhone action.

Name: iPhone

Price: 150

Cost: 100

Galaxy: This Proposition Data component outputs the Price and the Cost properties of the Galaxy action.

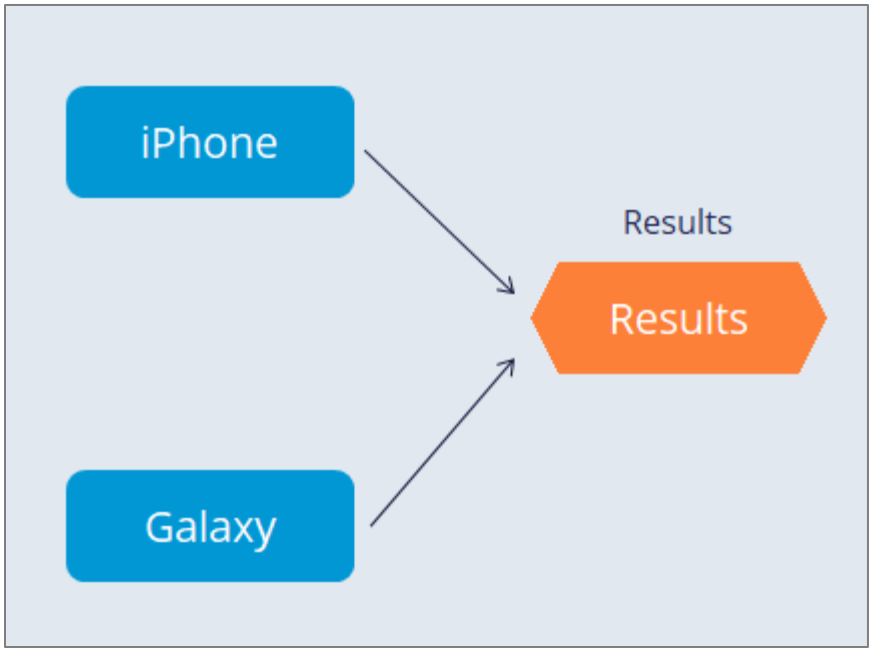
Name: Galaxy

Price: 250

Cost: 150

Screen3: Results component

Another decision component is the Results component. Each strategy always contains one Results component, which defines the output of the decision strategy.



How many actions do you think this strategy outputs?

A. 0
B. 1
C. 2

Feedback: Because both iPhone and Galaxy are connected to the Results component with a solid arrow line, this decision strategy outputs two actions.

Dynamic pricing

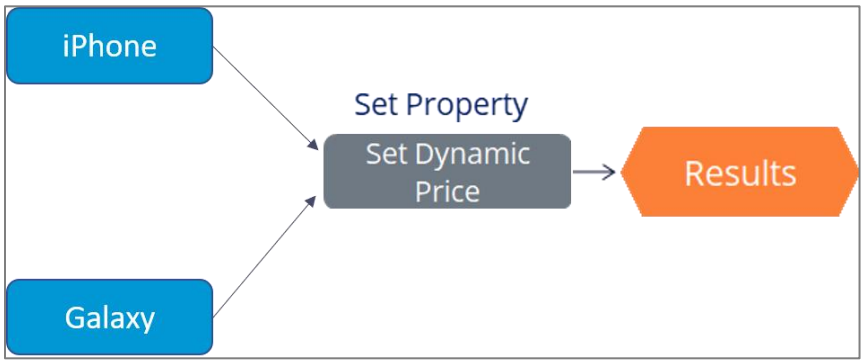
U+ Bank wants a dynamic Price for all offered actions. If the Customer value of a customer is higher than 60, the bank wants to offer a 10% discount to the customer.

To meet the new requirement, you must enhance the existing strategy to set the value of the Price based on Customer value. Changing the Price dynamically based on the Customer value makes the pricing customer-centric.

Set Property component

The Set Property component is used to dynamically alter the value of an action property based on a customer property. You use this component to set values to properties that are output by the strategy.

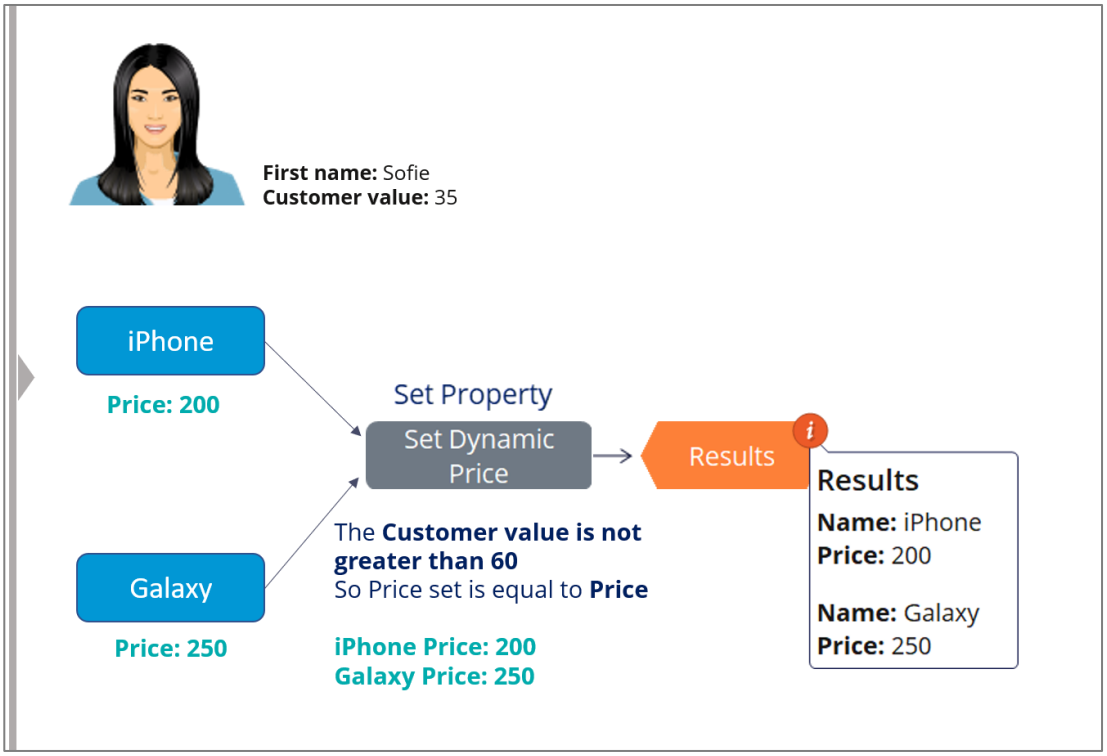
You can set properties to a **constant** or **calculated** value.

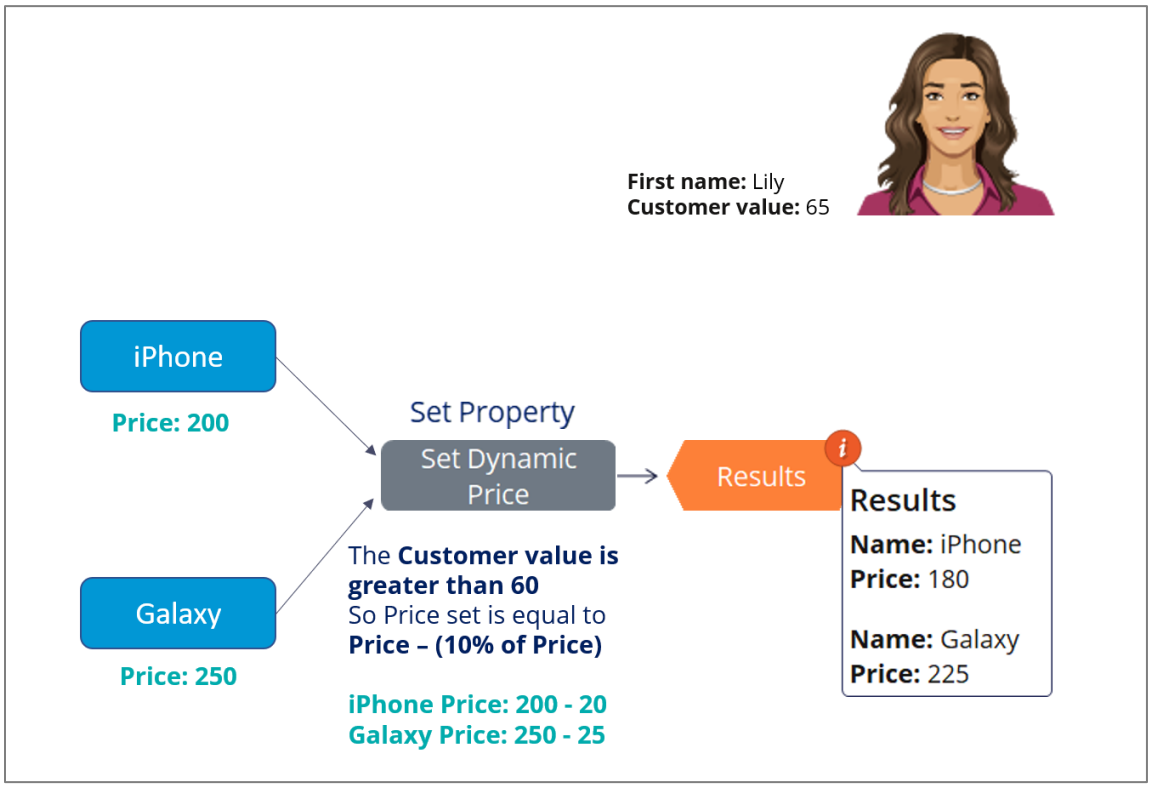


Example

Consider two customers: Sofie and Lily with customers value 35 and 65 respectively.

In the center of the following image, slide the vertical line to see how Sofie and Lily's **Customer value** affects the **Price** of the action offered to them.


First name: Sofie
Customer value: 35
iPhone
Price: 200
Galaxy
Price: 250
Set Property
Set Dynamic Price
Results
The Customer value is not greater than 60
So Price set is equal to Price
iPhone Price: 200
Galaxy Price: 250
Results
Name: iPhone
Price: 200
Name: Galaxy
Price: 250


First name: Lily
Customer value: 65
iPhone
Price: 200
Galaxy
Price: 250
Set Property
Set Dynamic Price
Results
The Customer value is greater than 60
So Price set is equal to
Price – (10% of Price)
iPhone Price: 200 - 20
Galaxy Price: 250 - 25
Results
Name: iPhone
Price: 180
Name: Galaxy
Price: 225

Action ranking

U+ wants to offer the most profitable action to its customers.

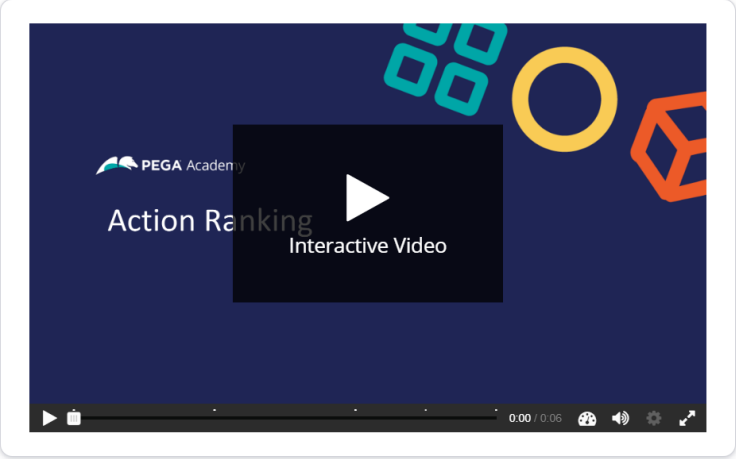
To enhance this strategy based on the new requirement, you need a new decision component that can rank the actions based on Profit and select the **highest ranked** action.

Profit is calculated based on **Price** and **Cost** action properties.

Prioritize component

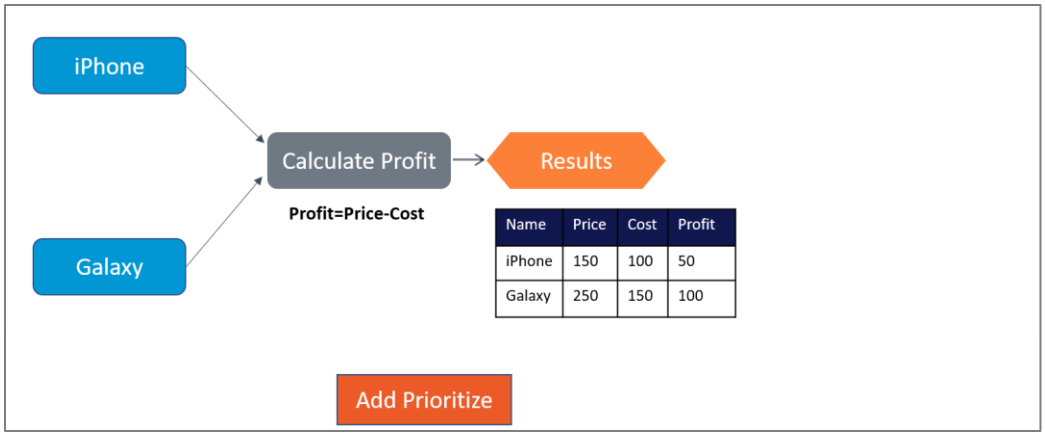
The Prioritize component is a decision strategy component used to rank actions. The Prioritize component is also used to select the top 1, top 2, or arbitrary top-*n* actions.

Click the Play icon to learn more about action ranking in detail with the help of a sample strategy.



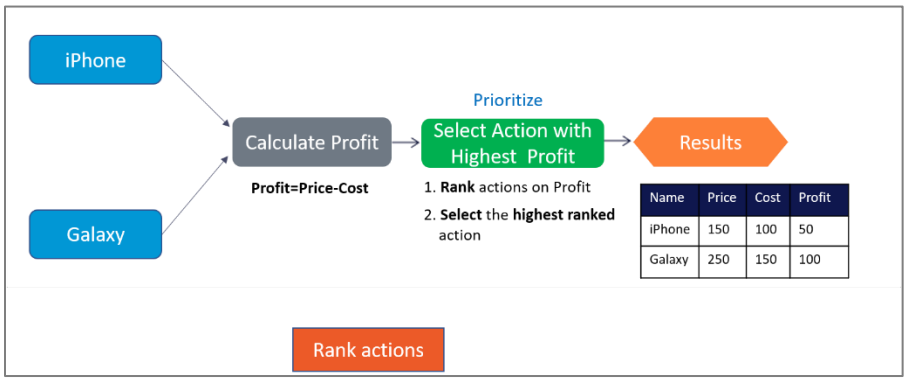
Screen 1:

The initial strategy outputs price, cost and the profit calculated by the Set Property component. Click **Add Prioritize** to add the Prioritize component to the strategy.



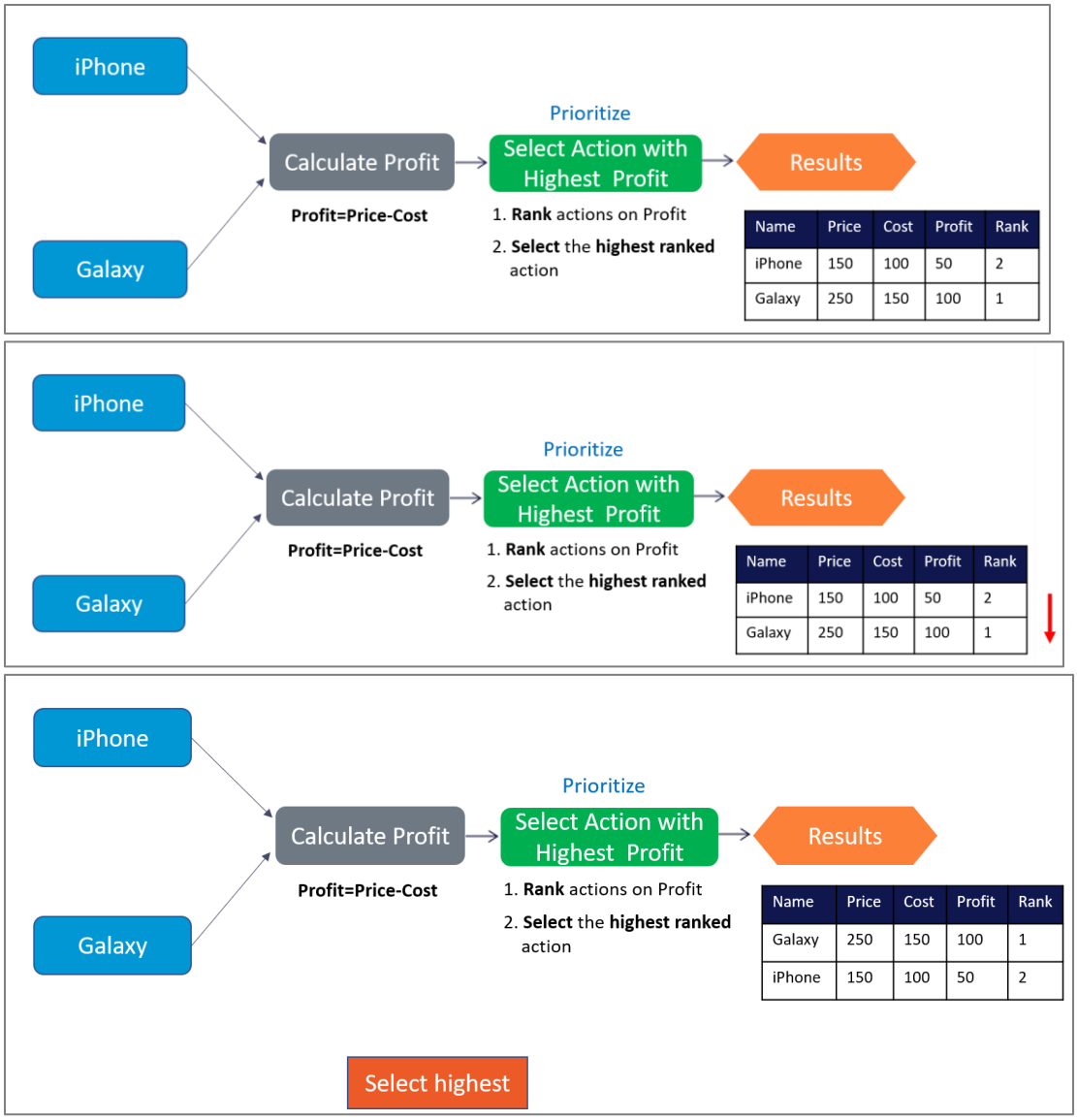
Screen 2:

The **Prioritize** decision component can perform two operations: **rank** actions based on an expression, and select the **highest** ranked action. Click **Rank actions** to see how the component rank the actions.



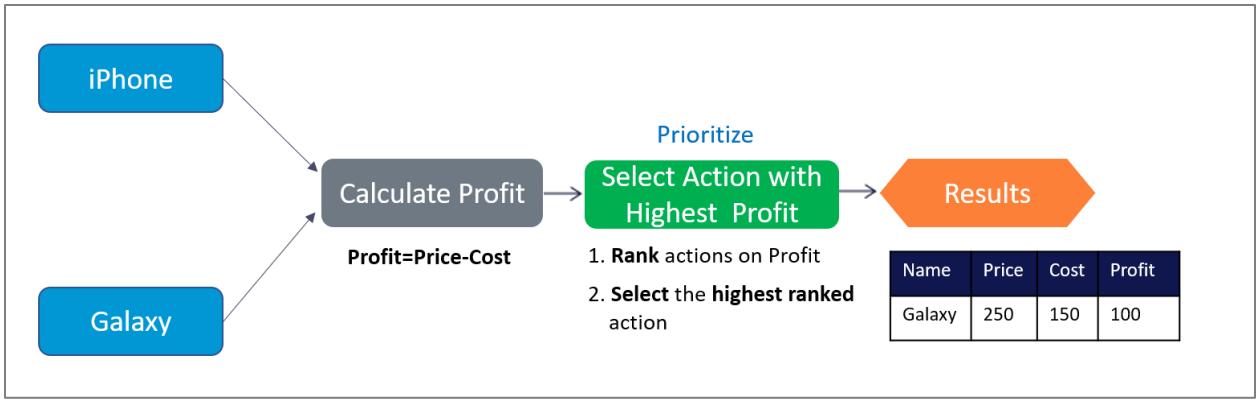
Screen 3:

Once the actions are ranked, the prioritize component selects the highest ranked action. Click **Select highest** to see the action selected.



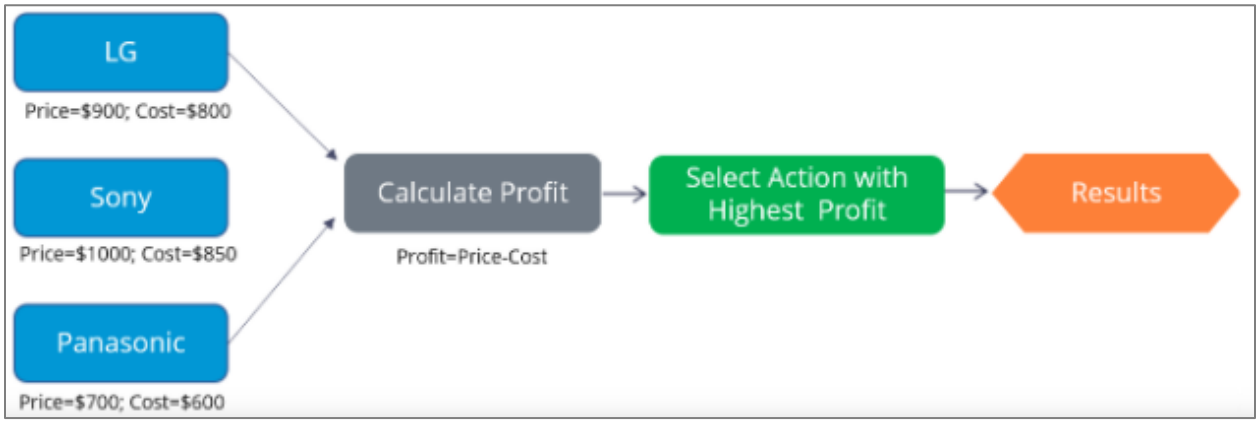
Screen 4:

You can see that the output of the result component is the highest selected action: Galaxy with a profit of 100.



Quiz

Examine this strategy and then answer the following question to check your knowledge on action ranking.



What does the **Results** component of the strategy contain?

- Sony with profit 150
- LG with profit 100
- Panasonic with profit 50

Feedback: The Prioritize decision component ranks the actions and selects the highest ranked action. Hence, the Results component of the strategy contains Sony with a profit of 150.

Creating a decision strategy

Introduction

Decision strategies drive the Next-Best-Action. Each strategy comprises a unit of reasoning represented by decision components. How these components combine determines which action (the Next-Best-Action) will be selected for a customer. Learn the types 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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

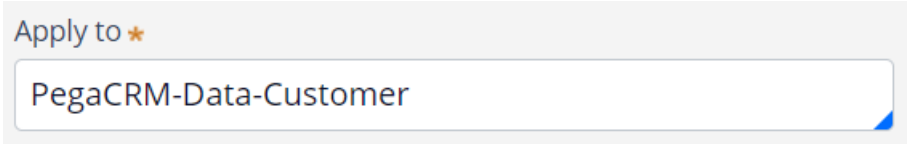
The Strategy-Results class limits the output of the strategy to the actions contained in the Business issue and Group.

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.

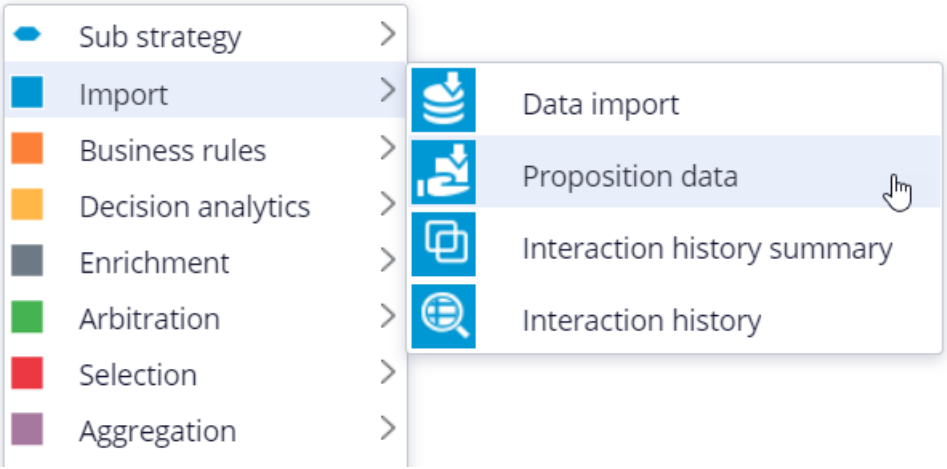


You can now start building the strategy. Right-click on the canvas to get the Context menu, which shows all component categories.

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.

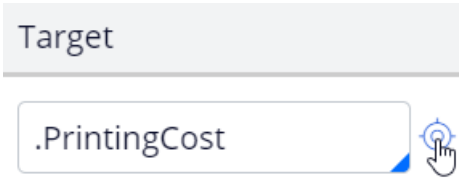
Click ‘Add Item’ to create the equation that will calculate the printing cost for each of the components.

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 BaseCost + 5 * 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.

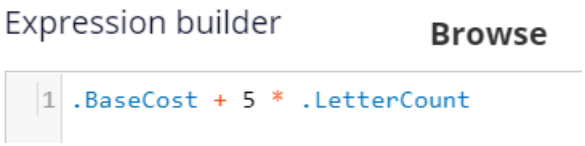
Suppose you want to use a customer property in your Expression, such as Age or Income.

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.

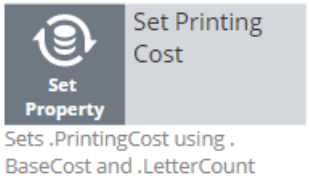
Finalize the Expression.



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.



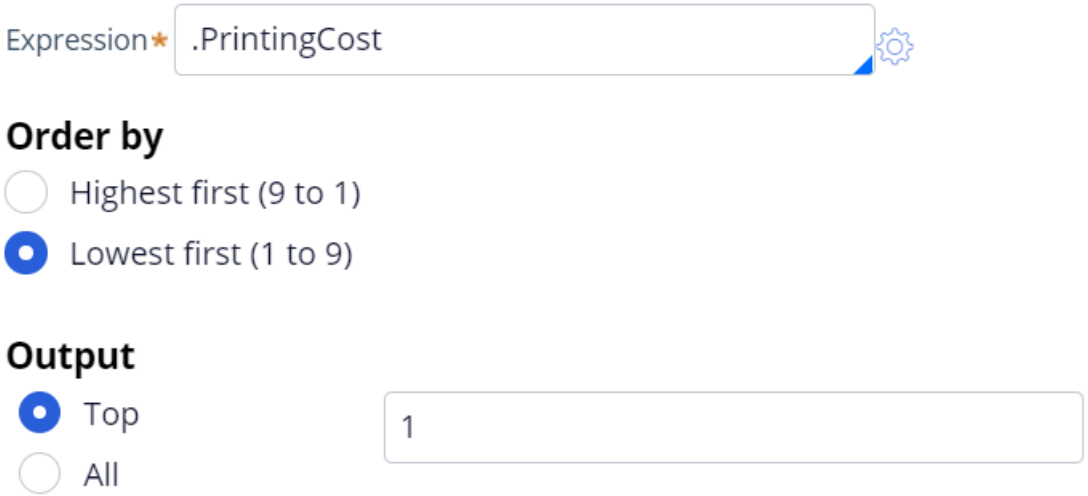
Now you want to ensure that the actions will be prioritized based on the lowest printing cost. So, you need to add the Prioritize component from the Arbitration category.

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.

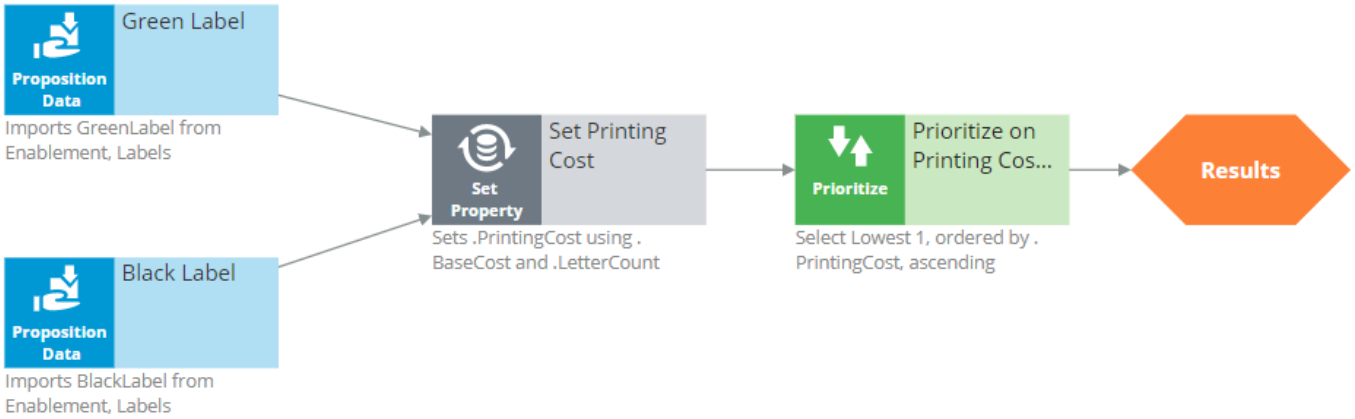
Here you can select the order in which the top actions are presented. Since you are interested in the lowest printing costs, configure it accordingly.

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.

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

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.

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

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.

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

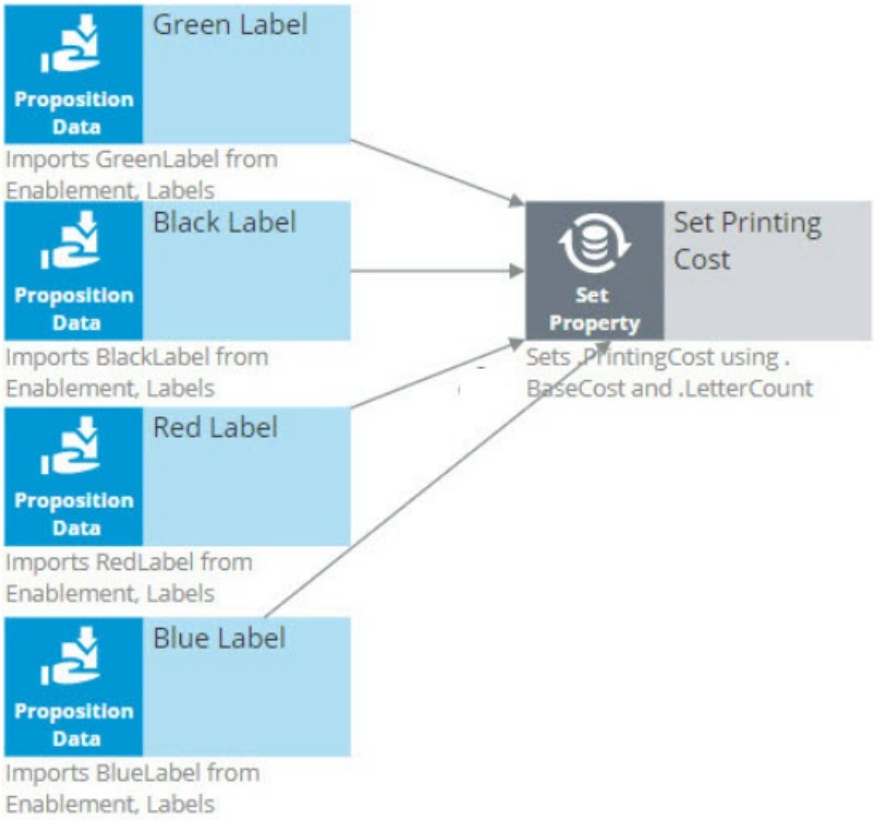
Rank	Name	BaseCost	LetterCount
1	Black Label	20	10

Rank	Name	BaseCost	LetterCount
1	Red Label	30	8

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.

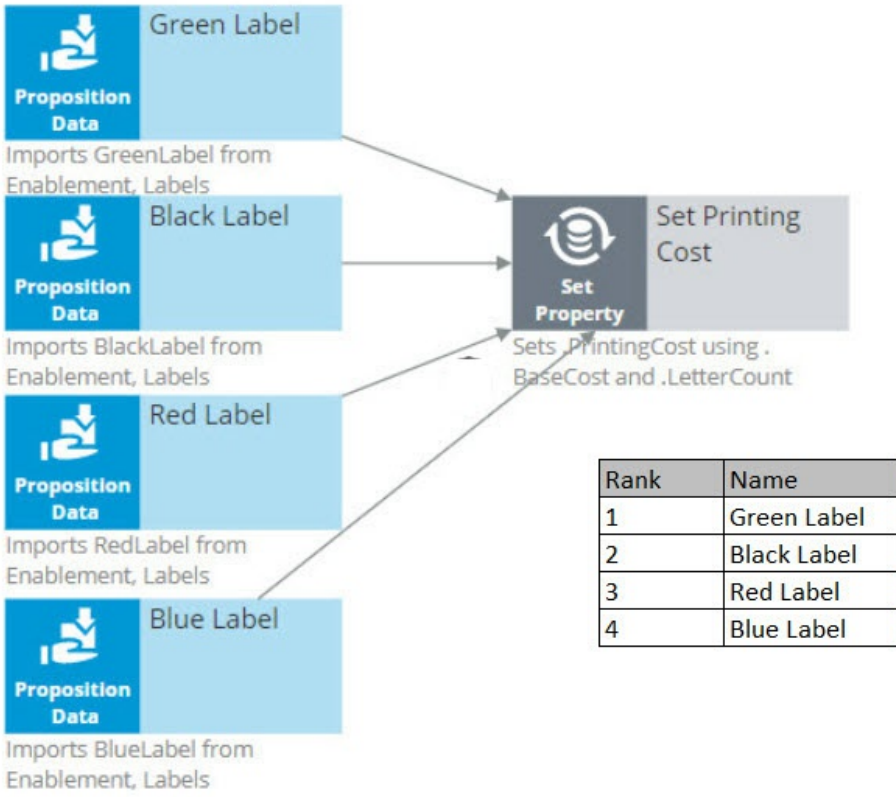
On the strategy canvas, components are connected by drawing arrows from component to component. So, what do these arrows mean exactly?



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.



Rank	Name	BaseCost	LetterCount
1	Green Label	10	10
2	Black Label	20	10
3	Red Label	30	8
4	Blue Label	40	9

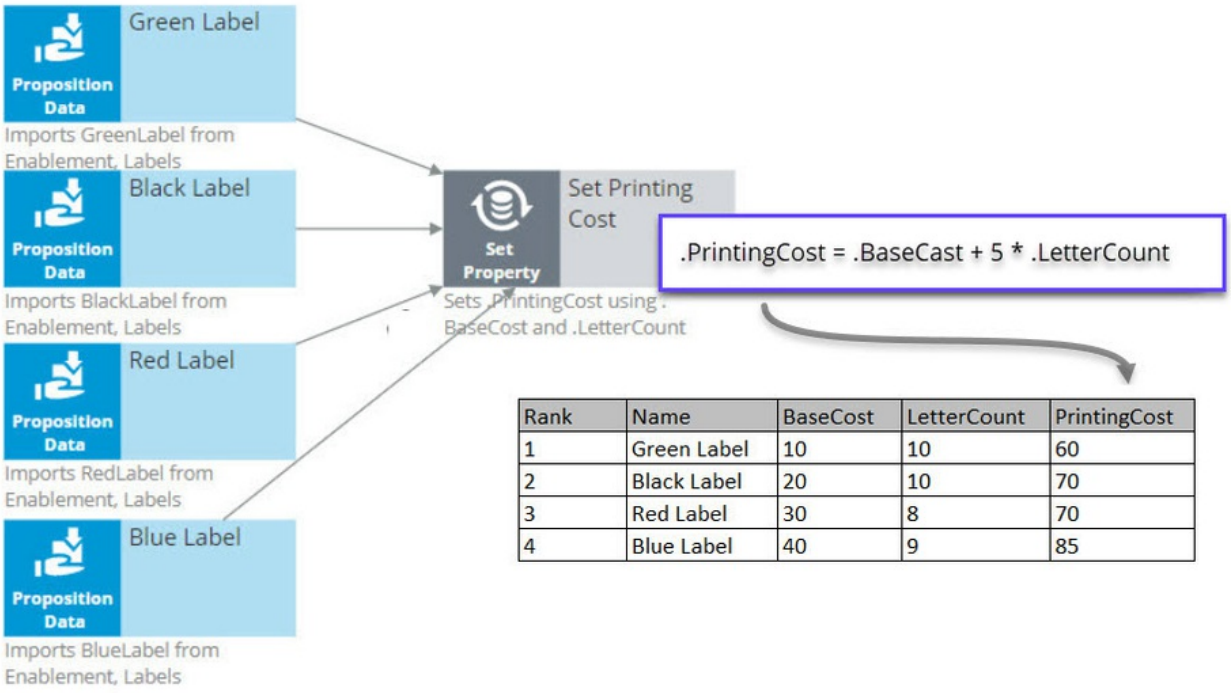
In the Set Property component, the Rank is determined by the order in which the actions are received by the component.

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.



Rank	Name	BaseCost	LetterCount	PrintingCost
1	Green Label	10	10	60
2	Black Label	20	10	70
3	Red Label	30	8	70
4	Blue Label	40	9	85

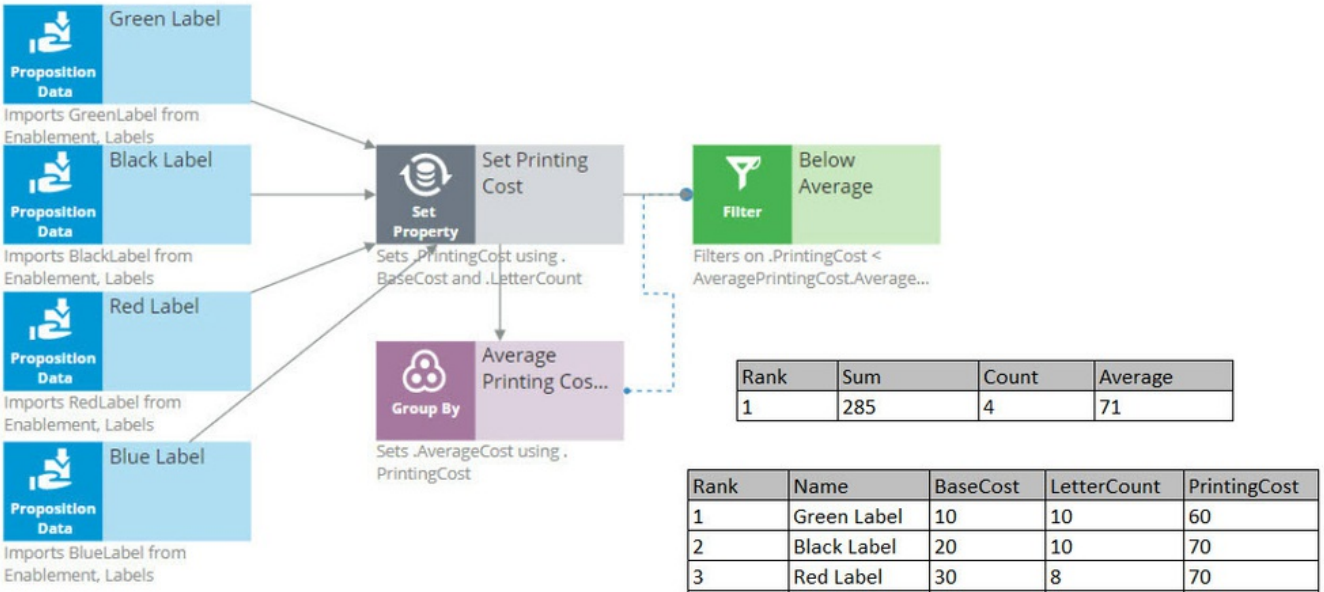
Because we are combining the data in our four Proposition Data components into one Set Property component, we only need to add one PrintingCost property to the new component, and it automatically computes the printing cost for all four actions.

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.



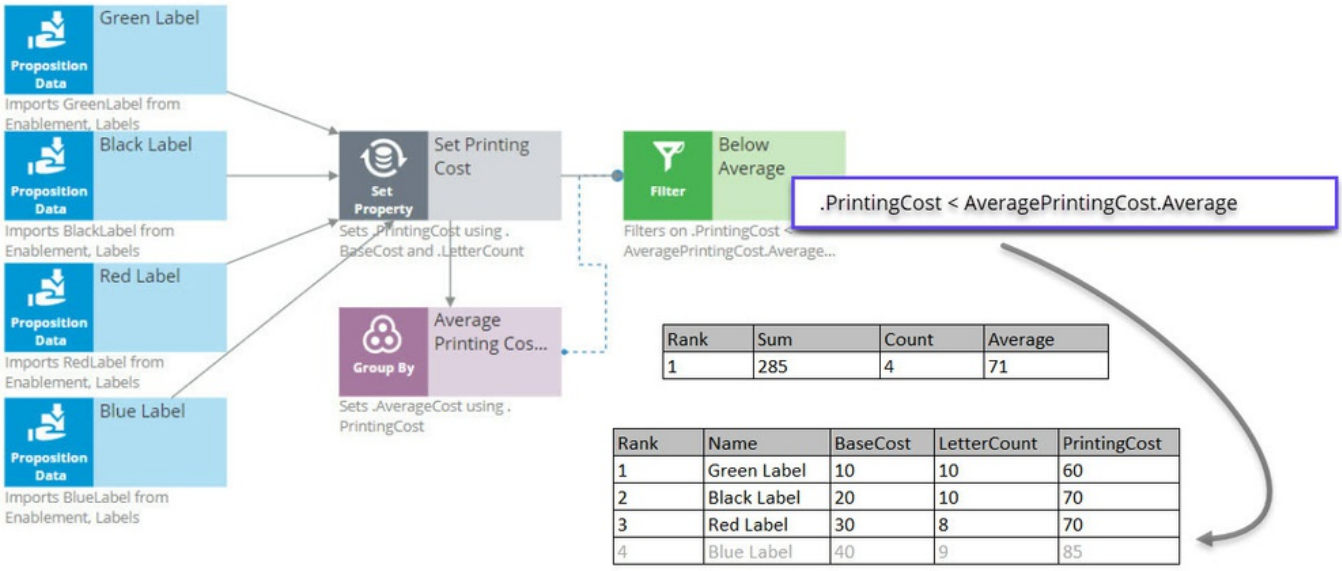
Rank	Sum	Count	Average
1	285	4	71

Rank	Name	BaseCost	LetterCount	PrintingCost
1	Green Label	10	10	60
2	Black Label	20	10	70
3	Red Label	30	8	70
4	Blue Label	40	9	85

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 lower 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.

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 AverageCost’.

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.

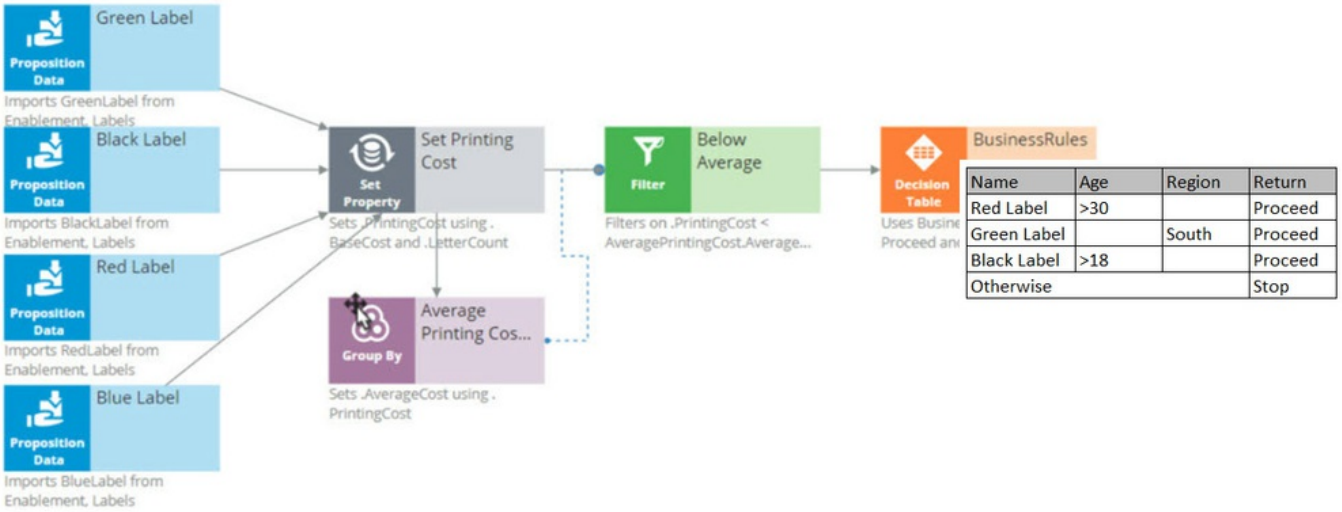
The Rank 1 average equals 71 in the ‘Group by’ component. This means that the filter will allow Label actions through that have a printing cost lower than 71.

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.



Name	Age	Region	Return
Red Label	>30		Proceed
Green Label		South	Proceed
Black Label	>18		Proceed
Otherwise			Stop

The Decision Table receives information about the remaining actions via the solid arrow from the Filter component.

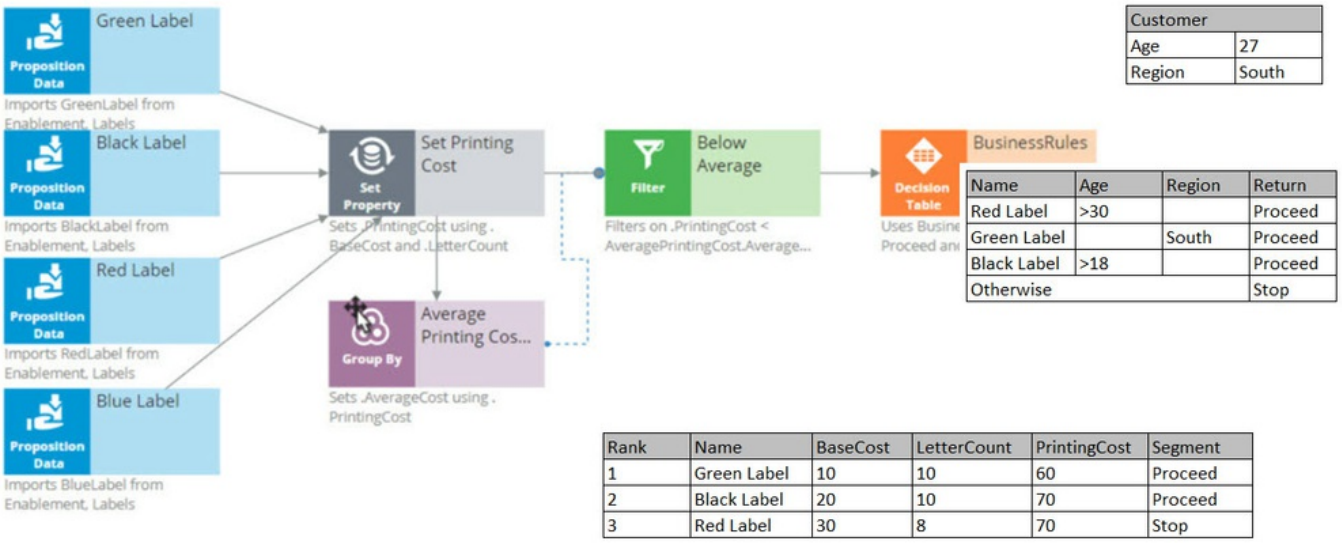
The business criteria say that the Red Label action can be offered if the customer's age is over 30 and they are from any region. If these criteria are met, the Return value is 'Proceed'.

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.



Customer	
Age	27
Region	South

Name	Age	Region	Return
Red Label	>30		Proceed
Green Label		South	Proceed
Black Label	>18		Proceed
Otherwise			Stop

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment
1	Green Label	10	10	60	Proceed
2	Black Label	20	10	70	Proceed
3	Red Label	30	8	70	Stop

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.

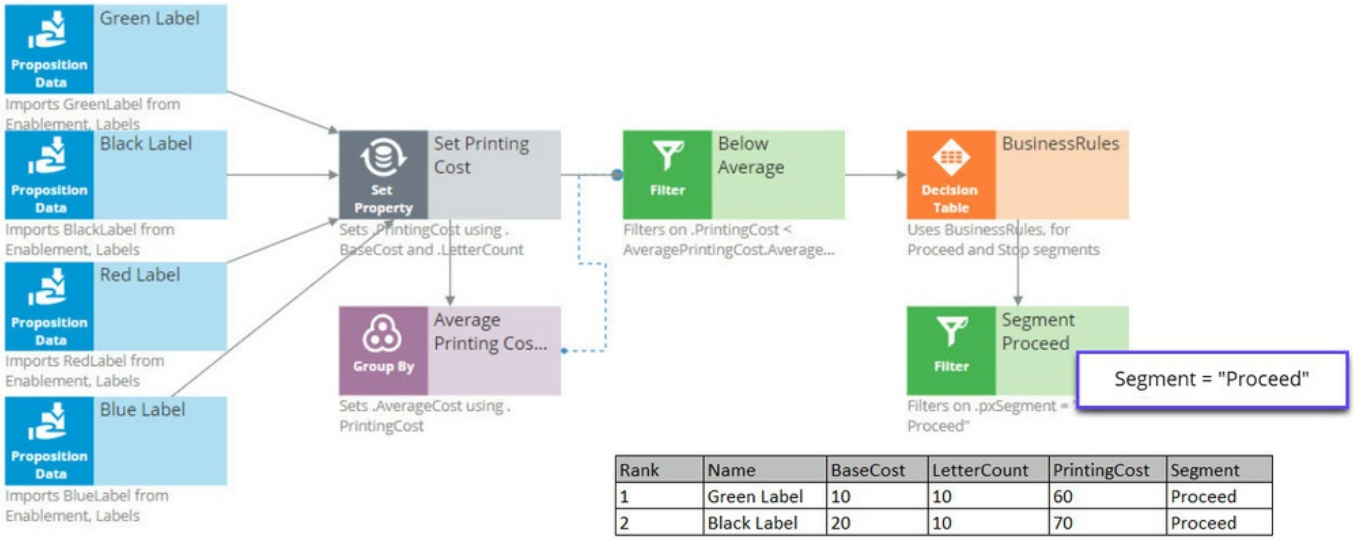
Next, it looks at the Black action and sees that the criteria for Black is that the customer's age is greater than 18. This customer is 27.

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.



Rank	Name	BaseCost	LetterCount	PrintingCost	Segment
1	Green Label	10	10	60	Proceed
2	Black Label	20	10	70	Proceed

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.

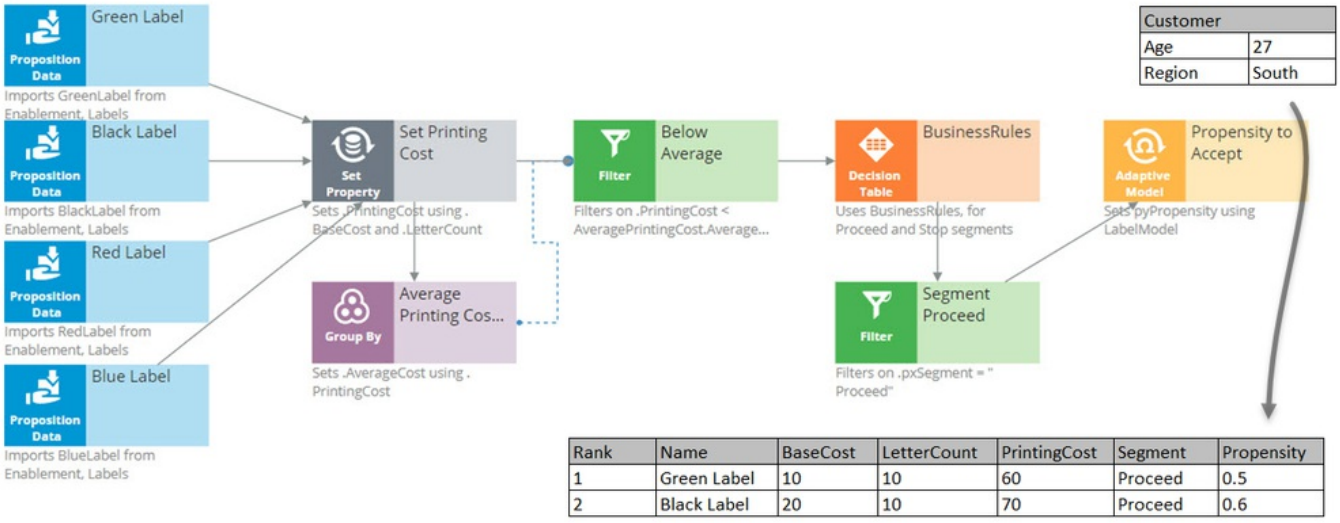
First is the Green Label. Green is allowed through, which means its properties will be available in the new component.

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.



Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity
1	Green Label	10	10	60	Proceed	0.5
2	Black Label	20	10	70	Proceed	0.6

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.

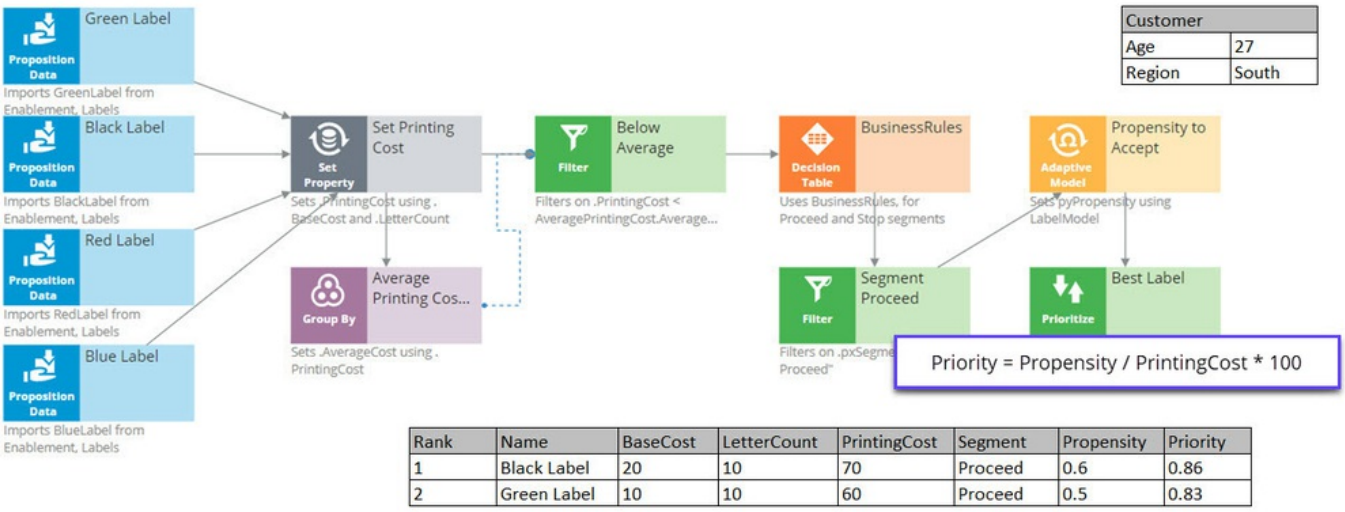
For the Black Label action, the likelihood turns out to be 0.6.

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.



Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity	Priority
1	Black Label	20	10	70	Proceed	0.6	0.86
2	Green Label	10	10	60	Proceed	0.5	0.83

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.

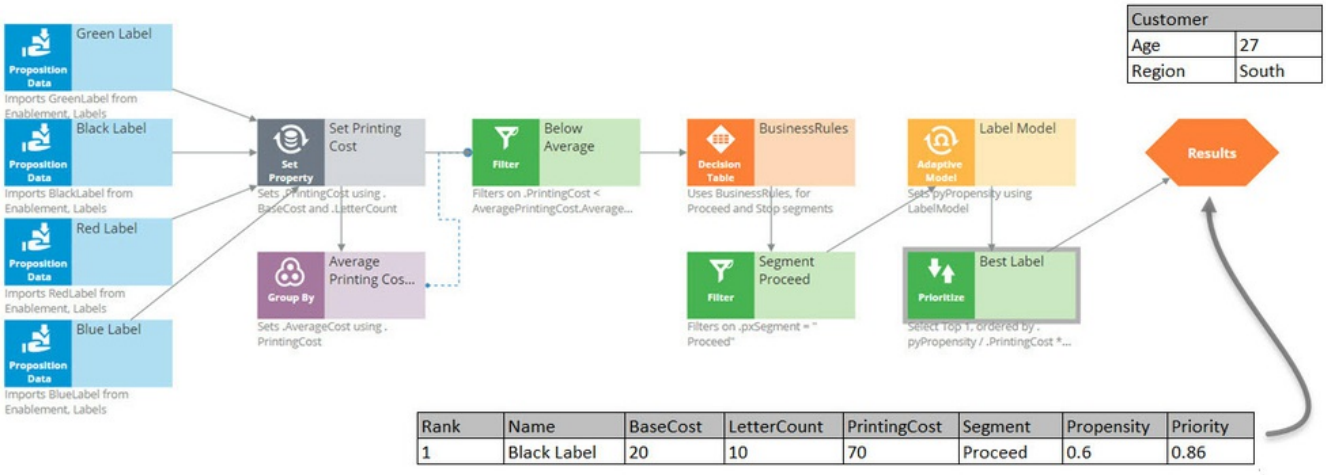
Because 0.86 is higher than 0.83, the Black Label action is now ranked number one.

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.



Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity	Priority
1	Black Label	20	10	70	Proceed	0.6	0.86

Decision strategy execution -- Fri, 07/24/2020 - 06:23
To get the full experience of this content, please visit https://academy.pega.com

Building a scorecard to calculate the credit score

Duration

10 mins

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This demo will show you how to build a scorecard.

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.

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.

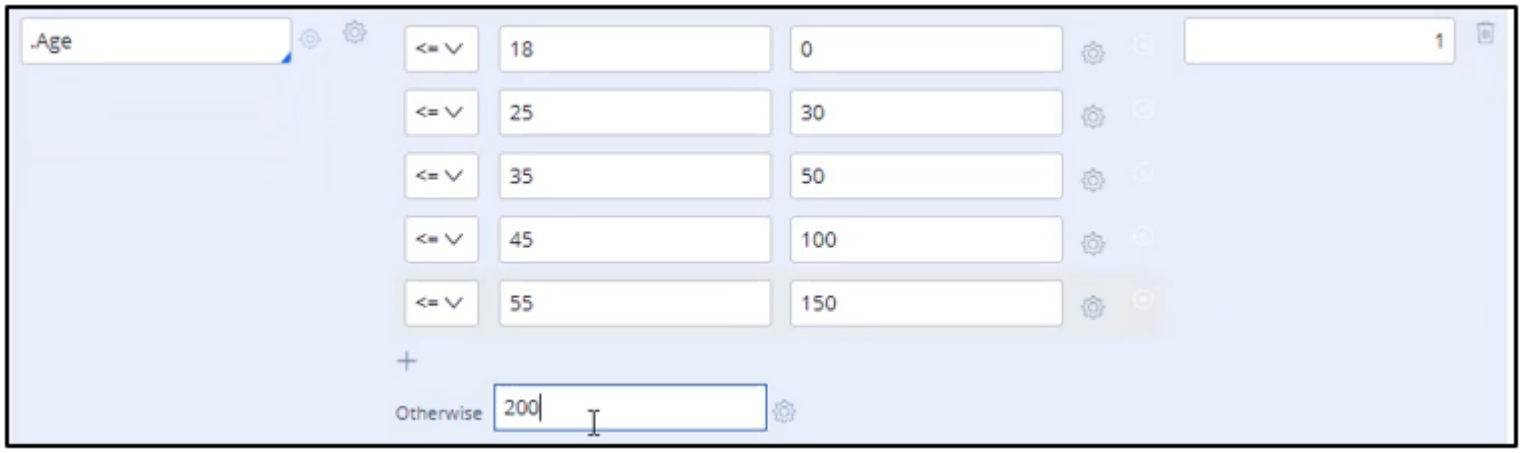
The data model contains an Income property, which contains the monthly income of the customer. The scorecard allows you to use this property to build the expression to compute the customer’s yearly income.

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.

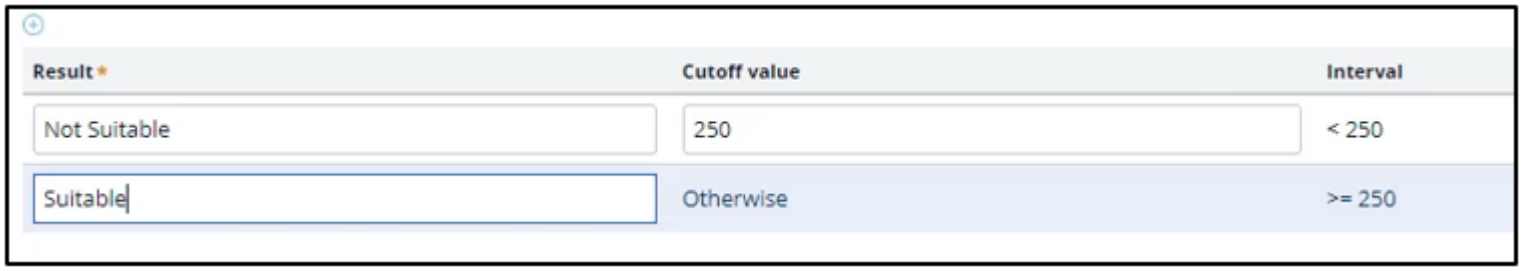
In a similar way, split the values for Age into five ranges and assign a score to each range. The higher the range, the higher the score you assign.



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’.



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.

Building a scorecard to calculate the credit score -- Tue, 06/23/2020 - 02:09
To get the full experience of this content, please visit https://academy.pega.com

Creating an engagement strategy

Duration

10 mins

Introduction

Scorecard results can be used to describe which actions are appropriate for a customer in the form of eligibility, applicability, or suitability rules. Learn how to build a decision strategy that leverages a scorecard to define engagement policy rules. Gain experience defining eligibility rules using a decision strategy in Next-Best-Action Designer.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

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.

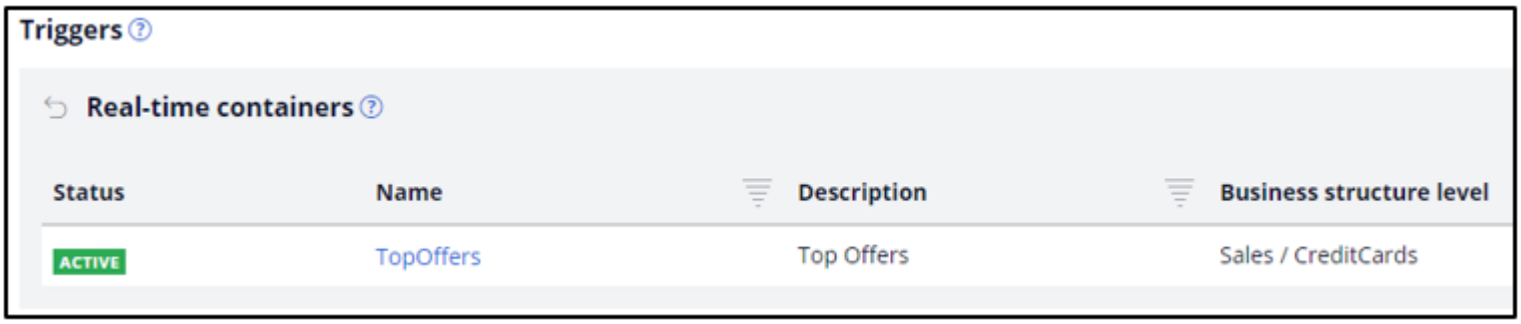
In this video, we will demonstrate the Engagement Policy capability without altering the use case.

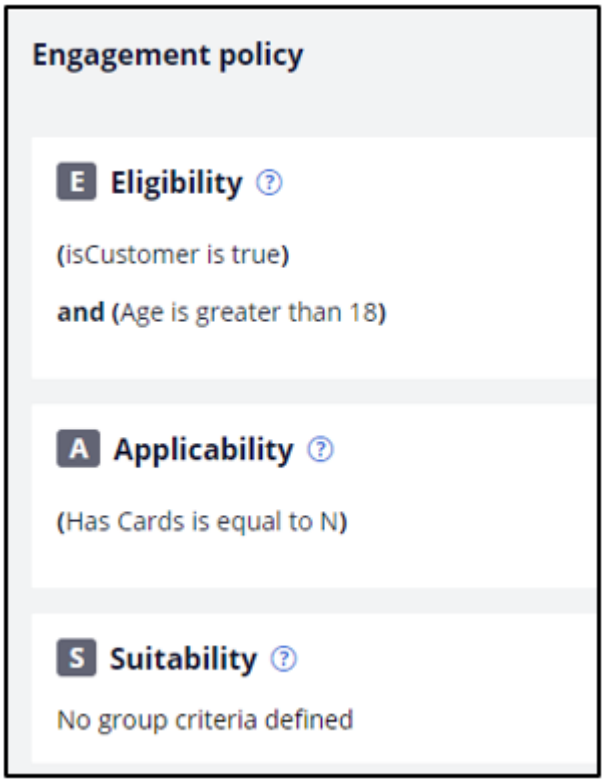
Currently, U+ Bank is doing cross-sell on the web by showing various credit cards to its customers. Every time Troy logs in to his accounts page, a credit card offer is shown.

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.

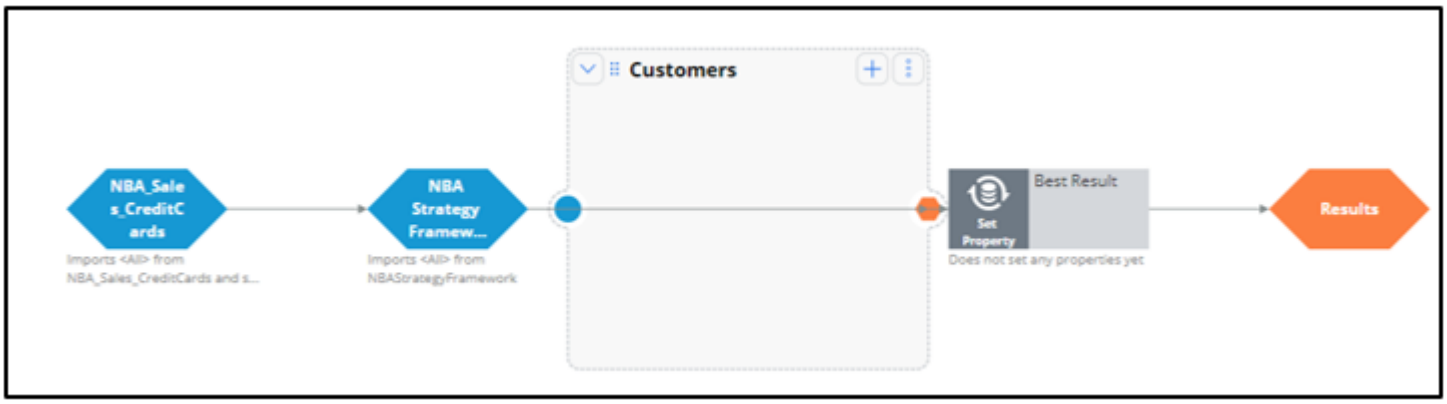




In the Engagement Policy, the Eligibility and Applicability rules have been defined to reflect the bank's current requirements using properties in the criteria.

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.

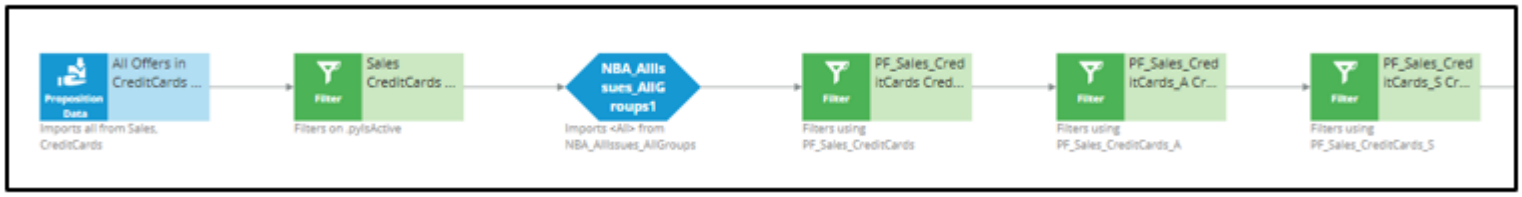


The first component of the strategy is responsible for all Engagement Policy rules.

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.

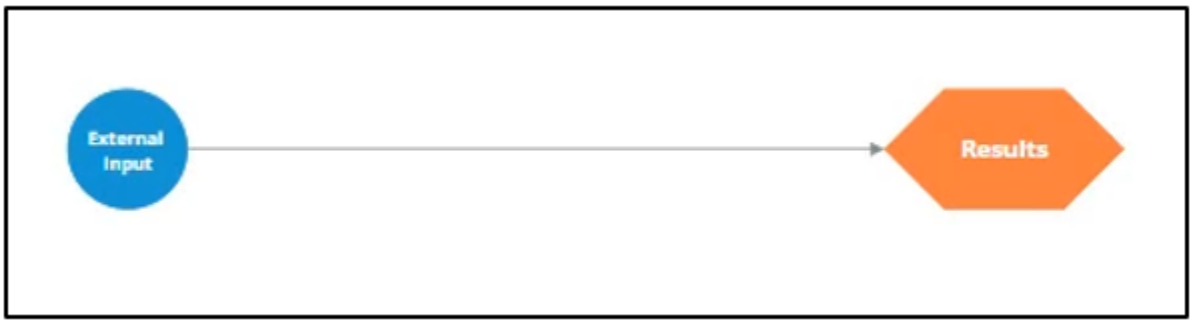
Enter a short description for the new strategy.

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.

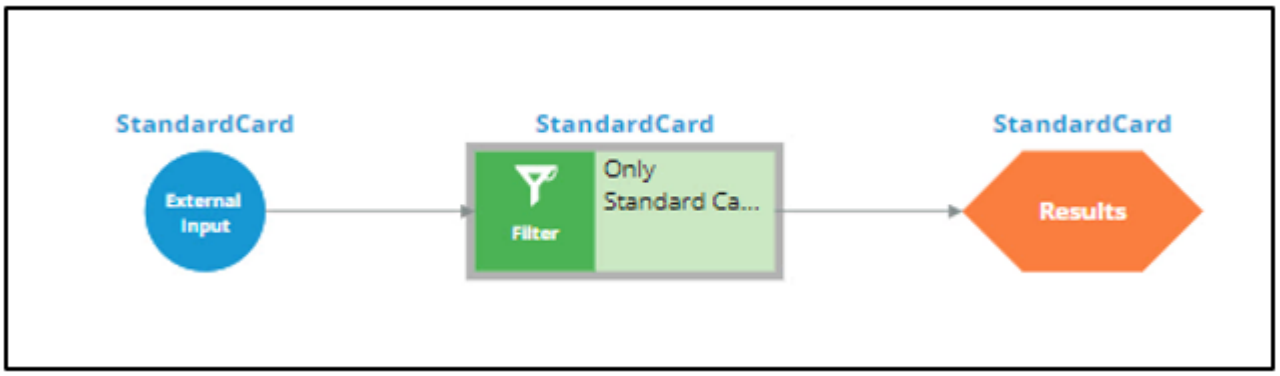
Start by naming the Filter component.

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.

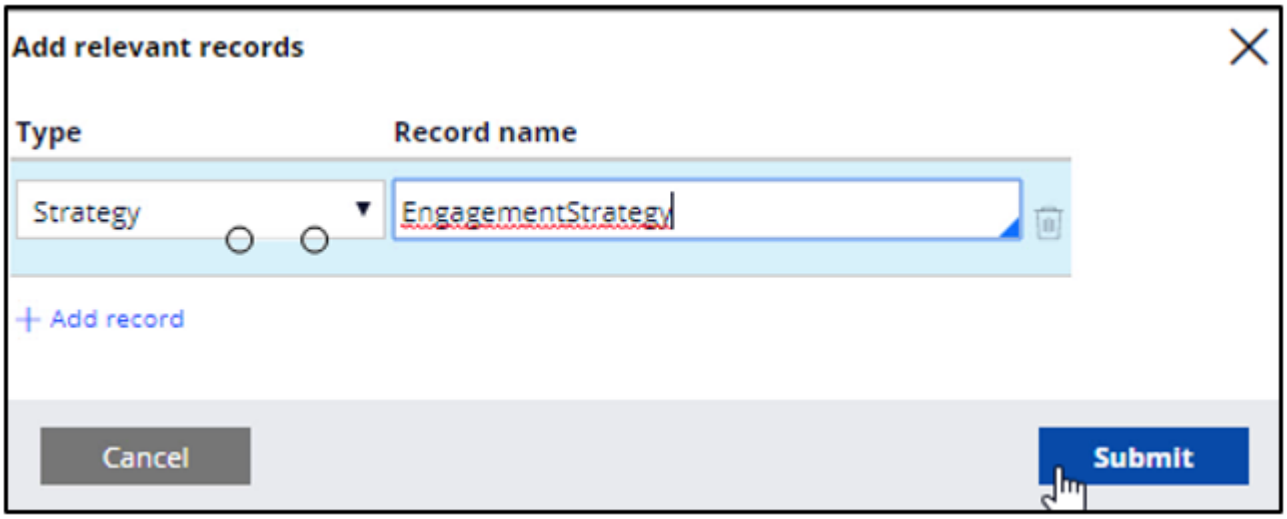


To make this strategy accessible in Next-Best-Action Designer as an Engagement Strategy, log in to the DEV STUDIO.

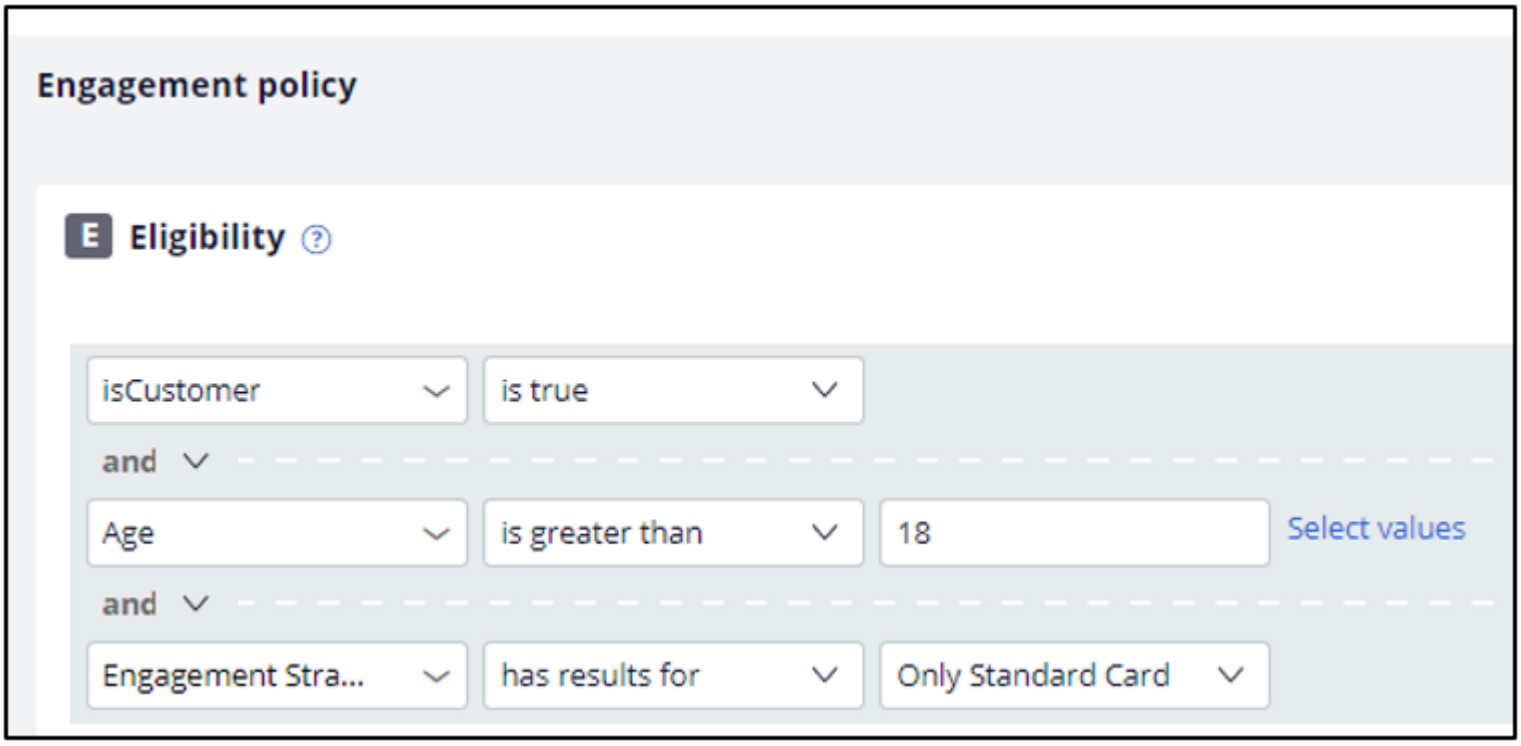
Navigate to the Relevant Records section.

Select the appropriate Class name for adding a Record.

Select the appropriate Type and Record name. In this case, you want to make the strategy accessible in NBA Designer, so set the Type to Strategy and the Record name to EngagementStrategy, the actual name of decision 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 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.

Creating an engagement strategy -- Tue, 06/23/2020 - 02:11
To get the full experience of this content, please visit https://academy.pega.com

Using a scorecard in a decision strategy

Duration

10 mins

Introduction

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.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

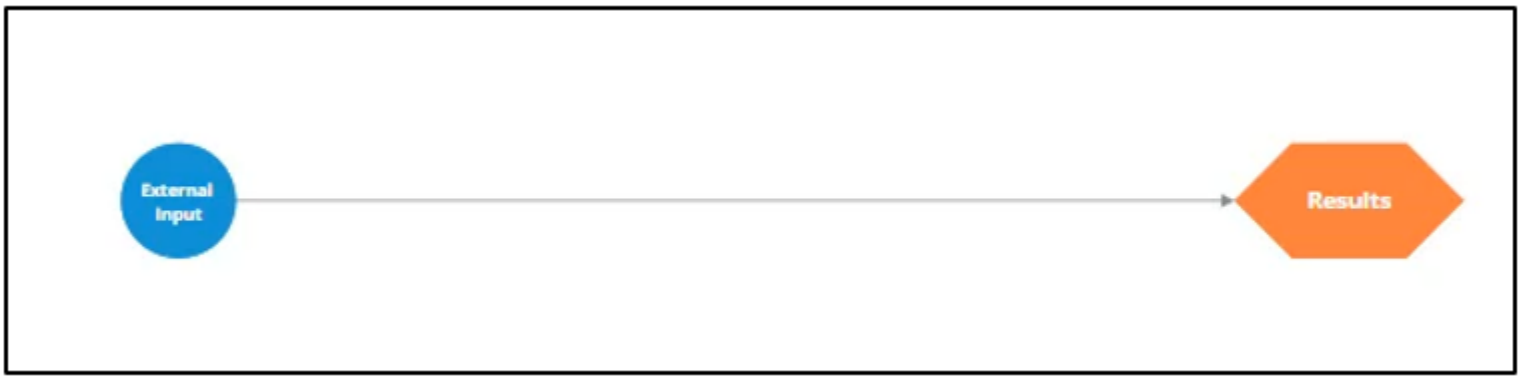
Transcript

This demo will show you how to use a Scorecard in a decision strategy to determine customer suitability for a credit card.

Currently, U+ Bank is doing cross-sell 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 who have a credit score greater than or equal to 250.

To implement this requirement, use an Engagement Strategy.



U+ already created a Scorecard that computes the customer credit score. To use the Scorecard in the decision strategy, add a Scorecard component to the canvas and configure it.

Select the Scorecard model **DetermineCreditScore**, which U+ already created**.**

If you open this Scorecard, you can see how the credit score is computed. Note that the 3 customer properties are used and a score is assigned to the value or range of values the property can have.

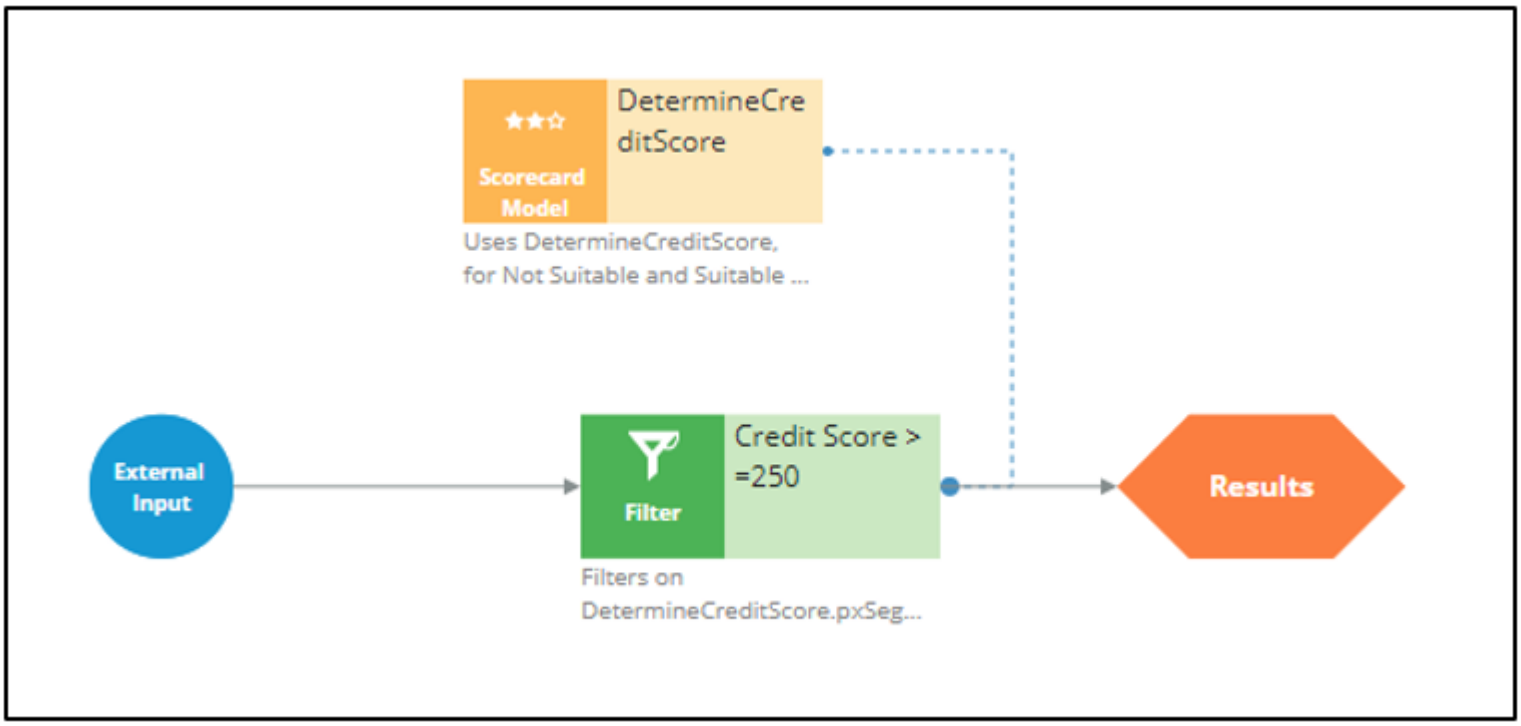
You can also see the segment results. The Scorecard returns value **Not Suitable** if the credit score is less than 250. Otherwise, it outputs the result **Suitable**.

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 result of the Scorecard is stored 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.

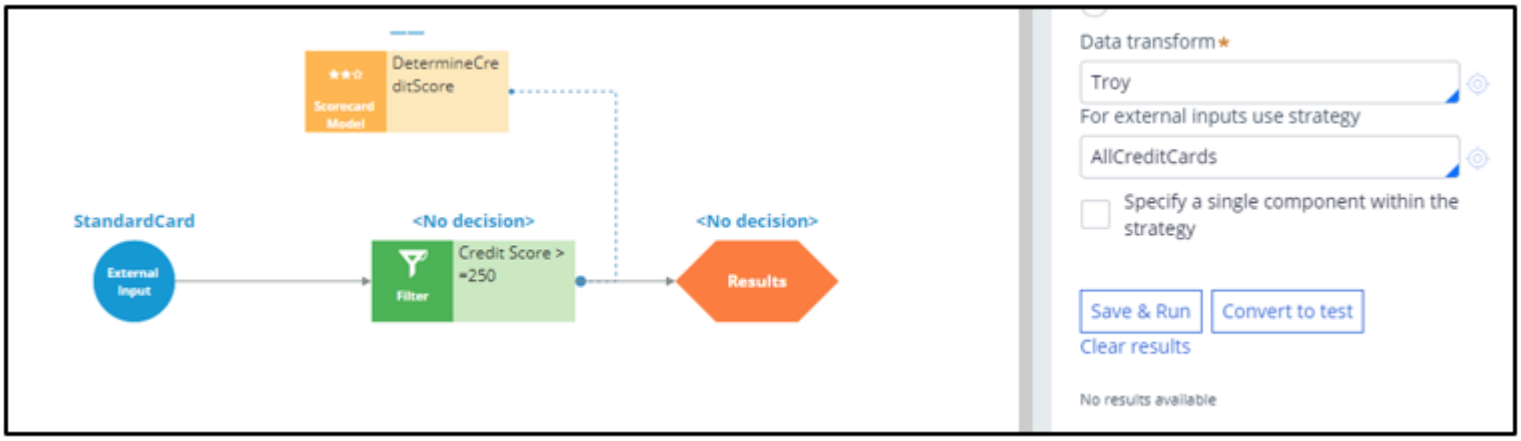


Note that the usage of the External Inputs component also gives you the ability to create more complex conditions, where Action attributes are also used.

Save the configurations.

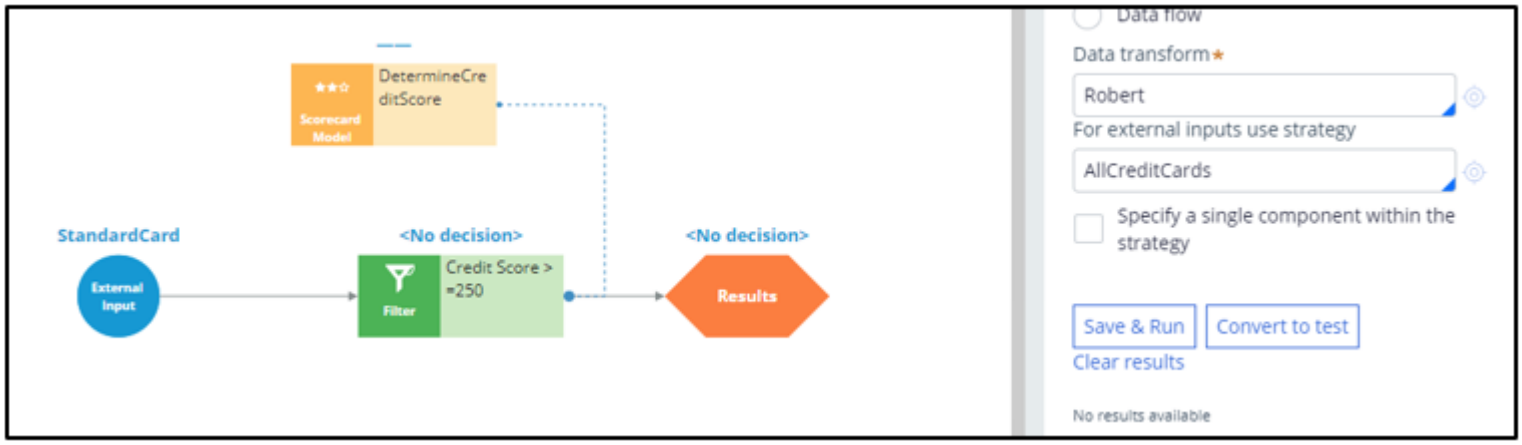
Now, test the strategy using the customer profiles, Troy and Robert.
For external inputs, consider all credit cards available.

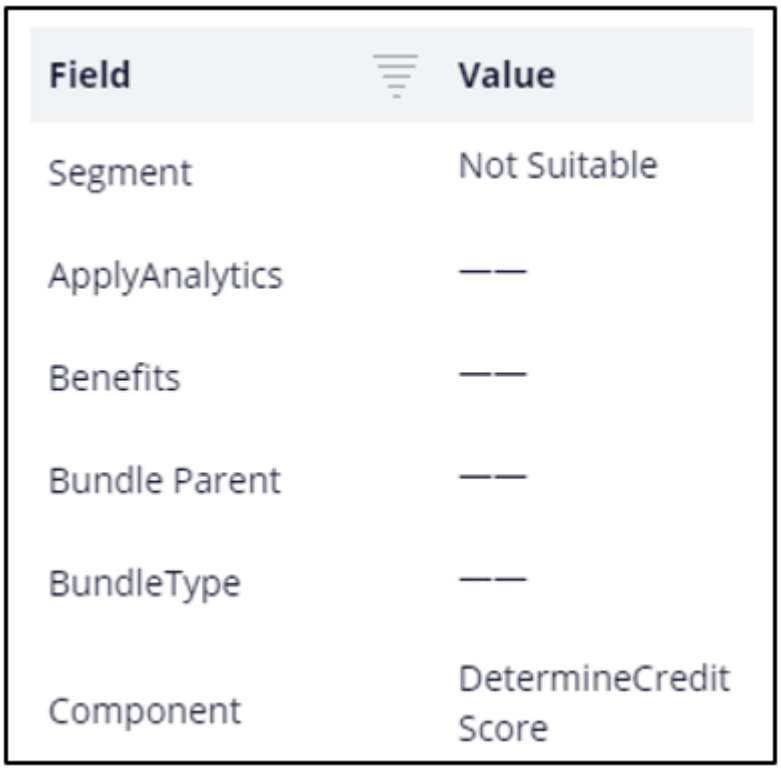
The result of the Scorecard is: **Not Suitable**. Therefore, the strategy did not output any results for Troy.



Field	Value
Segment	Not Suitable
ApplyAnalytics	——
Benefits	——
Bundle Parent	——
BundleType	——
Component	DetermineCredit Score

Now, repeat the test to verify results for the Robert data transform.



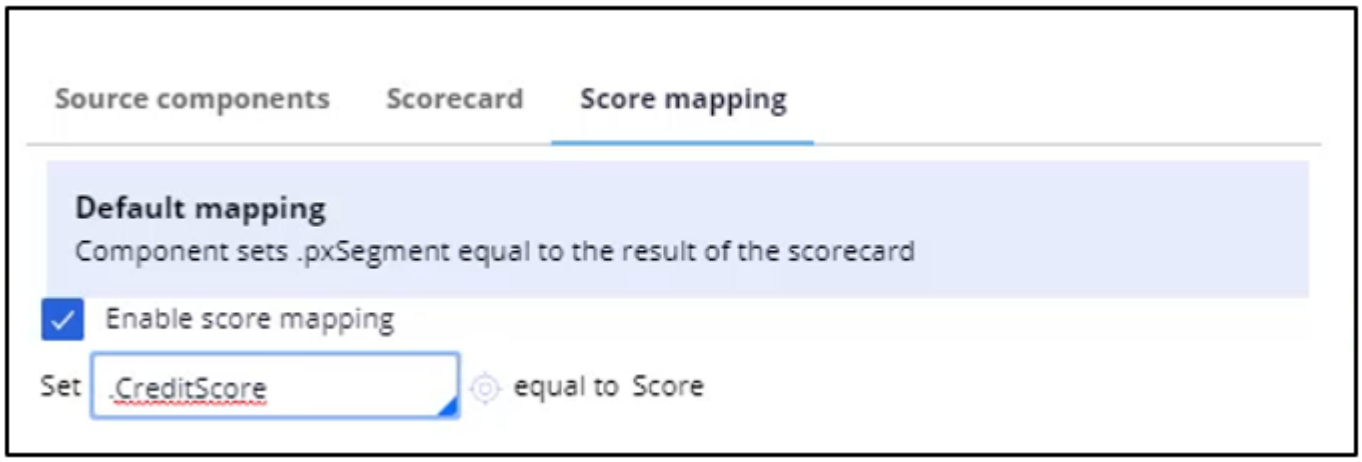


Field	Value
Segment	Not Suitable
ApplyAnalytics	——
Benefits	——
Bundle Parent	——
BundleType	——
Component	DetermineCredit Score

The strategy is also not outputting any results for Robert, as the Segment value is **Not Suitable**.

Now, assume the credit score value for customers is already computed and presented in the Customer data model. However, this value is not set for certain customers, in which case you want to use the credit score determined by the Scorecard itself.

To make these adjustments, start by opening the Scorecard component and mapping the score computed by the Scorecard to the CreditScore property.

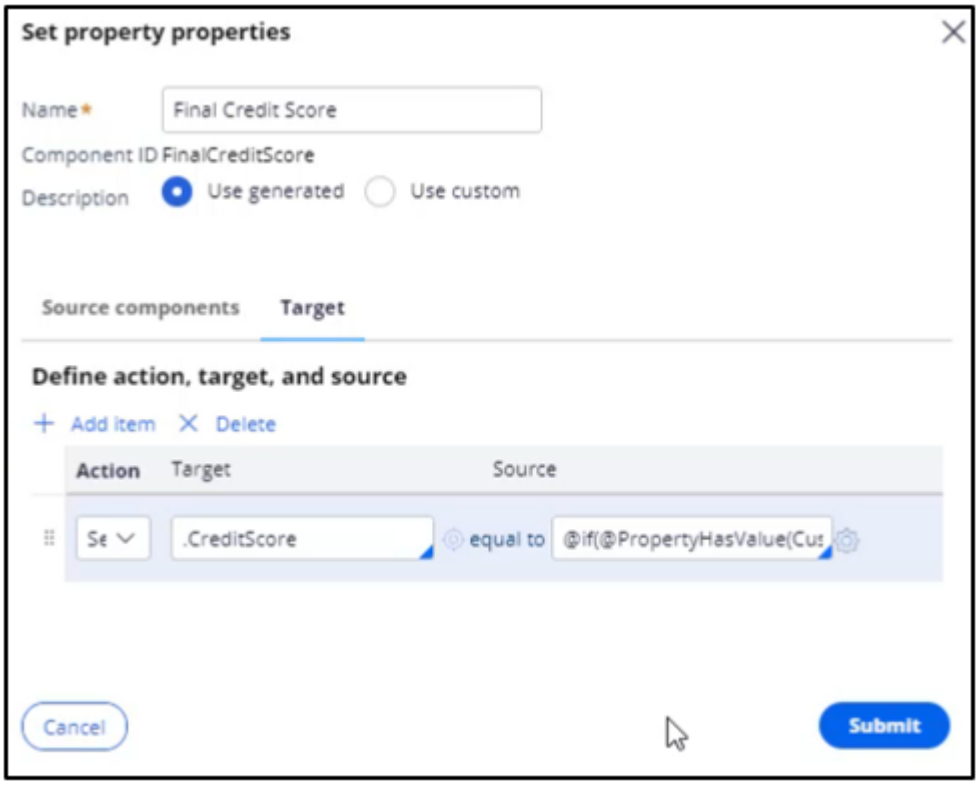


Then, add a Set Property component to determine the actual value of the credit score, given that the credit score is already available for certain customers.

Use ‘Add item’ to set the CreditScore to either the available credit score value from the Customer data model, or to the value computed by the Scorecard.

Define the Expression as *if(@PropertyHasValue(Customer.CreditScore), Customer.CreditScore, .CreditScore).*

If set, this Expression will result in the credit score from the Customer data model. If not, the credit score value will be computed by the Scorecard.

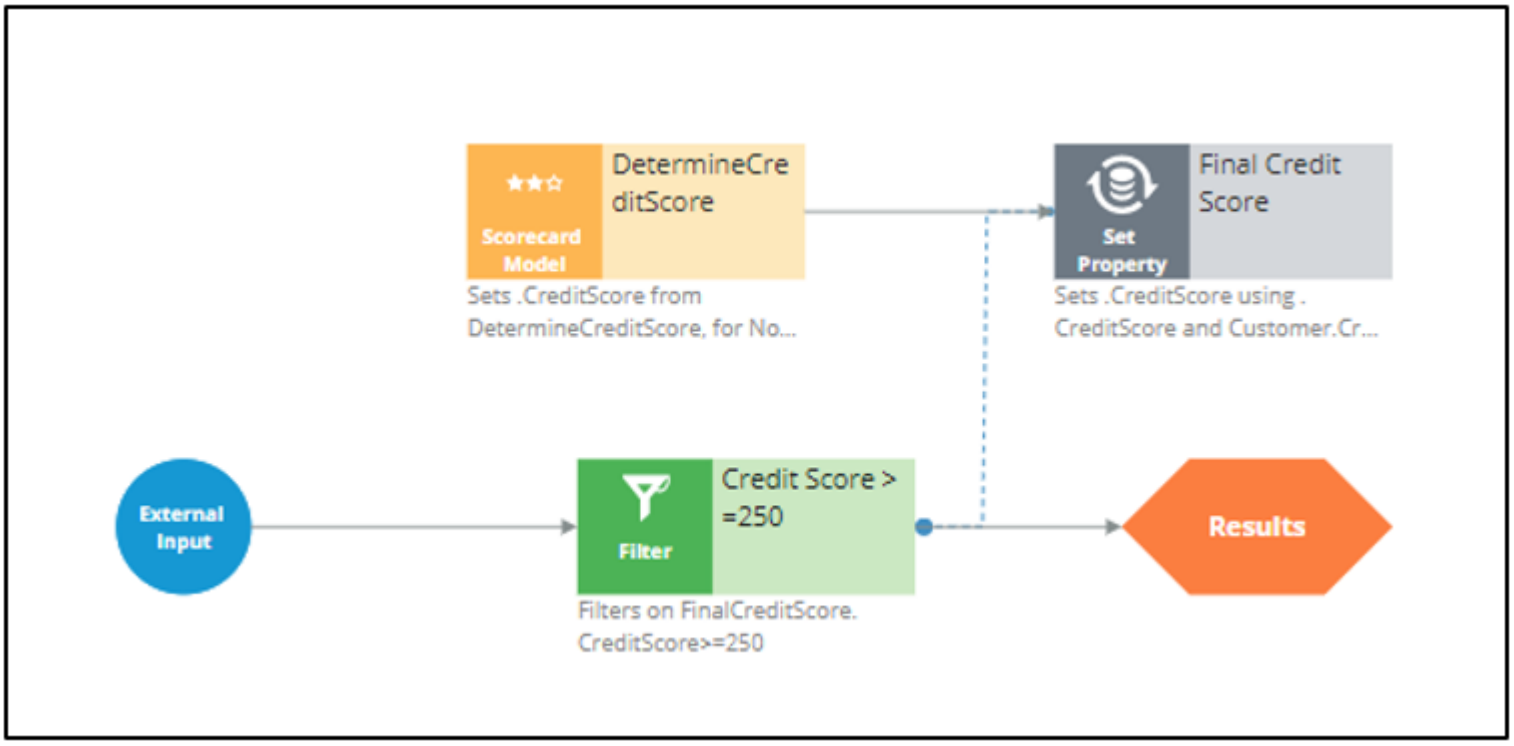


Once the Set Property component is configured, open the **Filter** component properties to modify the Filter condition. In this scenario, the bank has decided to present various credit cards to suitable customers who have credit scores greater than or equal to 250.

So, open the ‘Expression builder’ to modify the condition as *FinalCreditScore.CreditScore>=250*.

Now, connect the Scorecard Model component to the Set Property component to ensure the CreditScore value determined by the Scorecard is copied to the Set Property component.

Save the configurations.



Re-test the strategy for the Troy and Robert data transforms.

Note that the strategy is not outputting any results for Troy, as his credit score, 200, is less than 250.

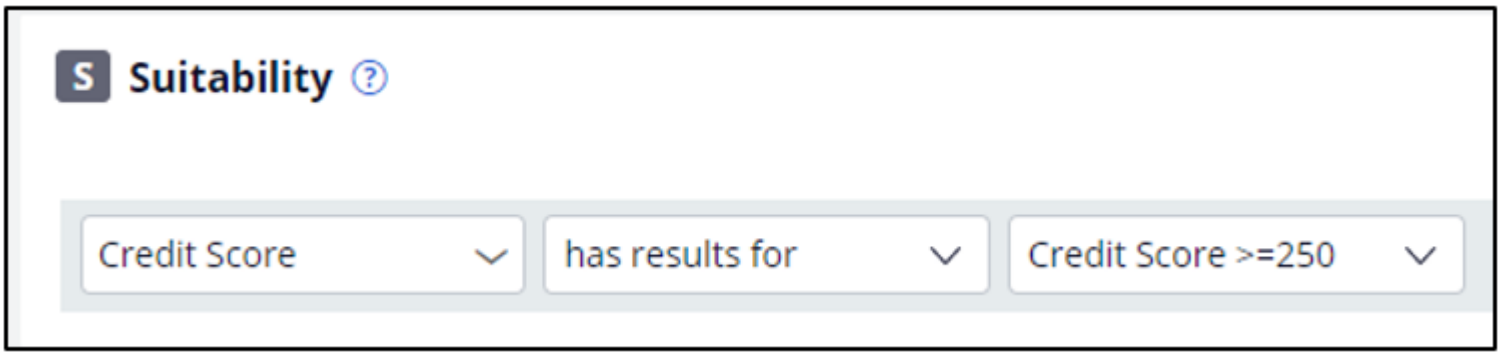
For Robert, the strategy is outputting results.

If you open the Robert data transform, note that the CreditScore in the data model is set to 600.

The Set Property component picks up the credit score value available in the data model (that is, CreditScore = 600) and not the value computed by the Scorecard (CreditScore = 200). That is why the strategy is outputting results for Robert.

Since you have completed configuring the strategy, check it in, so that the strategy will be available to the U+ bank website.

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 >=250.*

Saving this completes all the required configurations.

On the U+ bank website, if you log in as Troy, notice that no credit card offer is displayed. This is because no credit cards are suitable for Troy.

Now, if you log in as Robert, notice that the credit card offer is displayed.

This is because credit cards are suitable for Robert.

This demo has concluded. 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 if a Customer property has value or not.

Creating customer risk segments using a decision table

Business scenario

U+ Bank is currently doing cross-sell on the web by showing various credit cards to its customers. The bank already uses a customer's credit score to determine their suitability for a credit card.

The new eligibility rules require customers to be divided into risk segments varying from AAA to CCC. To start, customers in the risk category BB- and CCC are not eligible for credit cards. Customers from all other risk segments are eligible.

The risk segments are determined by two parameters: **Outstanding loan amount** and the customer **Credit score**.

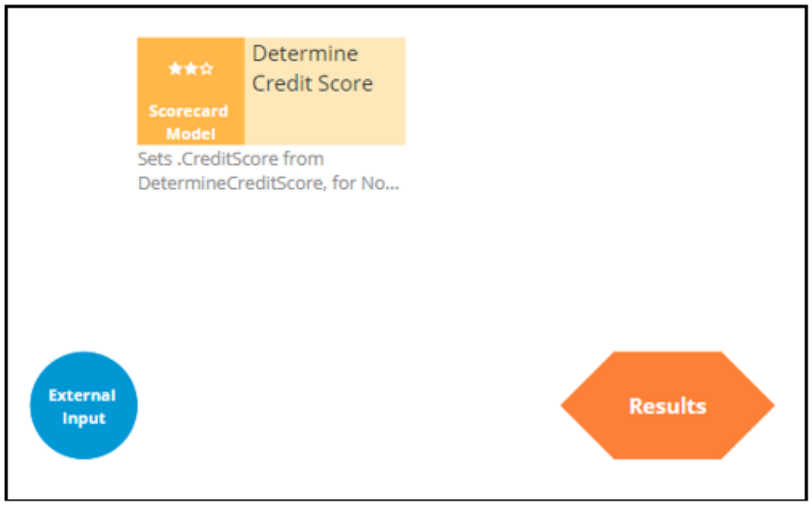
Condition	Risk Segment
If **Outstanding loan amount** < $10000 AND **Credit score** is > 600	AAA
If **Outstanding loan amount** between $10000 and $25000 AND **Credit score** is > 600	AAA-
If **Outstanding loan amount** between $25000 and $50000 AND **Credit score** is > 600	AA
If **Outstanding loan amount** > $50000 AND **Credit score** is > 600	AA-
If **Outstanding loan amount** < $25000 AND **Credit score** is between 400 and 600	BBB
If **Outstanding loan amount** between $25000 and $50000 AND **Credit score** is between 400 and 600	BBB-
If **Outstanding loan amount** > $50000 AND **Credit score** is between 400 and 600	BB-
If **Credit score** is between 200 and 400	CCC
If **Outstanding loan amount** AND **Credit score** falls in any other range	CCC

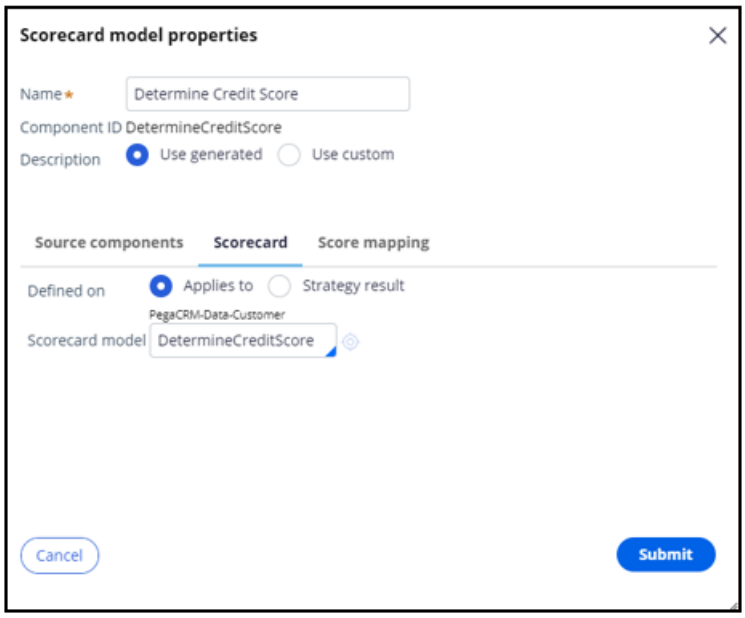
To implement the new bank regulations, use an Engagement Strategy to:

1. Calculate credit score.
2. Define risk segment.
3. Filter credit cards based on risk segment.

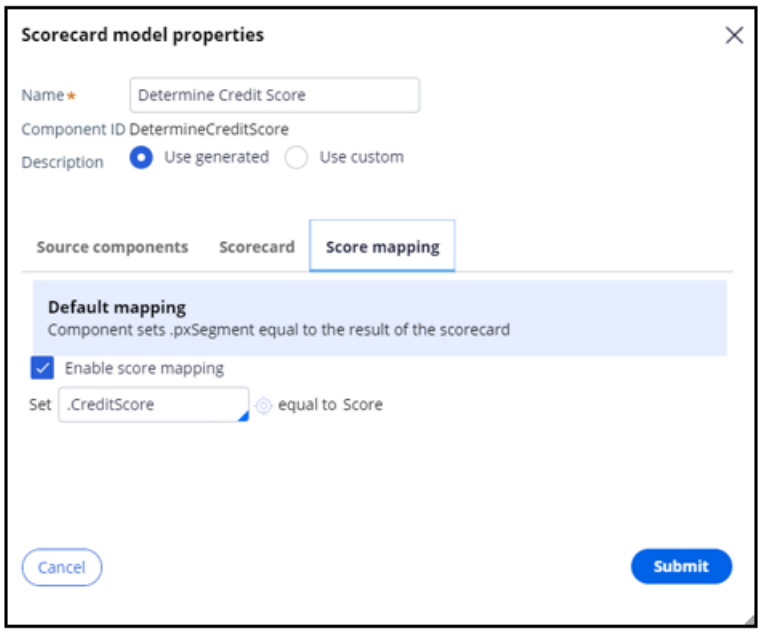
Calculate credit score

U+ has already created a Scorecard, **DetermineCreditScore**, which computes the customer's credit score. You can use the Scorecard in the decision strategy by adding a Scorecard component to the canvas.



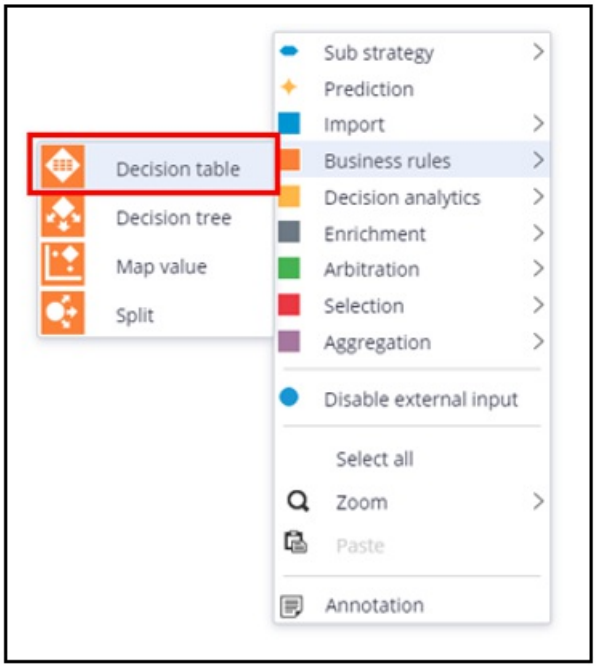


Since you need the raw score, enable **Score mapping** and set the Score value in the Credit Score property.

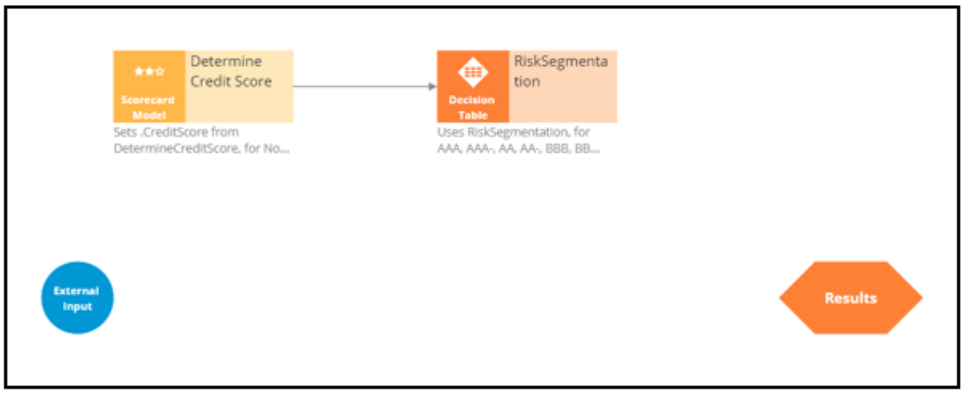


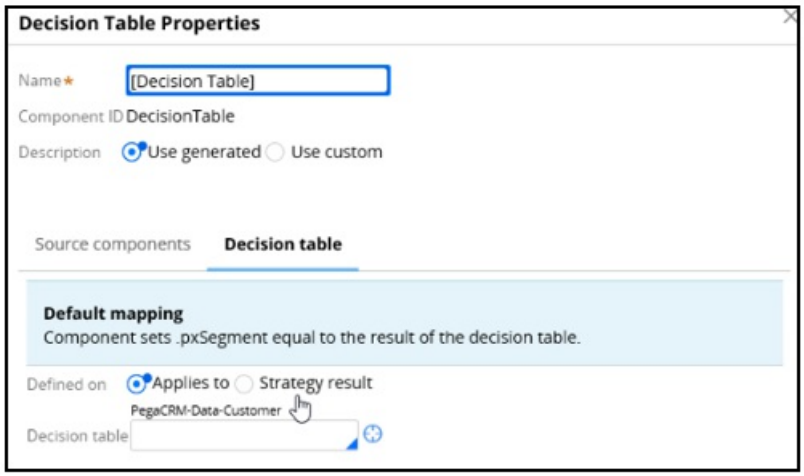
Define risk segment

To implement the risk segmentation requirements, use a **Decision table** component from the **Business rules** category, which outputs the customer risk segment.

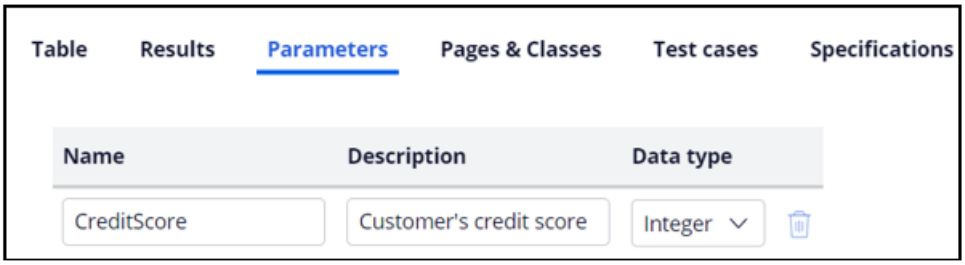


The Decision Table can be defined on the Strategy Result, SR class, or a Customer class. The choice often depends on where the properties used in the Decision Table are located. In this case the outstanding loan amount is a Customer property and the Credit Score calculation is an SR property, so both options are valid.



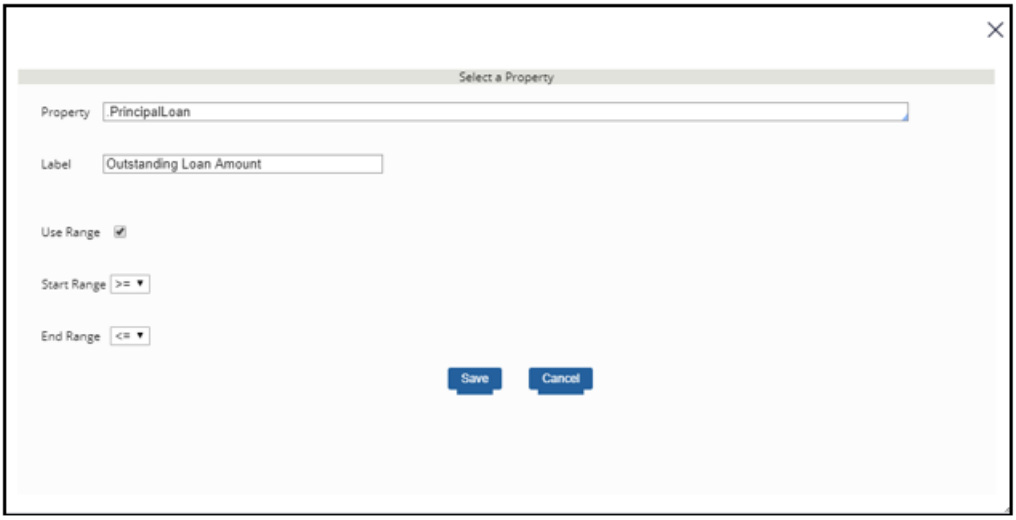


To use Credit Score as a parameter, you first need to define it on the **Parameters** tab. For this scenario, define the Customer class and reference it from the DT component.



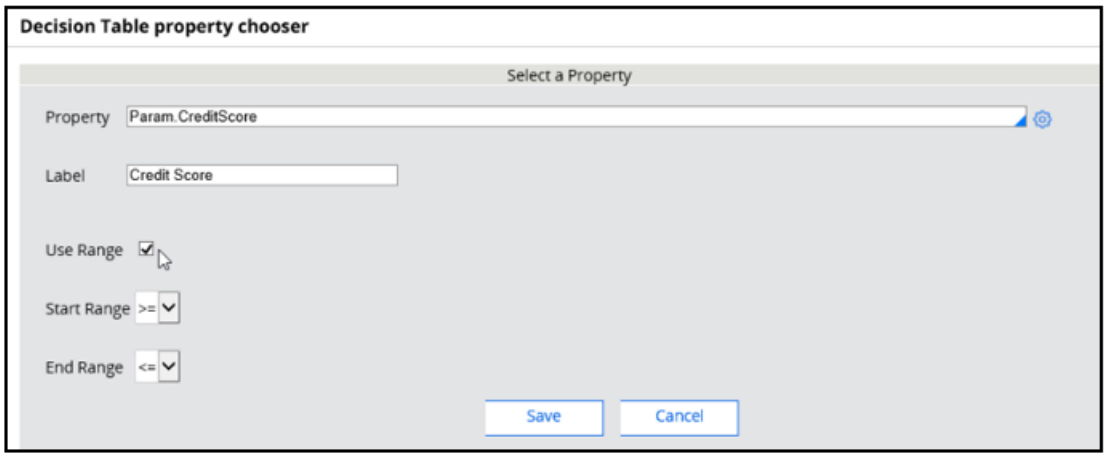
Creating the table requires two condition columns. First is the **Principal Loan**, which is the property in the data model representing the outstanding loan amount.

The Operator represents the condition applied to the column. In this case, **greater than**, which allows you to express the **Outstanding loan amount** condition from the requirement.

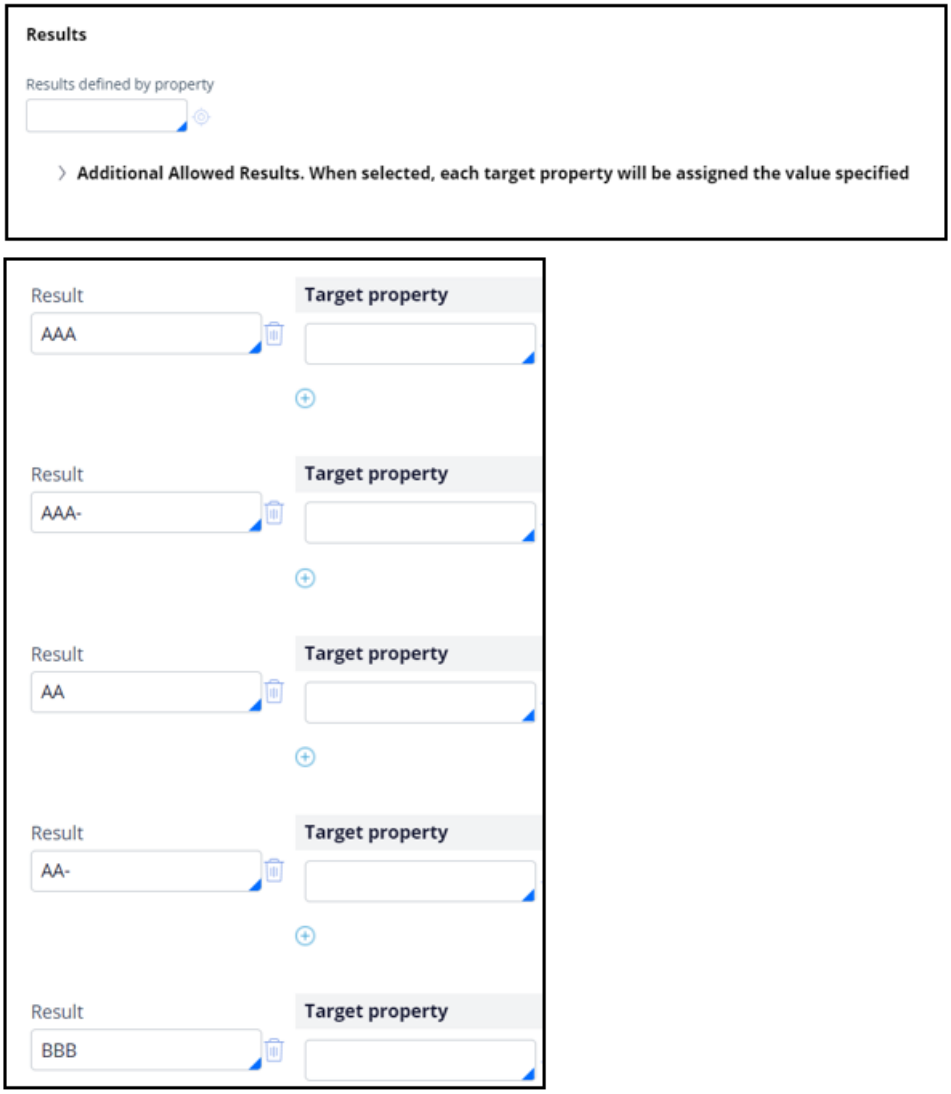


The second condition column is the Credit Score property. The Credit Score is a parameter, so you need to use the **Param.PropertyName** construct.

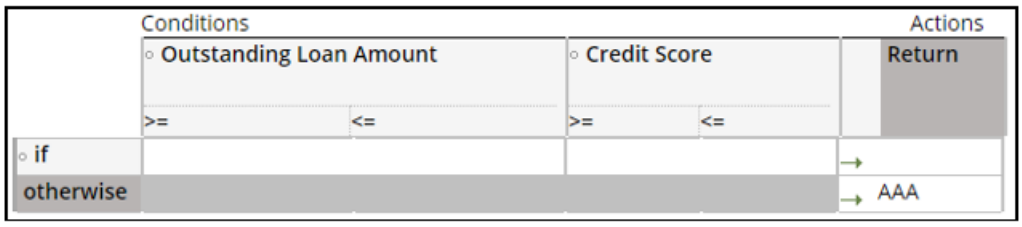
The **Use Range** checkbox groups the credit scores together.



Next, you need to specify the possible outputs of the Decision Table in the **Results** tab.



After adding the **Target properties** and **Results**, the Table will look like the following.



Fill in the bank’s requirements and specify a **Return** value for each row in the table.

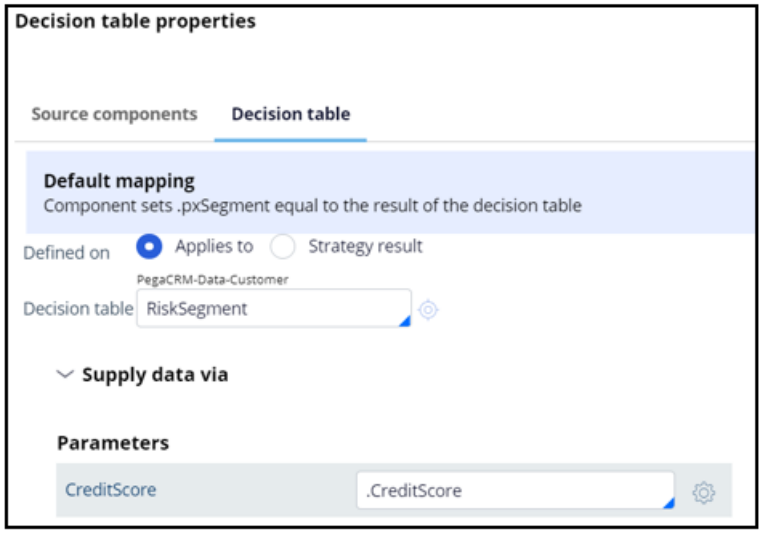
	Conditions				Actions
	Outstanding Loan Amount		Credit Score		Return
	>=	<=	>=	<=	
if		10000	600		AAA
else if	10000	25000	600		AAA-
else if	25000	50000	600		AA
else if	50000		600		AA-
else if		25000	400	600	BBB
else if	25000	50000	400	600	BBB-
else if	50000		400	600	BB-
else if			200	400	CCC
otherwise					CCC

Note, if you leave a value blank it is ignored by the Decision Table. For example, leaving the Credit Score value blank, means that credit score comparison always returns a value of True.

If none of the conditions are met, for example, if the loan amount is zero and the credit score is 50, the Otherwise path is taken; in this example the result will be CCC.

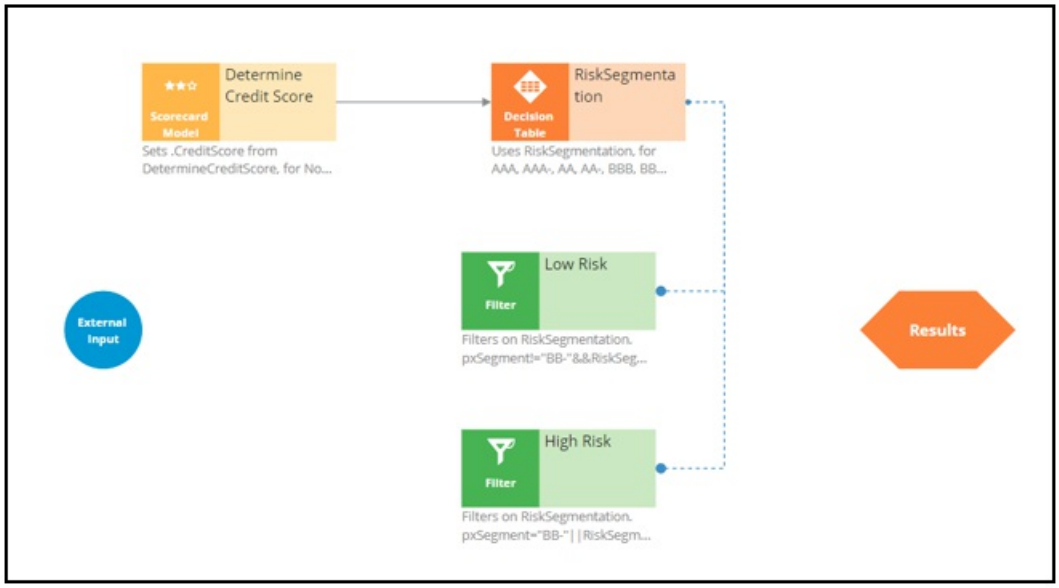
It is important to note that once a Decision Table condition is satisfied, the processing stops. For example, if the loan amount is 30,000, and the credit score is 650, the processing stops at row three returning the result “AA”. The other rows are not processed.

After the Table is configured, go back to the property panel of the Decision Table component and map the Decision Table parameter to a Strategy property.



Filter credit cards based on the calculated risk segment

Configure two Filter components to ensure that you identify High Risk and Low Risk customers.



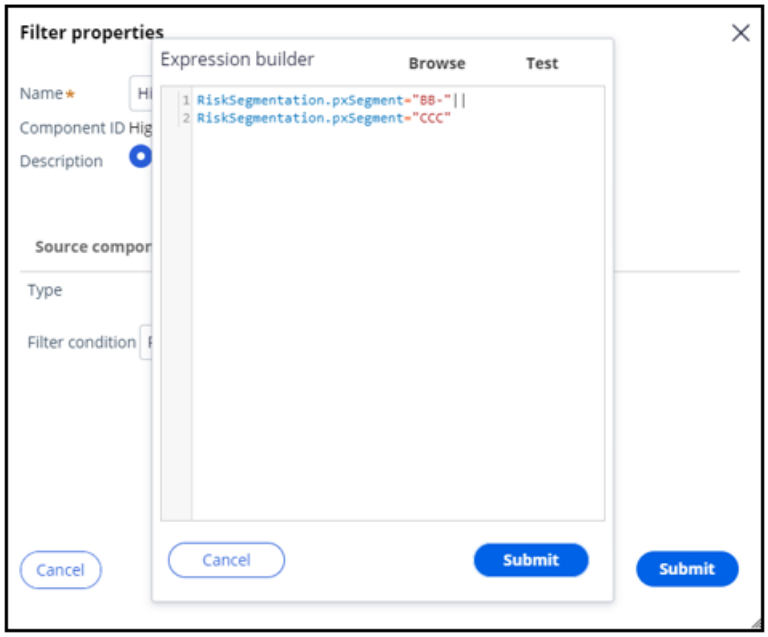
Customers in the risk category BB- and CCC are not eligible for credit cards, but all other customers are eligible. The **pxSegment** Strategy property contains the output of the Decision Table. So, create relevant expressions.

Low Risk Filter: *RiskSegmentation.pxSegment!="BB-"&&*

RiskSegmentation.pxSegment!="CCC"

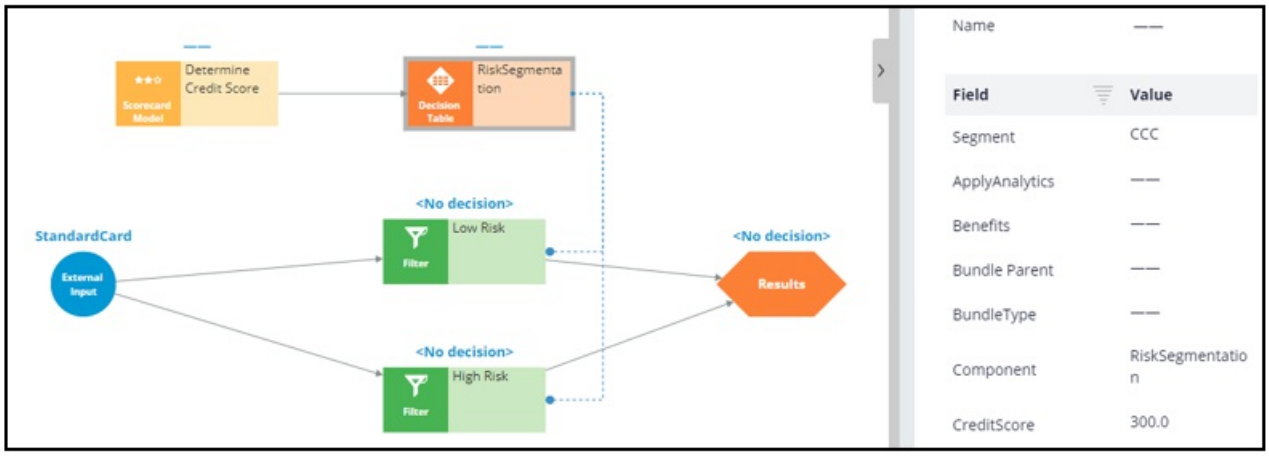
High Risk Filter: *RiskSegmentation.pxSegment="BB-"||*

RiskSegmentation.pxSegment="CCC"

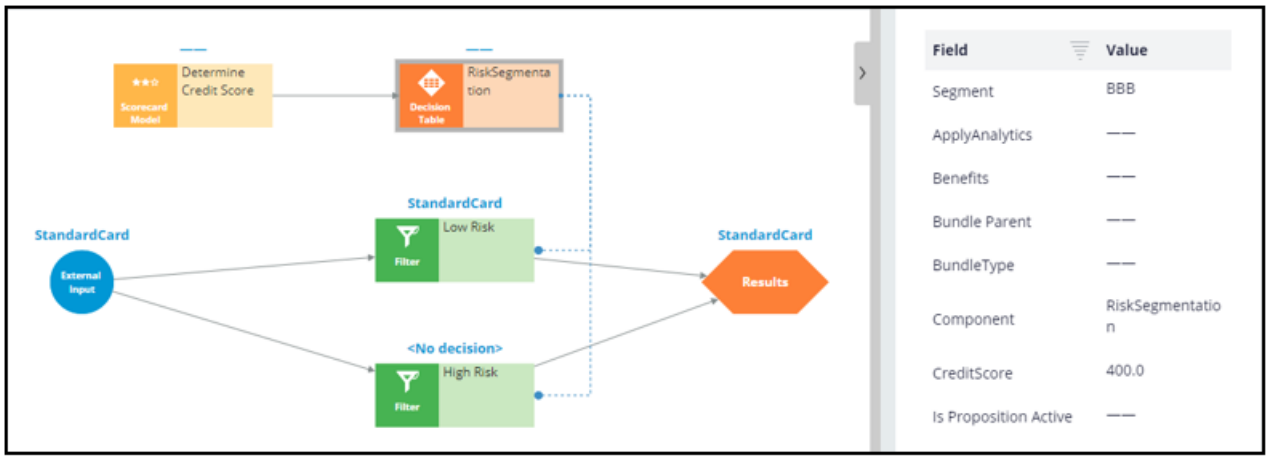


When you test the strategy using the Robert data transform, you will see that Robert falls under the category of high-risk customers, as he has a huge outstanding loan amount, over 50,000.

Therefore, you can expect his risk segment to be CCC.

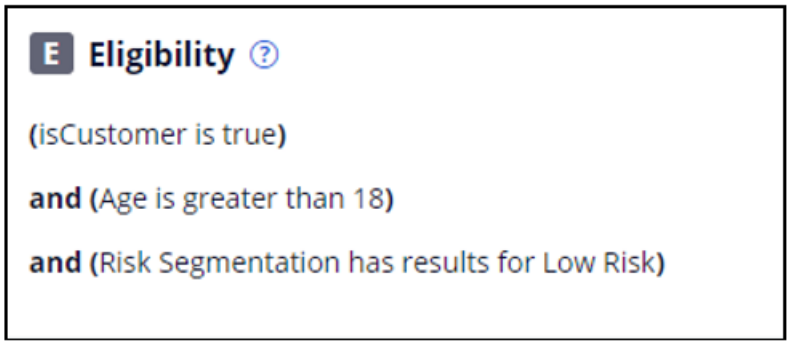


When you test the strategy for Arnold, who has an outstanding loan of 8000 and a credit score of 400, the Decision Table correctly classifies Arnold in the BBB risk segment and allows all credit cards for him.



Define the Eligibility criteria

Once the decision strategy is ready, you can complete the Eligibility criteria definition in Next-Best-Action Designer.



Creating customer risk segments using a decision table -- Fri, 07/24/2020 - 06:25
To get the full experience of this content, please visit https://academy.pega.com

Using predictive models in engagement strategies

Introduction

A predictive model is used to predict customer behavior such as offer acceptance and churn based on characteristics such as credit risk, income, product subscriptions, etc. Learn how to arbitrate between different groups of actions to display more relevant offers to customers. Gain experience using a predictive model in a decision strategy and learn how applicability rules can be defined to reflect the bank's requirements in a decision strategy.

Video

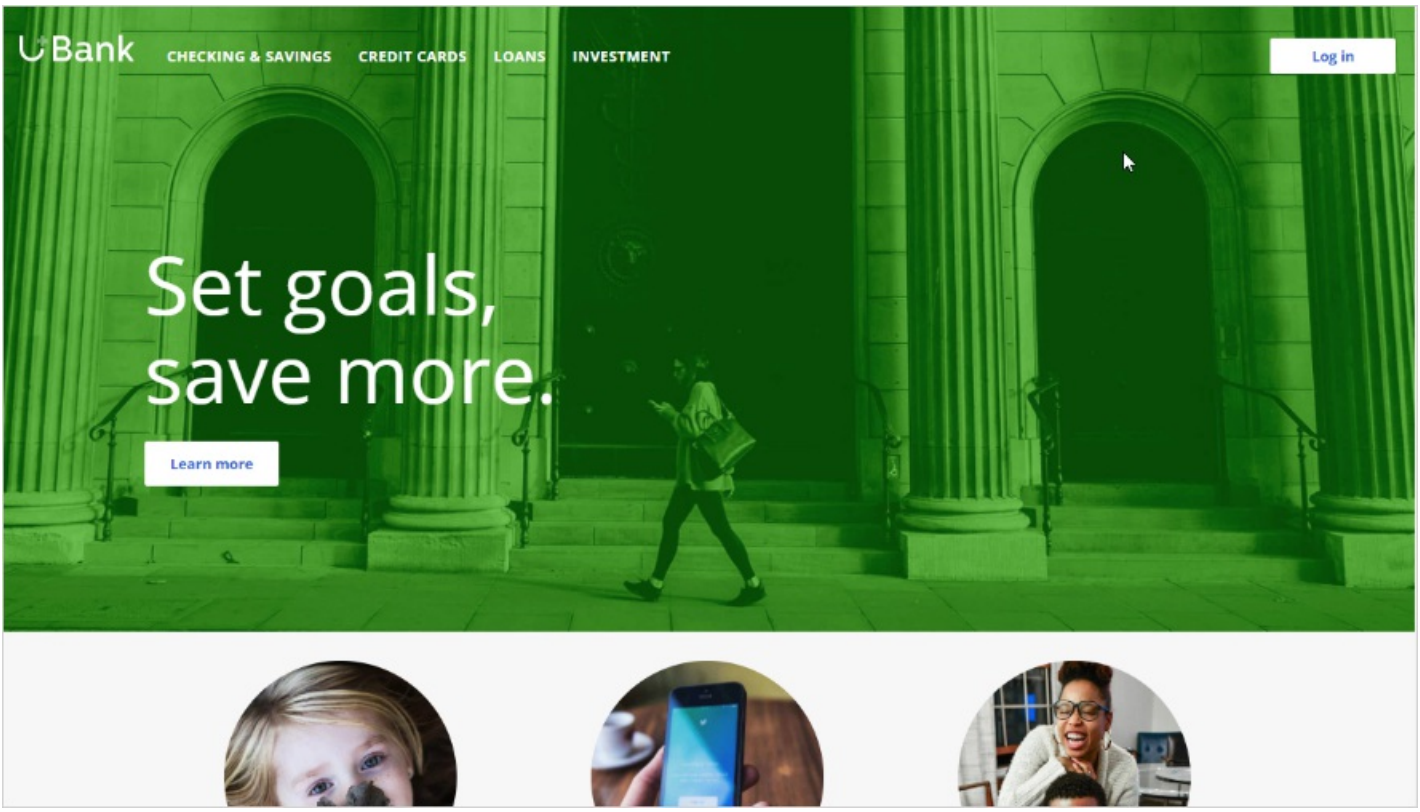
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This demo will show you how to use a predictive model in an engagement strategy to determine customer applicability for a retention offer.

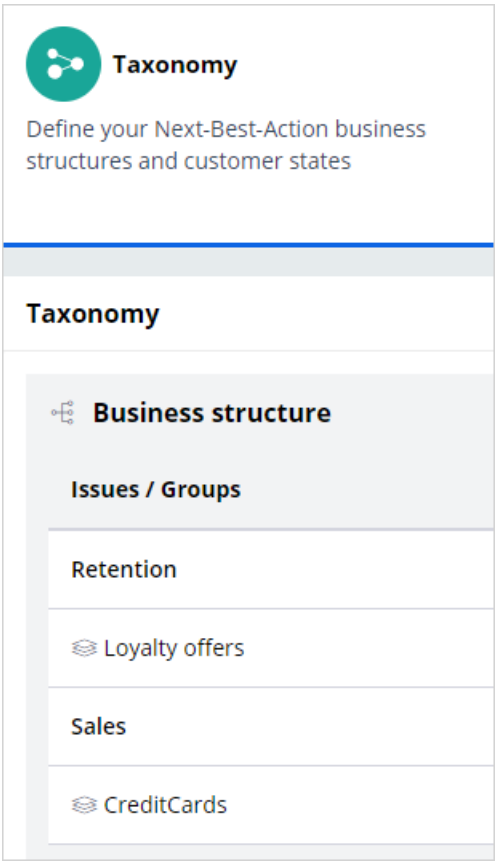
Currently, U+ Bank is doing cross-sell on the web by showing various credit cards to all customers who log in to their website.

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 will only be shown to loyal customers.

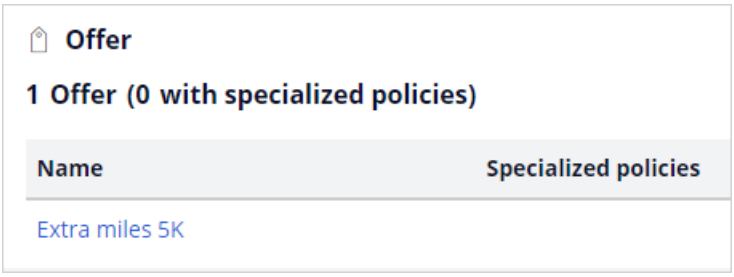


To meet this business requirement, a decisioning administrator has already set up the new business structure by defining a

new Issue/Group, the Retention/Loyalty offers under Taxonomy.



The retention offer, Extra Miles 5K, has also been created for this issue/group.



The next step is to create an applicability condition that makes a customer qualify for a retention offer when there is a churn risk. Since the churn risk is predicted using a predictive model, you need to use a decision strategy to define this condition.

A decisioning administrator has already created the RetentionStrategy. You can use this strategy as basis for the applicability condition.

A data scientist has already created the predictive model using Pega's built-in machine learning capabilities to identify the customers who are likely to churn in the near future.

To use the predictive model in the decision strategy, add a predictive model component to the canvas and configure it to reference the model. Note that this list contains all the predictive models that are available in the system. These include models created using Pega machine learning, imported PMML or H2O.ai models, and externally executed models built in Google ML or Amazon SageMaker.

Select the desired predictive model from the list.

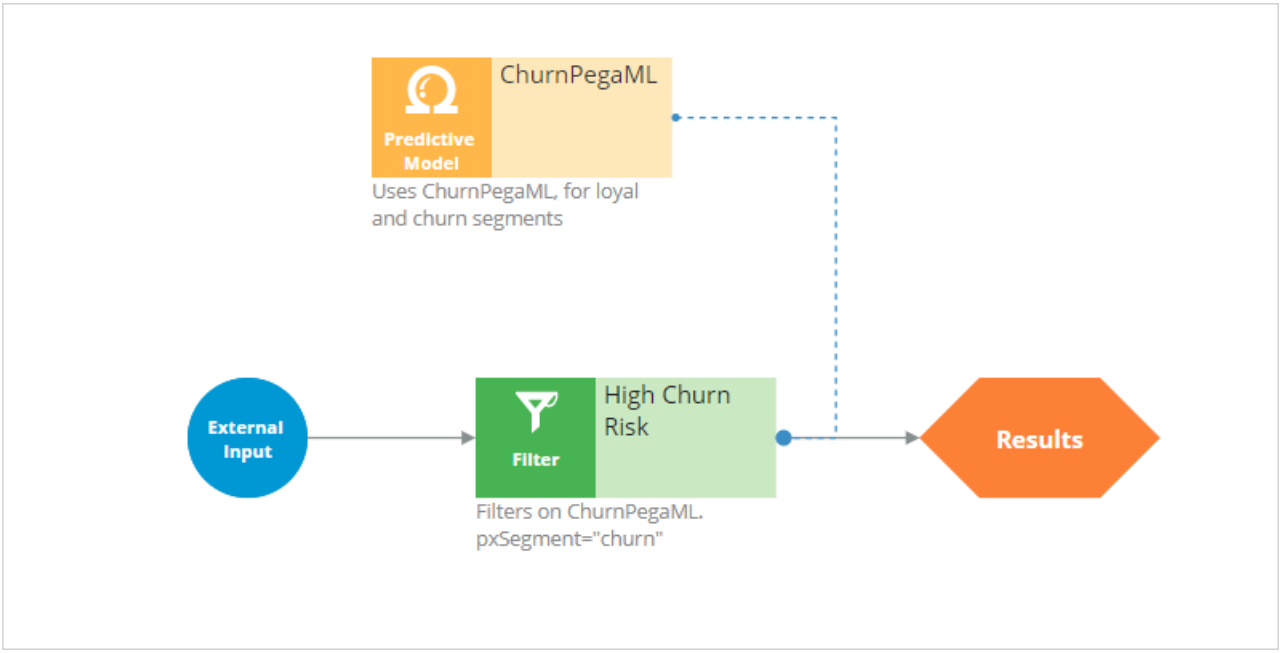
When you open the predictive model, notice that in the**Classification groups** section, the classes are divided into two groups, and their results are labeled "loyal" and "churn" to predict customer behavior.

You can now use a filter component to express the condition under which the retention offer is applicable.

You want this strategy to output a retention offer only if the result of the predictive model is “churn”. The result of the predictive model is stored in the pxSegment property.

Therefore, define the filter condition to output a retention offer for which the pxSegment property is equal to “churn”.

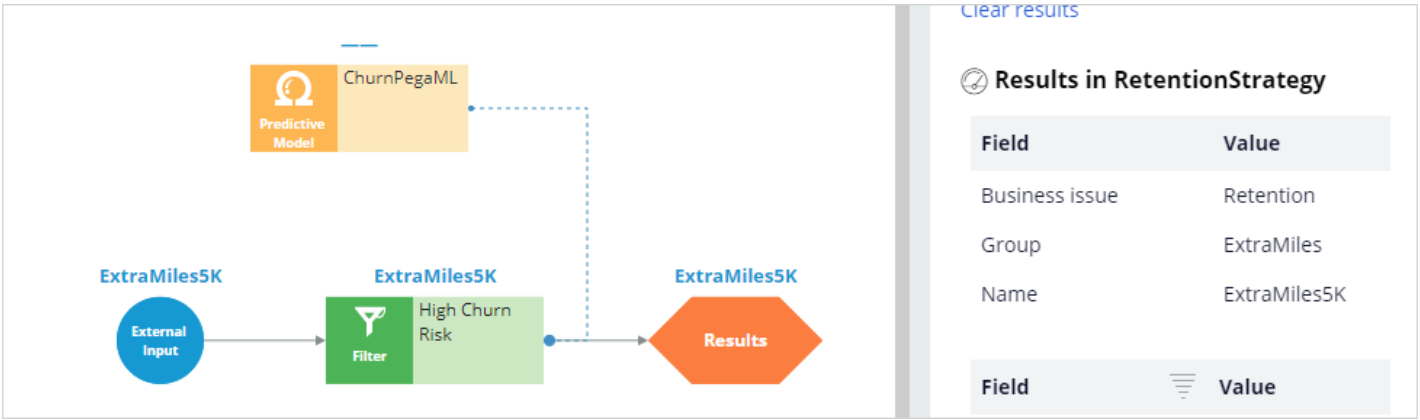
Save the strategy.



Now, test the strategy using the customer profiles,**Troy** and **Barbara**.

For external inputs, consider all available retention offers.

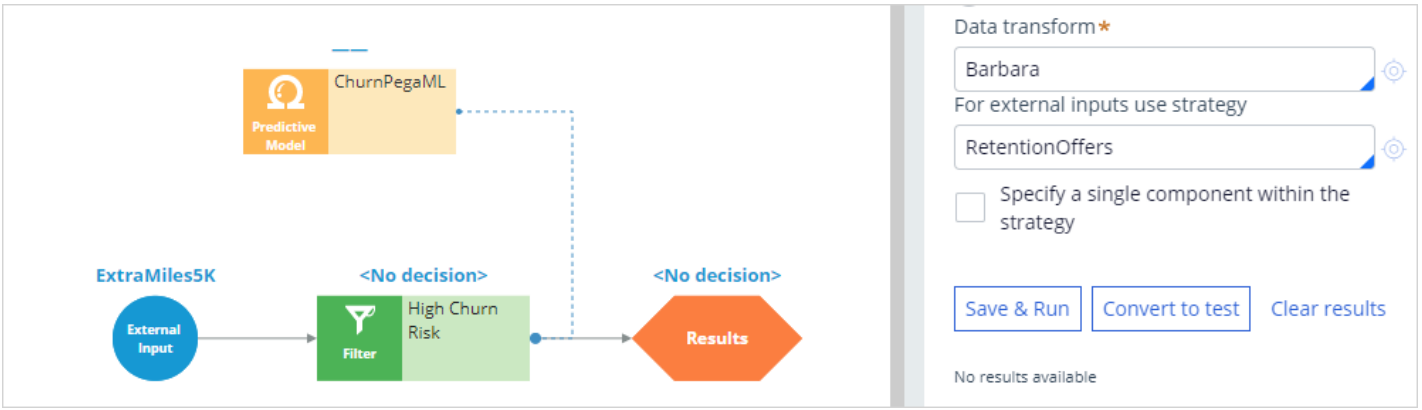
The strategy outputs a result for **Troy** because the result of the predictive model is "churn".



Field	Value
Segment	churn
ActionContext	——
ApplyAnalytics	——
Benefits	——
Bundle Parent	——

Now, repeat the test to verify results for the**Barbara** data transform.

The strategy does not have a result for **Barbara**, because the Segment value is "loyal".



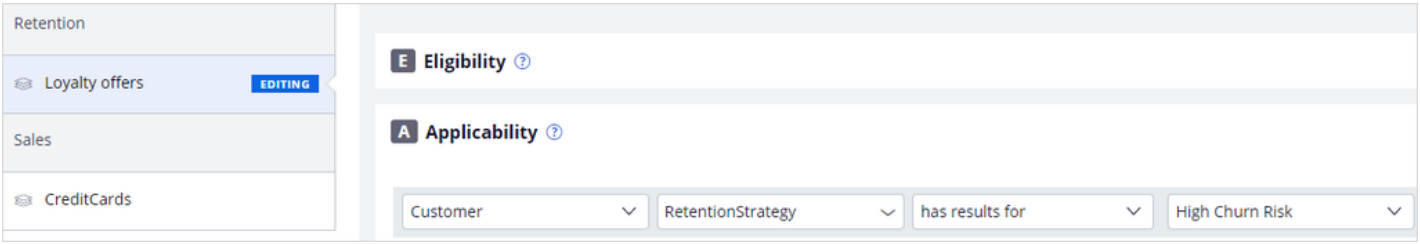
Field	Value
Segment	loyal
ActionContext	——
ApplyAnalytics	——
Benefits	——
Bundle Parent	——

The strategy is not available to the U+ website. By checking it in, you are committing your changes, so they will be put into effect.

You can now use this strategy in the Next-Best-Action Designer engagement policy as an applicability condition for the **Loyalty offers** and **CreditCards** groups.

The first business rule you need to implement is: the**Loyalty offers** group is applicable only for high risk customers. To implement this rule, in the Applicability section, define a condition for the customer field.

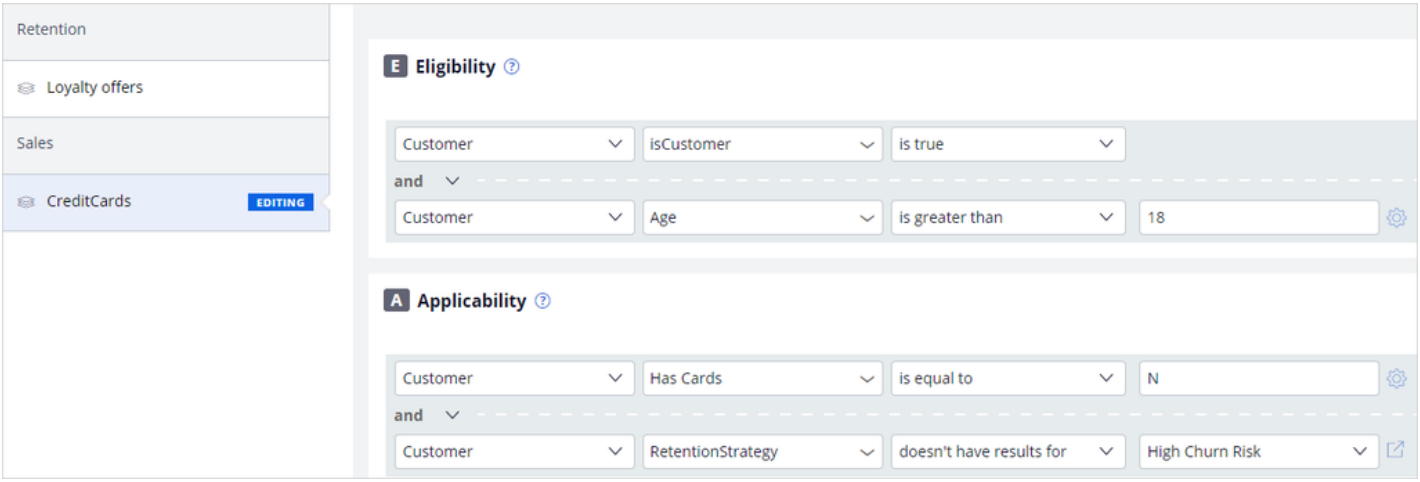
Select Strategy and then select RetentionStrategy. The condition is: RetentionStrategy has results for the High Churn Risk.



The second business rule you need to implement is: U+ Bank wants to show credit card offers only to customers who remain loyal for now; meaning the **CreditCards** group is not applicable for high risk customers.

To implement this rule, modify the Applicability section of the CreditCards group.

Select the **RetentionStrategy**. The condition is: RetentionStrategy doesn't have results for the High Churn Risk.

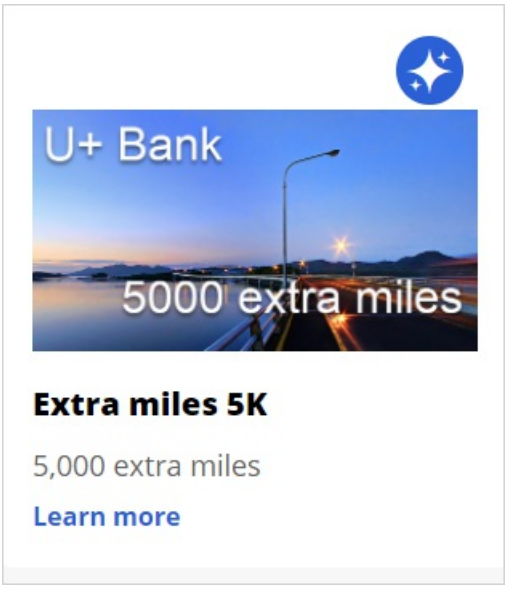


Save the configurations.

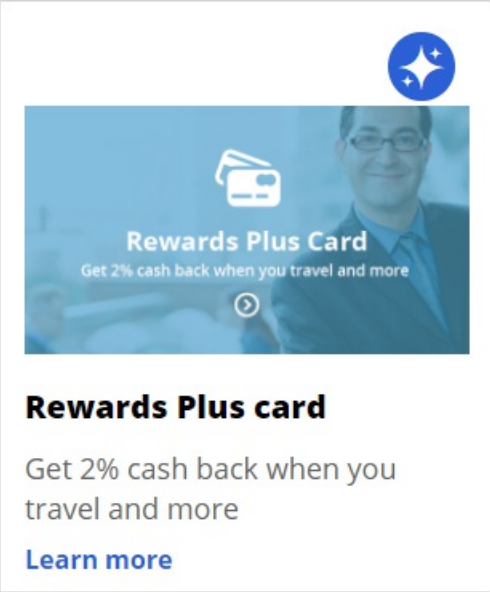
Once the applicability conditions are defined, you need to amend the channels configuration. Since U+ Bank introduced a new group, **Loyalty offers**, which belongs to a new business issue,**Retention**, you need to select the results from the appropriate business structure level. In this case, the bank wants to arbitrate between two different business issues: Sales and Retention. Therefore, select All Issues/All Groups from the business structure level.

Saving this completes the required configurations.

On the U+ bank website, when you log in as**Troy**, notice that the retention offer is displayed. This is because**Troy** is predicted to churn in the near future.



Now, when you log in as **Barbara**, notice that the credit card offer is displayed because she is predicted to remain loyal for now.



This demo has concluded. What did it show you?

- How to use a predictive model 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.

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.

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.

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.

Adding more tracking time periods for contact policies

Introduction

Suppression rules determine when and for how long an action or group of actions can be shown to a customer. These rules put an action on hold after a specific number of outcomes are recorded for some or all channels. Learn how to use suppression rules to track customer behavior for time periods that you define.

Video

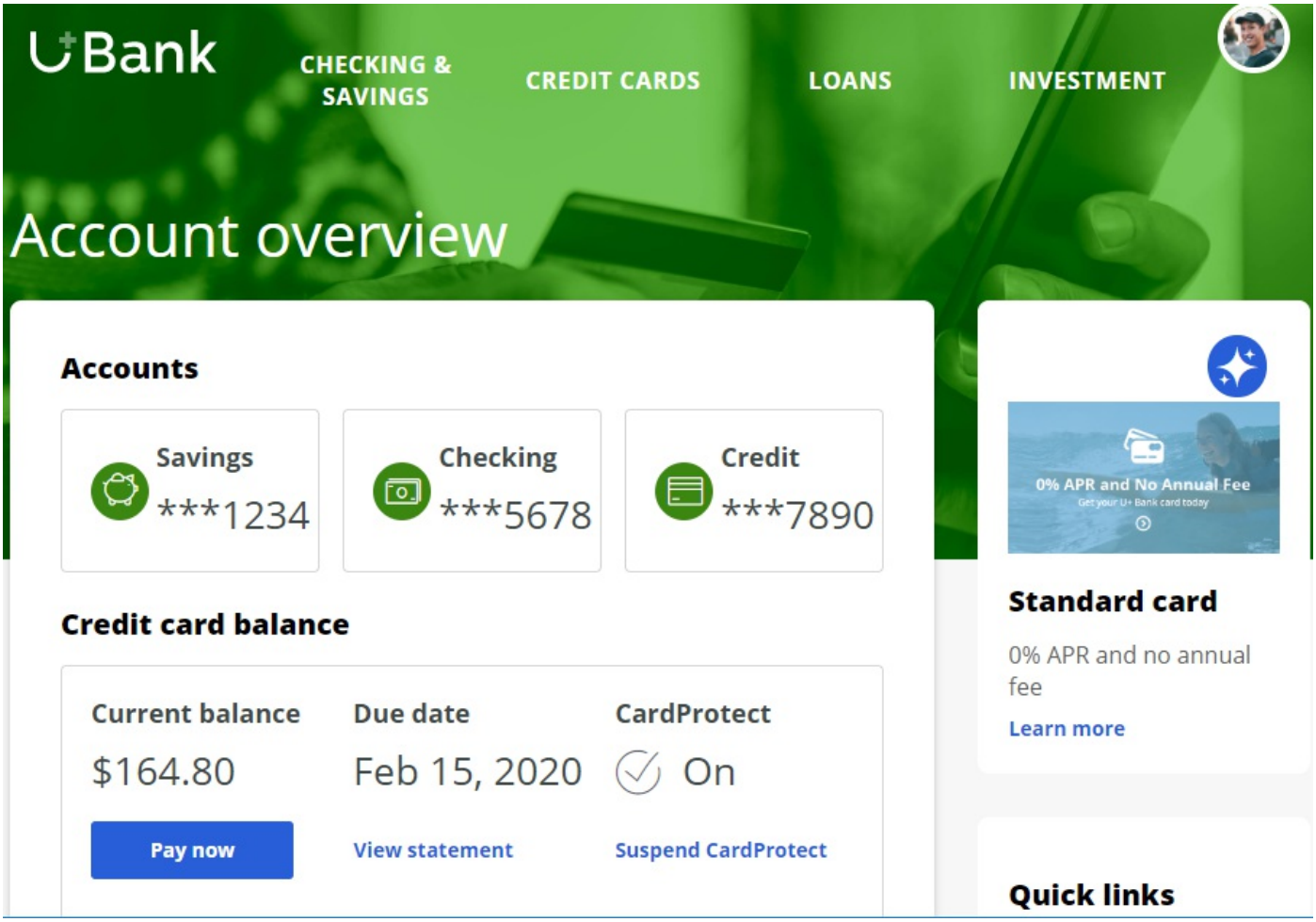
A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This demo illustrates how to add more time periods to contact policies to allow customer behavior tracking for periods other than the default 7 or 30 days.

U+, a retail bank, is using the Pega Customer Decision Hub™ to display marketing offers to customers on its website.

Currently, U+ Bank displays multiple credit card offers to each customer who logs in to the website. For example, if customer Troy logs in to his accounts page, the Standard card and Rewards card offers are displayed based on eligibility criteria defined by the business. If Troy clicks on the banner and then returns to his accounts page, the same set of offers is still displayed.



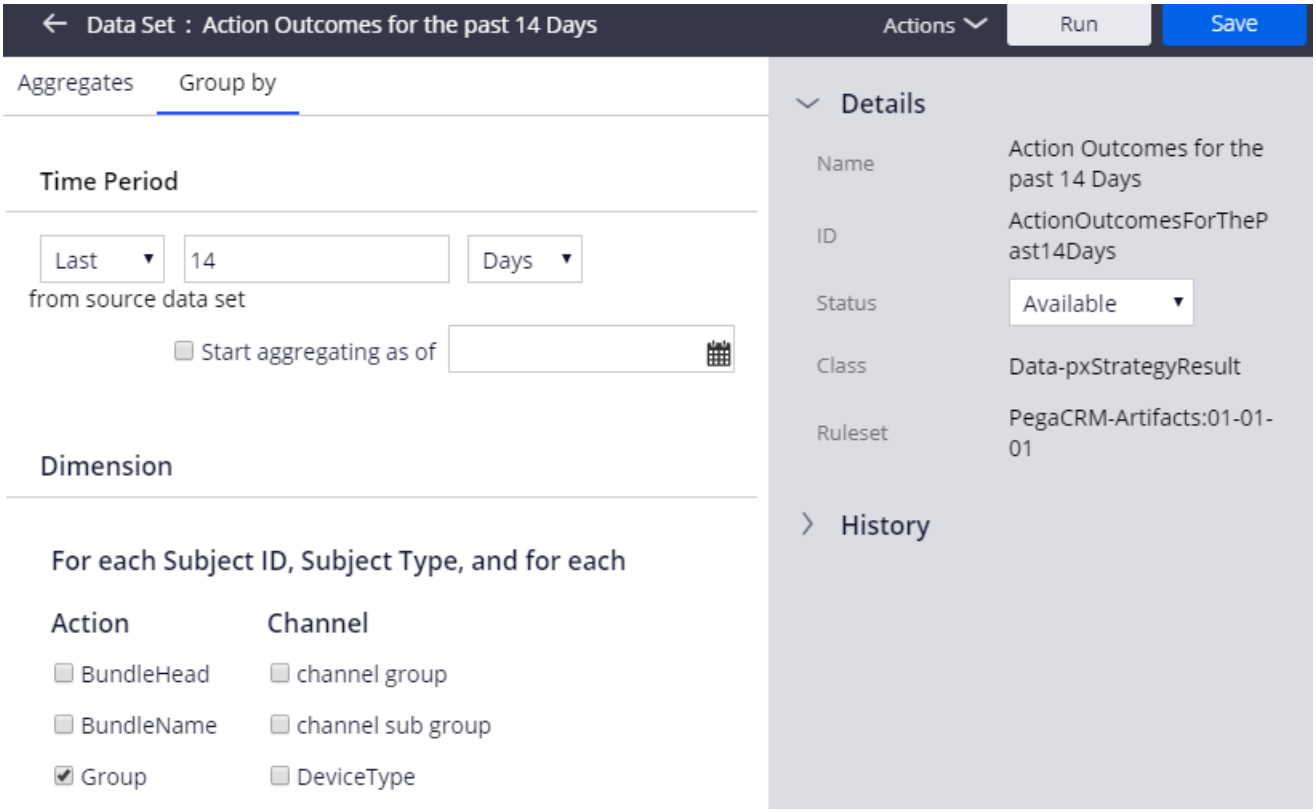
The bank does not want to show the same offer to customers who have clicked on it three times in the last 14 days.

This is the Pega Customer Decision Hub portal. Navigate to Next-Best-Action Designer to create a new contact policy to track customer responses for a period of 14 days. Notice that by default, responses are tracked for a period of 7 or 30 days.

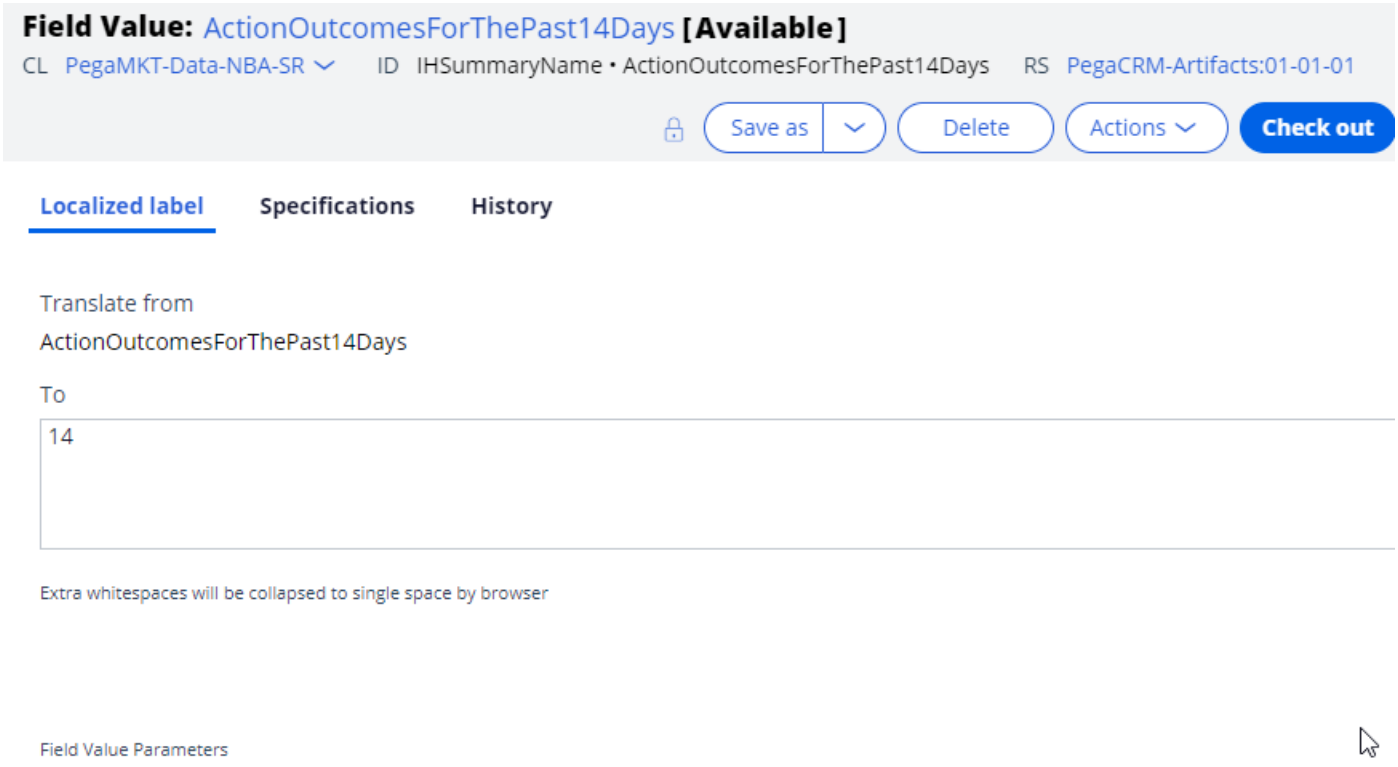
The values in this drop-down come from an artifact called Field Values in Pega. These are a set of valid values for a field displayed in the application. In this case, these values represent the existing IH Summary rules.

You must now add an additional tracking period of 14 days based on your business requirement.

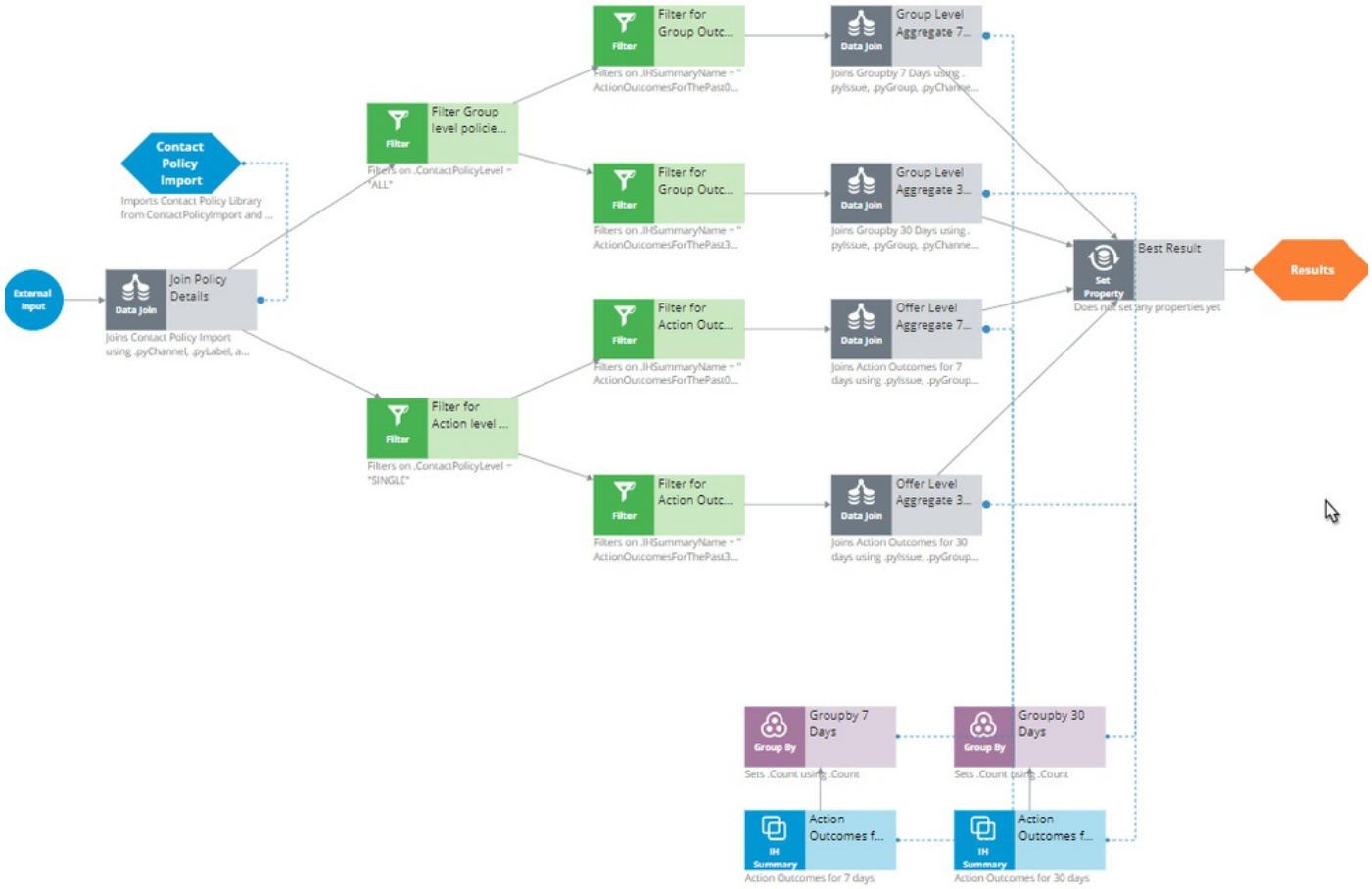
For this, you need to create a new Interaction History Summary rule that determines a customer’s responses over a 14-day period. The easiest way to do this is by creating a copy of an existing rule, such as “Action Outcomes for the past 7 Days”. Modify the rule to aggregate the customer responses over 14 days.



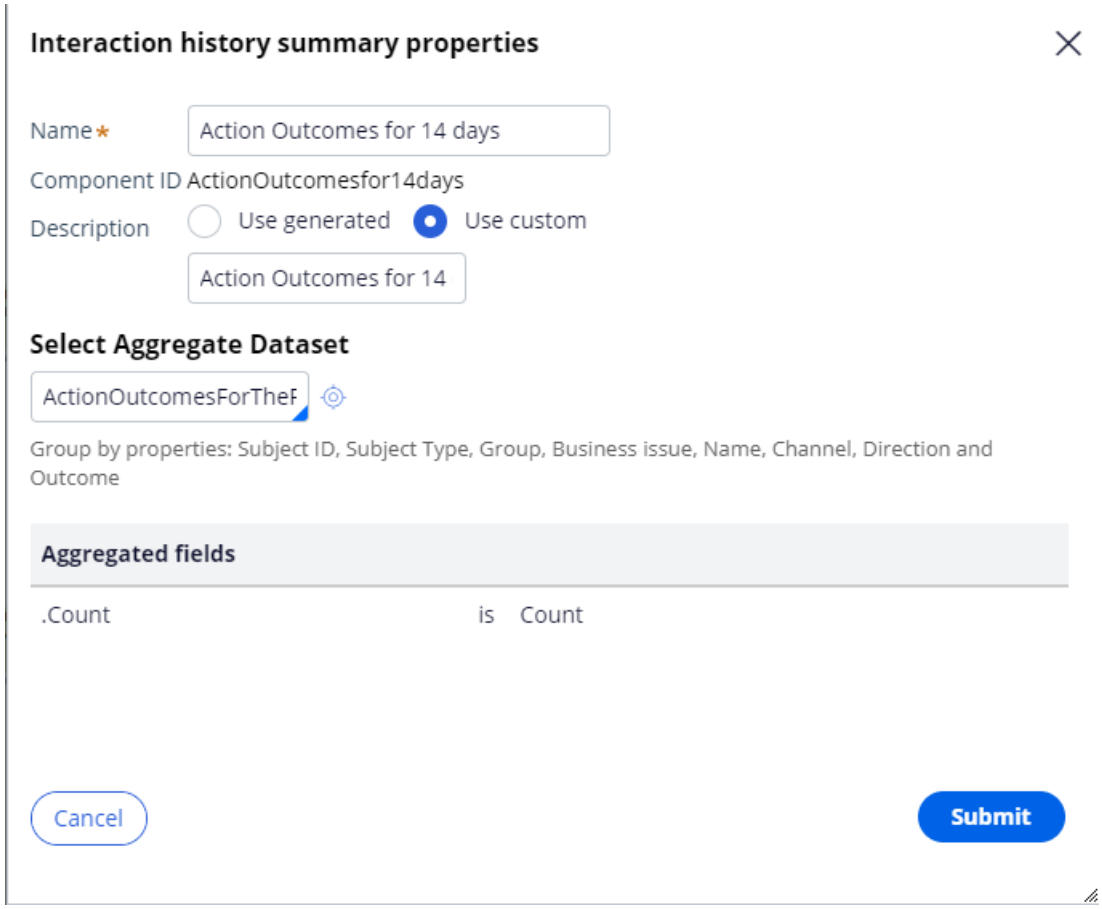
Then, create a copy of an existing field value. This will represent the newly created IH Summary rule in a selection list that will be presented to the user when they create the new contact policy.



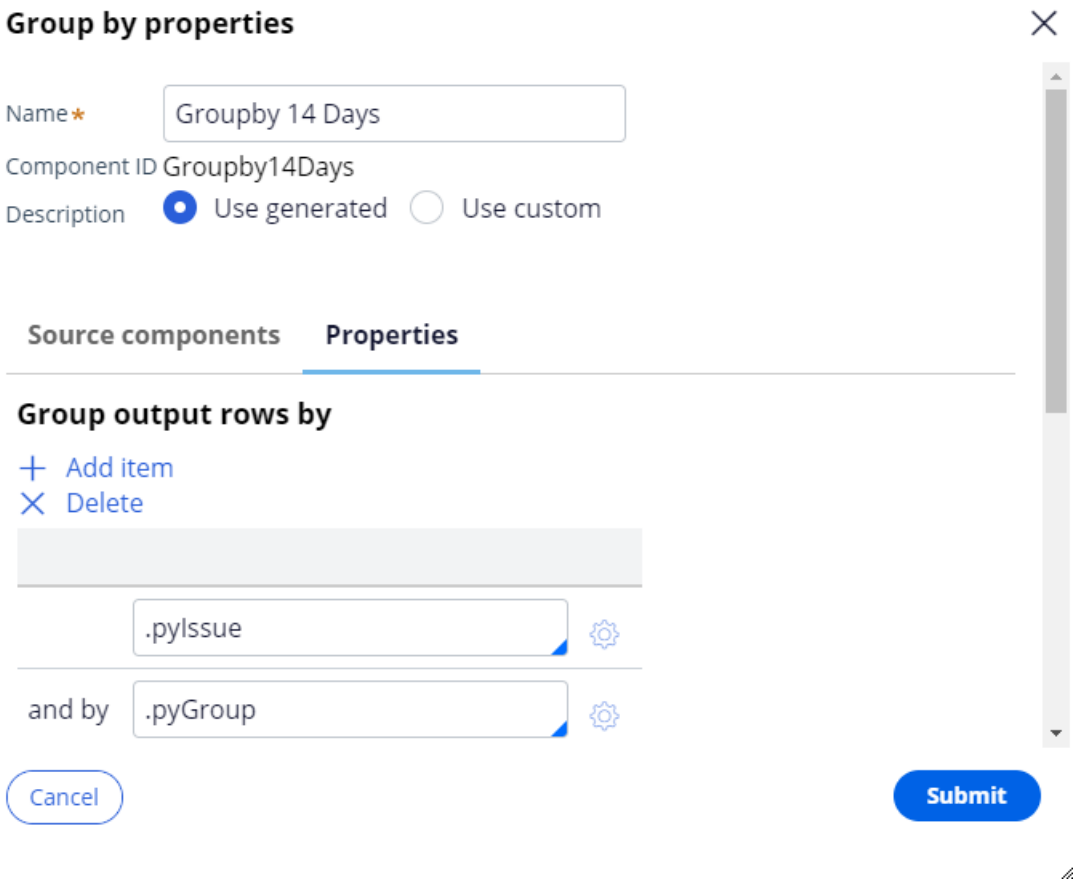
The CheckSpecificChannelLimits decision strategy is used to implement the contact policies. This strategy takes into account the number of responses from customers and suppresses Actions based on the contact policy configuration. It also checks the contact policy threshold for specific channels, such as web, email, or SMS. It is a sub-strategy of the BehavioralLimits strategy.

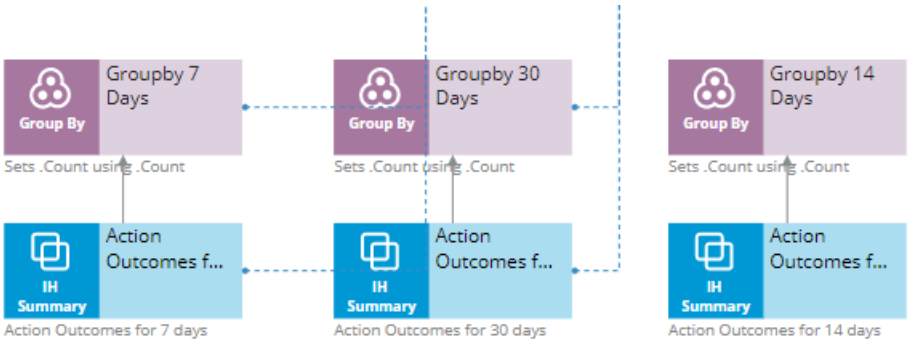


Now, let’s see how to extend the strategy to incorporate the new tracking period. First, add an Interaction History Summary component to retrieve the customer’s interactions with the company over the past 14 days. This IH Summary component must include the IH Summary rule to track the customer’s responses over the 14-day period.

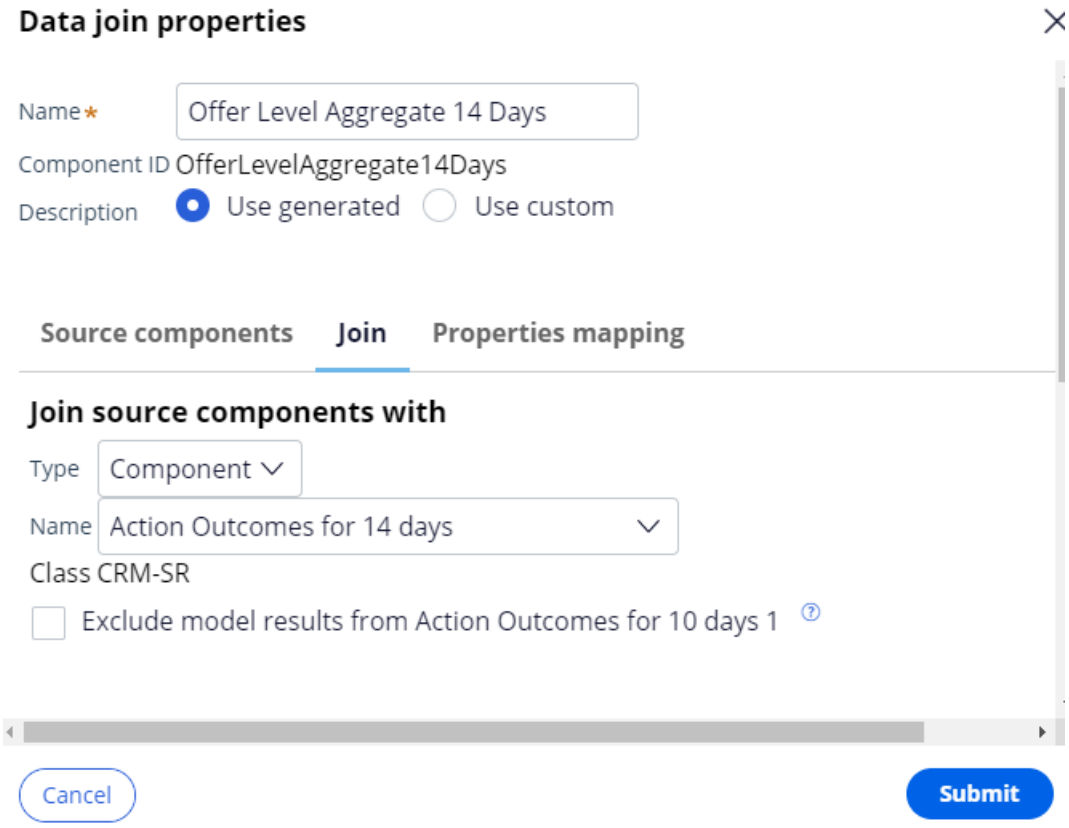


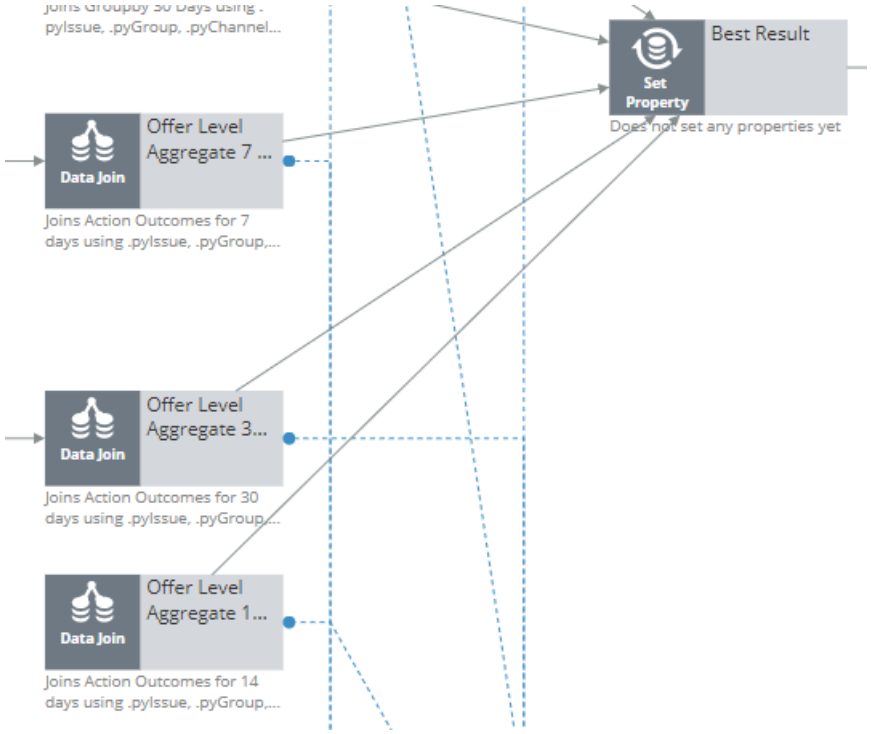
Then, add a Group By component to calculate the sum of customer responses for a particular Issue/Group as received from the IH Summary component.



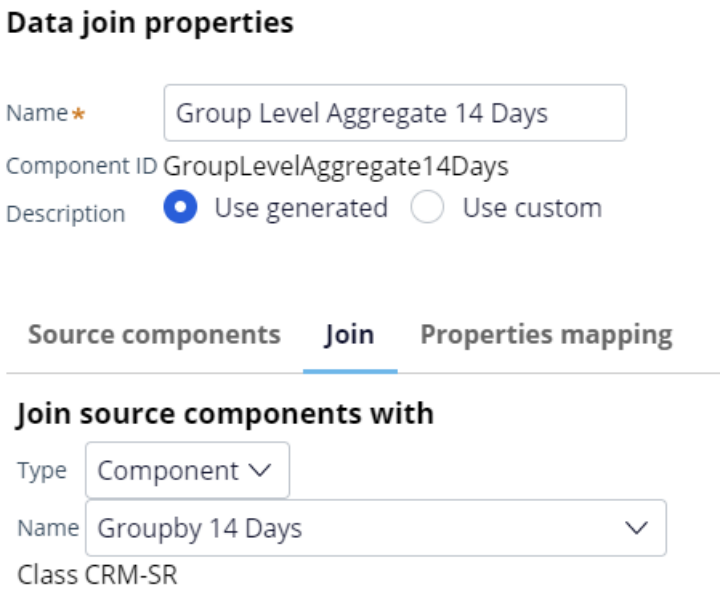


Next, add a Data Join component to join the data received from the IH Summary component to the strategy. That is, the data from the IH Summary component is referred to and used in the strategy via the Data Join component.

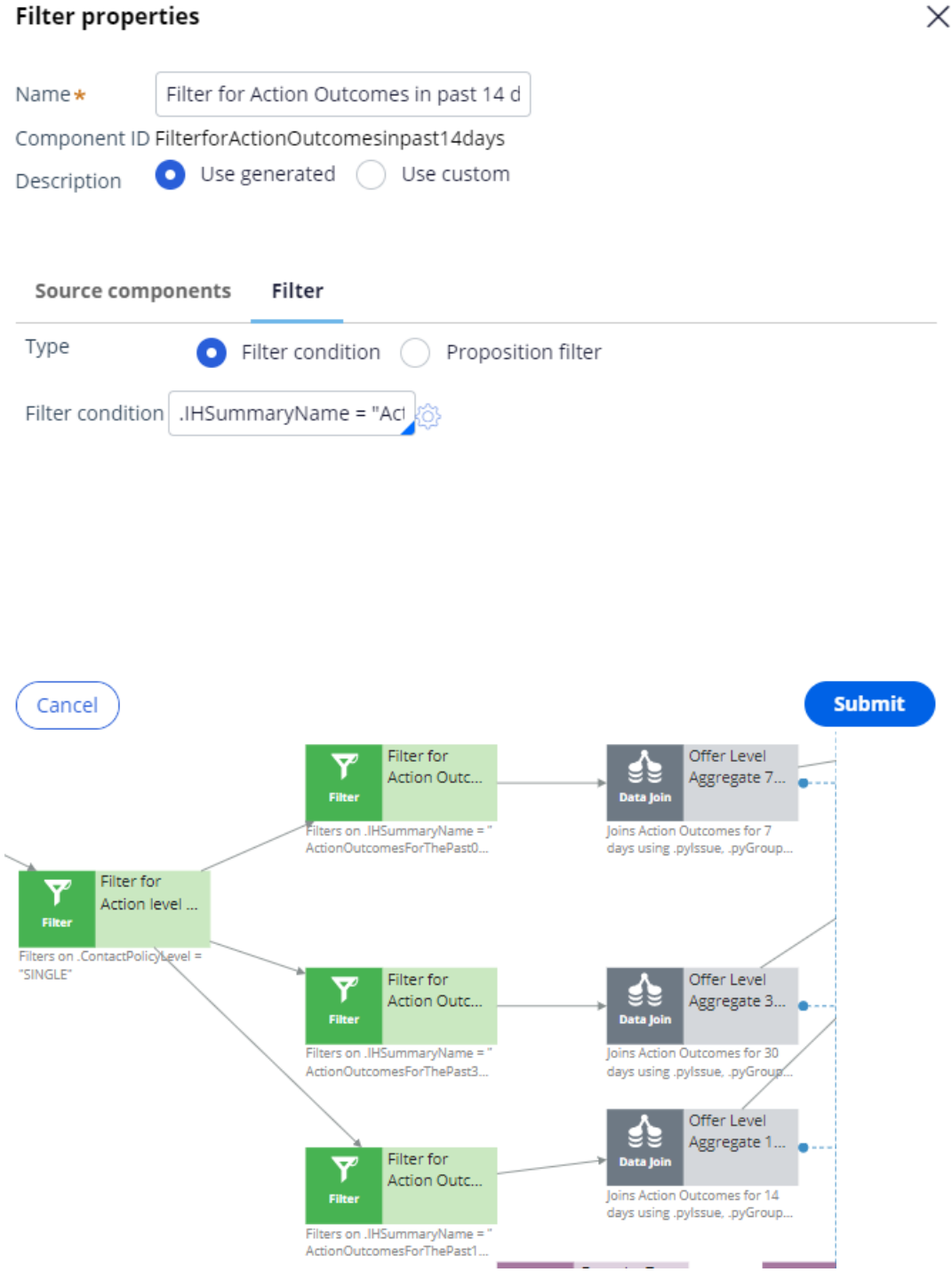




Then, add a Group-level Data Join component to join the data received from the Group By component to the strategy. That is, the data from the Group By component is referred to and used in the strategy via the Data Join component.



Then, add a Filter component to filter Action-level outcomes for a particular channel.



Finally, add a Filter component to filter Group-level outcomes for all channels.

Filter properties

Name * Filter for Group Outcomes in past 14 days

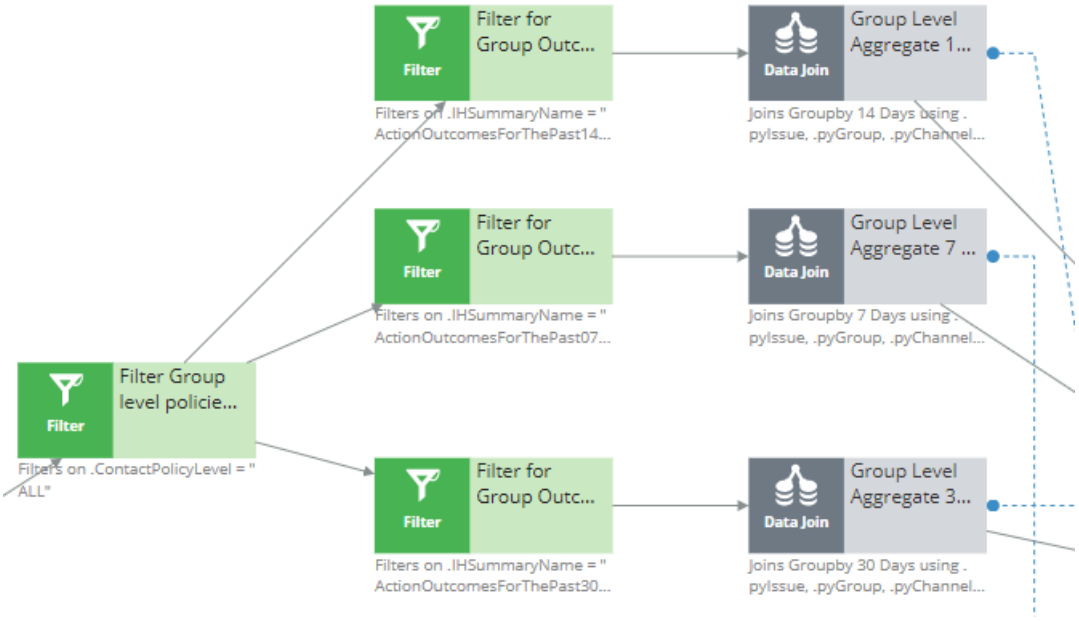
Component ID FilterforGroupOutcomesinpast14days

Description (•) Use generated ( ) Use custom

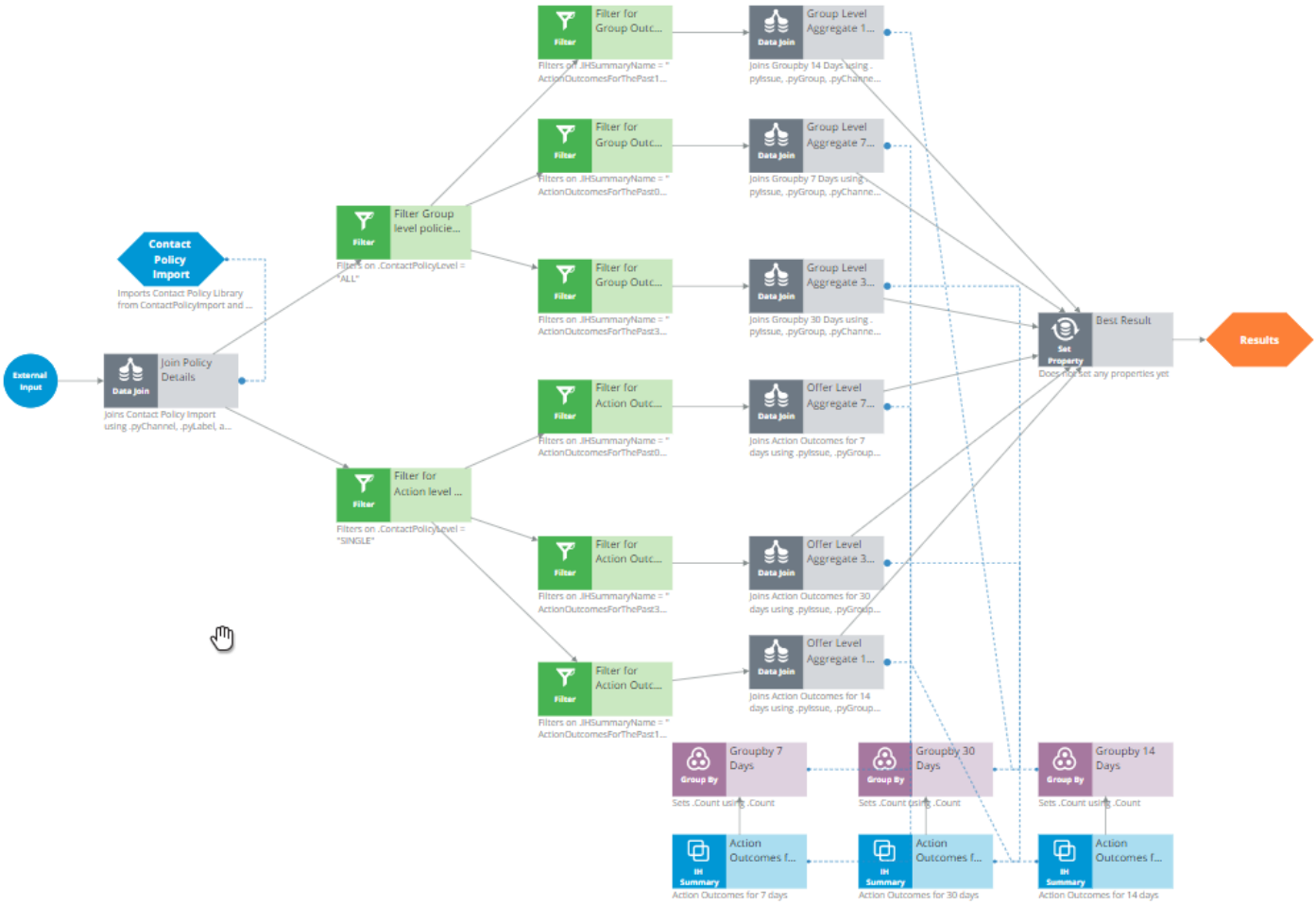
Source components | Filter

Type (•) Filter condition ( ) Proposition filter

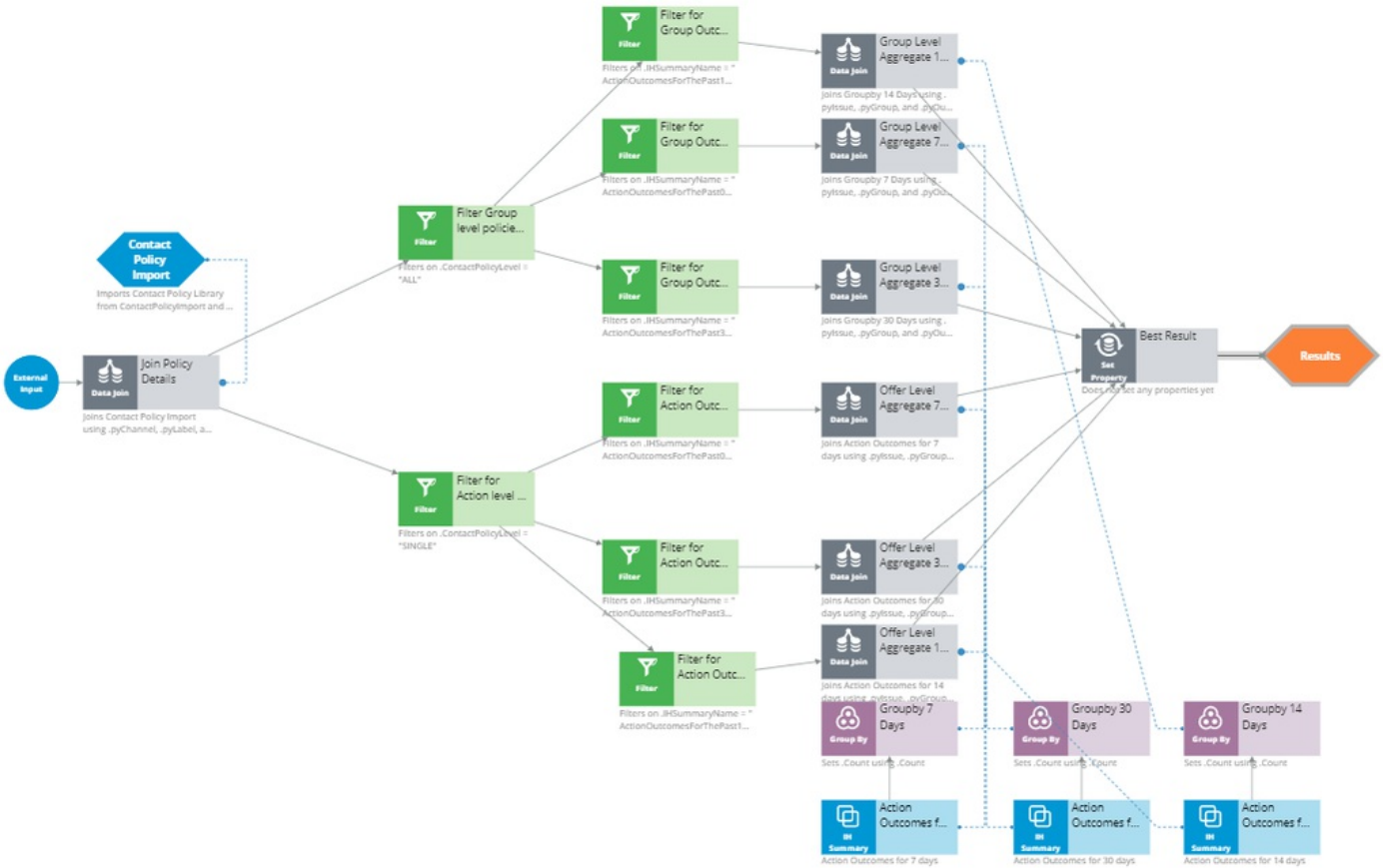
Filter condition .IHSummaryName = "ActionOutcomesForThePast14Days"



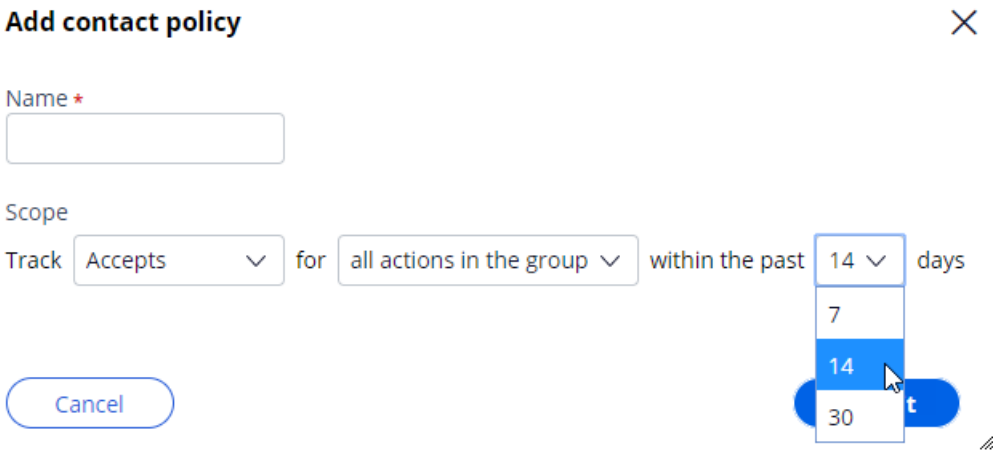
The extended strategy must look like this.



Now you must extend the CheckAllChannelLimits strategy in the exact same way. This strategy checks the contact policy threshold for all channels. It is a sub-strategy of the BehavioralLimits strategy.



Now, navigate to Next-Best-Action Designer to verify that the new tracking time period for the contact policy is available.



This demo has concluded. What did it show you?

- How to use an IH Summary rule to track time periods in a contact policy.
- How to extend the strategy for a specific channel to add a new tracking time period.
- How to extend the strategy for all channels to add a new tracking time period.
- How to use the new tracking time period in Next-Best-Action Designer.

Adding more tracking time periods for contact policies -- Fri, 07/24/2020 - 06:26
To get the full experience of this content, please visit https://academy.pega.com

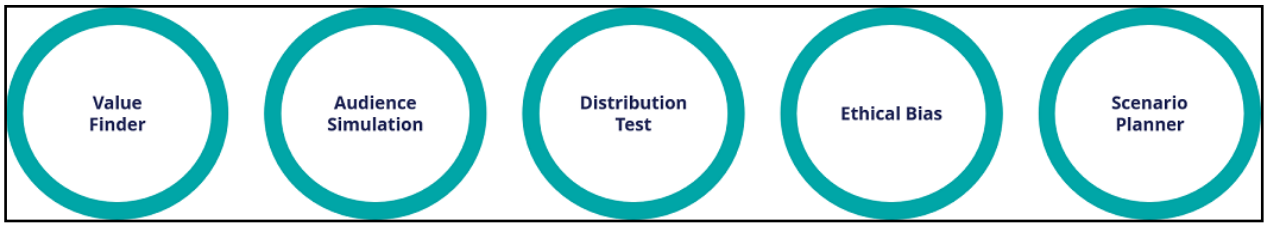
Simulation testing

Simulation types

By running simulation tests in Pega Customer Decision Hub™, you can understand the effect of business changes on your next-best-action strategy framework. The Pega Customer Decision Hub portal offers a large variety of simulations. The simulation capability ranges from simulations that help identify under-served customers to simulation tests that allow you to investigate how an introduction of a new engagement policy might affect actions offered across a segment of customers.

The different types of simulation testing available in Pega Customer Decision Hub are: Value Finder, Audience Simulation, Distribution Test, Ethical Bias, and Scenario Planner.

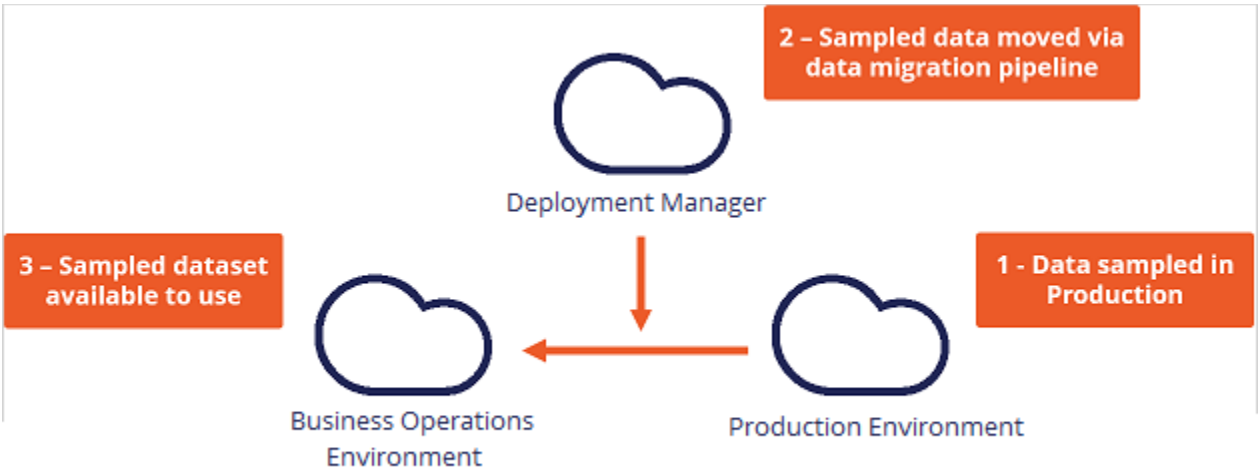
Click on the hotspots to learn more about each of the simulation types.



1. Value Finder allows clients to engage more empathetically by identifying and profiling under-served customers, then suggesting actions for improvement.
2. Audience Simulation tests the configuration of engagement policies against a segment of the audience.
3. Distribution Test enables you to unit test a decision strategy.
4. Ethical Bias tests the engagement policies for unwanted bias.
5. Scenario Planner enables users to easily simulate “what-if” scenarios so they can more accurately forecast results, optimize strategies to hit specific goals, and explore the potential tradeoffs of each option.

Business-as-usual

Simulations are run in a Business Operations Environment (BOE) that is specifically designed to build, simulate and optimize changes. A sample dataset, which includes interaction history and adaptive models from the production environment, is created via a pipeline into the BOE. This dataset is used as the basis for the simulations.



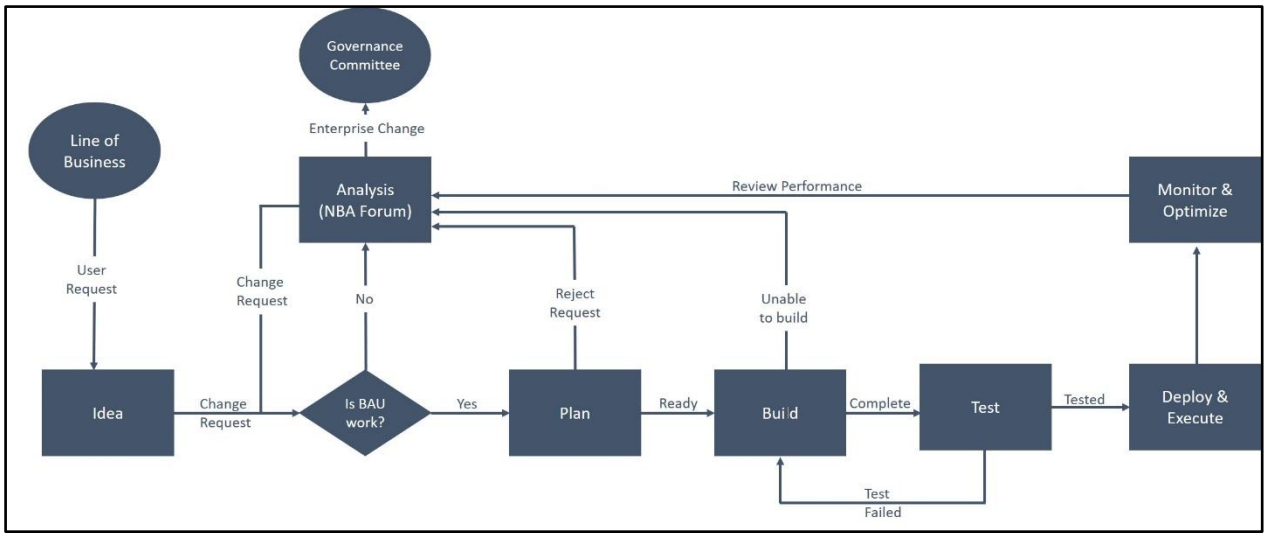
Let's see how each of these simulations can be used in a business-as-usual use case.

U+, a retail bank, has recently implemented a project in which credit card offers are presented to qualified customers when they log in to the web self-service portal. Now U+ would like to leverage the simulation testing capabilities of Pega Customer Decision Hub to:

- Market credit card offers to qualified customers
- Verify if their engagement policy conditions are presenting offers to customers as expected
- Check the offer distribution and prioritization ranges
- Check if business policies and regulations are being violated
- Show improvements in projected value

Each of these simulation types plays a specific role in a business operating model. This is U+ bank's operating model. An operating model supports businesses in the planning, development, testing, monitoring, and optimization of changes. The simulation capability can be utilized throughout the lifecycle but has greater significance when run at certain points.

Click the hotspots to see in which phase of the lifecycle U+ bank can use each of these simulation types.



1. In the ideation and analysis phase, Scenario Planner and Value Finder help the business come up with new ideas and engagement policies to improve offer marketing.
2. During the build phase, Audience Simulation provides detailed information about the pass rates of the engagement policies. It also helps you understand if you need to tighten or loosen any criteria during the build stage. In this way, audience simulation is similar to unit testing.
3. During the testing phase, the Distribution Test, Ethical Bias, and Scenario Planner simulations help you in performing system or acceptance testing. These simulations give you an idea of the distribution of offers across a sample customer base, help identify bias, and look for what is in target and what is not.
4. After deploying, you can use Scenario Planner and Value finder to look for more ideas and opportunities.

Pega Value Finder

Pega Value Finder

A value finder simulation allows you to engage more empathetically by identifying and profiling “under-served” customers, then suggesting actions for improvement – like adjusting engagement policies or creating new actions and treatments.

Using Pega Value Finder, you can discover areas in which you can improve the next-best-action strategy by monitoring scenarios in which customers are presented with no actions or only low-propensity actions. This is particularly useful when planning new changes or optimizing existing parameters.

Value Finder identifies and profiles “under-served” customers. These are customers that either do not receive actions, or only receive actions they have a low propensity to accept. It analyzes what happens at every stage of the next-best-action decision funnel, enabling you to:

- View offer distribution to well-engaged, under-engaged, or not-engaged customers
- Identify top opportunities for improvement
- Review details of under-served groups at each level of arbitration

Consider an example in which a bank runs a value finder simulation and identifies groups of not-engaged and under-engaged customers. The details of the groups and how to address the findings are as follows:

Not-engaged customers: There are 7000 under-served customers, none of whom own a credit card. They all have credit scores over 650 but are blocked because the eligibility rules in place keep them from seeing specific offers they have a high propensity for. In this case, you might want to tweak the engagement policy to present them with more appropriate offers.

Under-engaged customers: There are 5000 customers that have good credit scores and own a credit card but have no propensity scores higher than 5%. You might need to create a new offer with different terms and test it to see if you can capture their attention.

Analyzing customer distribution using a value finder simulation

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

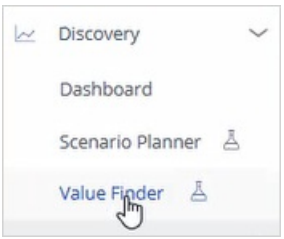
Transcript

This demo will show you how to analyze customer distribution using a value finder simulation.

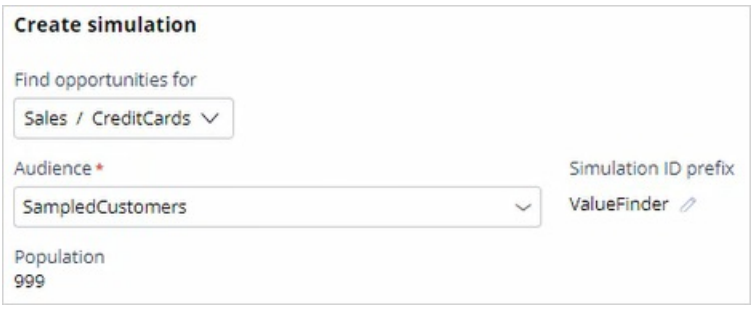
U+, a retail bank, has recently implemented a project in which credit card offers are presented to qualified customers when they log in to the self-service web portal. The bank would like to check if there are any un-served or under-served customers and find ways to serve them better.

This is the Pega Customer Decision Hub™ portal.

In **Discovery**, you can create a new value finder simulation.

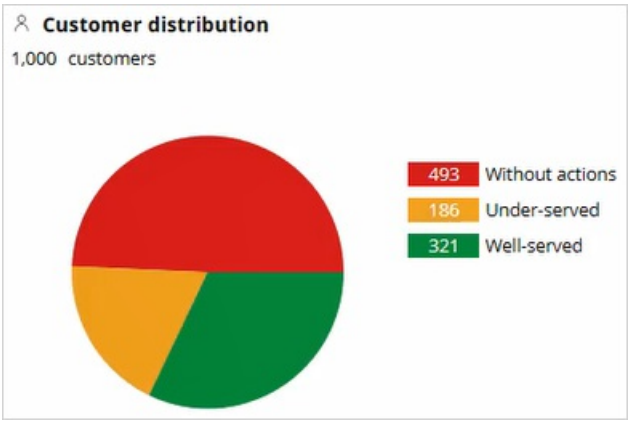


To create a value finder simulation run, select the issue and group in which you would like to find opportunities. Thenselect an audience on which you would like to do the simulation run. The audience is a list of potential target customers. You can modify the simulation name as required to easily identify the specific runs.



Once the simulation run is complete, Value Finder indicates several opportunities.

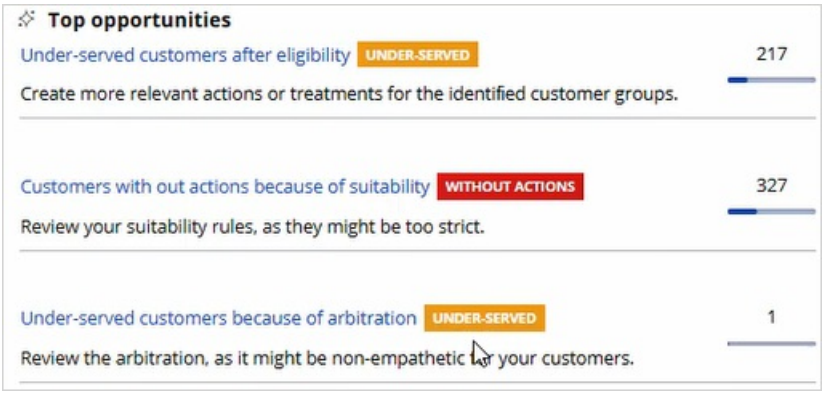
The pie chart displays the cumulative numbers of customers without actions, under-served customers, and well-served customers. The **Without actions** category indicates the number of customers who received no actions. The**Under-served** category indicates the number of customers who received low propensity actions. The **Well-served** category indicates the customers who received high propensity actions.



Value Finder identifies a number of opportunities for improving the Next-Best-Action Strategy. However, it displays the top three opportunities only.

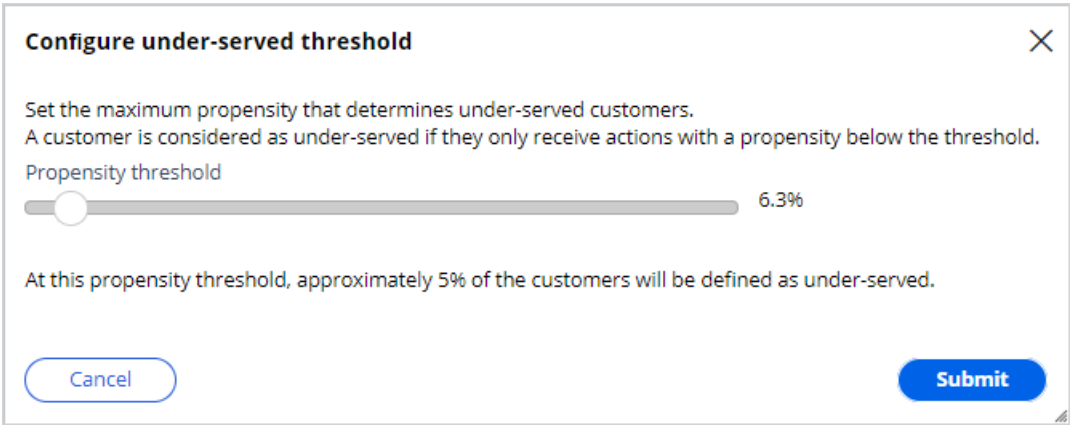
The first opportunity represents a group of customers that are under-served due to eligibility conditions. The second opportunity indicates the stage of your Next-Best-Action strategy at which customers are prevented from receiving actions. In this example, it is after suitability. The third opportunity shows the number of customers who are under-served due to low propensity actions.

In this case, notice that there are 217 customers who are under-served because they have no relevant actions or treatments, and 327 customers who have no actions because the suitability condition may be too strict.



Value Finder identifies customers as under-served if the propensity of every action and treatment available to the customer is below the threshold propensity. To provide a convenient starting point, the propensity threshold value is initially chosen in such a way that 1 out of 20 customers is defined as under-served. In this case, this results in an under-served threshold of 6.3%.

If the business would like to present offers with a higher propensity, this threshold value can be changed.



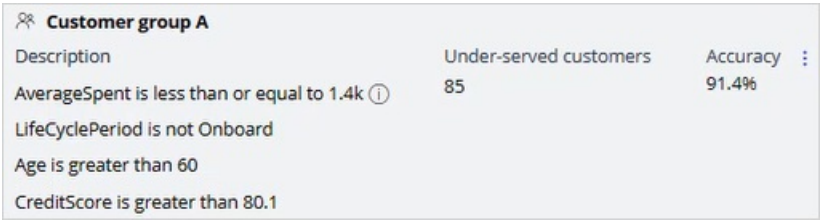
Details are provided for ‘Under-served groups after eligibility’ to show which eligibility conditions are contributing to the issue.

Value Finder provides the following information about each group:

Description: Group characteristics such as Age or LifeCycle Period.

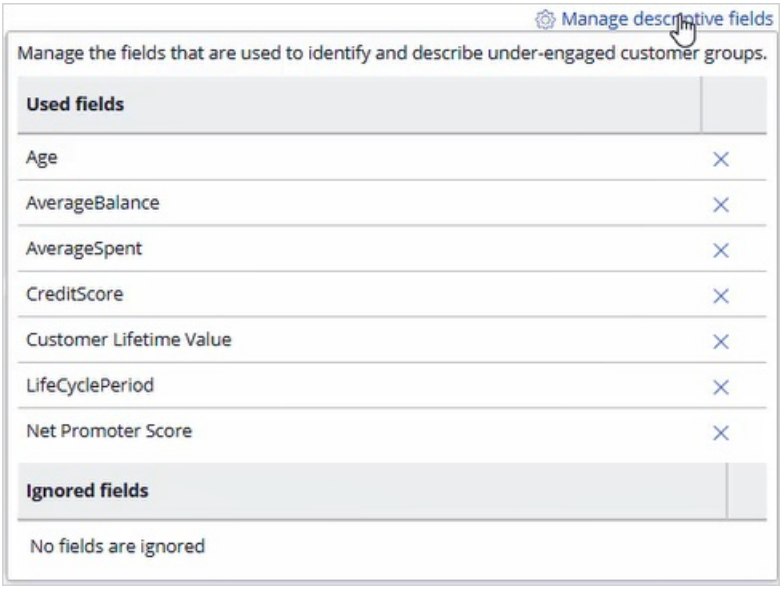
Under-served customers: The number of under-served customers in the group.

Accuracy: The number of under-served customers in the group divided by the total number of customers in the group. If the accuracy is 100%, this means that all customers in the group are under-served. If the accuracy is lower, for example, 91.4%, this means that 91.4% of the customers in the group are under-served. The remaining 8.6% have at least one action above the propensity threshold.

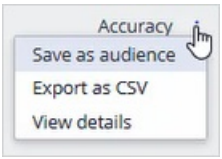


You can also manage the group by removing a field from the description. Typically, the system lists all potential fields available for the Next-Best-Action Strategy. You can decide to remove a field from the list if it’s not a required field. For example, if the business does not want to categorize customers based on age to avoid discrimination, they can remove the

Age field from the list. If a field is removed, Value Finder then recalculates the values for **Under-served customers** and **Accuracy**.



Also, you can save or export the customer groups identified as under-served as audiences.



You can then run distribution tests to get more insight into the current actions that these audiences receive. Use the Value Finder recommendations and your distribution test results as feedback for business stakeholders to inspire them to create new actions and treatments that will be relevant to these customers.

Let's now look at the **WITHOUT ACTIONS** customer category to understand how the customers are filtered.The filtration that happens in this simulation is similar to a funnel filtration for every engagement policy condition type. In this case, the total number of customers in the audience is 1000. Thus, the input population considered for the eligibility conditions is 1000. When the eligibility conditions are applied, 11 customers do not receive any actions. Thus, 1000-11 = 989 customers who pass through the eligibility level. The output population of the eligibility level is passed on as the input to the next level.

The input population for the applicability conditions is therefore 989. When the applicability condition is applied, 155 customers do not receive an action. Thus, 989-155 = 834 customers who pass through the applicability level.

The input population for the suitability conditions is therefore 834. When the suitability condition is applied, 327 customers do not receive an action.

In this specific scenario, 327 customers are presented with no actions due to the suitability condition. When you click on **Suitability** under the **WITHOUT ACTIONS** customer category, you can view the suggested recommendation for providing more customers with actions. Use this information from the Value Finder to run a funnel filtration on an audience simulation to detect at which level the suitability condition is preventing customers from receiving an action. That is, whether it is the suitability condition at the group level or the action level.

This demo has concluded. What did it show you?

- How to configure and run a value finder simulation.
- What are the top three opportunities identified by the Value Finder.
- How to interpret the customer group in the under-served customer category.
- How to interpret the without actions customer category.
- How to interpret the under-served customer category.

Audience simulation

Introduction

You can improve the performance of your next-best-action strategy by testing the configuration of engagement policies against a set of customers. In this way, you can check how many potential actions are filtered out by each component of the policy and discover if a particular criterion is too broad or too narrow for your requirements.

Video

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

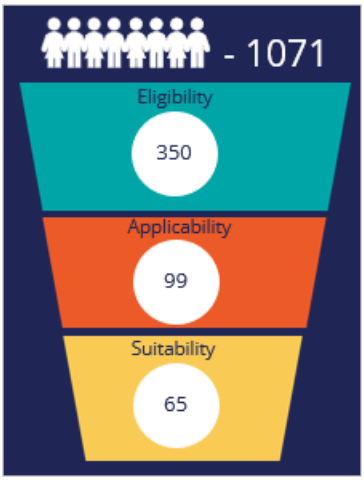
Transcript

This video explains what an audience simulation is and how it improves the performance of your Next-Best-Action strategy.

Audience simulation tests the configuration of engagement policies against a set of customers.

Consider an example in which an audience comprises 1071 customers. Let's see how the audience will be filtered based on the engagement policies.

When the eligibility conditions are applied, only 350 out of 1071 pass through. These 350 customers become the input for the next stage, applicability conditions. Only 99 of the 350 pass through the second stage. In the final stage, suitability conditions, only 65 pass through. Of the original 350, only 65 customers qualify to receive at least one offer.

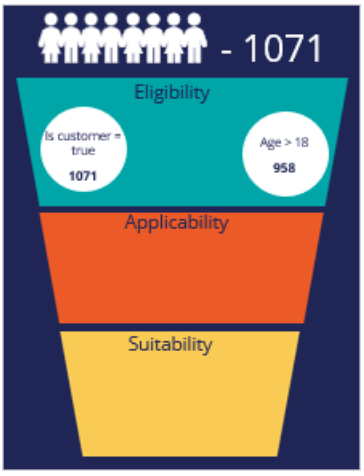


The filtering that happens in this simulation is similar to a funnel filtration for every engagement policy condition type. However, within an engagement policy condition type, each of the criterion is applied separately to the corresponding input audience.

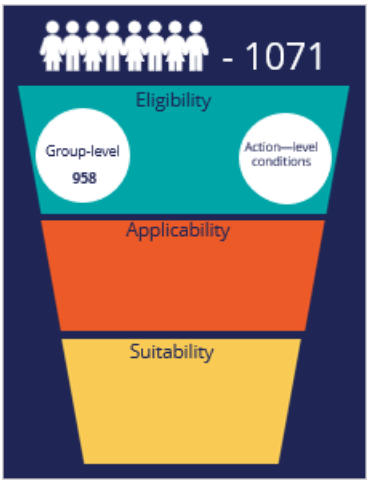
Let's now look at the same example in detail to see how the funnel filtration happens within each engagement policy type.

The engagement policies can be defined for a specific group within an issue and/or for individual actions. Group-level conditions are applicable to all the actions within that group while the action-level conditions are applied to that specific action only.

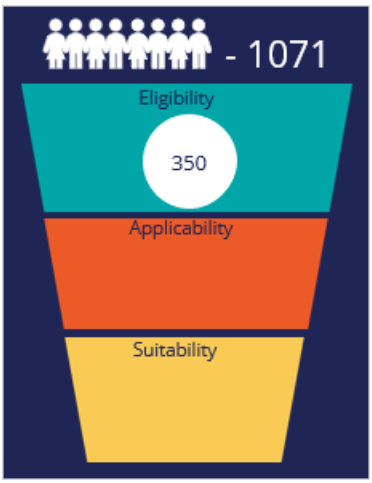
In this case, the total number of the customers in the audience is 1071. Within the eligibility criteria, the first condition does not filter any customers, as all qualify. When the second condition is applied, 958 customers pass through.



Since both conditions need to be met, the intersection of these two conditions, 958, is the final number of customers who pass through at the group level. Now the system checks for individual action-level eligibility conditions that also need to be met.

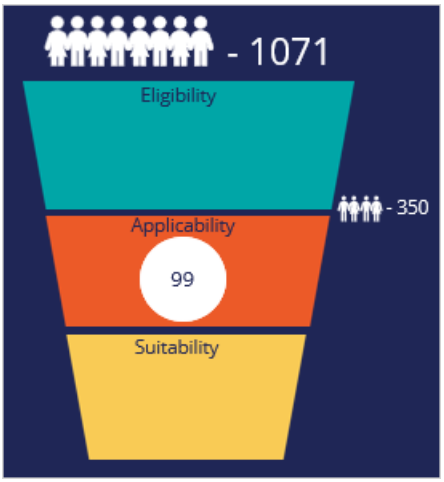


The intersection of the group-level and action-level conditions form the final eligibility-level output population of 350.



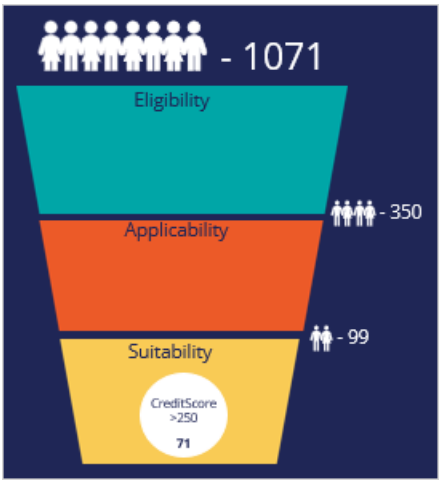
The output population of the eligibility level is passed on as the input for the next level.

The input population that will be filtered by the applicability conditions is therefore 350. When the applicability condition is applied, 99 of the 350 customers qualify for at least one offer. In this case, let’s assume there are no action-level conditions. Thus, the result from the individual group-level condition is the same as the overall applicability level, 99.

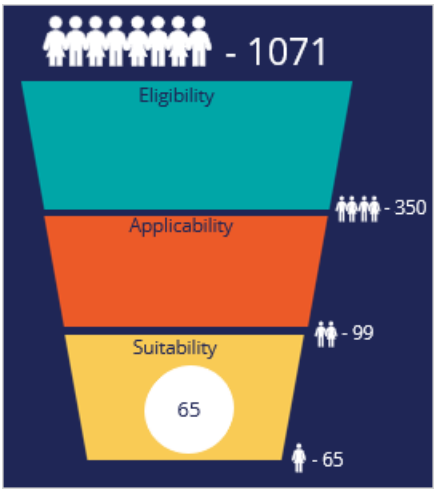


The output population of the applicability level, 99 customers, is passed on as the input to the suitability level.

When the suitability group-level condition is applied, 71 of the 99 customers qualify for the offers.



Now the system checks for individual action-level suitability conditions, and the intersection of both group-level and action-level conditions forms the final suitability-level output population, which is 65.



In summary, with the audience simulation test you can check how many potential actions are filtered out by each component of the engagement policy and discover if a particular criterion is too broad or too narrow for your requirements.

Running an audience simulation test

You can improve the performance of your next-best-action strategy by testing the engagement policy configuration against a set of customers. In this way, you can check how many potential actions are filtered out by each component of the policy and discover how a criterion might affect your requirements.

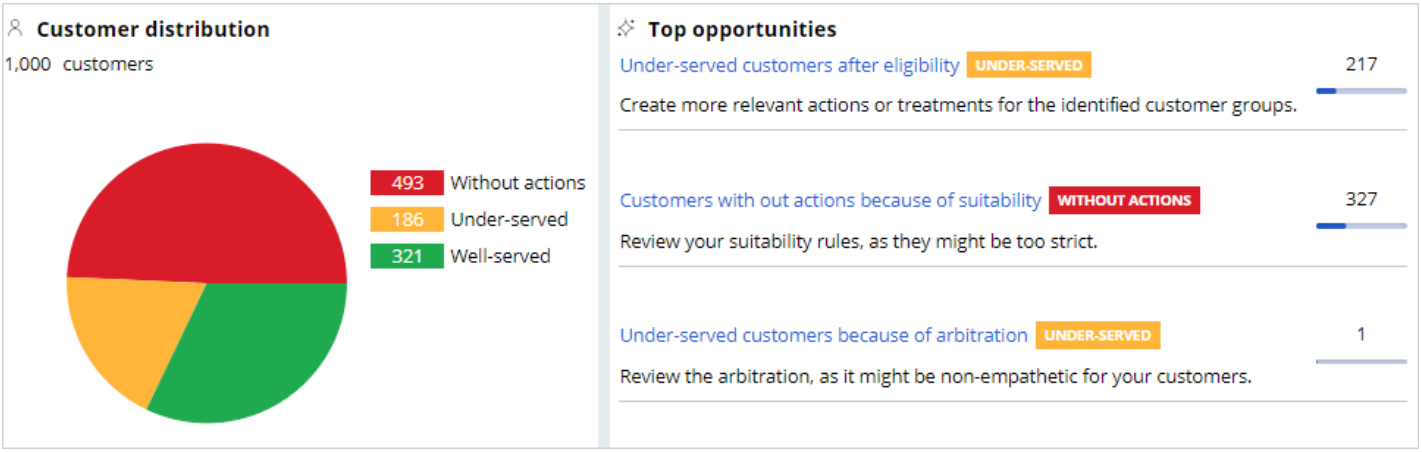
Audience simulation

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This demo will show you how to run an audience simulation test. It will also explain how many potential actions are filtered out by each section of the engagement policy and reveal if a particular criterion is too broad or too narrow for your requirements.

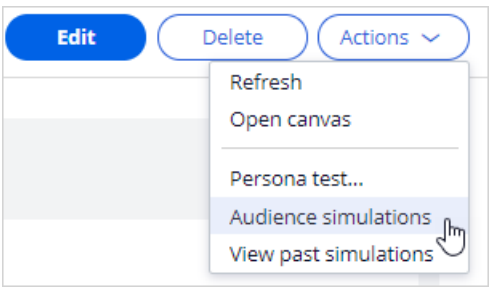
U+, a retail bank, has recently implemented a project in which credit card offers are presented to qualified customers when they log in to the web self-service portal. The bank ran a value finder simulation to find under-engaged customers. In that simulation run, a set of customers is identified with no actions after engagement policy conditions are applied.

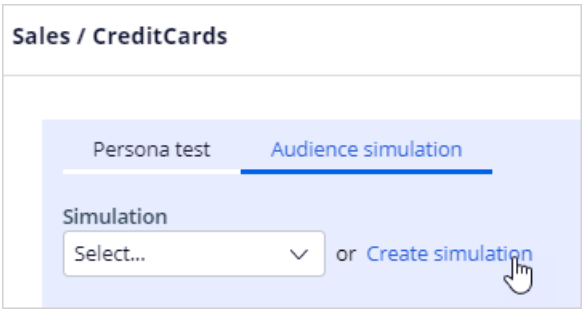


The bank would now like to run an audience simulation to investigate why these customers have no actions presented to them, and correct the condition that causes the deviation.

This is the Pega Customer Decision Hub™ portal. First, take a look at the set of eligibility criteria that has already been configured by U+.

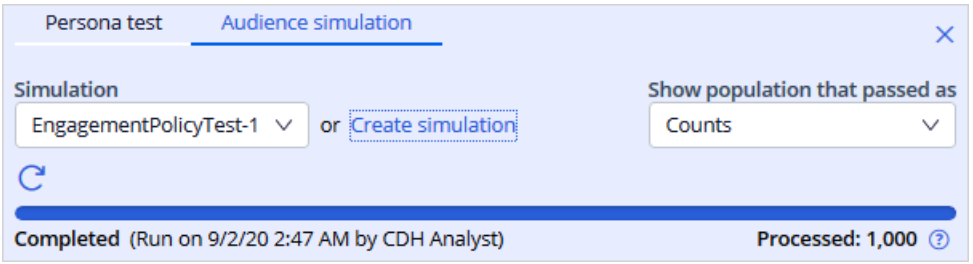
In **Audience simulations,** you can create a new simulation run.





To create an audience simulation run, select an audience on which you would like to do the simulation.The audience is a list of potential target customers. You can modify the simulation name as required to easily identify the specific runs. Then, choose the scope of the run. That is, you can choose to simulate only the engagement policies to validate the eligibility, applicability and suitability conditions. Or, you can simulate on both engagement policies and arbitration to understand how the conditions work when arbitration across all actions is also considered.

Once the simulation run is complete, you can view the details of how the audience is filtered at the group level based on the configured engagement policy conditions.



For each component of the engagement policy, the simulation test shows a numerical or percentage value of the audience that will receive the action based on current criteria. For example, if the result of a criterion that checks if the action is active returns 100%, the component did not filter out any audience members.

The filtering process that happens in this simulation is similar to a funnel filtration for every engagement policy condition type. However, within an engagement policy condition type, each of the criterion is applied separately to the corresponding input audience.

In this case, the total number of customers in the audience is 1000. For this audience, the number of customers who received offers is 182. This simulation run also provides details of the audience filtration that happens with each engagement policy condition. When the eligibility condition is applied, 979 customers qualify for the offers. Within the eligibility criteria, the first condition does not filter any customers, as all qualify. When the second condition is applied, only 979 customers pass through. Thus, the intersection of these two conditions, 979, is the final number of customers who pass through to the eligibility level.

In this case, there are no eligibility conditions defined at the action level. If there are, the final result is the result of the

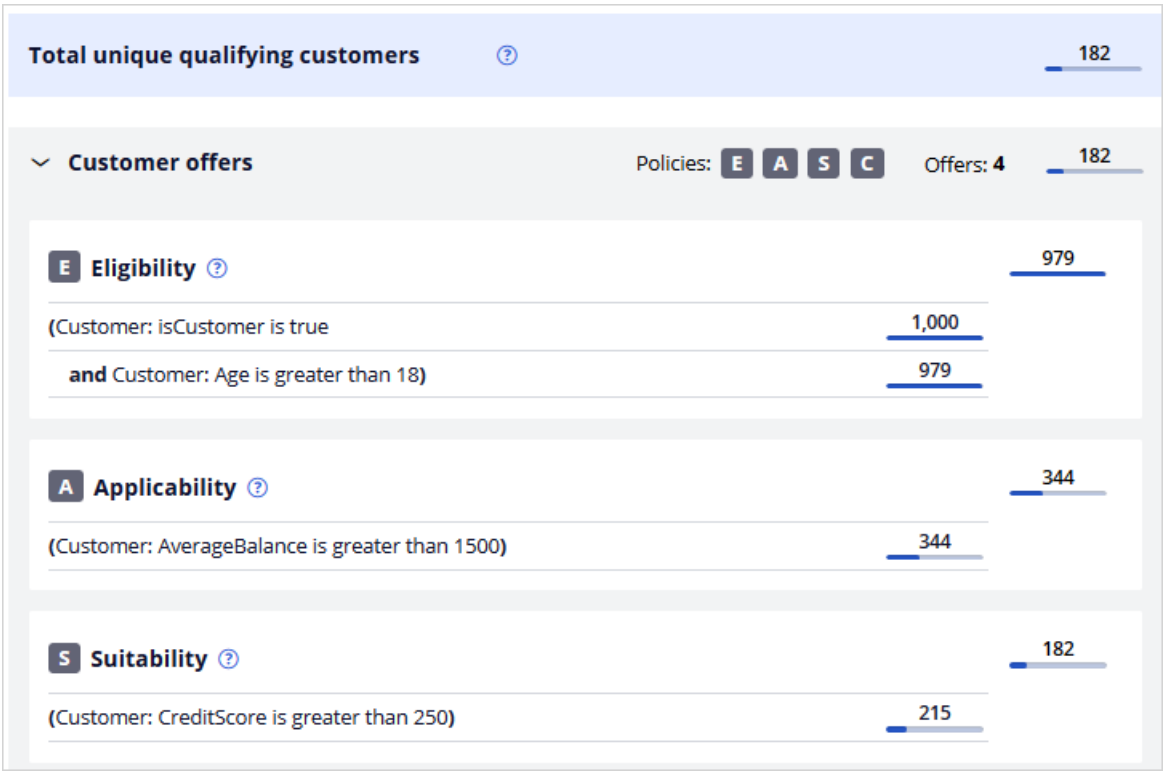
eligibility conditions at the group level, combined with the results at the action level.

The output population of the eligibility level is passed on as the input to the next level.

The input population considered for the applicability condition is therefore 979. When the applicability condition is applied, 344 customers qualify for at least one offer out of the 979 customers. Note, since there are no specific action-level applicability conditions, the result of the individual group-level condition is the same as that of the overall applicability component level, 344.

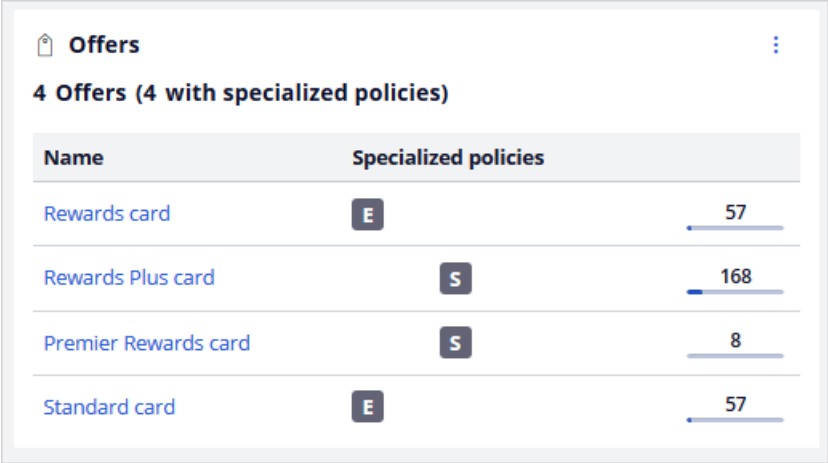
The output population of the applicability level, 344 customers, is passed on as the input to the suitability level.

When the Suitability group-level condition is applied, 215 customers qualify for the offers out of the 344 customers. If you check the engagement policies at the action level, notice that there are additional suitability conditions. When all of these conditions apply, the total number of customers who qualify for at least one offer is 182.

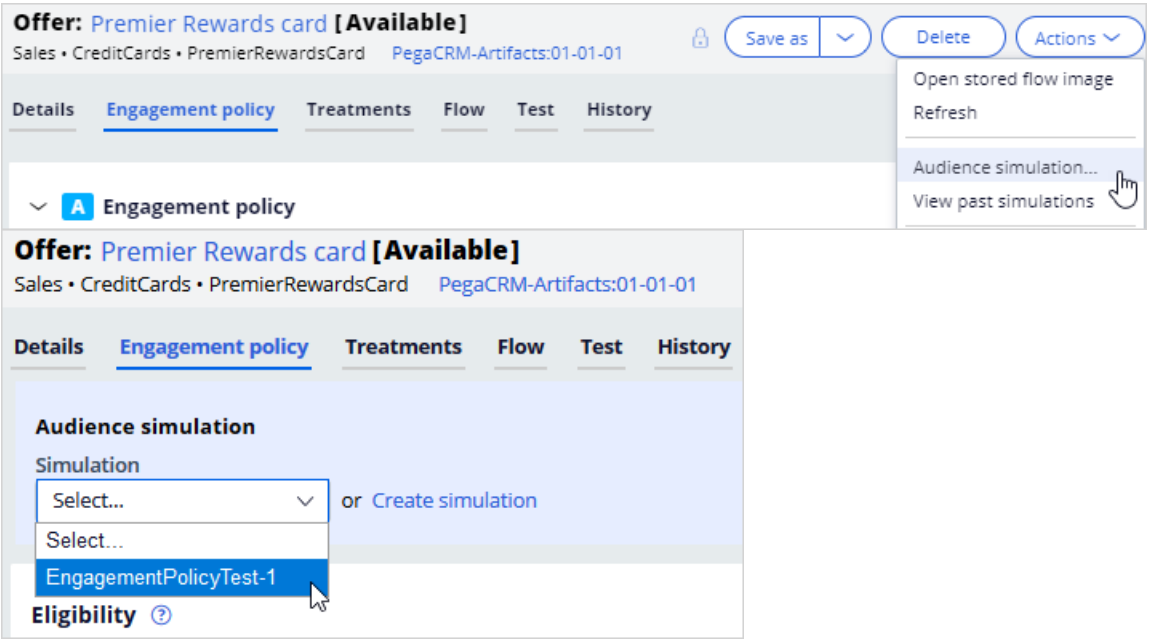


At the bottom, you can also view the number of customers who qualify for the offers. When only engagement policy conditions are considered in the simulation run, the number you see here is the number of customers who qualify for at least one action. Notice that the Premier Rewards card offer is rarely presented. Let's try to find out why this is so.

Open the Premier Rewards card to view the audience simulation filtering at the action level.



You can choose the same audience simulation to view the action-level filtration details and investigate which engagement policy condition is causing the current outcome.



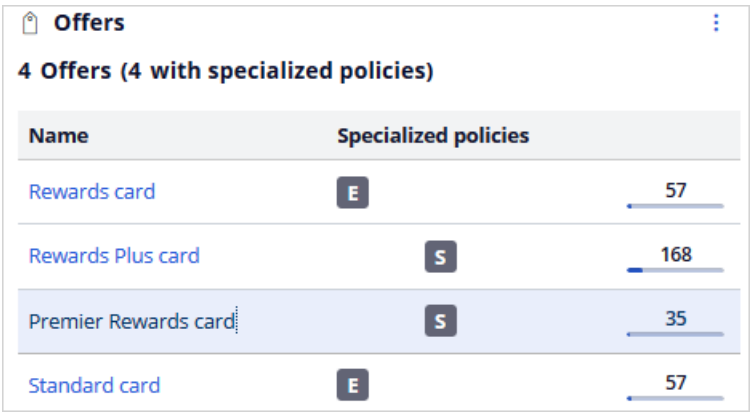
These numbers show how the engagement policies that are inherited from the CreditCards group are filtered. At the bottom, you can also view the final number of customers who qualify for the specific offers once the action-level engagement policy is applied.

Notice that at the suitability-level, only eight customers qualify for the Premier Rewards card offer.

This is because the configured engagement policy criterion is too narrow. The results of this simulation show us that from the 215 customers who passed the eligibility and applicability conditions, all qualify for the group-level condition, but only eight customers qualify for the action-level condition. As the action-level suitability condition is too narrow, a very small number of customers are presented with the offer. The bank then decides to reduce the CreditScore from 825 to 750 to ensure the

Premier Rewards card offer is also presented to the customers.

Now, modify the credit score condition to reflect the correct value. Then, rerun the same audience simulation at the group level. Notice that now, 35 customers are presented with the Premier Rewards card offer.



Let's now run an audience simulation with engagement policies and arbitration in scope.

Note that the simulation results are different. This is because the simulation includes arbitration, adaptive analytics, treatment and channel processing, and constraints. That is, this simulation result shows the number of customers who will receive an action as their top action. Thus, the numbers at the group level and the action level tally.

That is, the sum of all customers (12+118+22+30) is 182.

This demo has concluded. What did it show you?

- How to configure an audience simulation.

- How to view the simulationfiltering details at the group and action level.

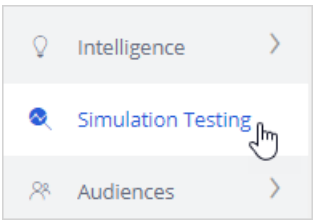
Decision funnel explanation simulation

A transcript for this video is not available. Please visit https://academy.pega.com to watch.

Transcript

This demo will show you how a decision funnel explanation simulation is auto-created for every audience simulation run at the Next-Best-Action Designer level.

Every time an audience simulation is run, in the background, a decision funnel explanation simulation is created. Now, navigate to the **Simulations** landing page to view the auto-created funnel simulation.



This is the auto-created funnel simulation. Alternatively, you can also create a simulation run from here and show the simulation results in the Engagement policies->Audience Simulation.

To view the funnel simulation, open the simulation. When a funnel simulation is created, a set of reports is added to the run results.

Assigned reports

Output	Report category	Report
ExplainDetails	Simulations	Top level view
ExplainDetails	Simulations	Champion Challenger - Drill down view
ExplainDetails	Simulations	Prioritization - Drill down view
ExplainDetails	Simulations	Switch - Drill down view
ExplainDetails	Simulations	Proposition filter - Drill down view

Notice that the **NBA_Sales_CreditCards** strategy is used to run the simulation. This is because you selected 'Engagement only' in the audience simulation, without the arbitration.

Setup details

Run simulation on revision version
—

Strategy
NBA_Sales_CreditCards

Context
Data-Decision-Request-Customer

Results in
CRM-SR-Sales-CreditCards

Audience
Data set

Simulation ID
EngagementPolicyTest-1

Rule name
SampledCustomers

Purpose
Explain

The simulation reports provide detail about the filtering that occurred. You can download these reports offline to analyze the outcome.

Top level view Edit report Actions

Filtered by:
(Report Sub Type = toplevel or (Report Sub Type = Overview and Rule Object Class = Rule-Decision-Strategy)) and Any Subject Type and Any Name

Is	Group	Name	Strategy	Component	Type	#	#	P...
SalesCreditCardsPremierRewardsCard	NBA_AllIssues_AllGroups_Customer	Results			Output	10001,000		100.00
SalesCreditCardsPremierRewardsCard	NBA_Sales_CreditCards			Customer Actions	Filter	10001,000		100.00
SalesCreditCardsPremierRewardsCard	NBA_Sales_CreditCards			NBA_Sales_CreditCards_AllActions_Primary	Sub Strategy	10001,000		100.00
SalesCreditCardsPremierRewardsCard	NBA_Sales_CreditCards_AllActions		NBA_Sales_CreditCards_A_All		Proposition Filter	10001,000		100.00
SalesCreditCardsPremierRewardsCard	NBA_Sales_CreditCards_AllActions		NBA_Sales_CreditCards_E_All		Proposition Filter	10001,000		100.00

This demo has concluded. What did it show you?

- How to view the auto-generated decision funnel explanation simulation.

Running an audience simulation test -- Wed, 10/14/2020 - 05:39
To get the full experience of this content, please visit https://academy.pega.com

Working of the always-on brain in outbound

What do customers expect from business?

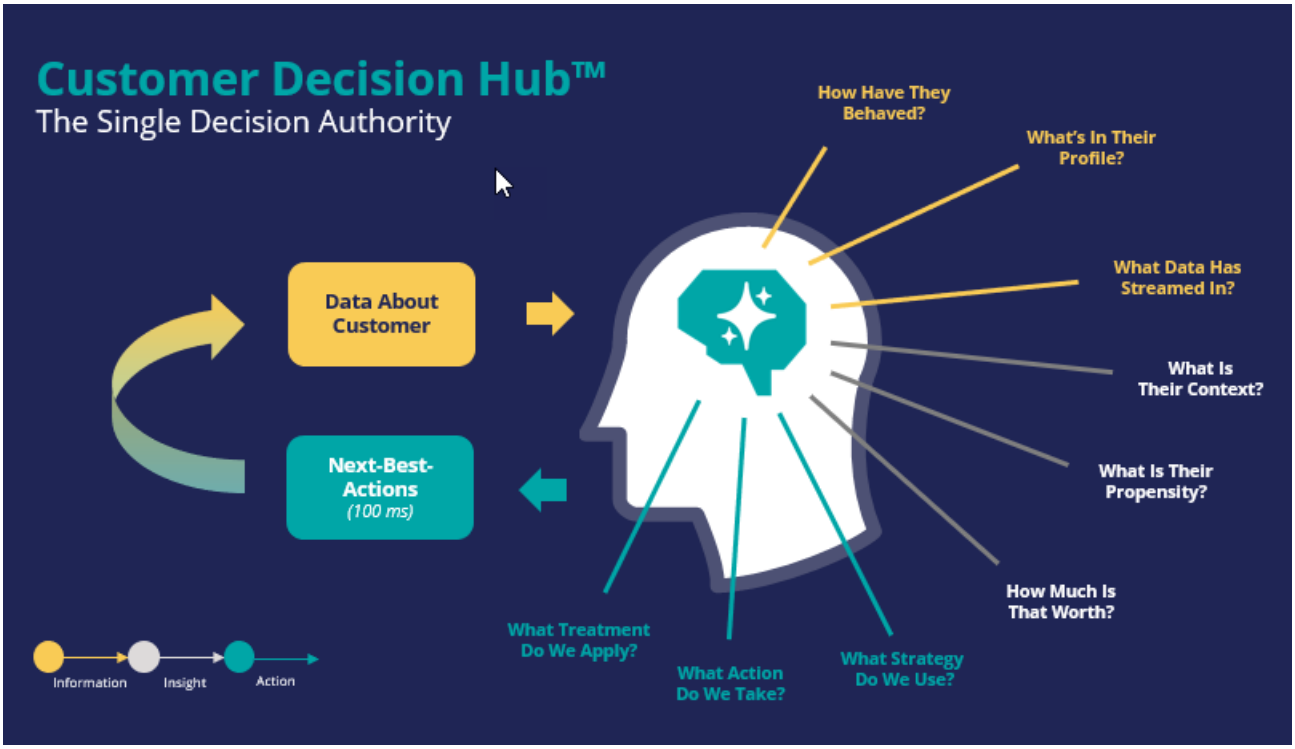
Customers are more empowered than ever before. As a result, they have very high expectations of the experiences they receive from their service providers. These experiences must be meaningful, consistent, and personalized across every channel they interact with.

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 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 behavior, 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

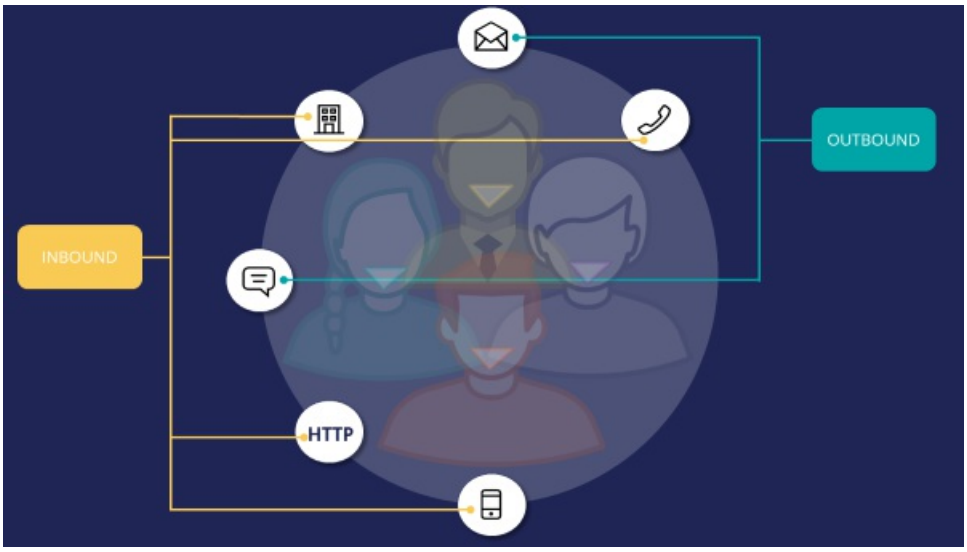
Customer engagement channels

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.



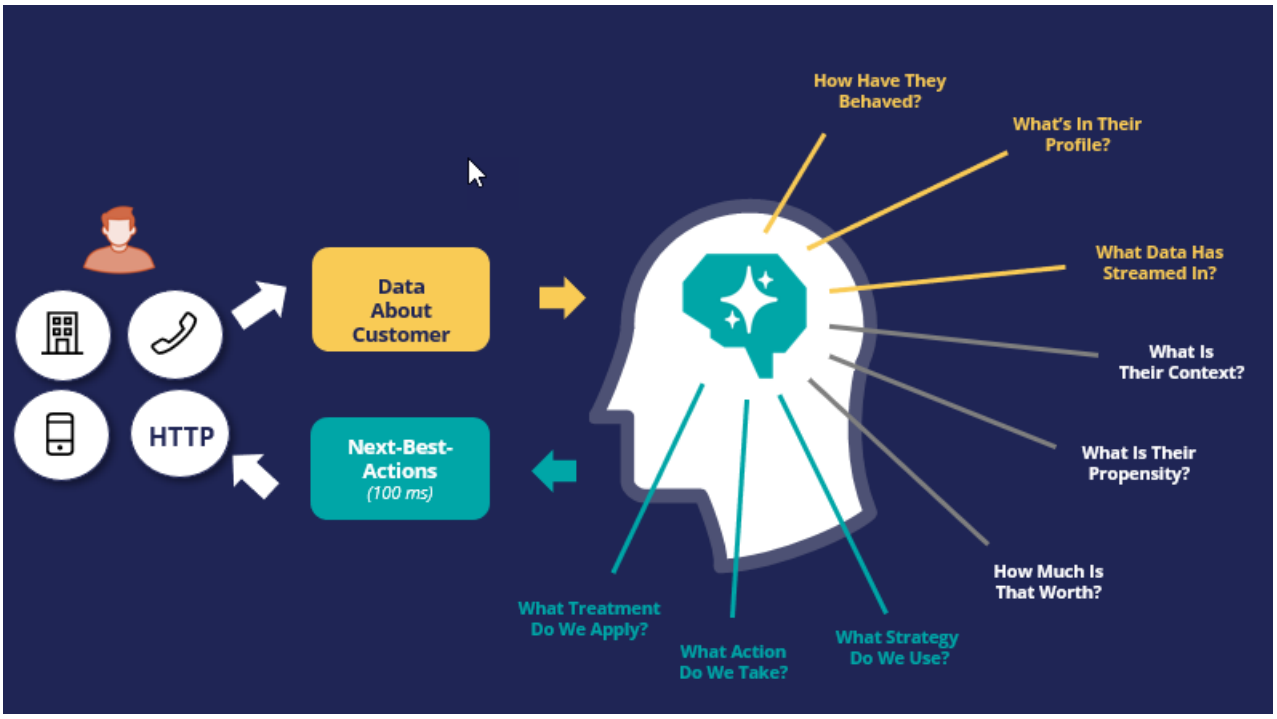
An inbound channel is used when a customer approaches the business, via, for example, a self-service web portal.

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.

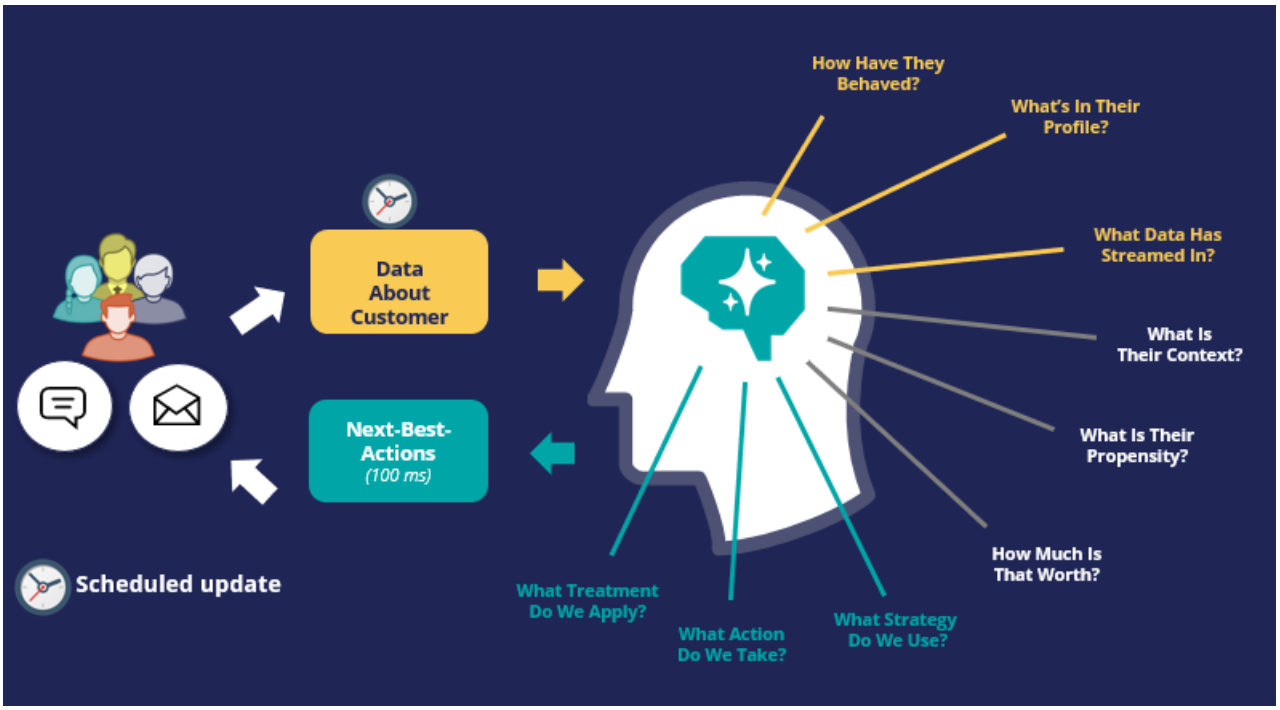


Outbound interactions

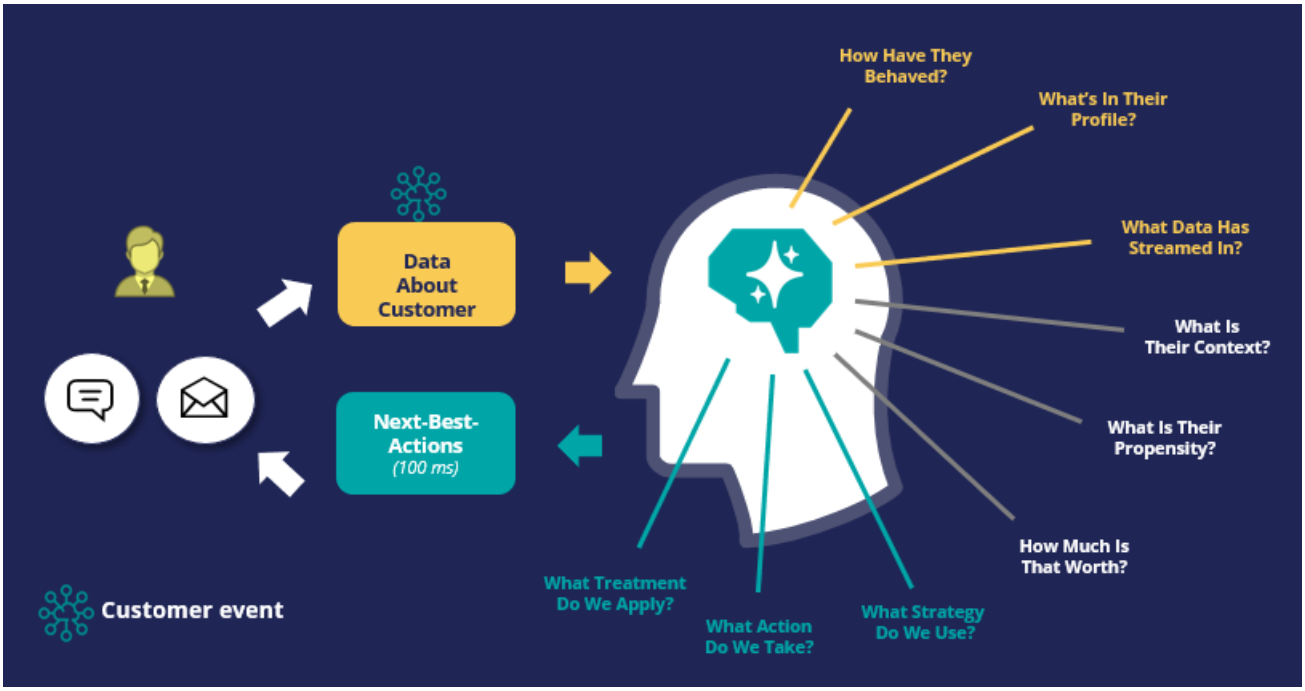
Always-on outbound means that the brain sends outbound messages, via email or SMS, to customers only when it's appropriate.

This can happen in three ways:

1) **Scheduled update**: 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 in the subsequent Next-Best-Action evaluation.



2) **Customer event**: When a customer event is detected that is of significance to the business, for example, a customer abruptly ends an online transaction, data is received via 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 outbound message is triggered only for the customer for whom the event occurred.



3) **Priority communication**: In extreme circumstances, you may want to contact all or a subset of your

customers outside the regularly scheduled update. In such cases, you would configure the brain to send out a high-priority communication to those customers. This would be a one-off communication and won't be repeated automatically by the system.

Working of the always-on brain in outbound -- Thu, 07/23/2020 - 01:04
To get the full experience of this content, please visit https://academy.pega.com

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

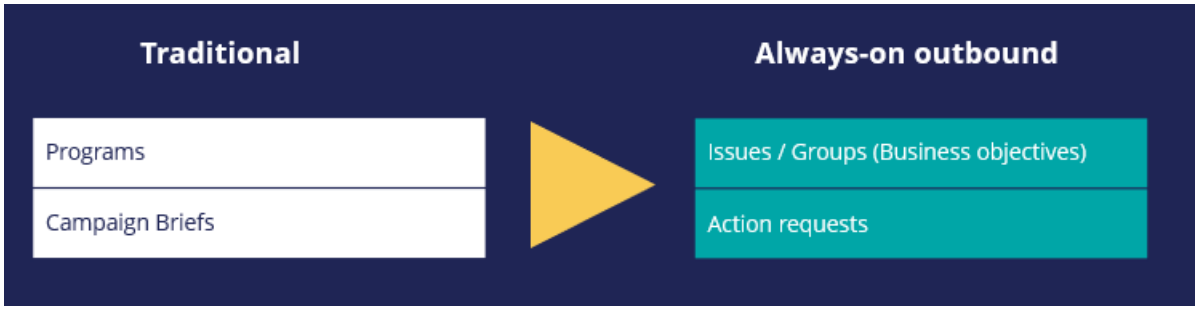
In traditional marketing, a marketing program is a set of activities designed to meet marketing objectives by grouping campaigns into logical business outcomes.

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).

By understanding the purpose of traditional segments, you can translate them into eligibility, applicability, suitability and contact policy rules.

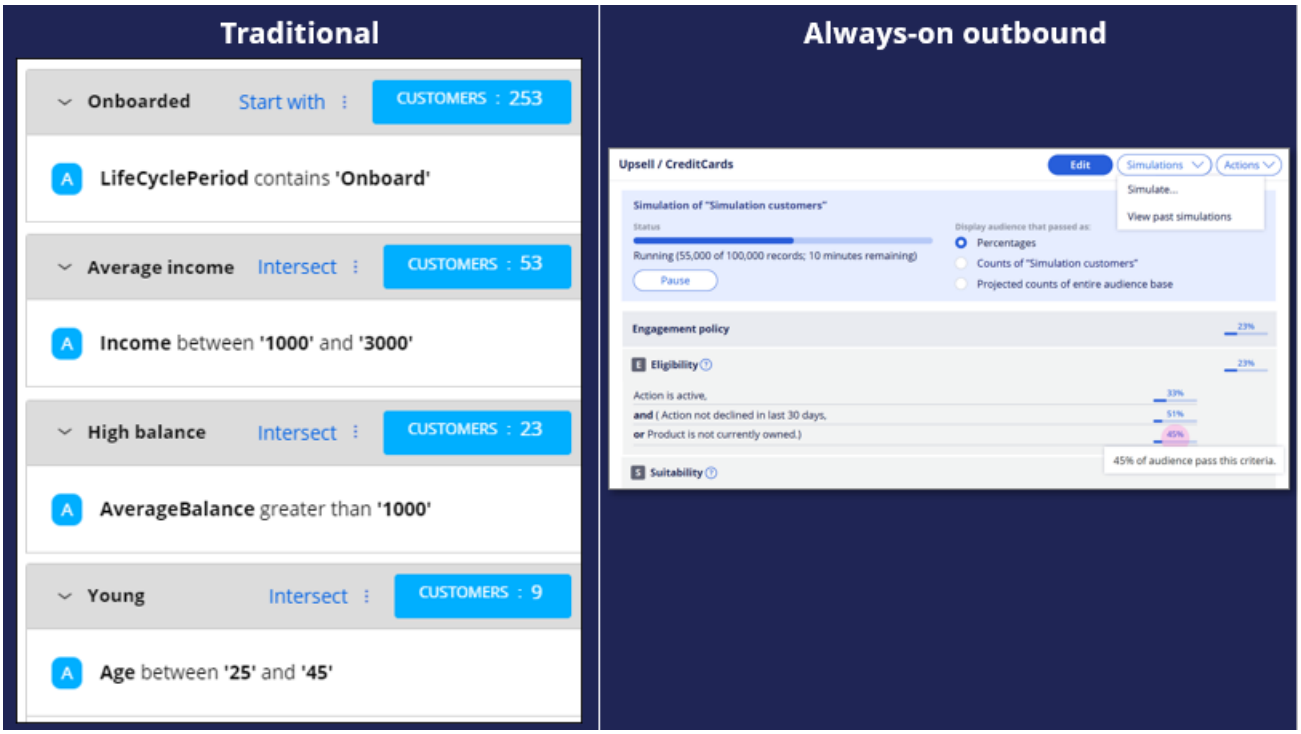
When selecting targets, any criteria that are not based on clear business rules should be avoided. For example, targeting a specific demographic such as an age range or income level based on market research or intuition is not necessary. In the always-on approach, it's the job of the AI to choose the 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 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. 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.



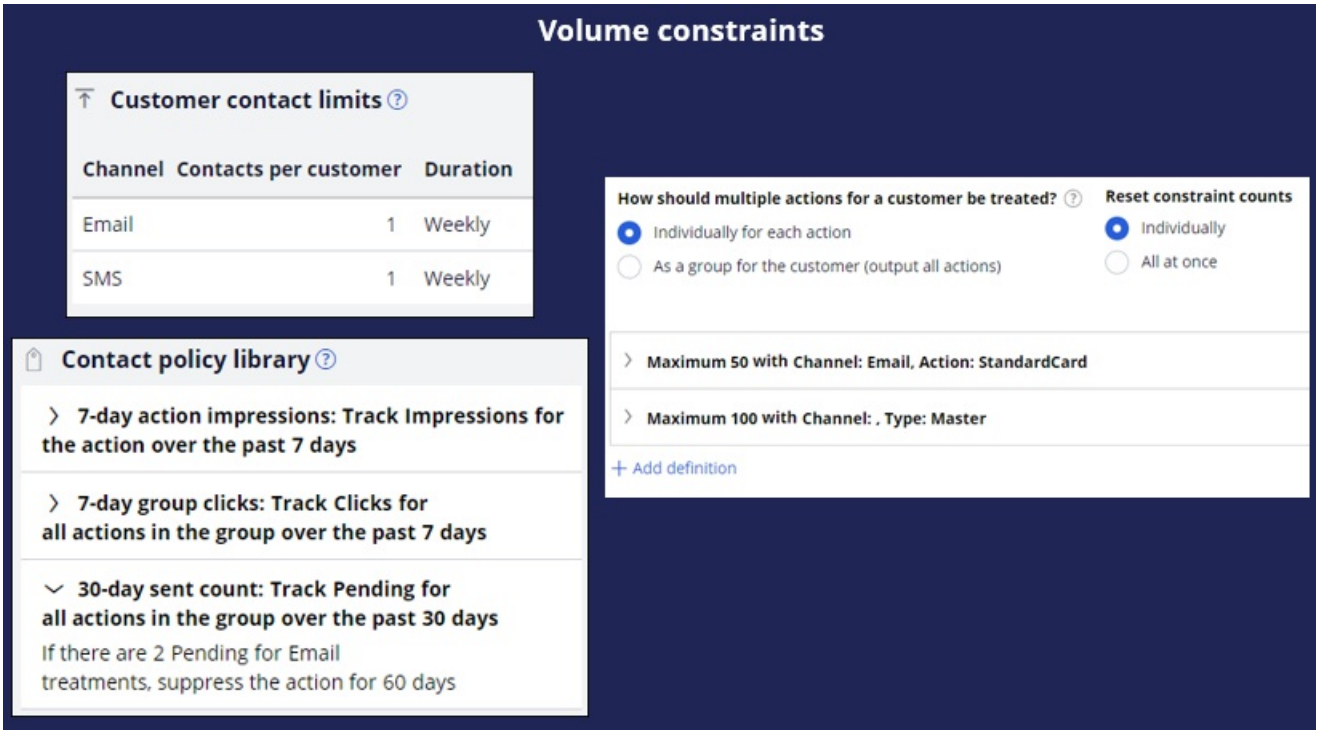
In the always-on approach, you use reports generated by the simulation facility in Next-Best-Action Designer to:

- 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.



Scheduling

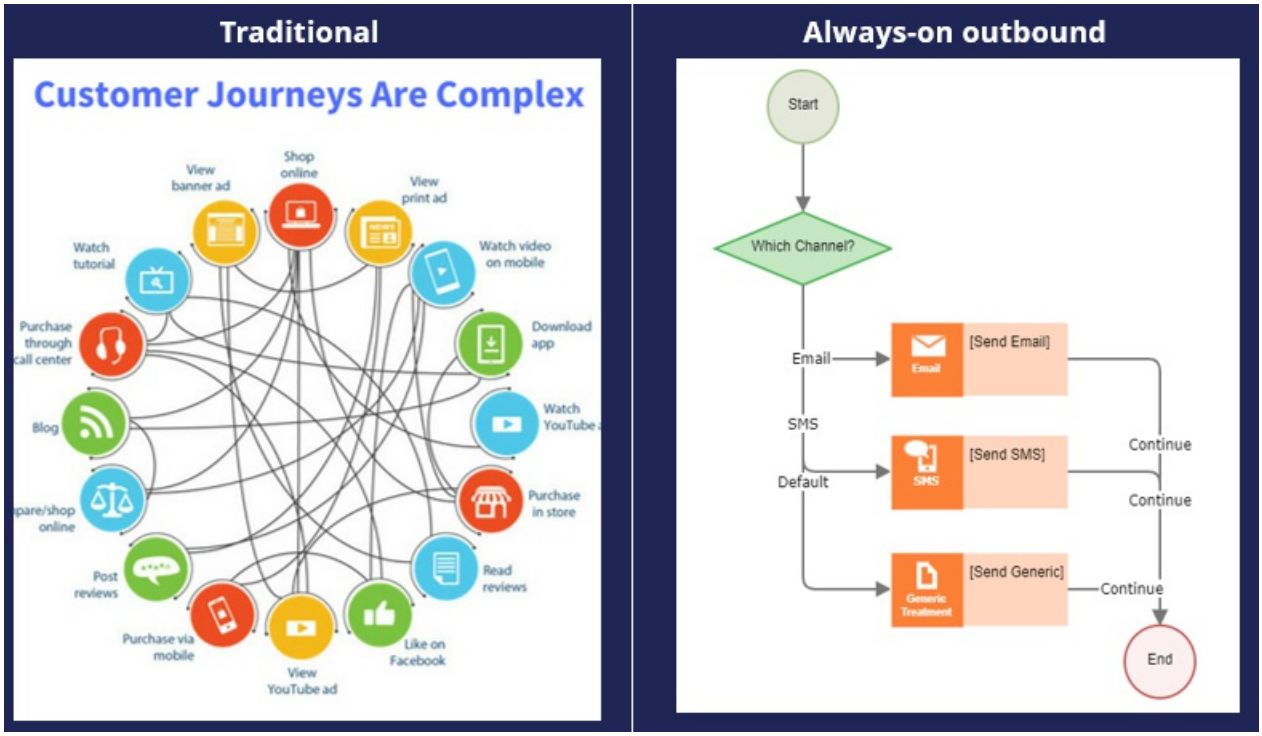
In a traditional marketing approach, there are several options to control the campaign schedule. For example, you can run a campaign as a one-off, or as recurring daily, weekly or monthly.

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

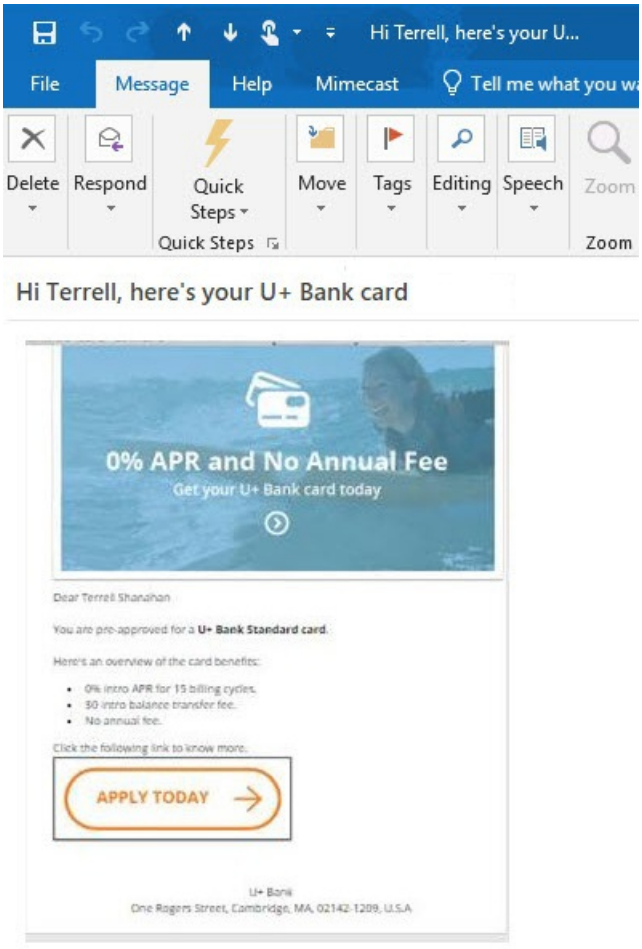
Introduction

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.

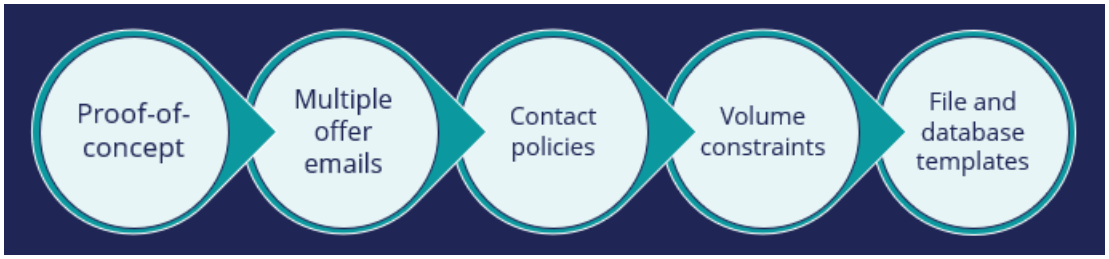
Satisfied with the results of the first microjourney, the bank now wants to extend these Next-Best-Action recommendations to be delivered via one of their outbound communication channels, email.

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 toapply 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.

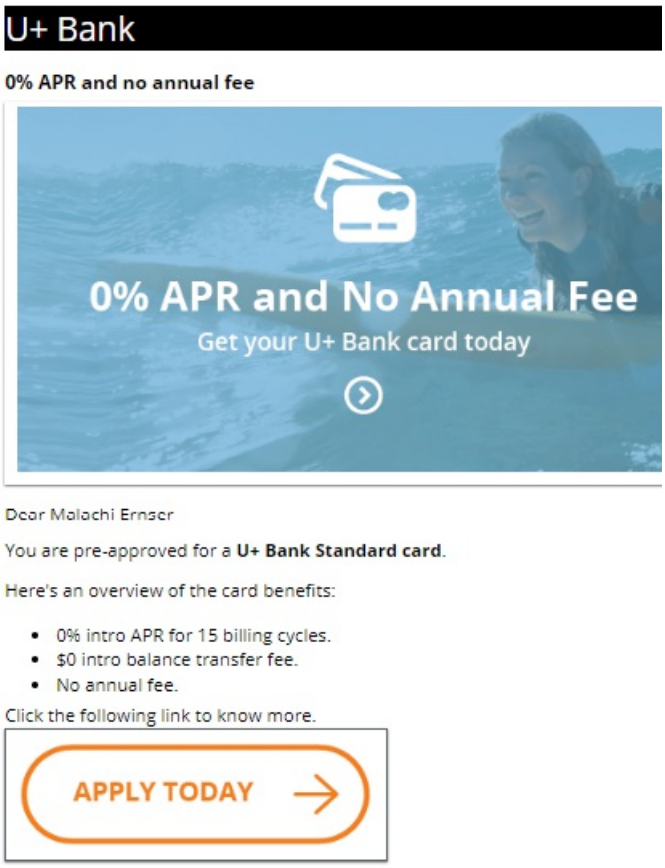
In summary, you will learn more about how U+ bank will leverage the always-on outbound capability in this phase of the project.

Sending offer emails

Business scenario

U+, a retail bank, would like to use Pega Customer Decision Hub™ to send emails related to a credit card offer to qualified customers.

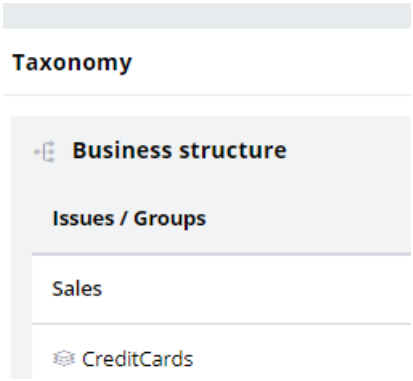
The email will contain the offer details and a call to action. When a customer clicks the**APPLY TODAY** call to action in the email, it will take them to the credit cards page on the bank's website where they can learn more about the offer and apply for it if interested.



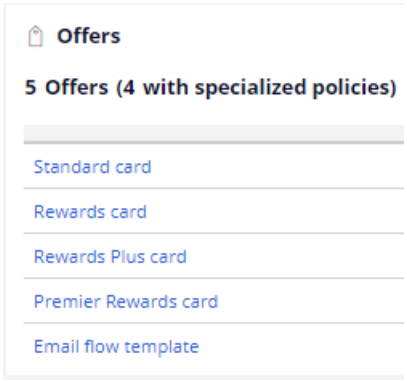
Next-Best-Action Designer review

To implement this business requirement, you need to configure Next-Best-Action Designer to perform always-on outbound customer engagement.

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/up-sell of credit cards, thus they have created one Issue called **Sales** and one Group under it called **CreditCards**.



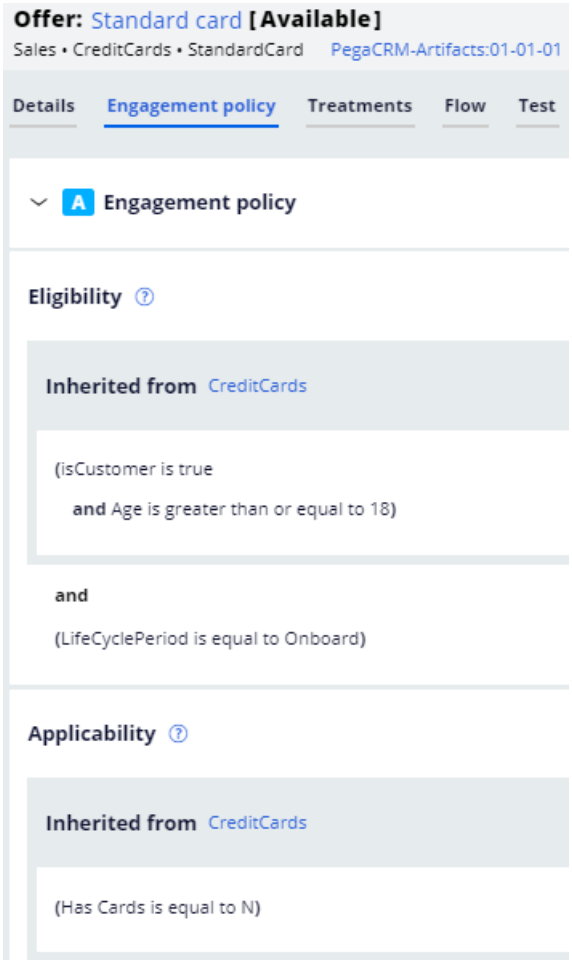
A consultant has already created a few actions under the Sales/CreditCards hierarchy.



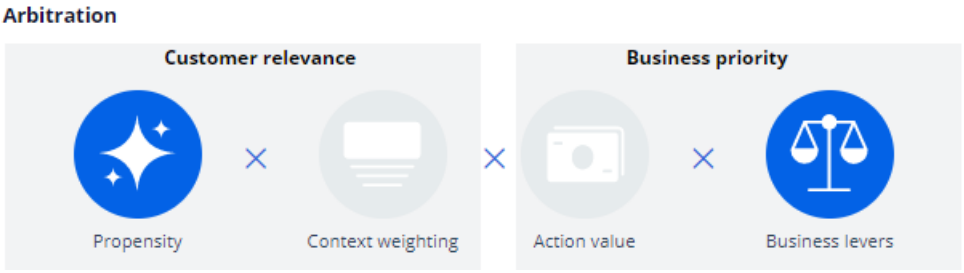
These actions correlate with the credit card offers that the bank wants to promote.

The engagement policy conditions determine which customers qualify for the actions.

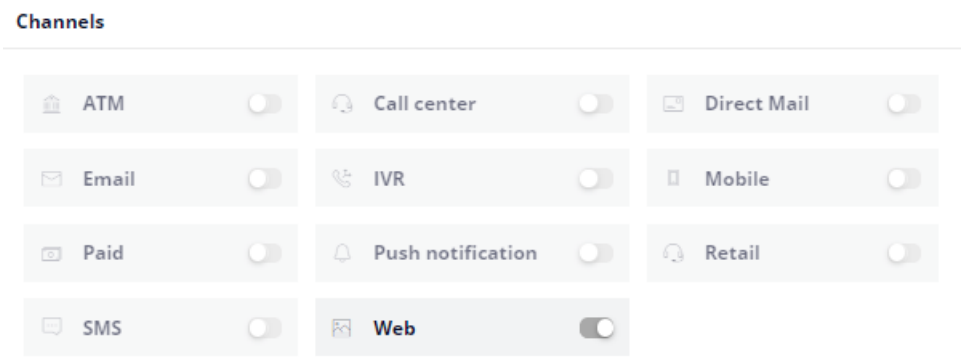
Some conditions apply to all actions under the group and some conditions apply to specific actions.



On the **Arbitration** tab, you define how the Actions should be prioritized when a customer qualifies for multiple Actions. Here you combine the predicted customer preference calculated by AI models with business objectives represented by **Action values** and **Business levers**. The Pega brain combines all four factors to select the best action.

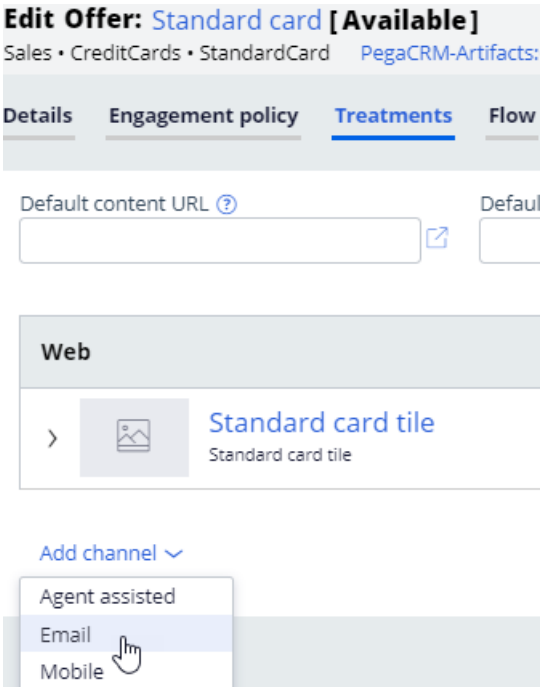


Currently, the bank promotes credit cards on the web channel only. They would like to extend this to the email channel.

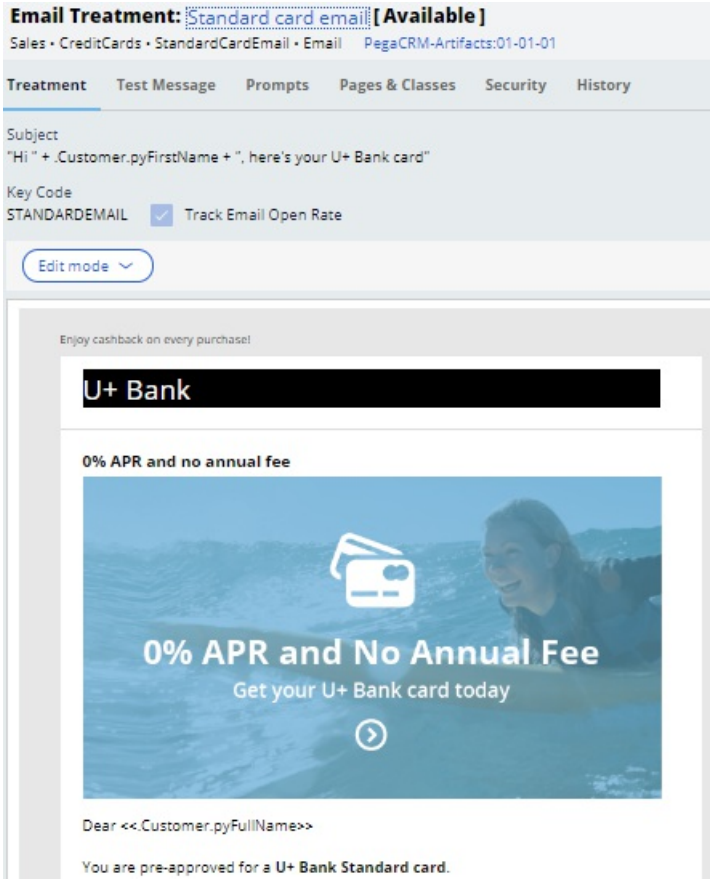


Email treatment

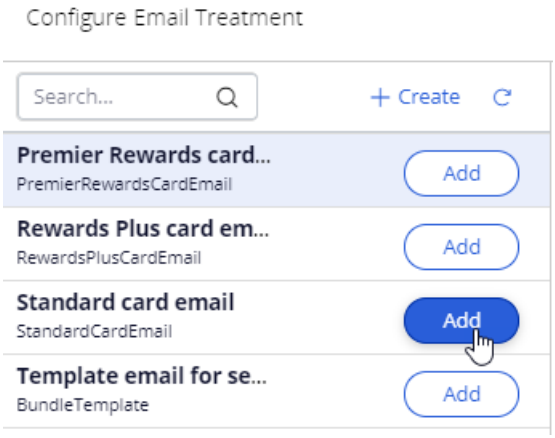
To send emails in an always-on outbound environment, you need to add an email treatment to the action.



An email treatment contains the message that you want to communicate to your customers on the email channel. It also contains the call to action, for example, a link or a button with the text *Apply now* or *Click here*. When a customer clicks on the call to action, the brain considers this a positive response.

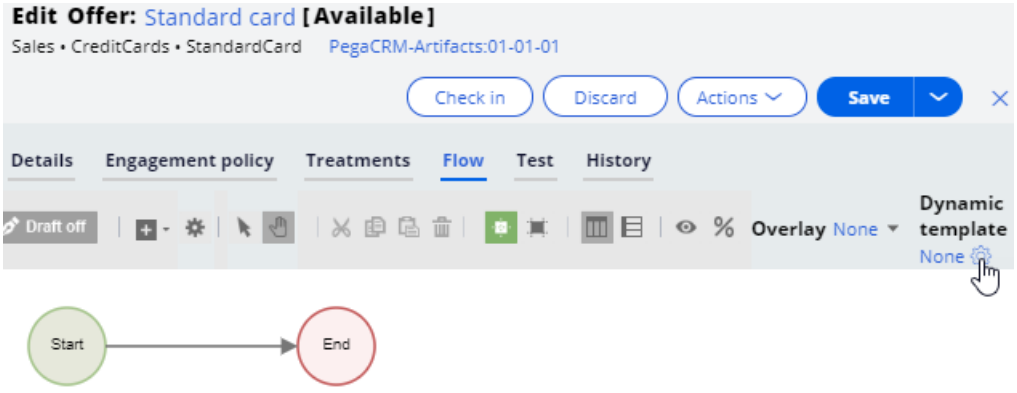


You can use an existing treatment or create a new one.

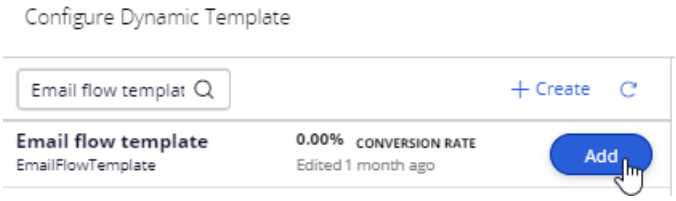


Action flow

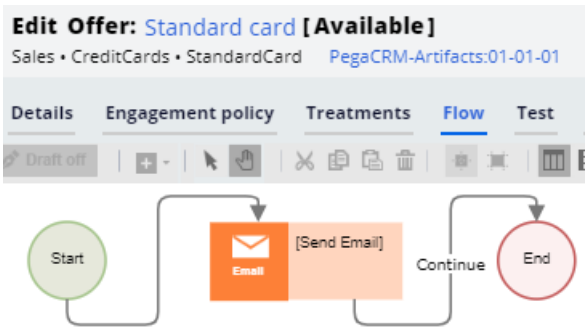
The action flow is a flow diagram that contains the sequence of steps carried out during the lifecycle of an action. The steps can deliver the action message via an outbound channel, wait for a response, update the Interaction History, send a follow-up message, and so on. A flow can consist of a variety of shapes that represent each of those steps. 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 brain decide the Next-Best-Action. Action flows can be created once and reused across Actions using the **Dynamic template** option.



A flow template is just another action with the flow configured in a generic pattern that is applicable to multiple actions.



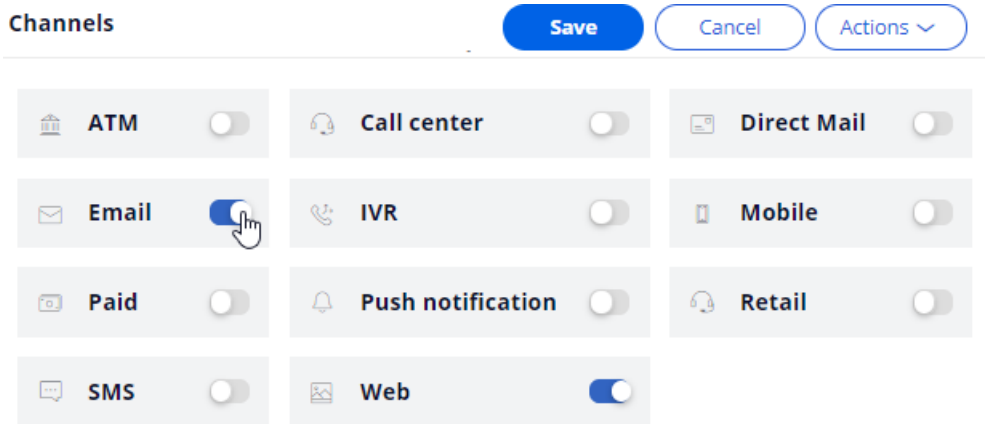
In the following flow there is only one key shape, the **Send Email** shape. It represents the outbound email delivery.



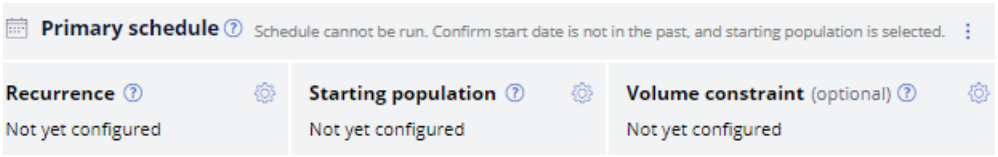
Next-Best-Action Designer outbound configurations

On the **Channels** tab, you can enable the channels and triggers that will invoke Next-Best-Action evaluation.

As U+ wants to send out emails, enable the **Email** channel.



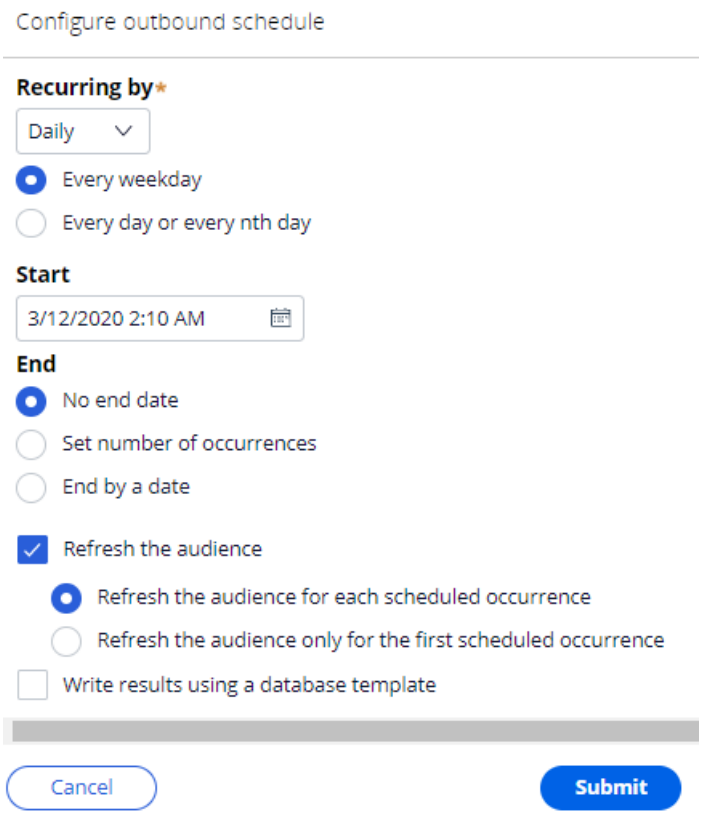
In the always-on outbound paradigm, the brain evaluates the Next-Best-Action for each of the customers identified in a segment, which contains the potential list of customers to whom you want to send messages. These messages can be sent at a regular frequency (for example, daily). You configure the **Primary schedule** to achieve this.



The schedule has two mandatory configurations, **Recurrence** and **Starting population**.

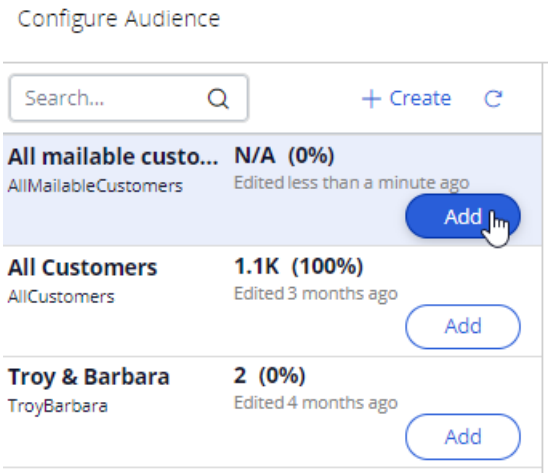
Recurrence

In **Recurrence**, you configure the outbound schedule. Enabling the **Refresh the audience** option ensures that the latest customer information is always used when running the schedule.



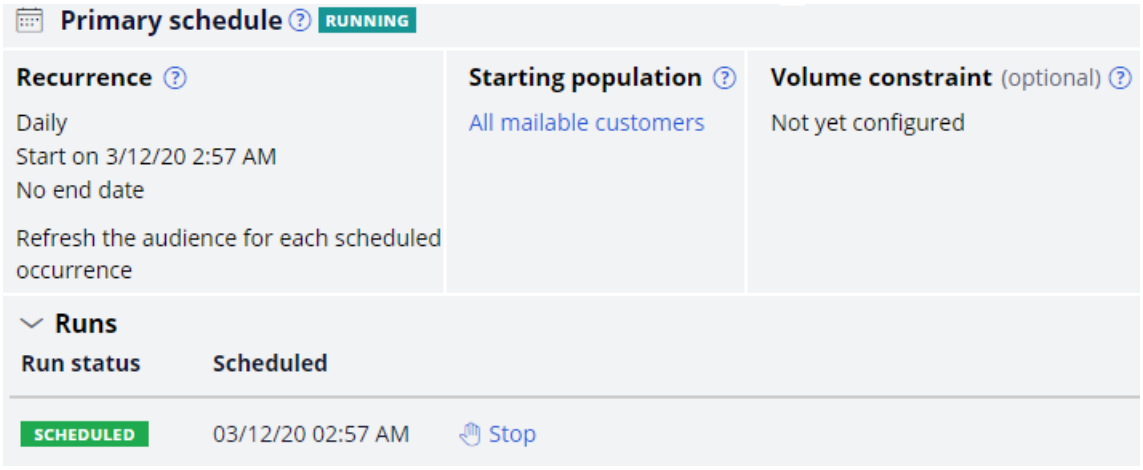
Starting population

For the **Starting population**, 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. The brain decides whether to contact a customer or not, and which specific action and treatment to send.

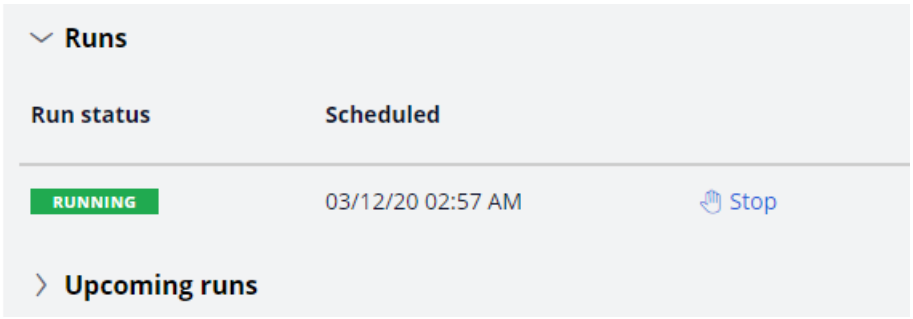


With that, all the necessary configurations for this scenario are complete.

Run the schedule. When the date and time for a scheduled run are reached, an entry appears with the status **Scheduled**, indicating that the run is about to begin.



The status changes to **Running** once processing begins. An entry for upcoming runs appears in the **Run Schedule** section.



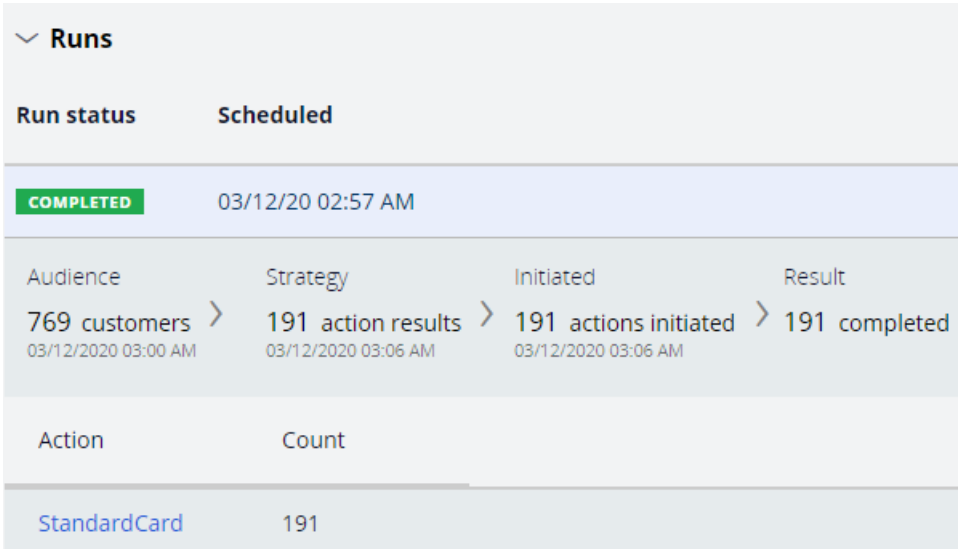
You can refresh to see the latest status. Once the run is complete, you can see the results of the run.

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 under **Initiated** represents the number of actions initiated for outbound messaging. The count under **Result** is the number of actions that have completed all the steps defined in their action flow.

Just below the state-level counts, you can see the actions in the current run, which were selected by the brain.



Qualified customers receive the email. When a customer clicks the **APPLY TODAY** call to action, it takes them to the pre-configured URL, which in this scenario is the Credit Cards landing page on the bank's

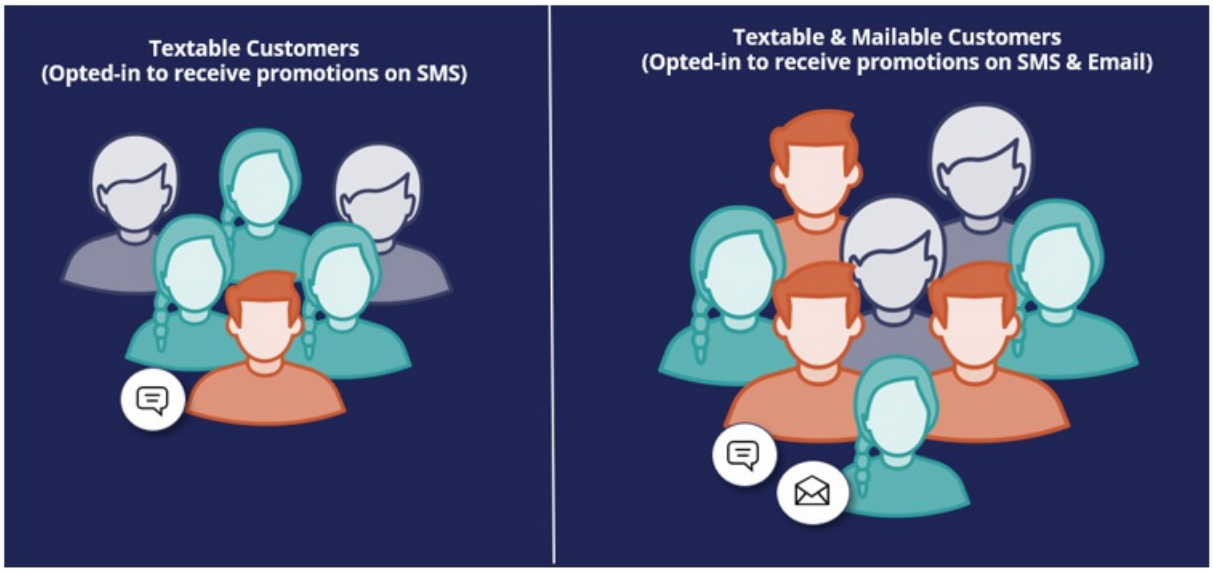
website.

Sending offer emails -- Thu, 07/23/2020 - 01:07
To get the full experience of this content, please visit https://academy.pega.com

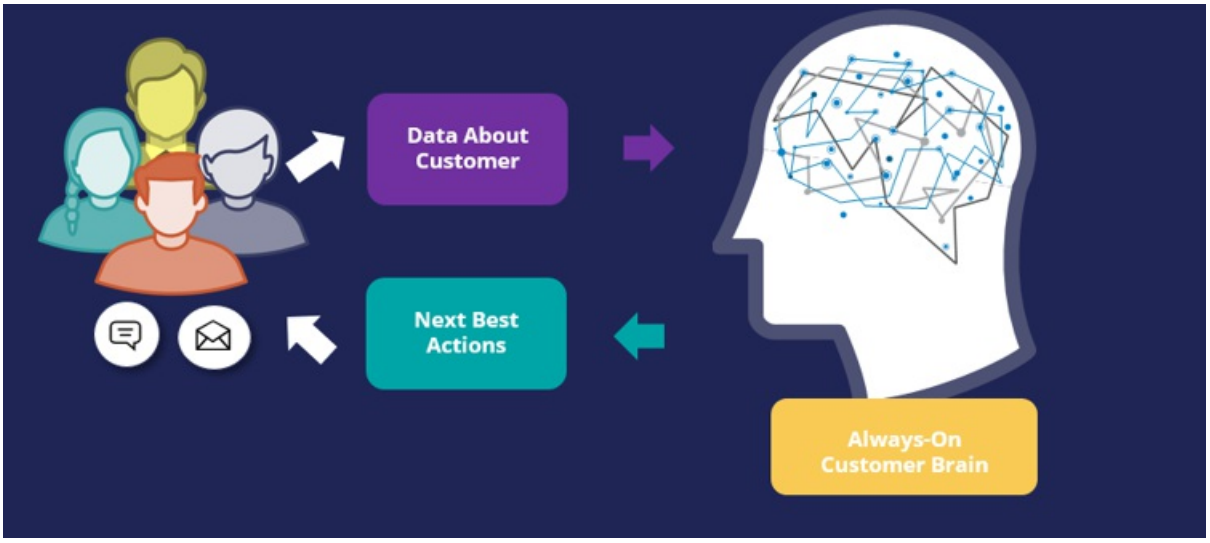
Segmentation

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.



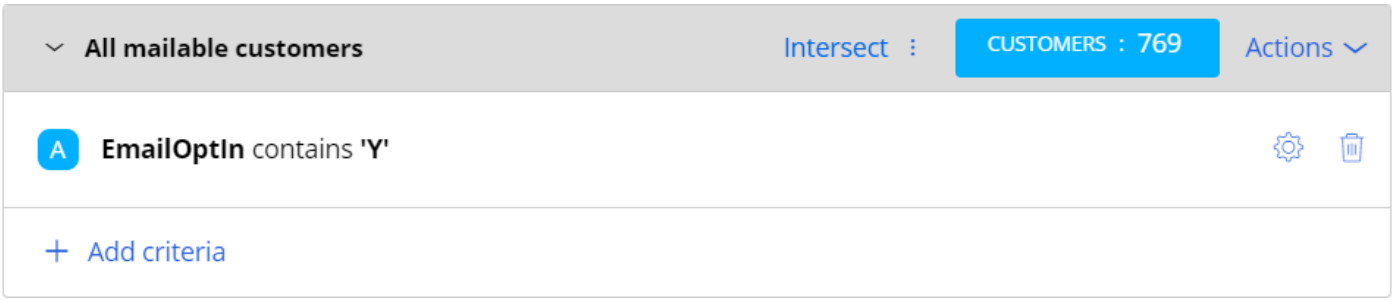
This set of customers is known as the starting population. The customers in this group will be further qualified through eligibility, applicability, and suitability criteria configured in Next-Best-Action Designer.

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.

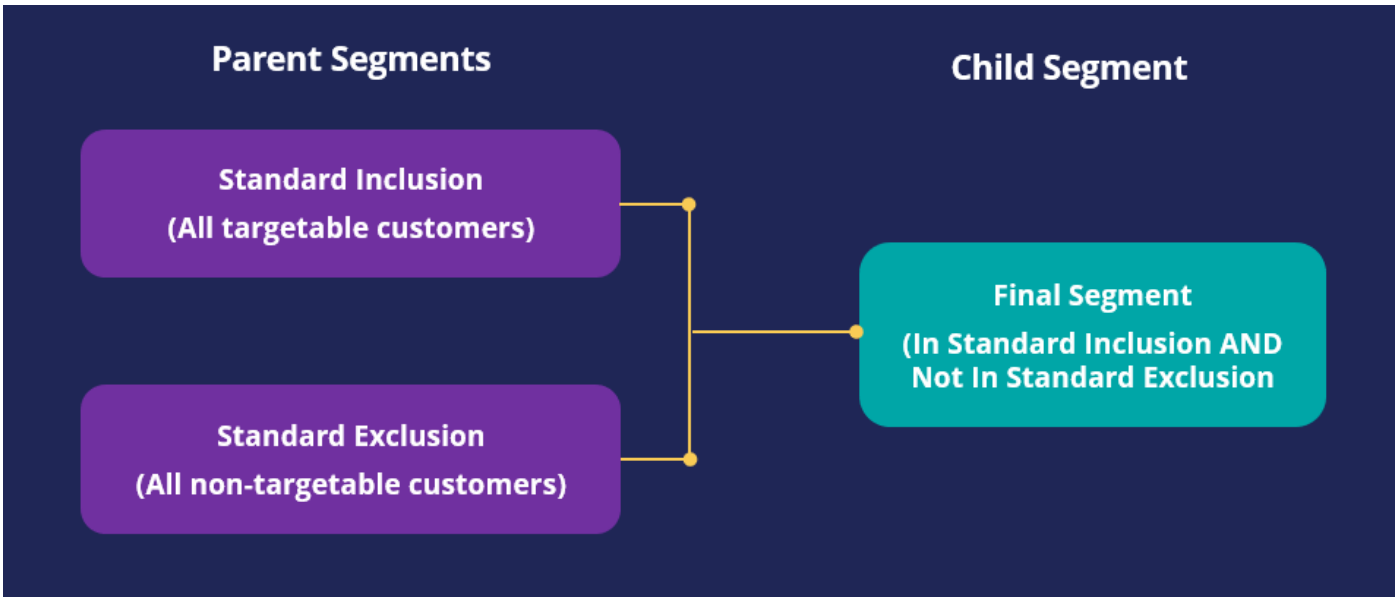
For example, to send emails related to a credit card offer to a target audience, you should first define a criterion that will filter for customers who have opted-in to receive promotional emails.



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

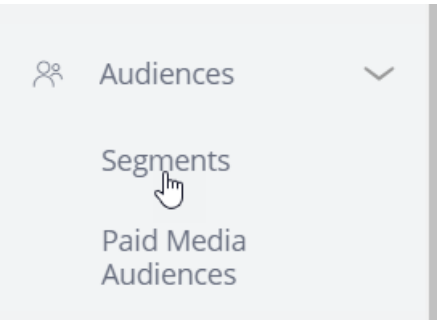
U+, a retail bank, wants to use the always-on outbound approach to send promotional emails to qualified customers. As the starting population for the outbound schedule, the bank wants to include all customers who have opted in to receive promotional emails.

Creating a segment

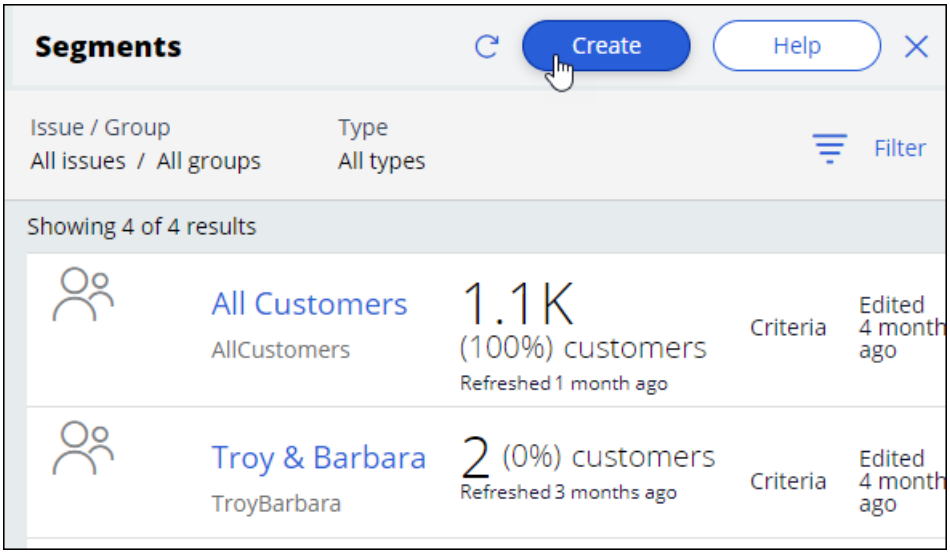
To meet the business requirement, you need to create a segment with two criteria.

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.

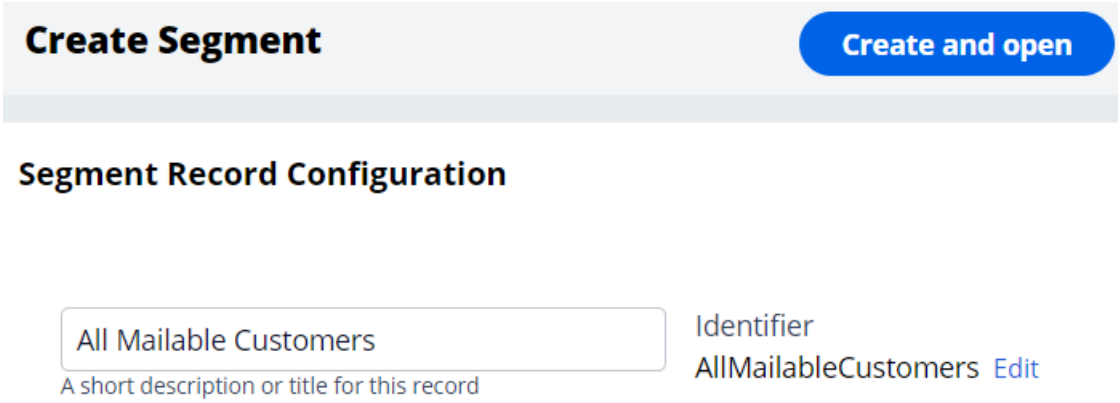
Segments are managed on the Segments landing page. The Segments landing page can be accessed from **Audiences** -> **Segments** in the navigation menu.



You can open existing segments or create a new segment from here.



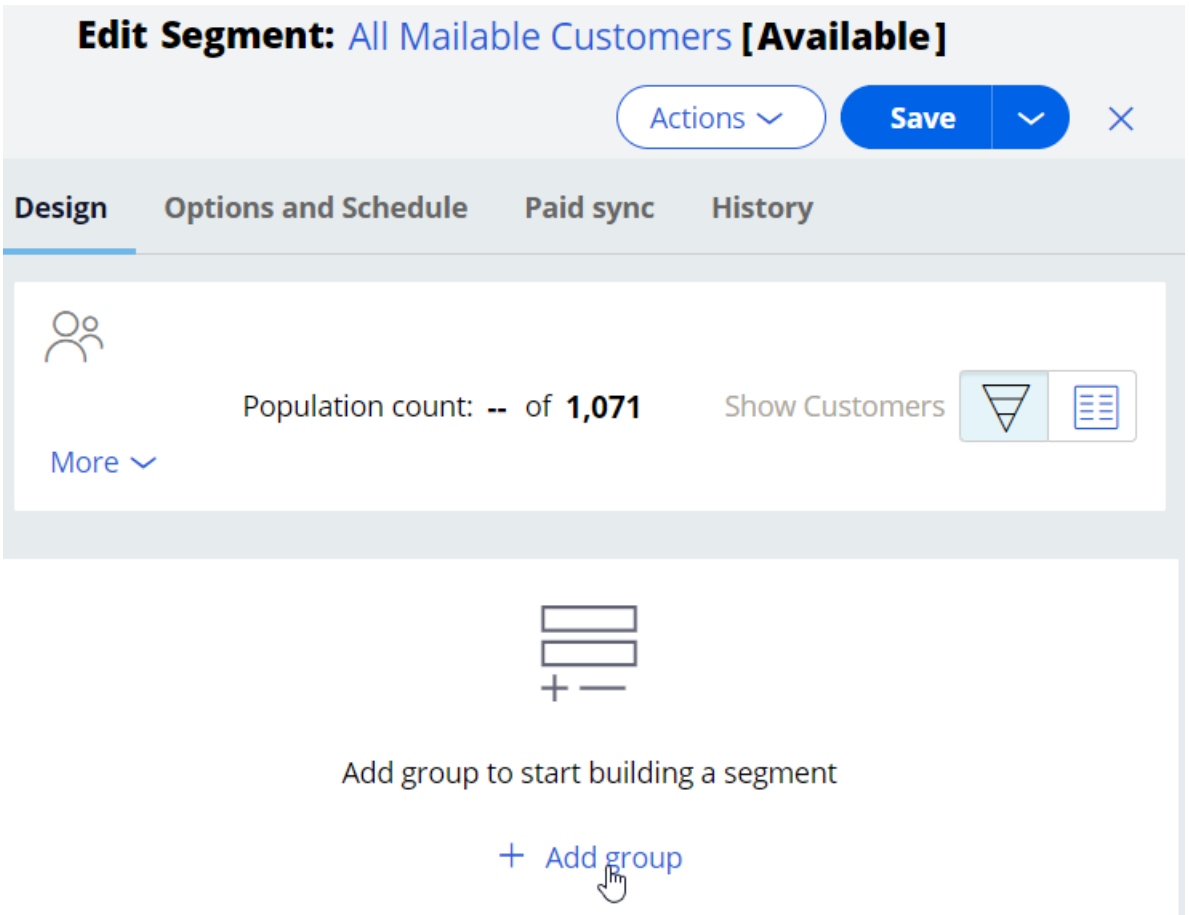
Provide a short description for the new segment, and the identifier is automatically populated when you tab out.



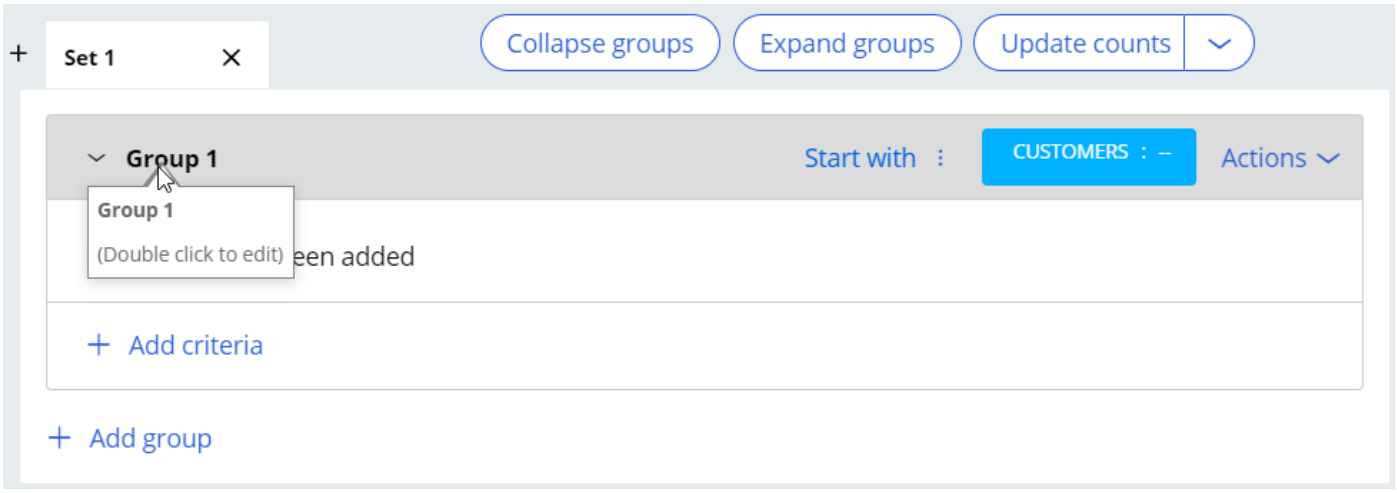
Entering the segment criteria

The **Design** tab is where you assemble the criteria required to identify customers of interest.

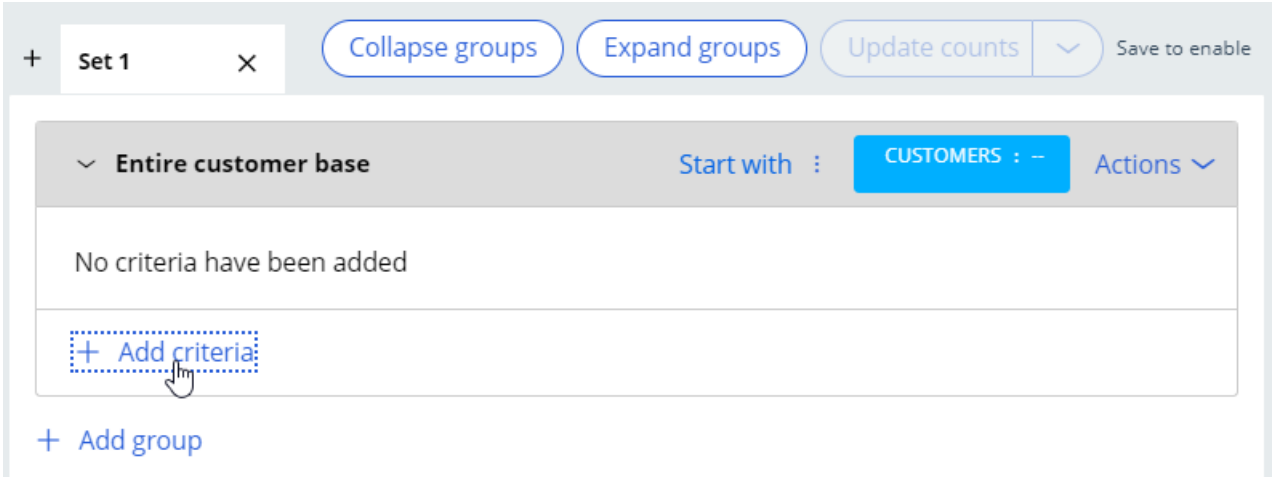
Segment criteria are grouped for ease of reading. So, before you can add the first criteria, add a criteria group.



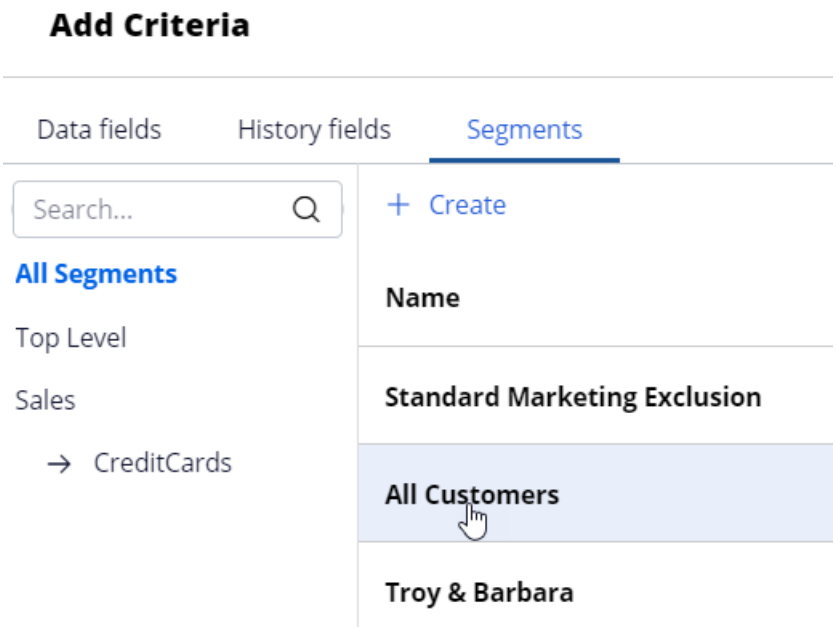
You can provide a more descriptive name for the criteria group.



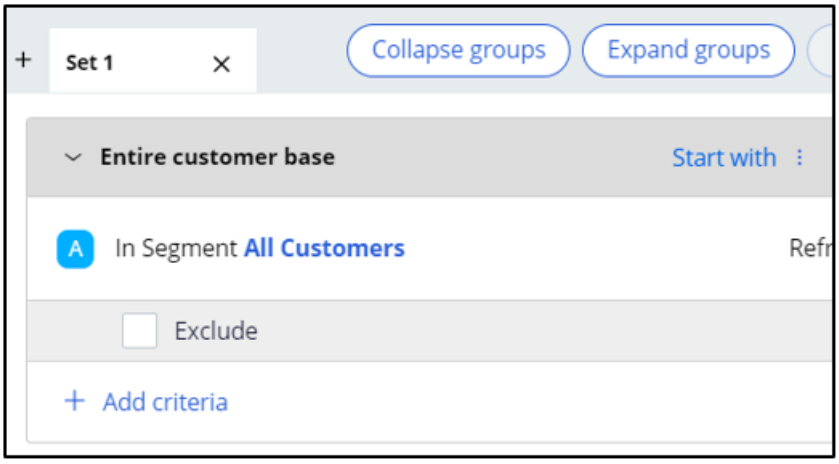
In this scenario, the bank wants to consider all customers in the customer base, then select the customers who have opted in to receive promotional emails. The bank has already created the **All Customers** segment, which contains the list of all customers, You can reference this segment in the first criterion.



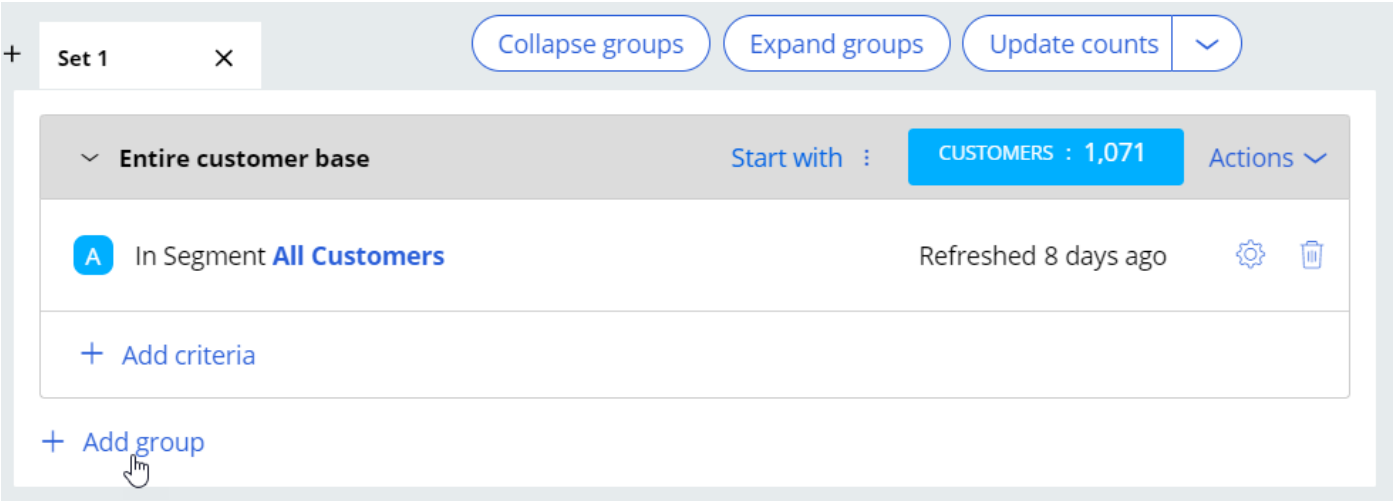
In the **Add Criteria** window, the **Segments** tab lists all available segments. Select the existing **All Customers** segment.



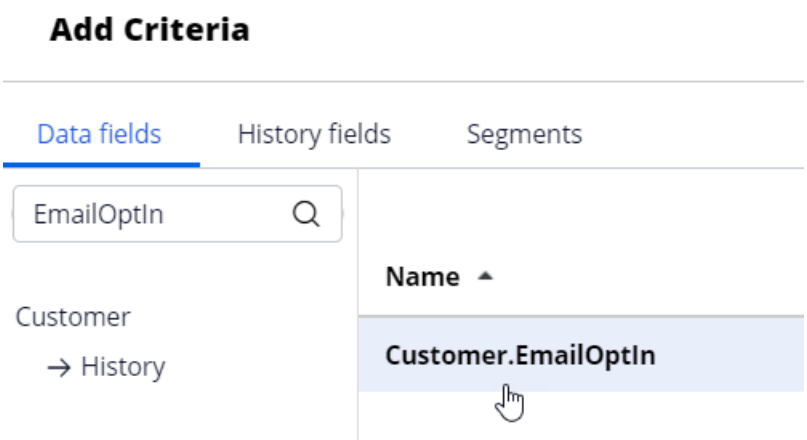
The **All Customers** segment is added as the first segment criterion. The **Exclude** option allows you to exclude the result of a criterion.



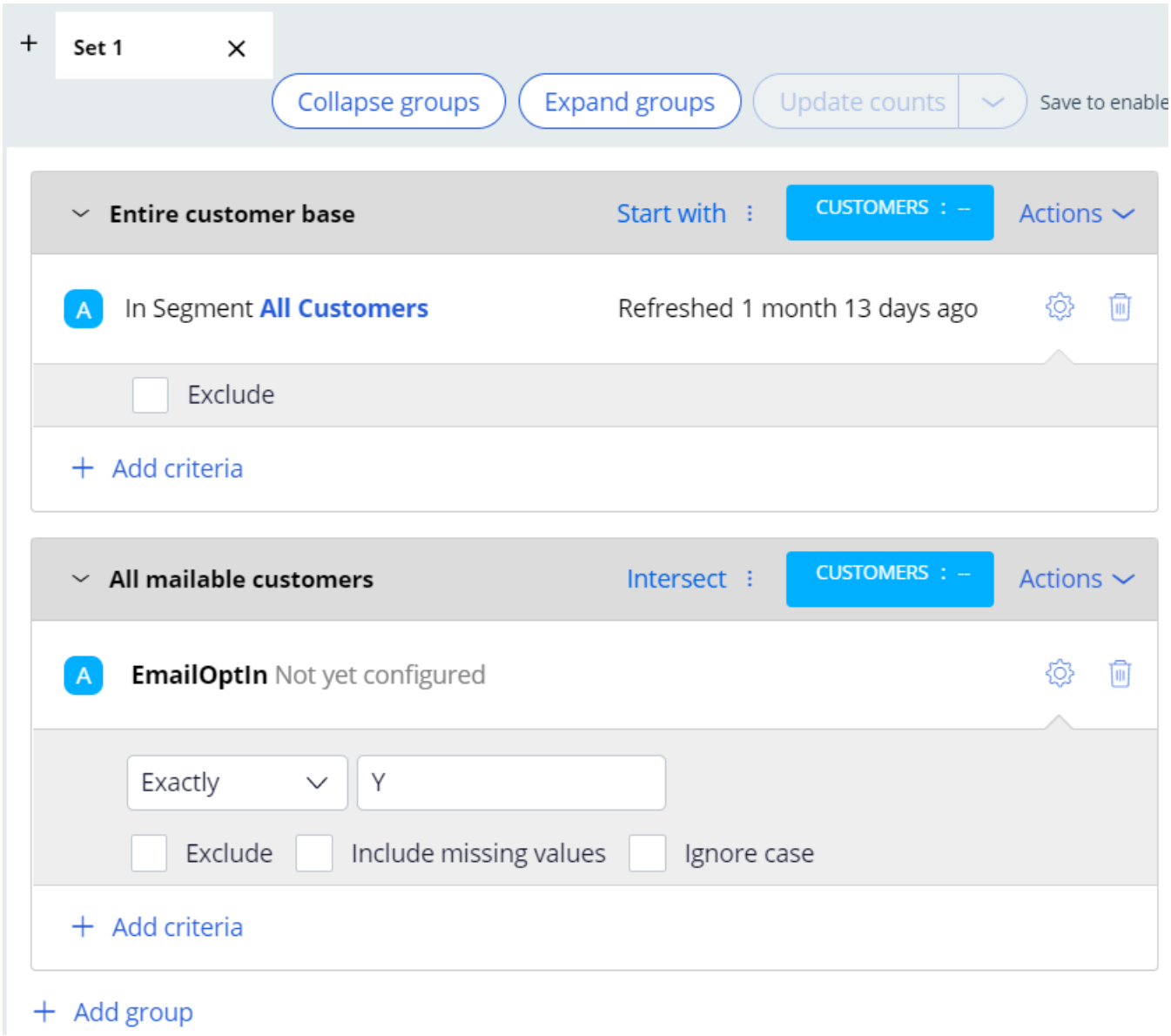
The next step is to add another criterion to identify customers who have opted in to receive promotional emails. You can add the new criterion in the existing criteria group or in a new group.



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.



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.



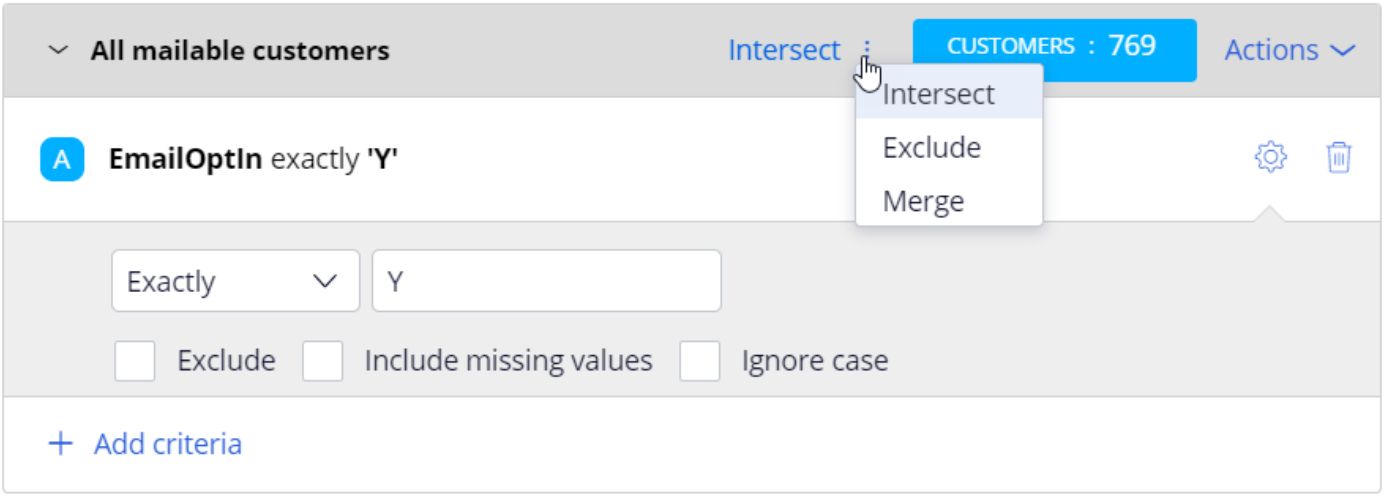
The **Include missing values** option is used to include results that do not have a value specified for the selected field.

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.

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.

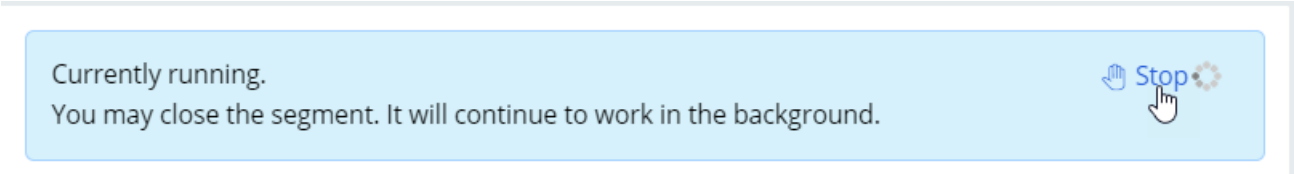
Exclude excludes the result of the current criteria group from the result above it.

Merge combines the results of both criteria groups.

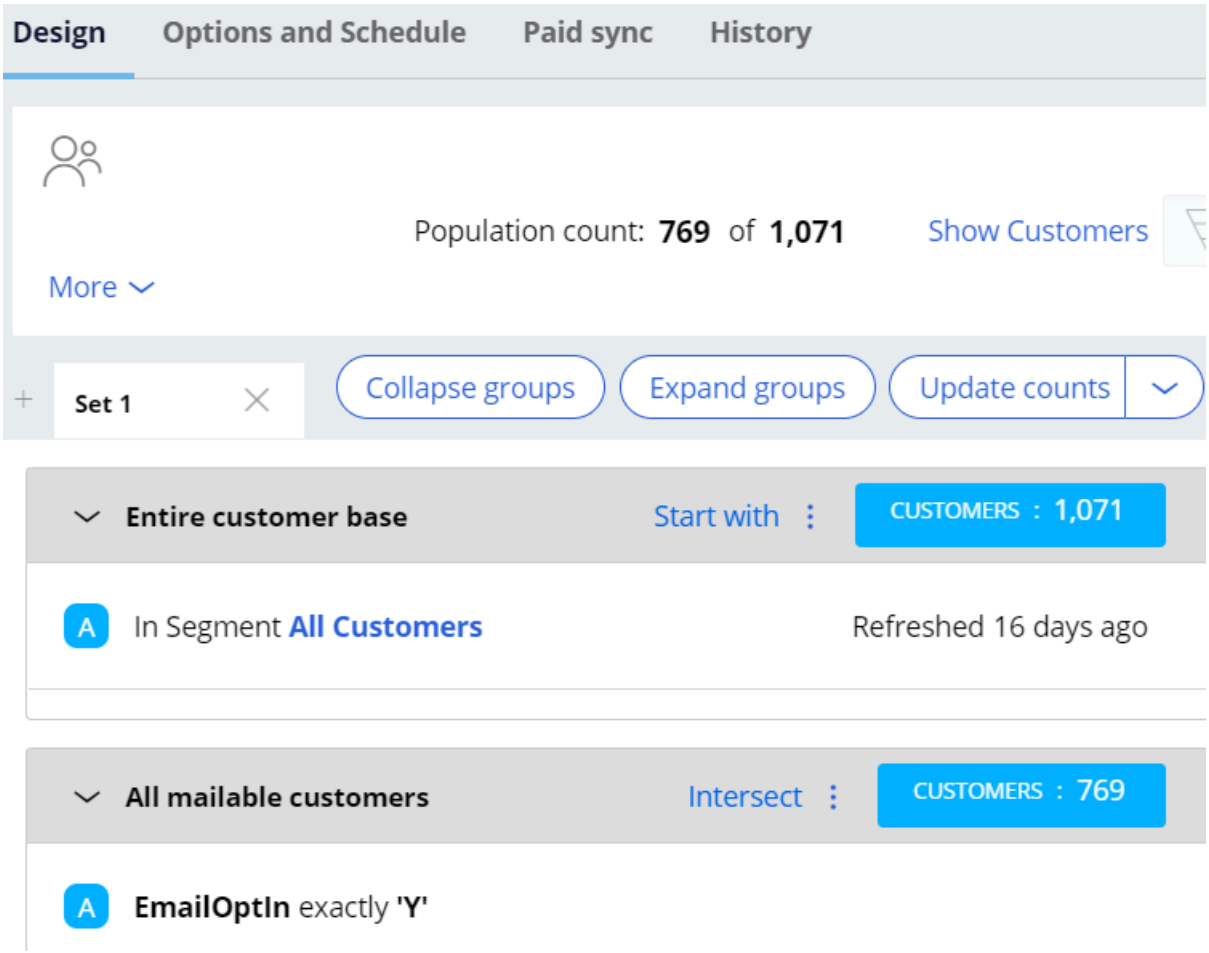
Save and run the segment

When you are done defining all the segment criteria, you can save and run the segment. 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**.



There are **1071** customers in the **All Customers** Segment, out of which **769** customers have opted in to receive promotional emails.

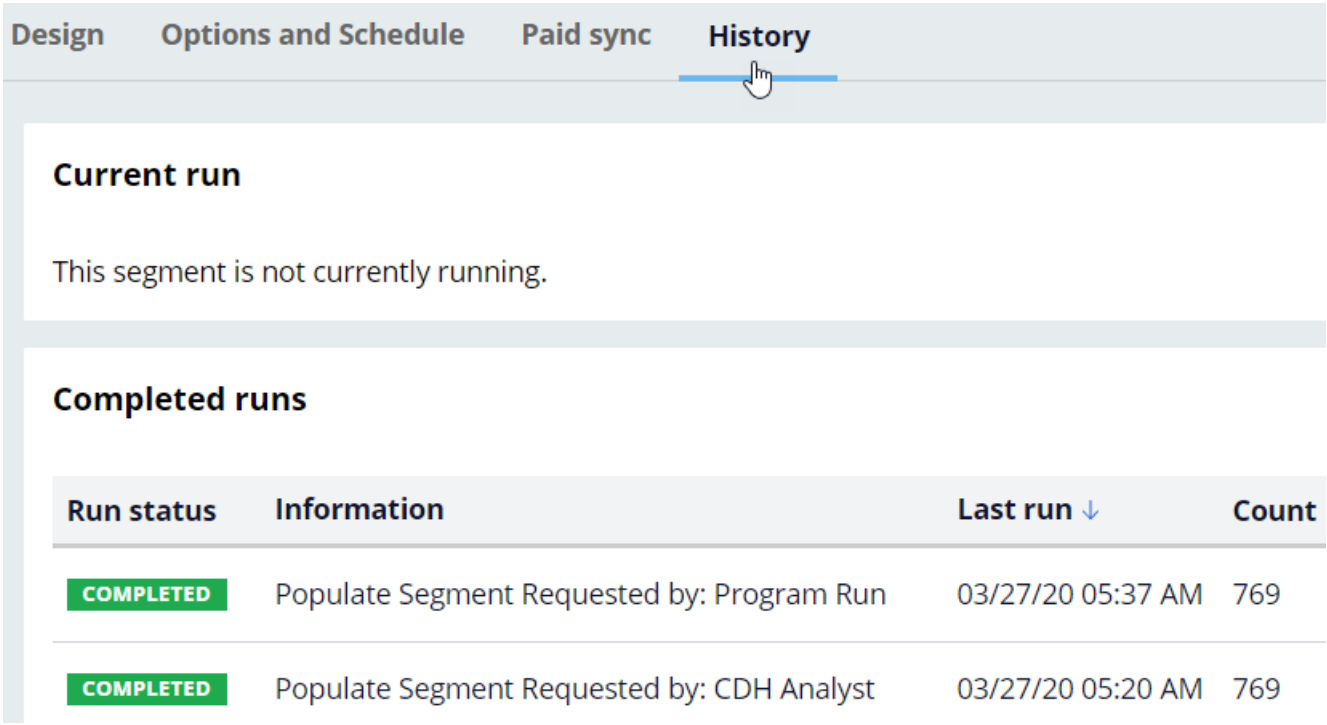
Click the **Show Customers** link to preview the list of customers selected for this segment.

The list is shown in a separate window.

CustomerID*	Full Name	Gender	Age	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

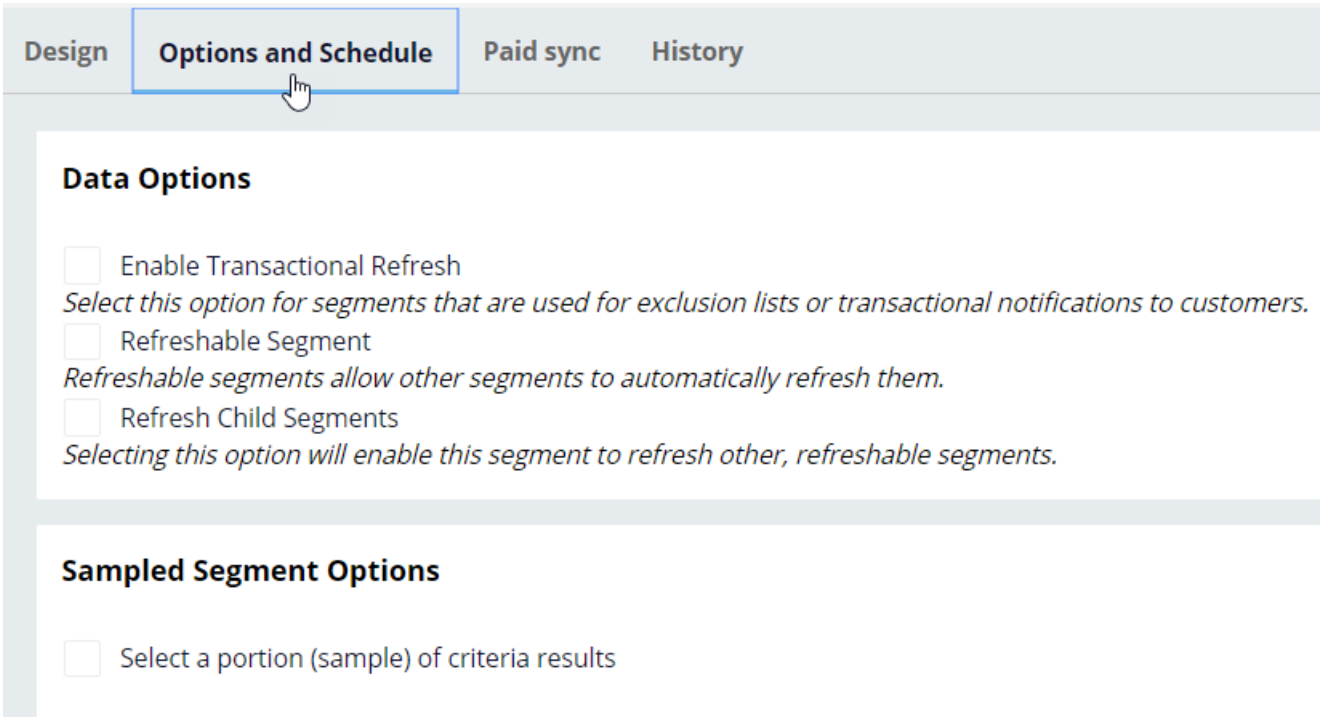
Segment run history

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**.



Key segment configurations

On the **Options and Schedule** tab, there are two key settings under **Data Options** that control how data in the segment is refreshed at run-time and what level of control a segment can exert over other segments.



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.

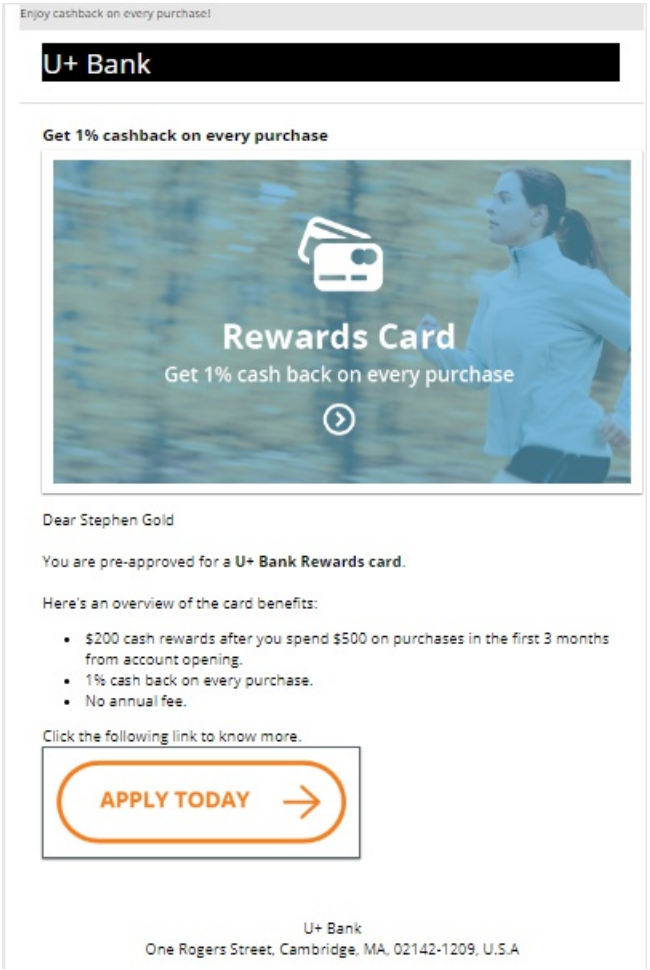
The **Sampled Segment Options** allows you to select a random sample of a segment result to be selected as the final segment result.

Creating an email treatment

Business scenario

U+, a retail bank, would like to promote their Rewards credit card via email. They want to send the email to qualified customers in an always-on outbound environment.

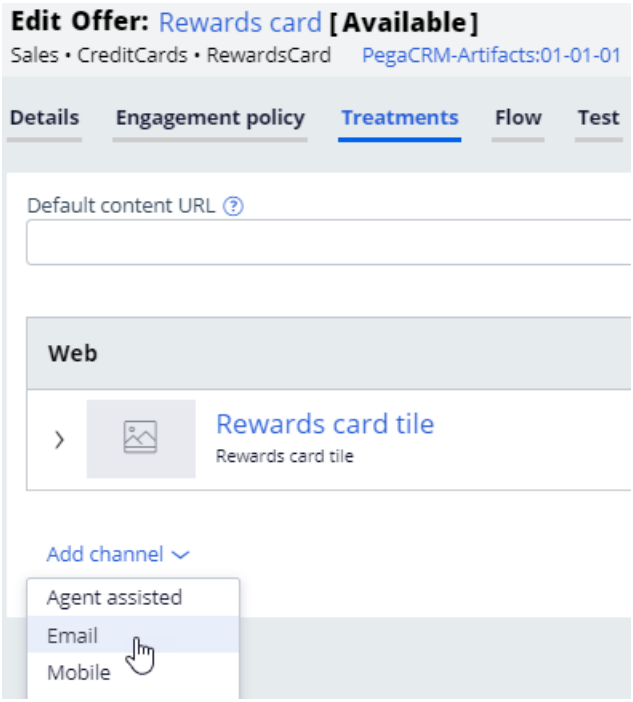
The email will contain the offer details and a call to action. When a customer clicks the **APPLY TODAY** call to action in the email, it will take them to the credit cards page on the bank's website where they can learn more about the offer and apply for it if interested.



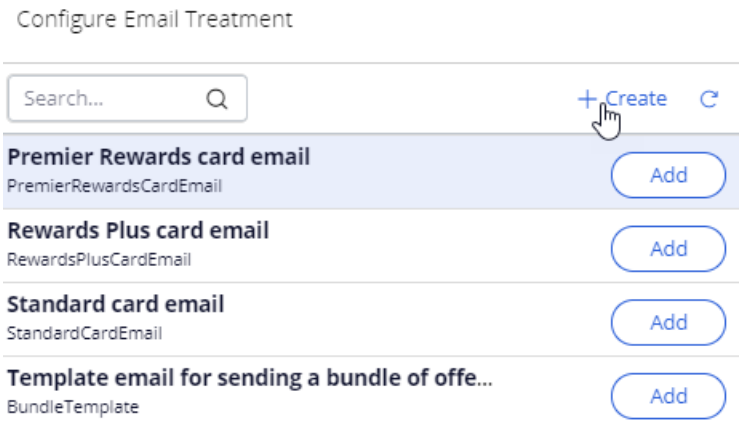
Creating an email treatment

To send emails in an always-on outbound environment, you need to add an email treatment to every action that you want to promote via email. An email treatment contains the message that you want to communicate to your customers.

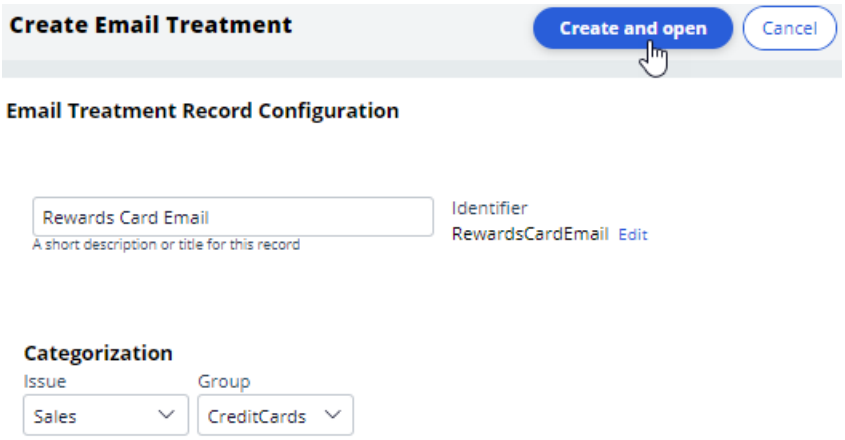
You add new treatments to an **action** on the **Treatments** tab.



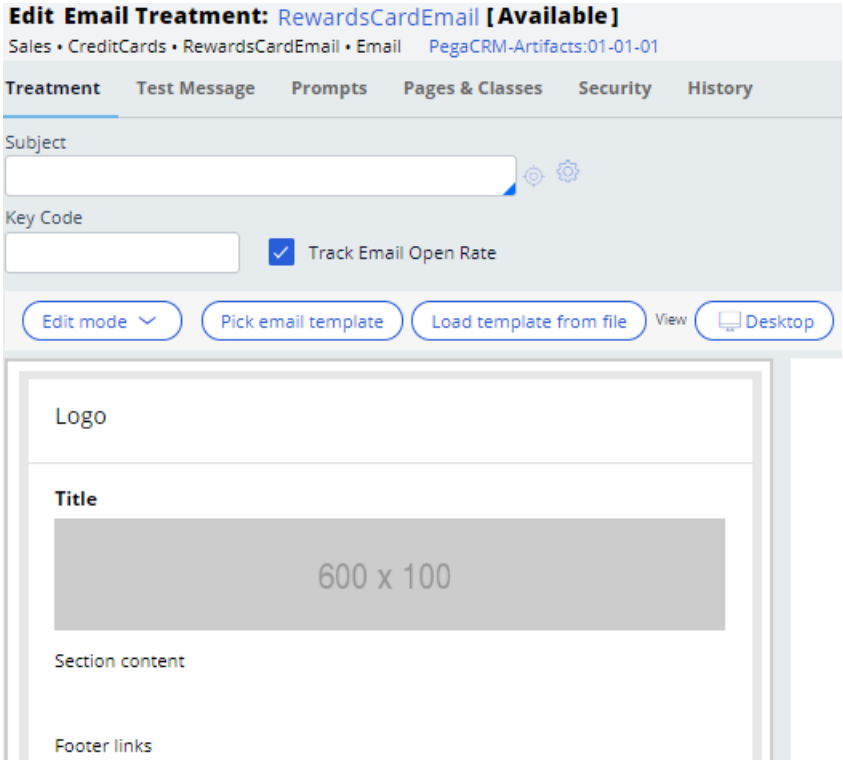
You can use an existing email treatment or create a new one.



Enter an appropriate **short description** to create the new treatment and tab out. The**Identifier** is automatically populated. It's a good practice to create the treatment in the same issue/group hierarchy in which it will be used; this will make it easier to access and manage.



Enter the email content on the**Treatment** tab.



Email subject

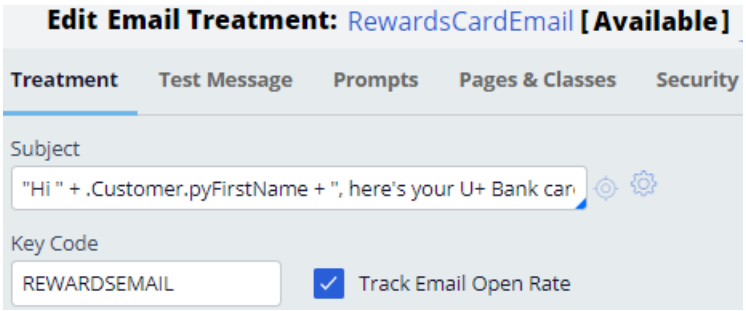
Enter the email subject in the **Subject** field. You can include both static text and dynamic placeholders in the subject. Dynamic placeholders allow you to personalize the subject and can be constructed with both customer and offer attribute properties.

Here is an example:

"Hi " + .Customer.pyFirstname + ", here's your U+ Bank card"

At runtime, the properties will be replaced with real values. **If the customer's first name is** *Stephen*, **the generated subject line will be:** *Hi Stephen, here's your U+ Bank card*

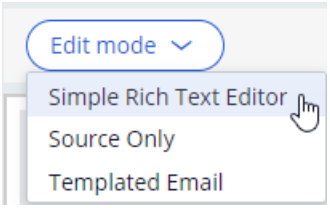
When the **Track Email Open Rate** option is enabled, the system records an impression when a customer opens an email.



Email content

You can design the email using one of the three edit modes.

1. Simple Rich Text Editor
2. Source Only
3. Templated Email

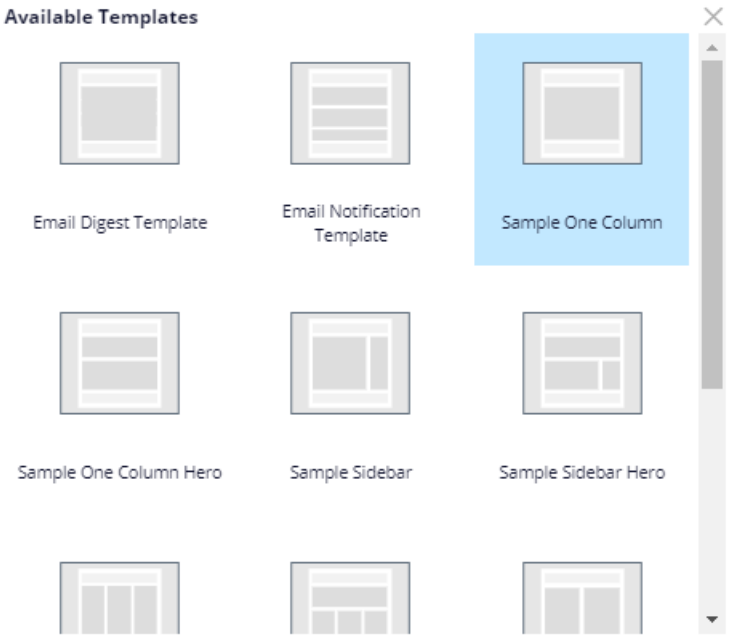


The **Simple Rich Text Editor** provides a blank page with a rich text editor, which allows you to compose an email in any format you like.

The **Source Only** mode allows you to work with HTML content only. If the content is designed in an external tool, you can copy-paste the HTML content in this mode.

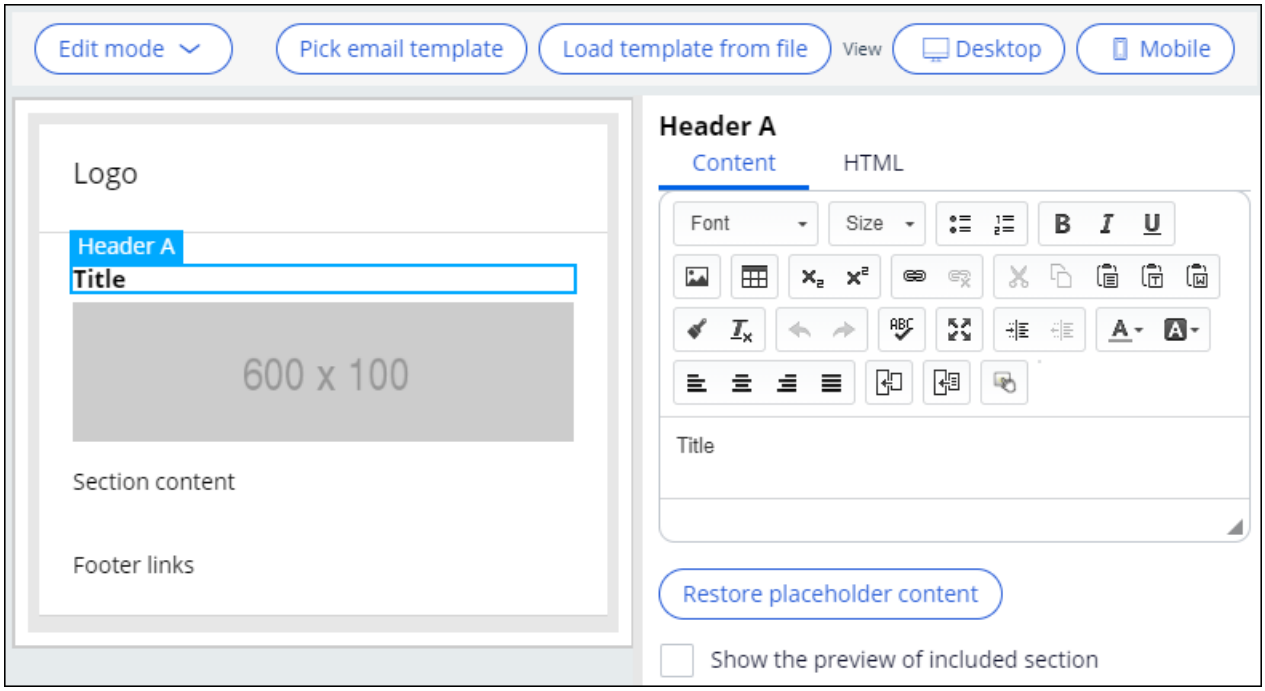
The **Templated Email** mode provides you with several easy-to-use templates for designing appealing emails. This is also the default mode.

Here's a sample of some of the template options available.



The **Simple One Column** template is the default template. Also, you can choose to upload a new template of your own.

When you choose one of the templates, a preview section is presented on the left and an editor on the right. The content is shown in a readable format as it's being built.



Composing the email using a template

In the email Treatment, you can create the content with dynamic placeholders, images, buttons, and links. The content is shown in a readable format as it's being built. Alternatively, you also can paste HTML code to generate the content you want.

The email structure is different in different templates. For example, the Simple One Column template contains 4 sections.

1. Preheader
2. Header
3. Body
4. Footer

The Preheader section can contain a short **data teaser for inbox** that will hopefully persuade the customer to read the email further. This text is hidden on a mobile device.

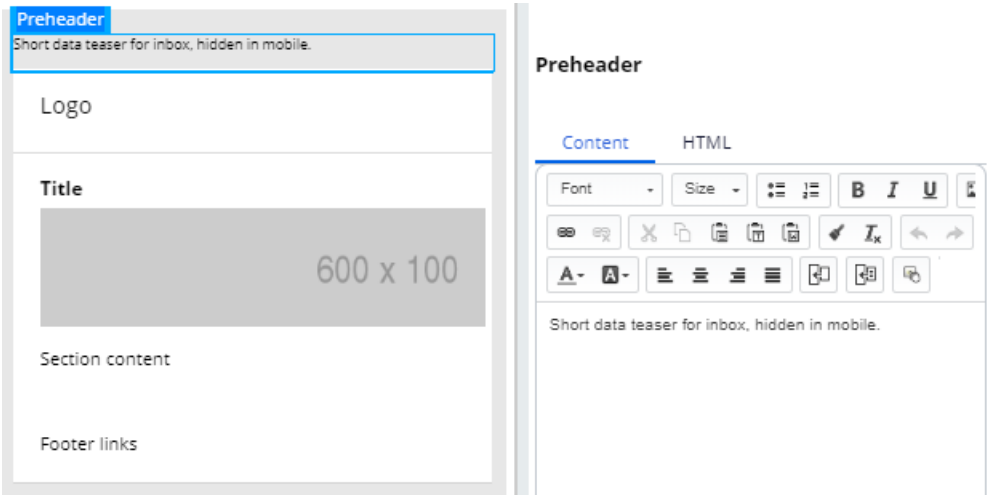
The Header section can contain the organization's logo.

The Body section contains a placeholder for a title, image, text, and/or buttons as required. This is where the main content goes.

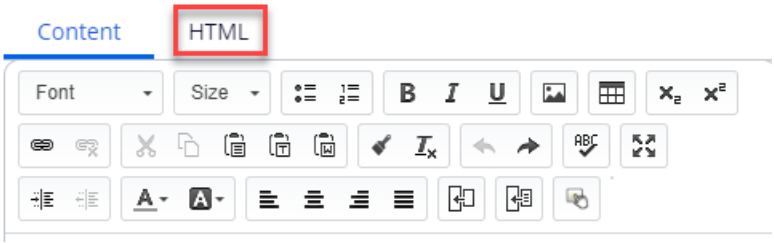
The Footer section can contain contact details or links, if any.

Entering the email content

When you click on the different placeholders or sections in the template, an editing pane opens on the right of the screen.

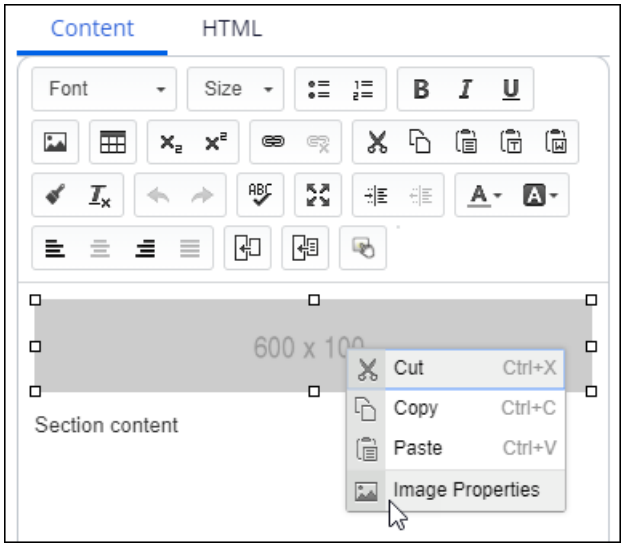


This editor also allows you to copy-paste HTML content from other tools using the **HTML** tab.

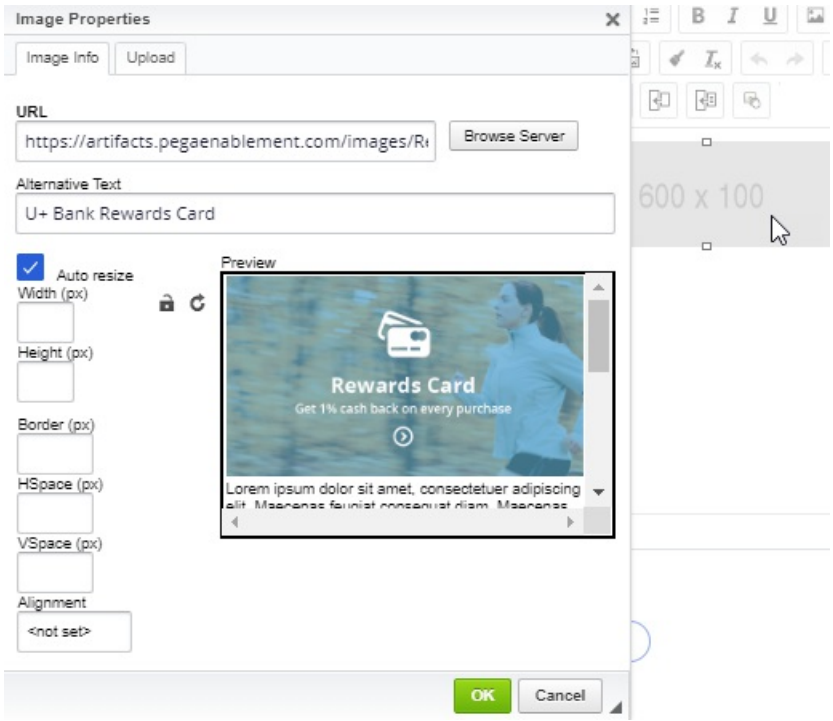


For example, here's how an HTML version of the U+ Bank logo is created using simple HTML code.

To add an image, you can double-click on the image placeholder, which opens the **Image Properties**.



In the **Image Properties** you can reference an image from a content server.



It is best practice to host an image on an external server and reference it in the email. Images should not be embedded directly in the email, as it makes the email bulky. Also, images should not be uploaded from the Image Library within the Pega application. These images will not be rendered on a customer's email client.

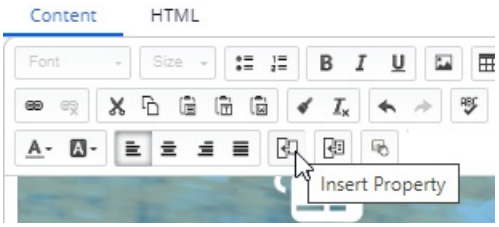
In the email content, you can add static content as well as dynamic placeholders to personalize the message.

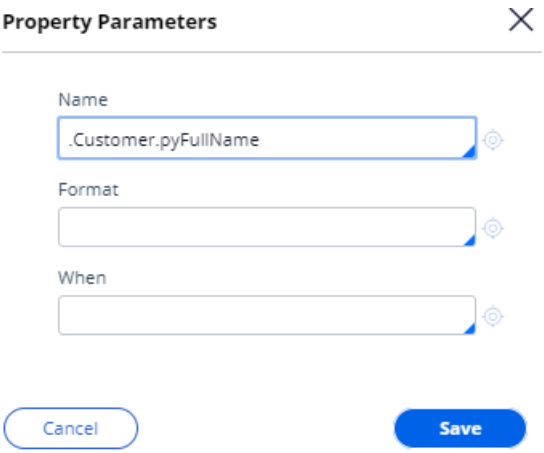
Adding personalization

For example, the property **.Customer.pyFirstName** can be added as a placeholder, and it will be replaced with the customer's real first name at the time of email delivery.

```
Dear <<.Customer.pyFullName>>
```

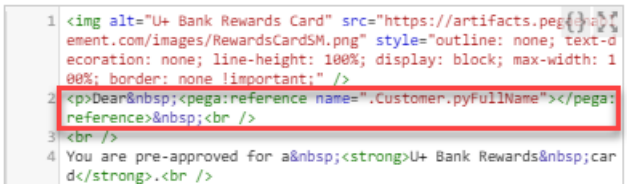
To add a customer or offer attribute as a dynamic placeholder, use the**Insert Property** button.





To view or edit an existing dynamic placeholder, select the placeholder and click on the Open icon that appears next to it.

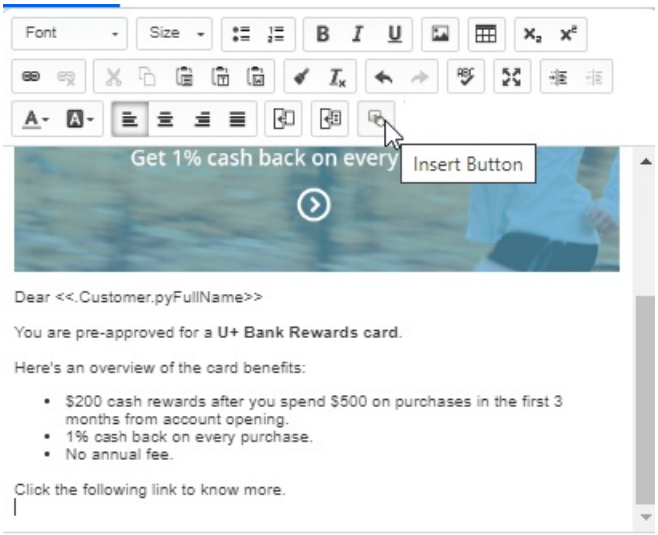
Any property you add in an email treatment has an equivalent Pega markup tag (**pega:reference)** to reference customer or offer attributes. After you insert the property, you can view the tag (which is automatically added) in the HTML tab. When you design the email in an external application, you can use these tags to add dynamic placeholders, so the personalization works when you import the HTML into the email treatment.



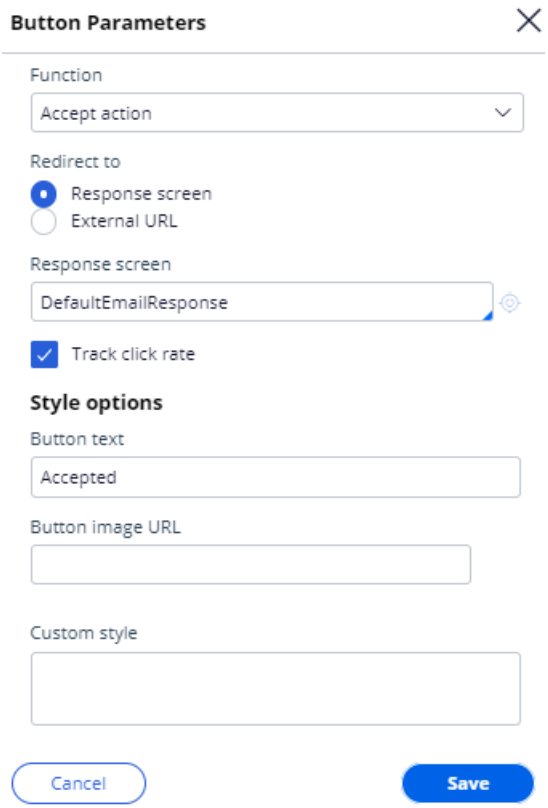
To add an offer attribute, you can add a dynamic placeholder, which is then replaced with the offer attribute at the time of email delivery. For example, you can use <<.OfferData.ActionName>> to reference the offer attribute.

Adding links / buttons

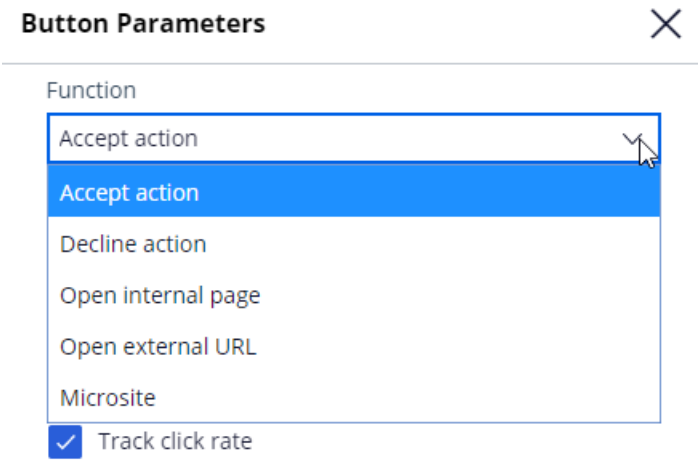
You can also add links or buttons that the customer can click to respond to an offer or view more details. Click **Insert Button** to add a button.



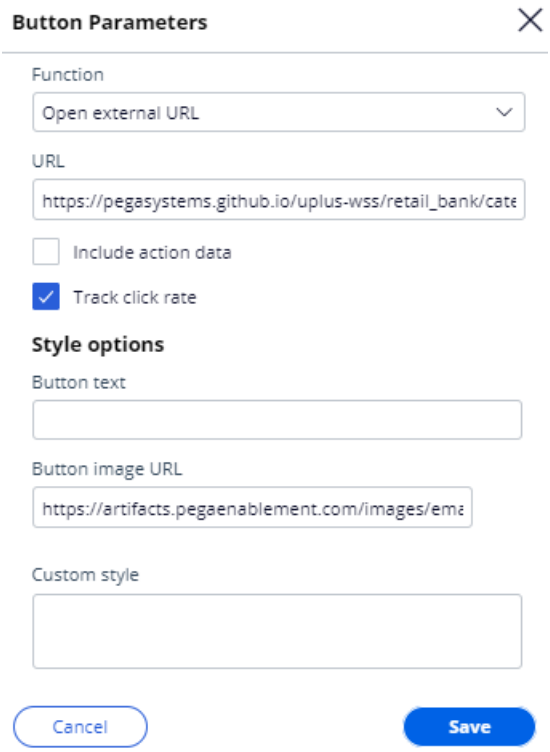
In the **Button Parameters**, the **Function** drop-down menu lists the possible functions that can be assigned to a button.



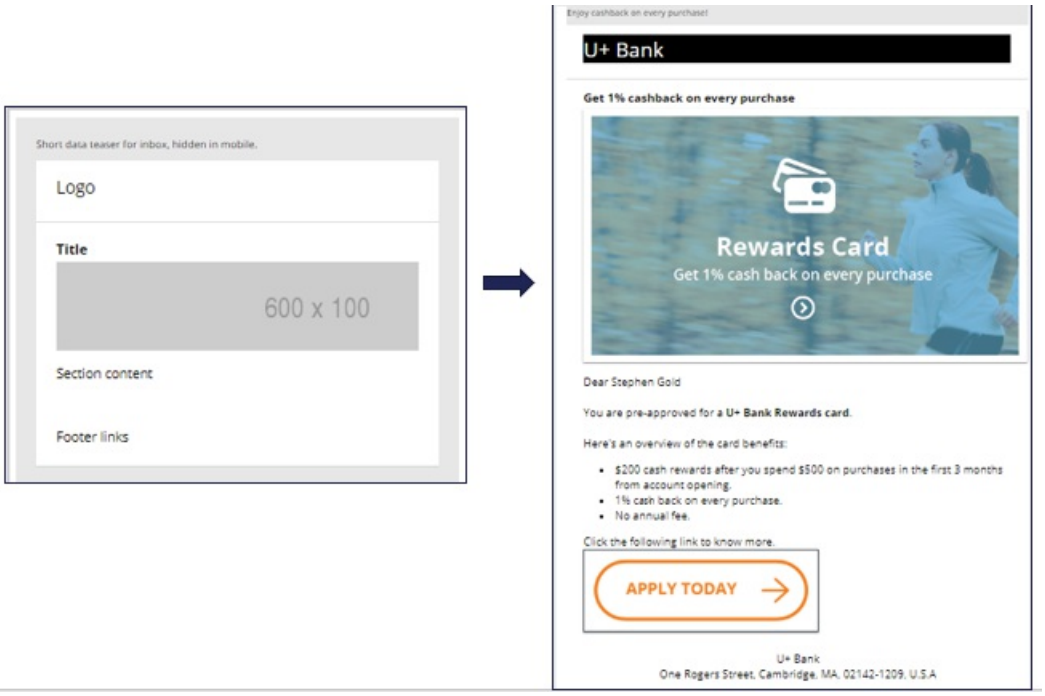
Accept and **Decline** buttons trigger the recording of customer responses to actions in the Interaction History when the customer clicks them.



The **Open external URL** button opens an external URL when the customer clicks it.



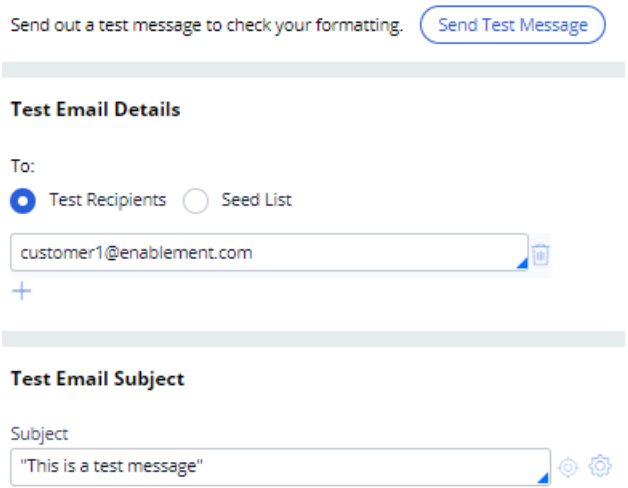
A composed email looks like the following.



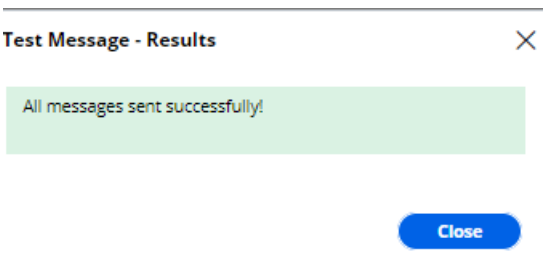
Testing the treatment

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 Message** tab. Here you can specify the target email addresses as well as the email subject specifically for the test.

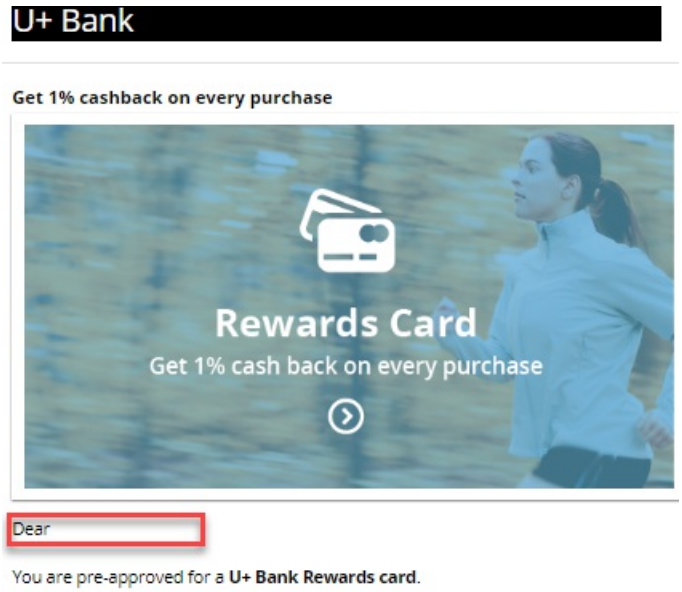


To test the Treatment, click the**Send Test Message** button. Once the test is complete, you can check the emails in the email client.

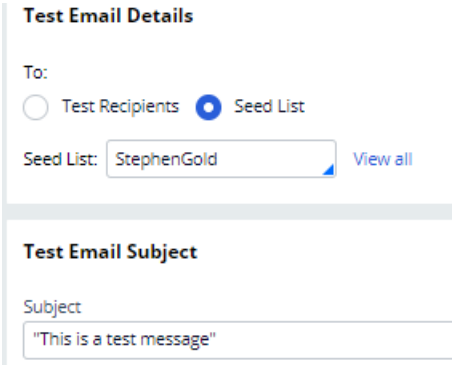


Note that none of the attribute placeholders are populated when you manually type in the recipients' email address.

However, you can test the functionality of the buttons and links.

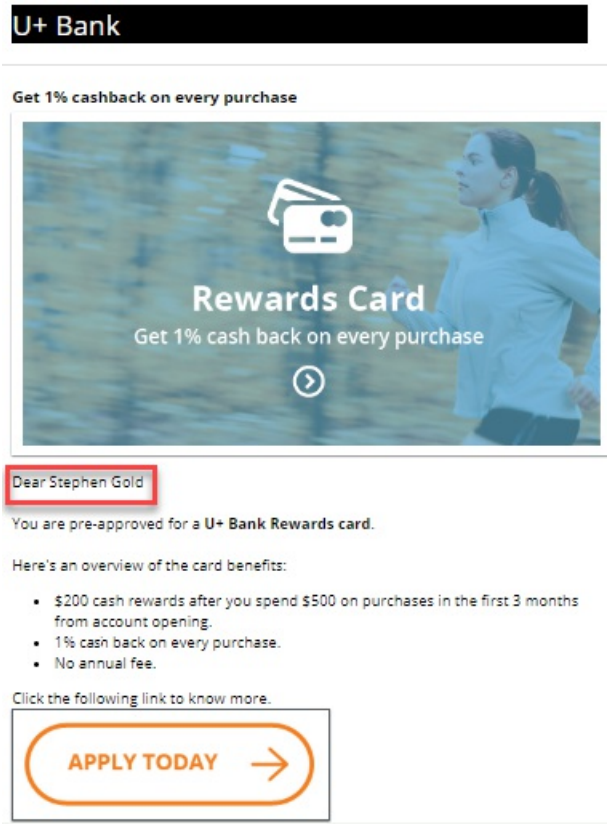


To populate the attribute placeholders, you can use a **seed list** as test input.



A seed list contains test customer data. It contains values for the customer attributes, including email address.

When using a seed list to test the Treatment, the dynamic placeholders are replaced with customer information from the seed list.



Creating an email treatment -- Tue, 10/27/2020 - 00:51
To get the full experience of this content, please visit https://academy.pega.com

Defining the action flow

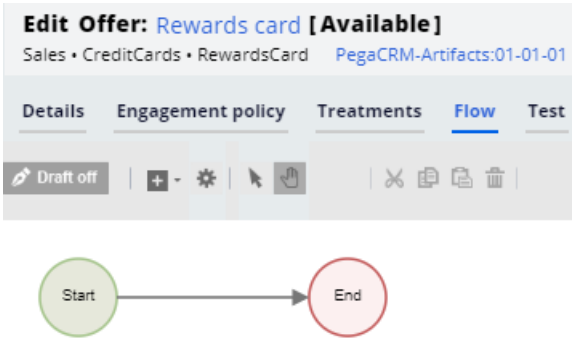
Business scenario

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.

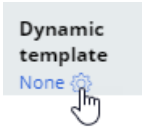
Action flow

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.

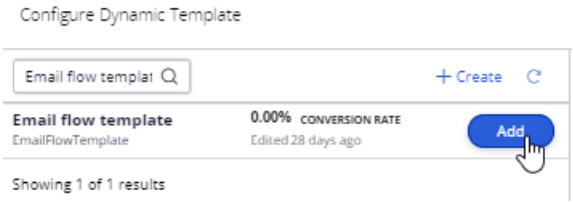
A new action contains only the default **Start** and **End** shapes. To send emails in an always-on outbound environment, you need to add a **Send Email** shape to the action flow.



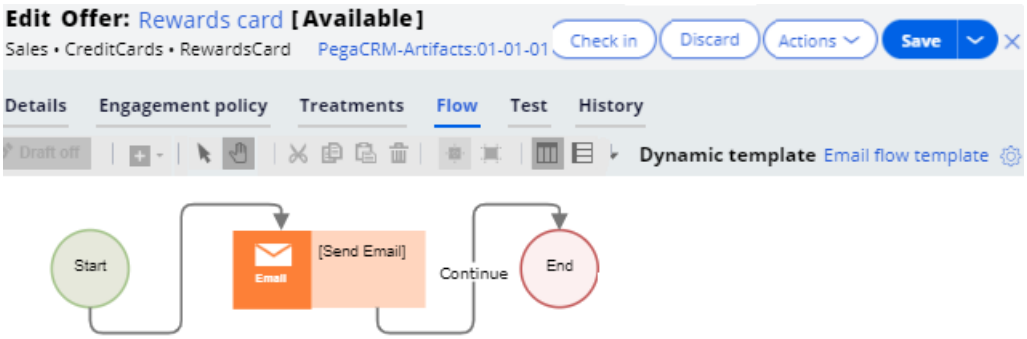
When there are several actions that need the same flow pattern, you can create that flow pattern once and reuse it for multiple actions using the **Dynamic template** option.



In the **Configure Dynamic Template** window, select the action that contains the desired flow pattern.



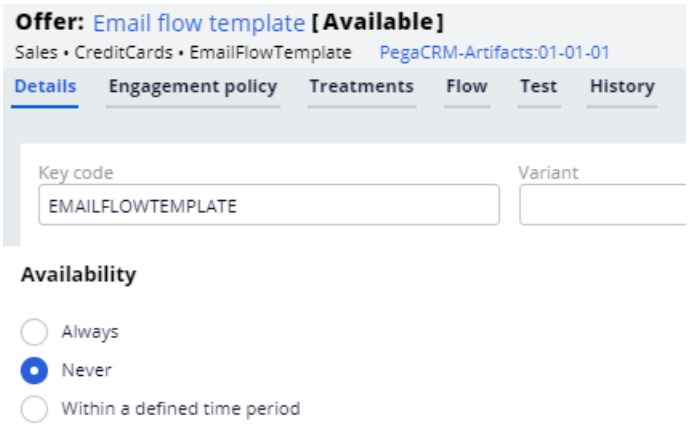
When you use the **Dynamic template** to configure the flow, you cannot change the flow in the referenced action. To edit the flow, you must edit the original action in which the flow was defined.



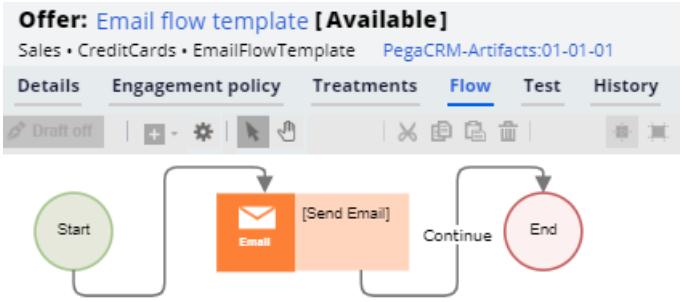
Basic email flow template

Let’s look at a basic flow pattern for email delivery.

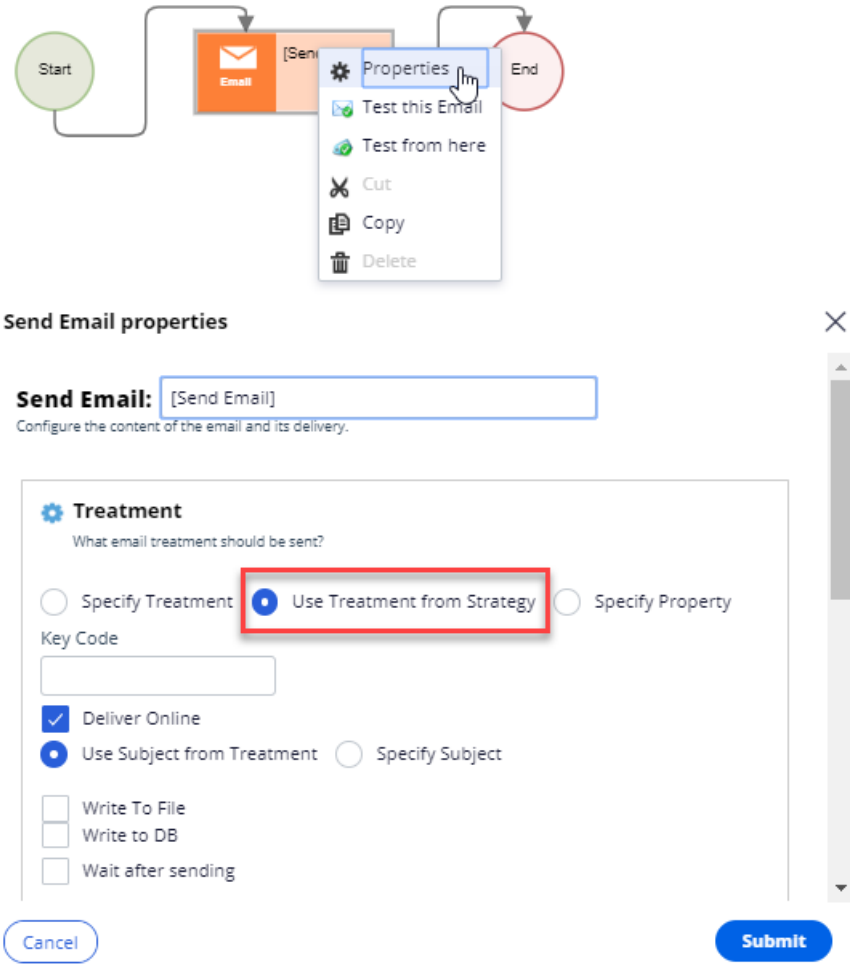
A flow template is just a normal action with the flow configured. In the following example, an action is used as a flow template within other actions to enable them to send emails. Thus, the **Availability** of the action is set to ***Never*** to prevent it from being presented to customers.



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.



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.

Defining the action flow -- Thu, 07/23/2020 - 01:10
To get the full experience of this content, please visit https://academy.pega.com

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.

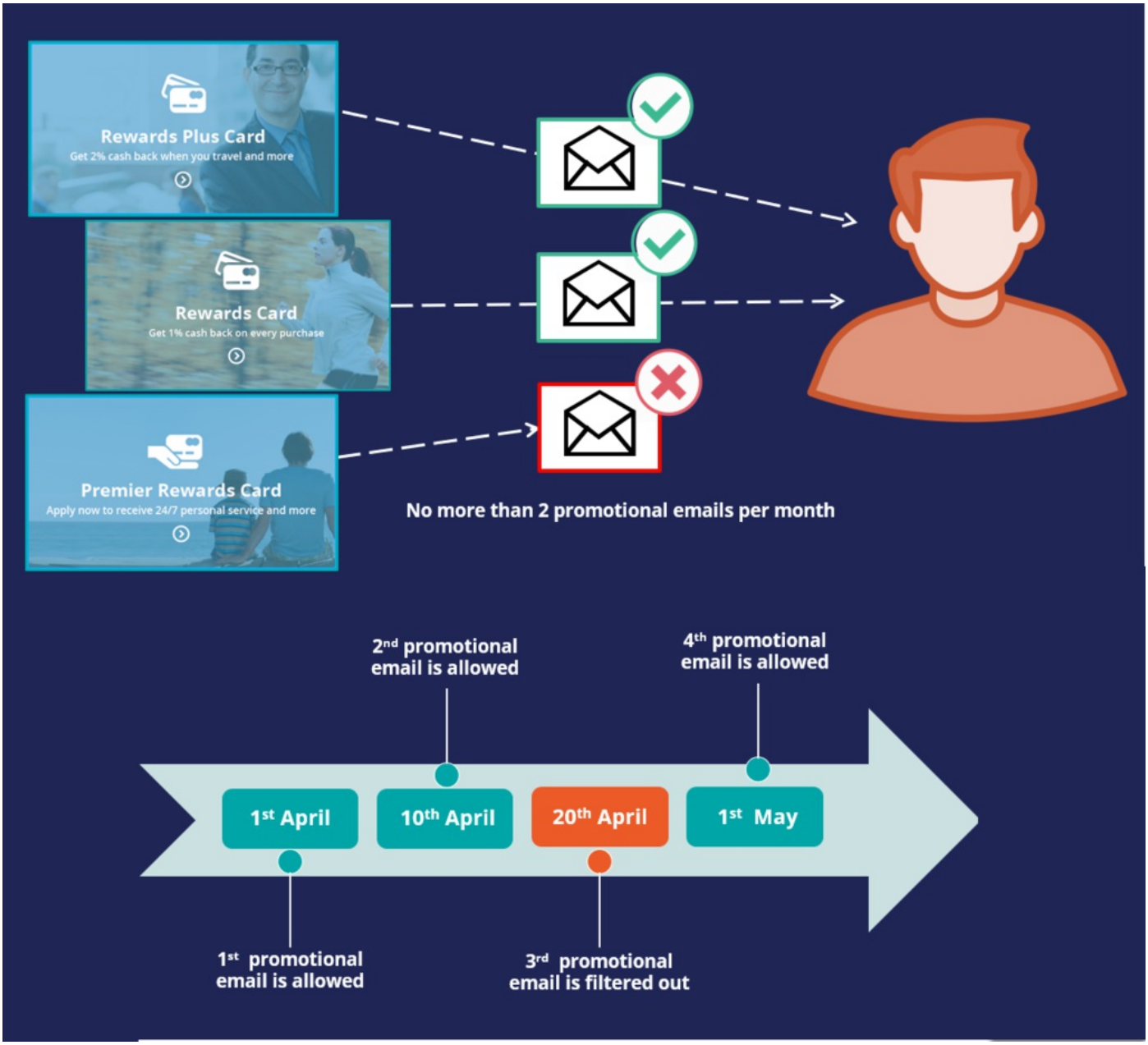
In Pega Customer Decision Hub™, there are two types of contact limits that allow you to limit actions over a given period of time.

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.

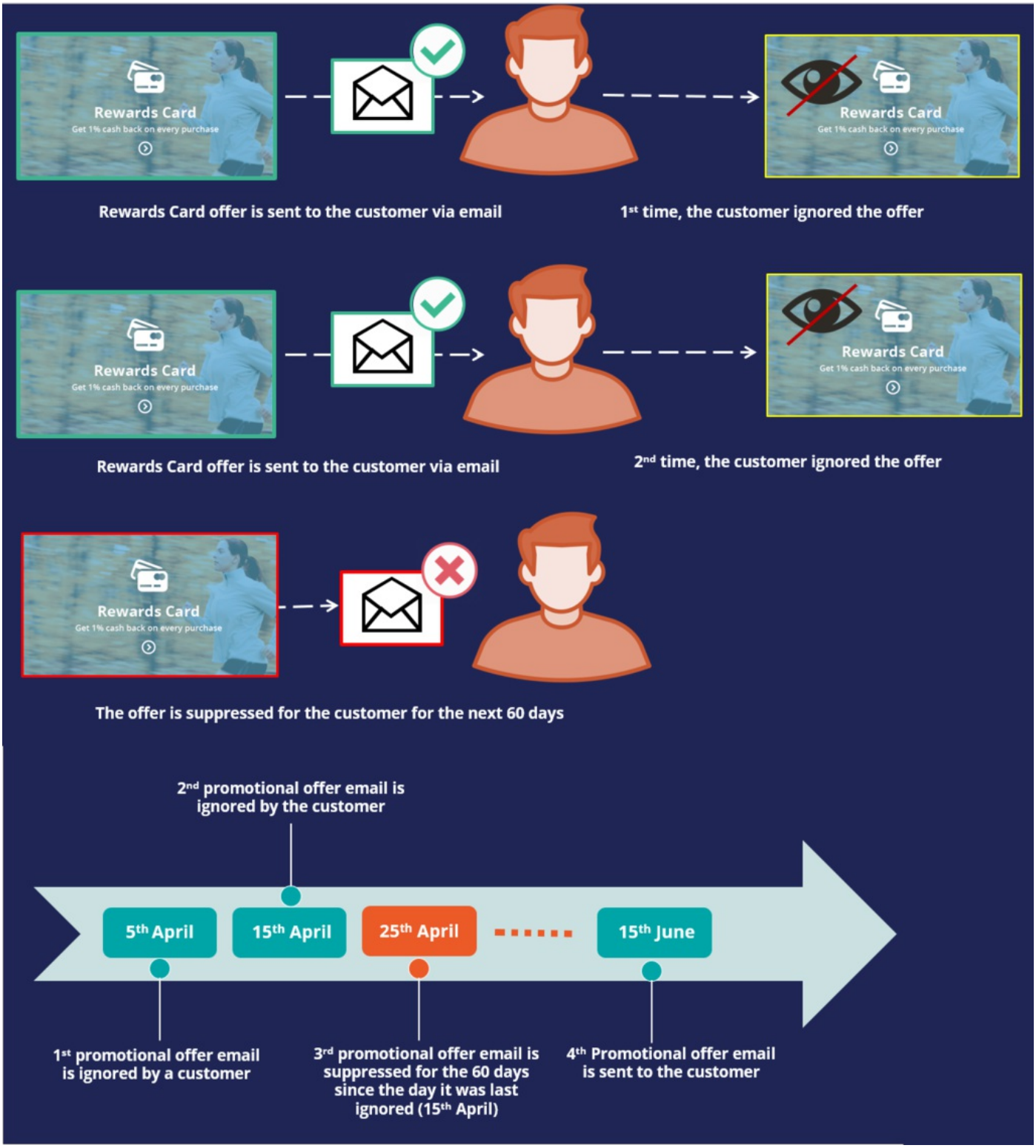


Suppression policies

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.



Contact policy types -- Thu, 07/23/2020 - 01:11
To get the full experience of this content, please visit https://academy.pega.com

Avoid sending the same action for a time period

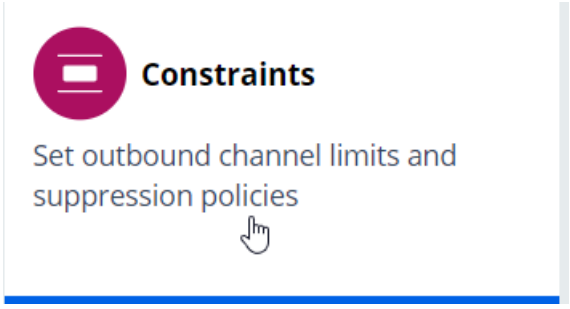
Business scenario

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

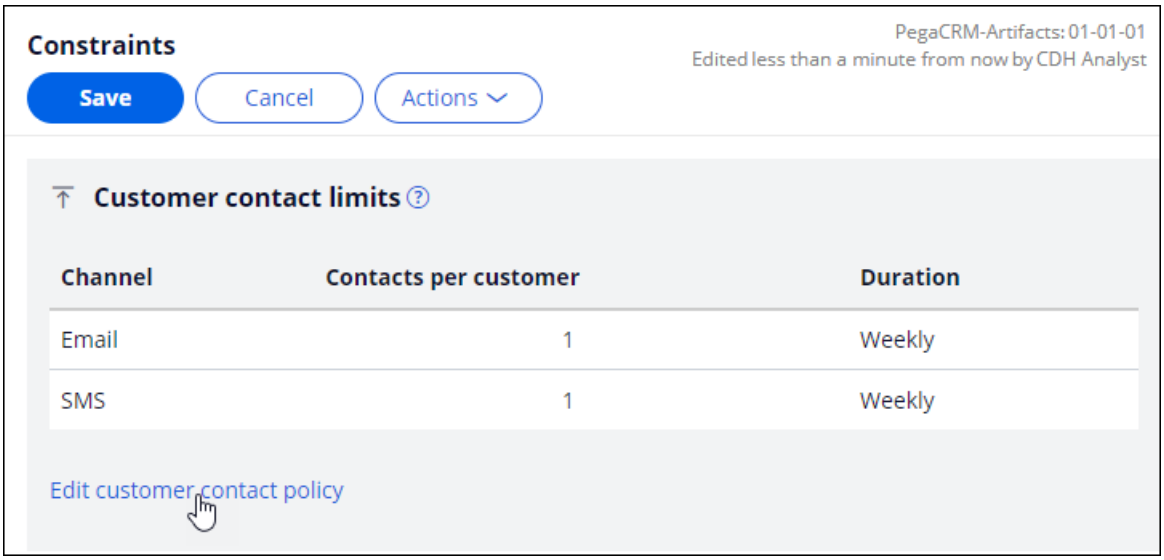
On the Next-Best-Action Designer **Constraints** tab, you configure outbound channel limits and suppression policies.



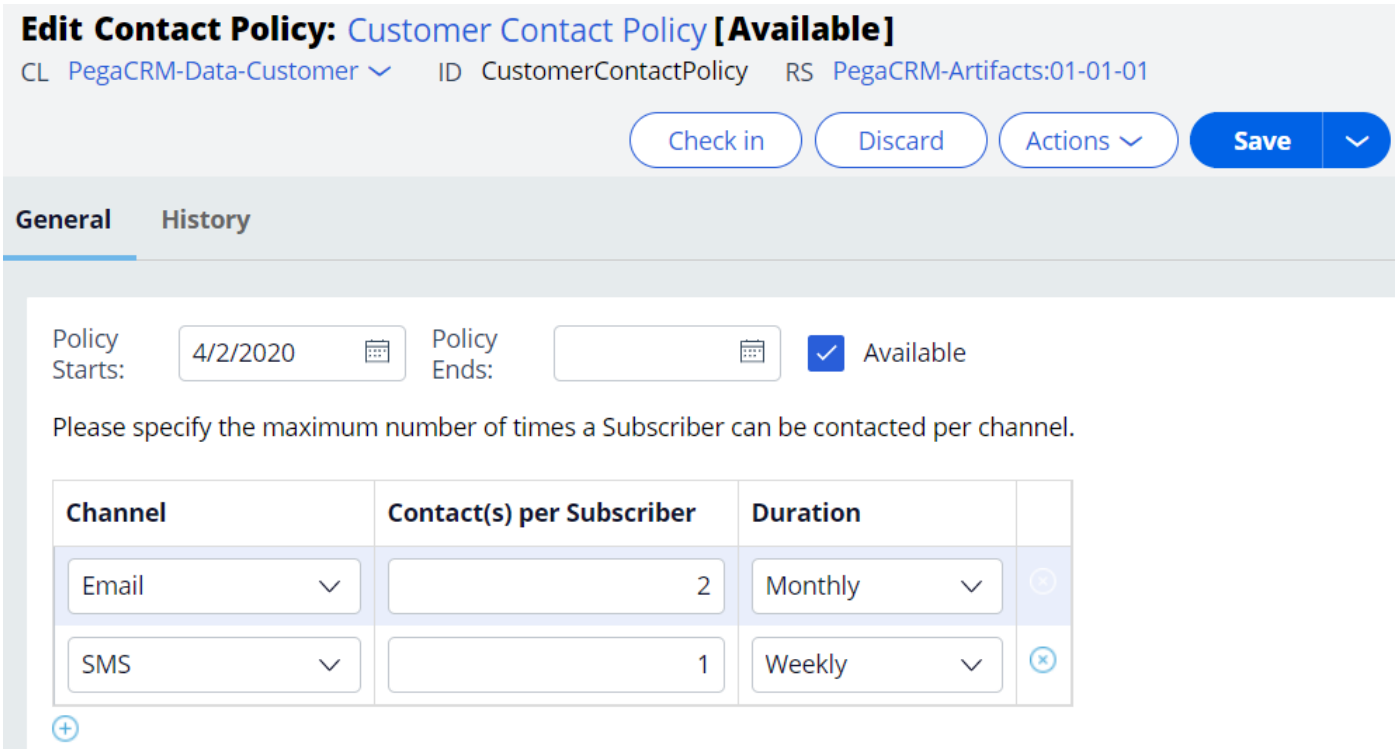
Edit **Constraints** to modify the limits.

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.



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.



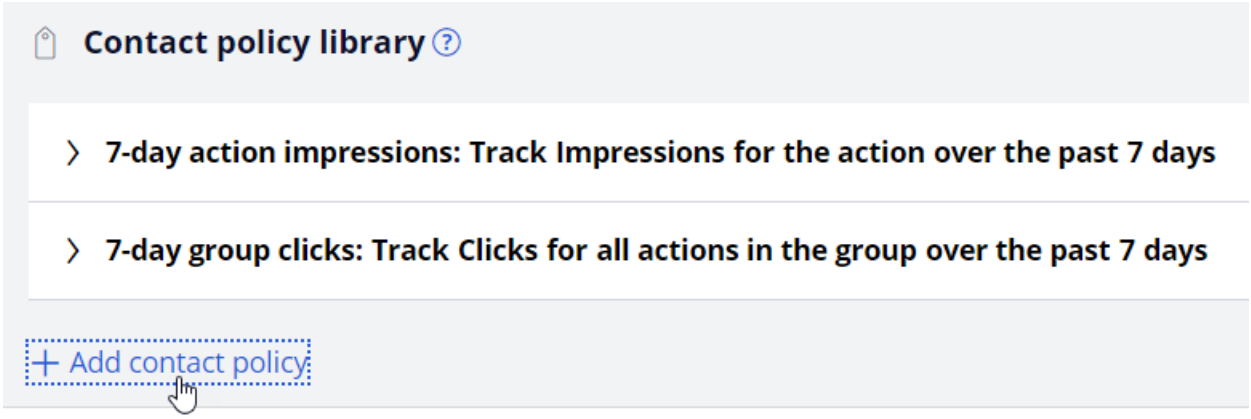
In the contact policy, you can update current limits or add new channel limits.

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.

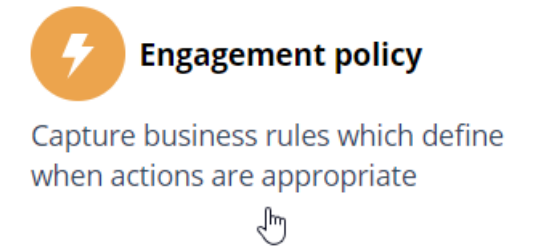
You can select the time period over which the responses should be tracked. In this case, responses should be tracked for the past 30 days.

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.

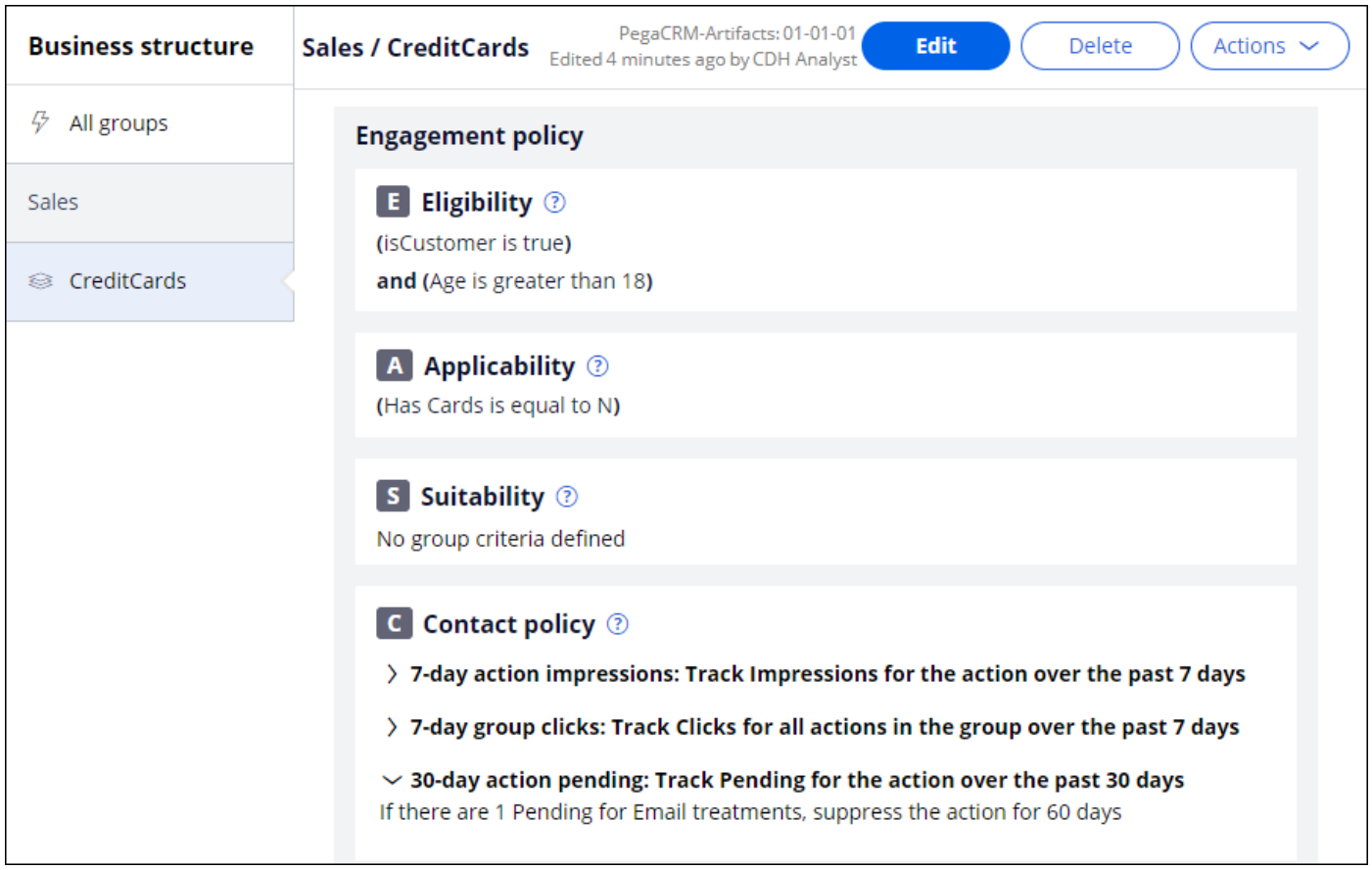
Then you need to select the channel for which the responses are tracked and enter the number of days for which an action should be paused when the suppression criteria are met.

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.



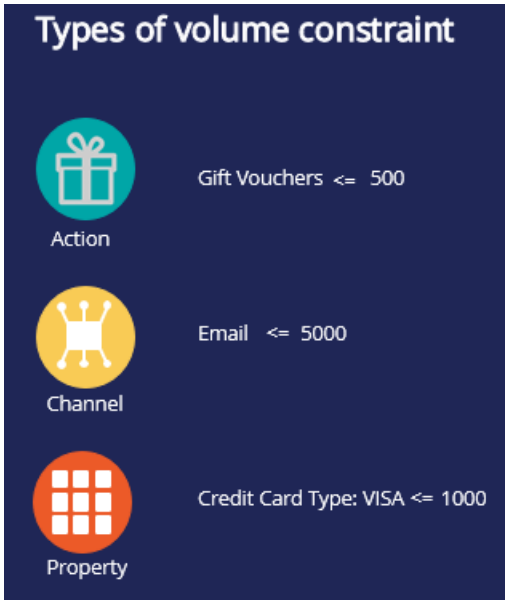
Avoid sending the same action for a time period -- Thu, 07/23/2020 - 01:11
To get the full experience of this content, please visit https://academy.pega.com

Volume constraints

What is volume constraint?

In Pega Customer Decision Hub™ you can limit the volume of actions delivered to customers via always-on outbound.

Volume constraints can be set at an action, channel, and/or property level.



Action constraint: It limits the volume of actions that can be sent to customers.

An action constraint is useful in a situation where you have a limited number of offers that you can send to customers, for example, 500 gift vouchers. Configuring an action constraint would ensure that you do not offer more offers than are available.

Channel constraint: It limits the volume of actions that can be delivered via a certain channel.

You can use channel constraint for channel capacity planning. For example, assume that you have a limited call handling capacity in your inbound call center, for example 1000 calls/day. When you send 50,000 emails on a day on a certain regulatory update, let's say you expect 2%-3% follow-up calls a day from your customers to your contact center, which is about 1000-1500 calls. So, you need to limit the outbound email volume to 35,000 emails a day to match the inbound call center capacity.

Property constraint: By default, actions are grouped under the issue / group business hierarchy. 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 offers by setting a property, CreditCardType, to VISA. Then you can define a constraint to limit the action volume where CreditCardType is VISA.

Volume constraint modes

Within a volume constraint, the combination of constraints can be defined for one or more actions with one or more channel types.

In situations where a customer is eligible for multiple actions, volume constraint can be applied in two different ways depending on the business requirement. That is, you may specify whether volume constraints should be applied to the actions as a group, or whether each action should be evaluated separately.

When volume constraint is evaluated separately, each action is checked for the customer separately against applicable channel and action constraints. A customer receives only one action, the one with the top priority, even though the limits for the other low priority actions, are not reached.The top action is selected based on factors like propensity, action value and business levers defined in Next-Best-Action Designer Arbitration.

For example, let's consider that a customer is eligible for Standard card, Rewards card, and Rewards Plus card credit card actions and Standard card is the top priority card. Also, assume that the remaining volume for all three actions are above zero. When you set volume constraints to check each action, the customer will be presented only with the top priority action, Standard card.

When volume constraint is applied for actions as a group, the highest priority action is selected to apply constraints on the channels and actions, without removing the other actions for a customer. That is, a customer receives all the actions he/she qualifies for, provided that the limit for the top priority action is not reached. The other low priority actions are selected even though the limits for those actions are reached.

For example, let's consider that a customer is eligible for Standard card, Rewards card, and Rewards Plus card actions and Standard card is the top priority action. Also, the remaining volume for Standard card is above zero, but for the other two actions it has reached zero. When you set volume constraints to check actions as a group, the customer receives all the actions.

Evaluating actions as a group allows Pega Customer Decision Hub to present all eligible actions to a customer.

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.

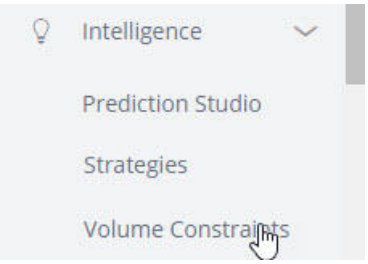
- The IT department has enforced a limit of 1000 emails per week to avoid being classified as spammers. Hence, the marketing team wants to adhere to the limits.
- The bank would like to restrict the number of Standard card offer emails to only 100 per week due to a business requirement.
- The bank would 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

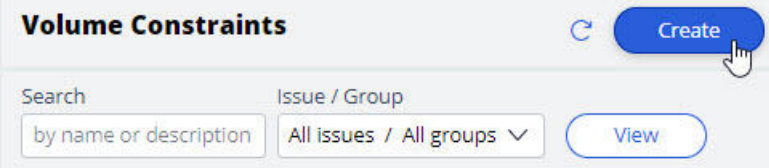
A constraint is an optional configuration element of always-on outbound. When configured, it will limit the volume of actions that will ultimately be initiated.

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.

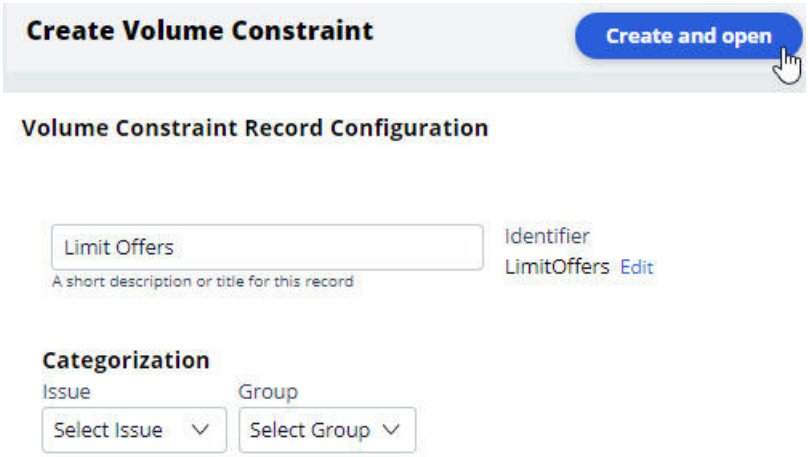


You can add a new volume constraint from this page.



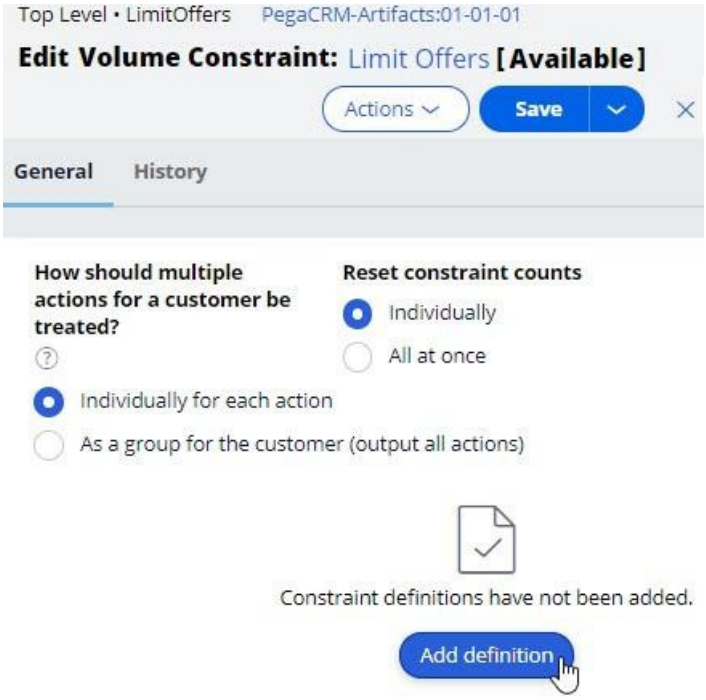
Provide a short description for the new volume constraint and the identifier is automatically populated when you tab out.

The volume constraint must be created at the top level for it to be used in the Next-Best-Action Designer outbound schedule

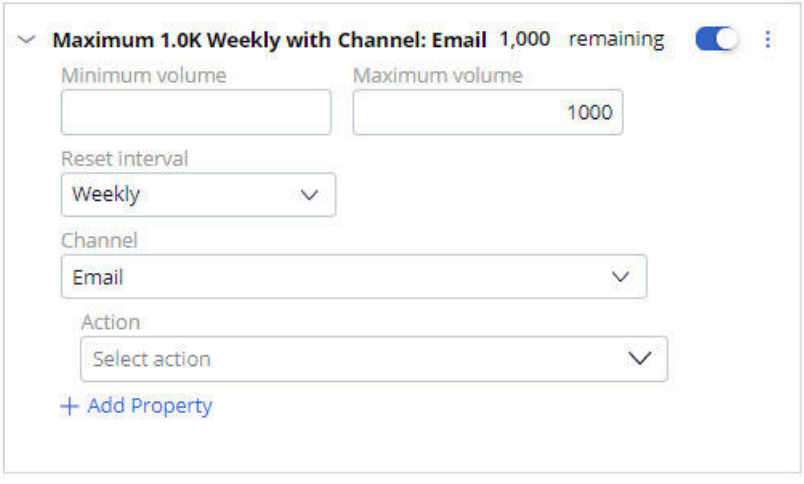


Configuring a volume constraint

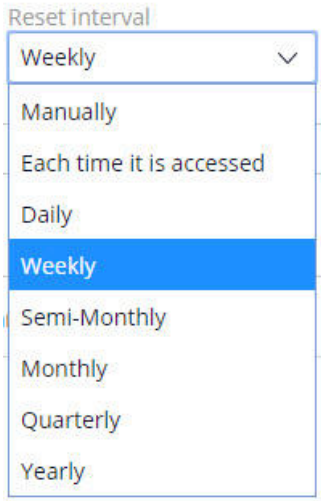
The **General** tab is where you configure the different types of volume constraints.



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 **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.

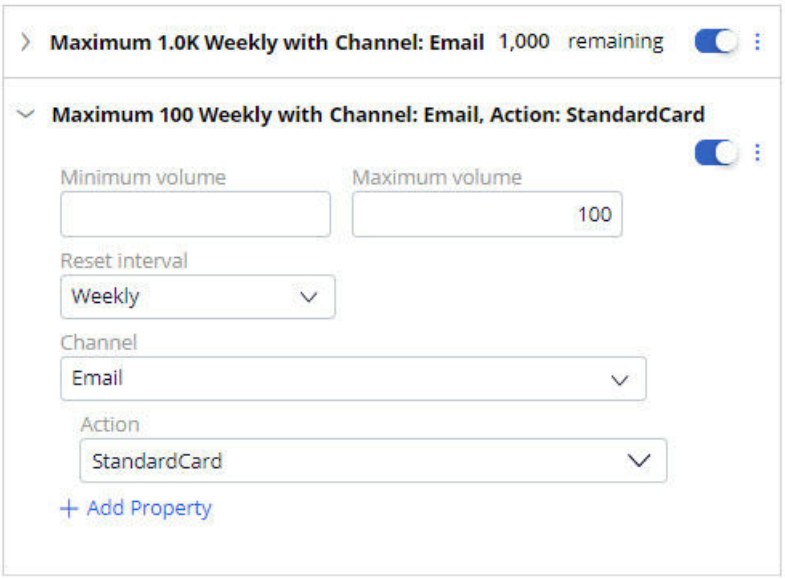


The other options in this list can be used when you want the system to automatically reset the constraint. For example, *Weekly* means that the system will reset the limits the first time they are applied in a particular week, Sunday being considered the first day of the week.

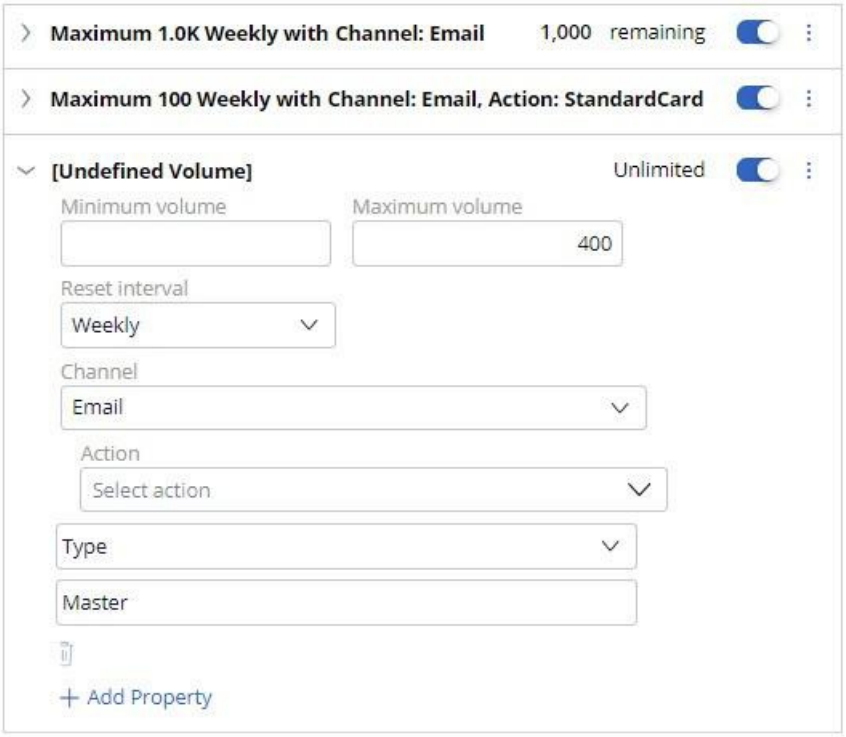
The remaining quota of the volume constraints, if any, in a given reset interval will be used in the following scheduled run. For example, if only 300 Rewards card emails can be sent in a week, the scheduled run might send 250 emails based on action prioritization. The remaining 50 emails will be sent 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.



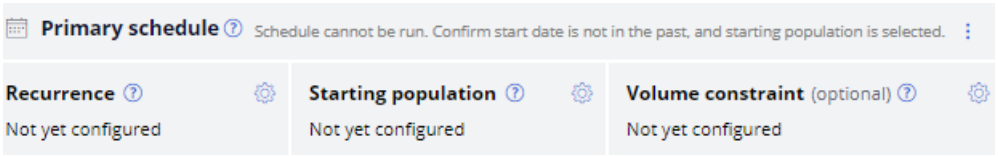
Finally, as per the requirements, add a property constraint that will limit offer emails related to credit cards of type “Master” to only 400 per week.



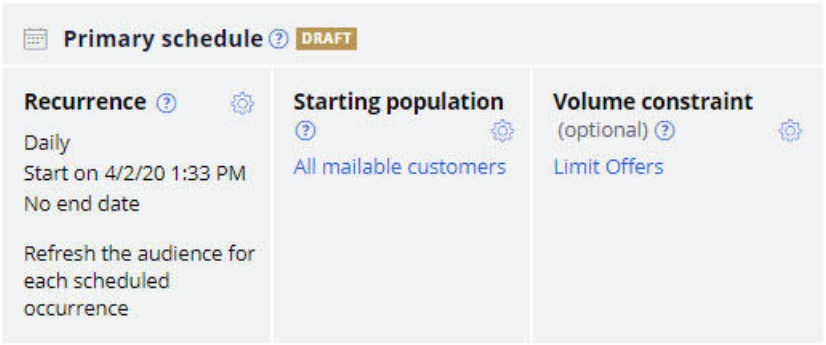
Configuring volume constraint in the Next-Best-Action Designer

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.

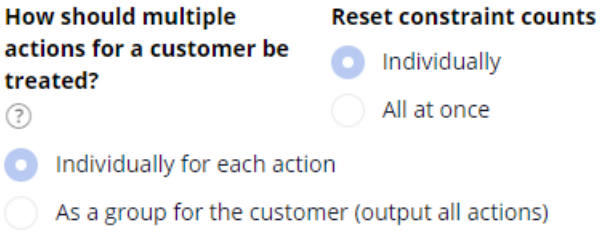


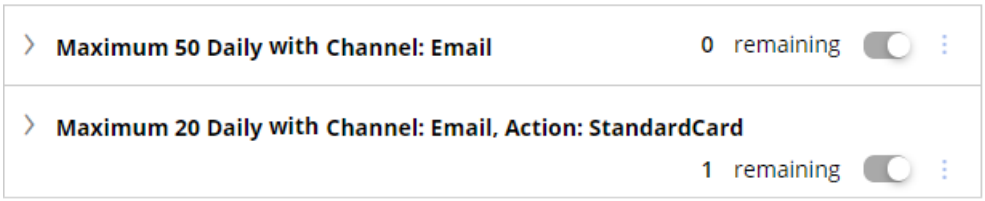
The schedule has two mandatory configurations, **Recurrence** and **Starting population**. To include the volume limits on outbound, you must configure the **Volume constraint**.



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.



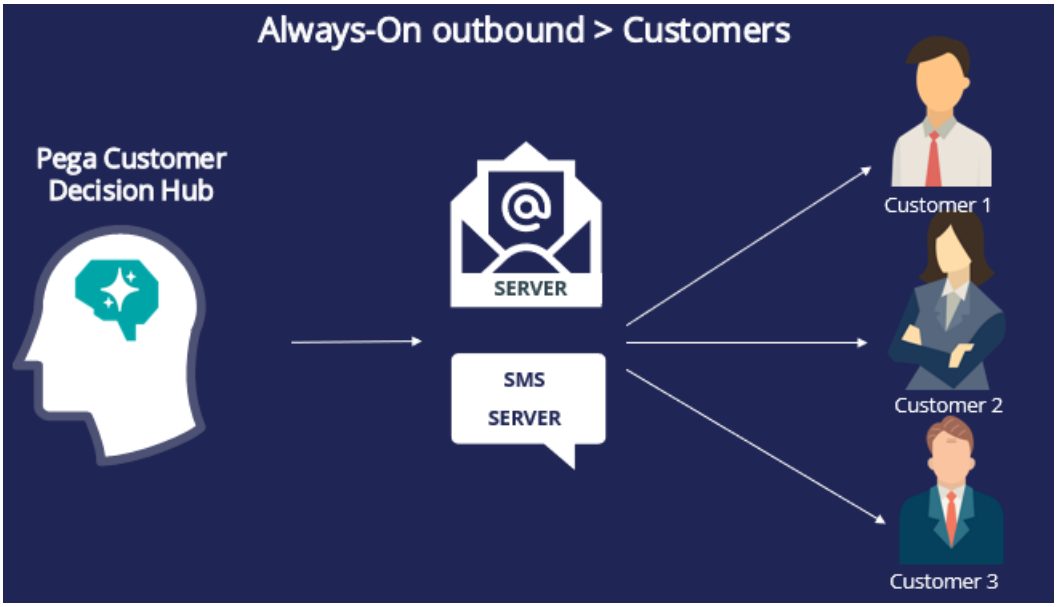


Configuring a volume constraint -- Thu, 07/23/2020 - 01:12
To get the full experience of this content, please visit https://academy.pega.com

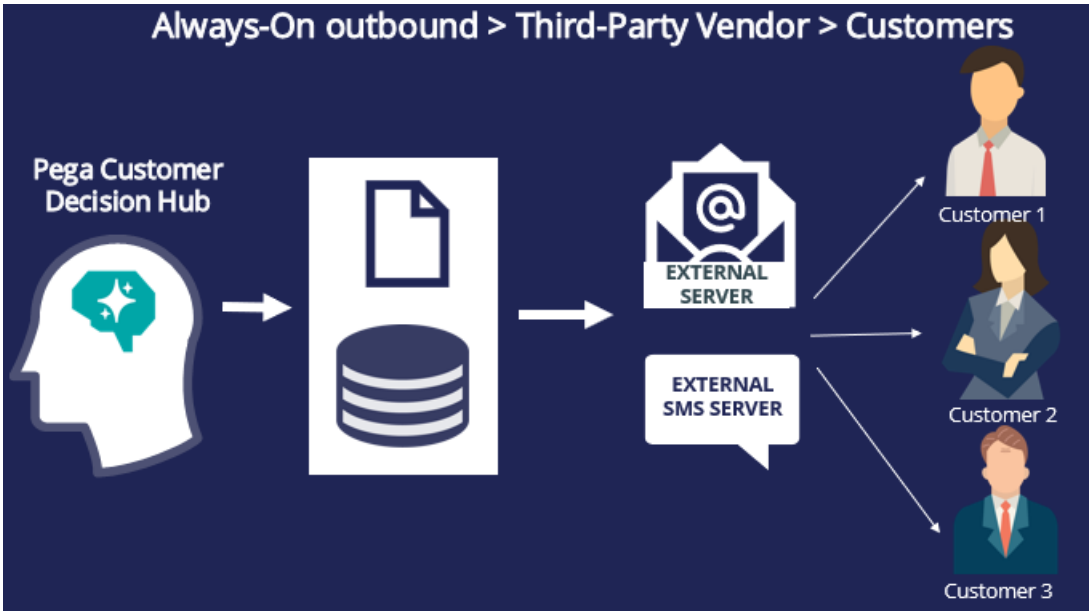
File and database templates

Why do you need output 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.

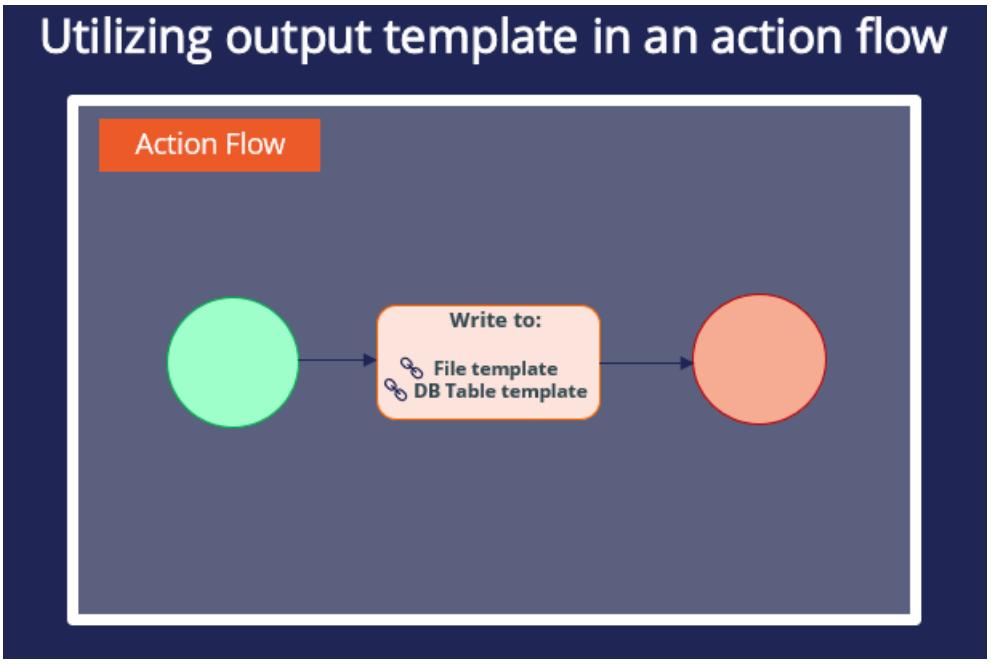


Pega Customer Decision Hub enables you to write the file in one of the following cloud-based destinations as well as the local filesystem.

- JFrog Artifactory
- Amazon S3
- Microsoft Azure

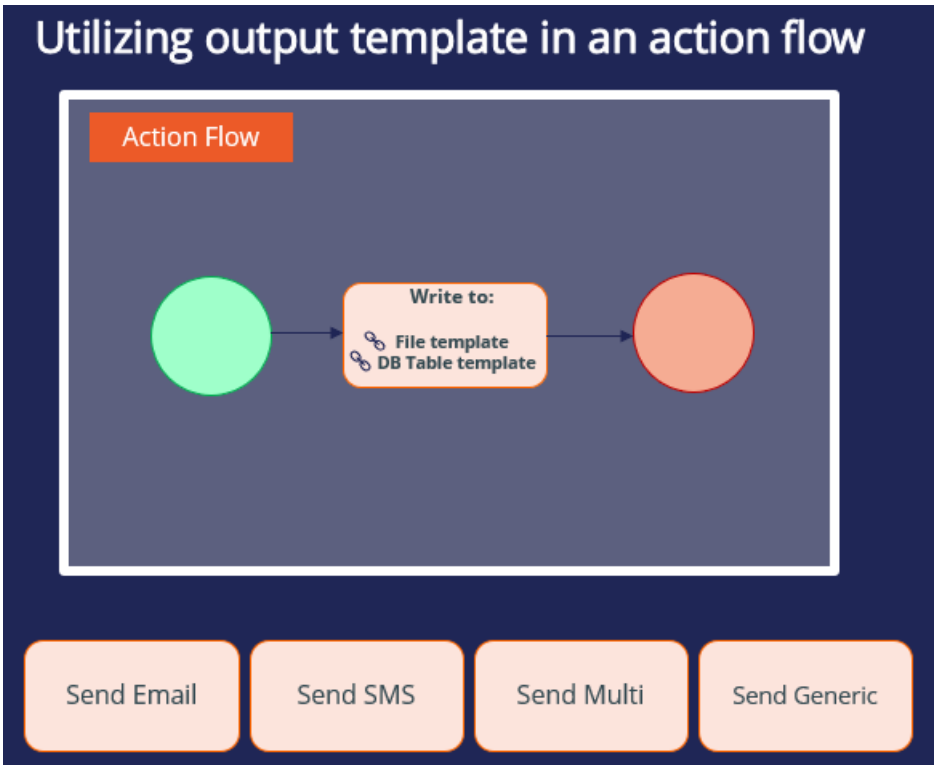
Writing action details to a file or database

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.



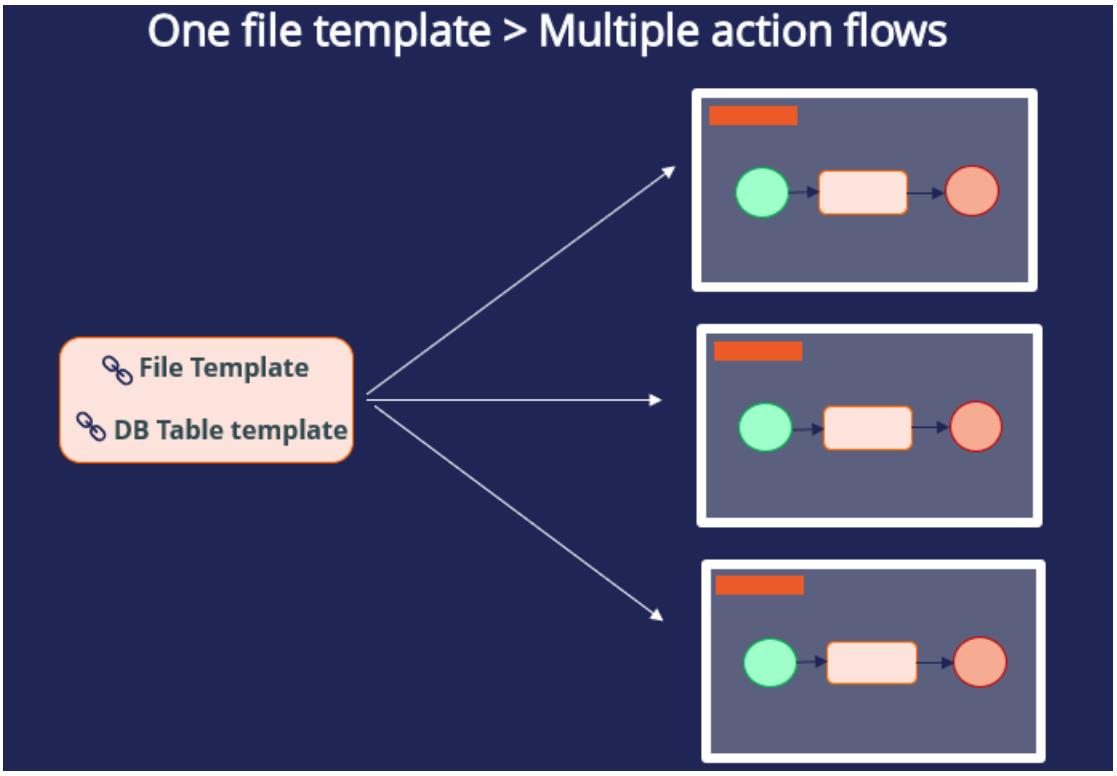
The four available Send shapes that can be used for this purpose are: **Send Email**, **Send SMS**, **Send Multi,** and **Send Generic**. Any one of these shapes can be used to write action details to a file or a database table. However, when you know that the action messages will be sent eventually 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.



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.



File and database templates -- Thu, 07/23/2020 - 01:13
To get the full experience of this content, please visit https://academy.pega.com

Writing action details to a file or database

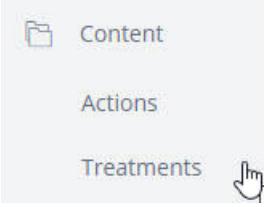
Business scenario

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 distributor.

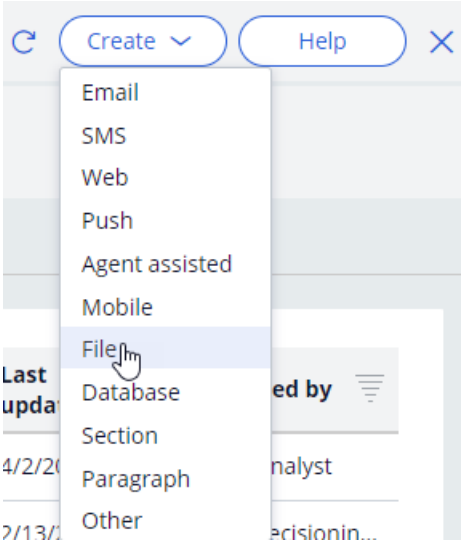
Creating a file template

Pega Customer Decision Hub™ enables you to easily write action details to a file that can be sent to a third-party vendor or application for distribution.

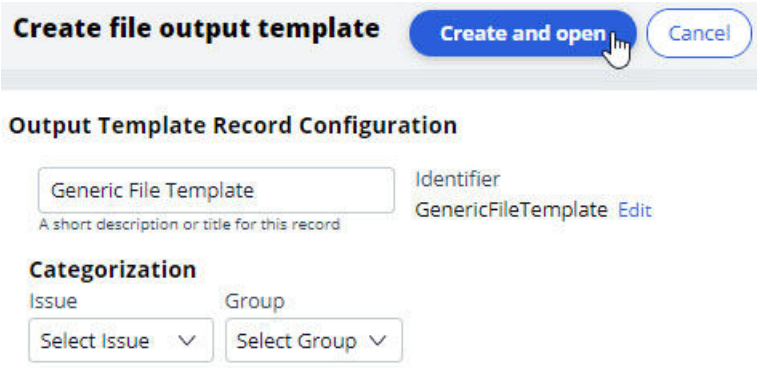
To implement this business requirement, you need to configure a file template. The file template landing page can be accessed from the navigation menu, **Content** -> **Treatments**.



You can create a new file template from this page.



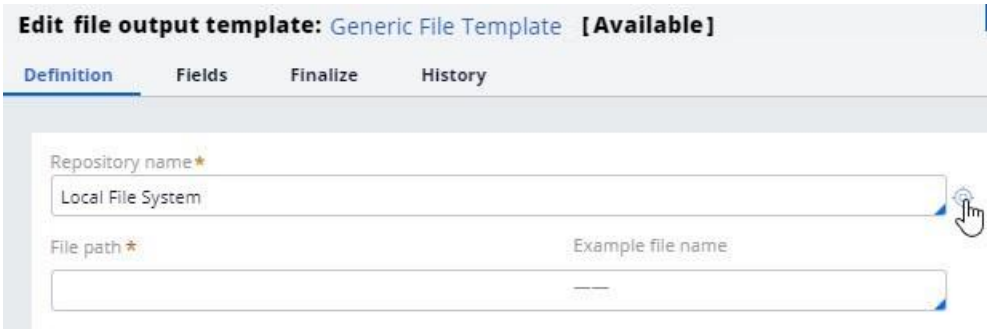
Provide a short description for the new file template and the identifier is automatically populated when you tab out.



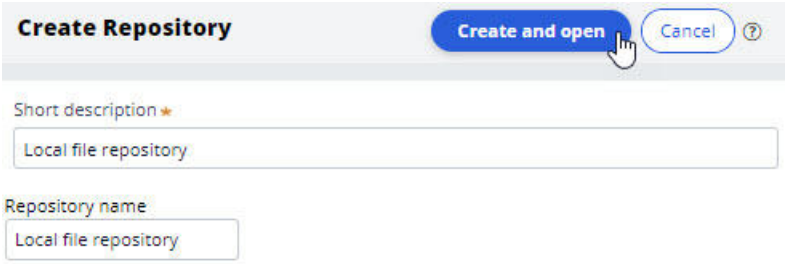
Creating a repository

When configuring a file template, provide details on an existing repository or to create a new repository where the file needs to be stored.

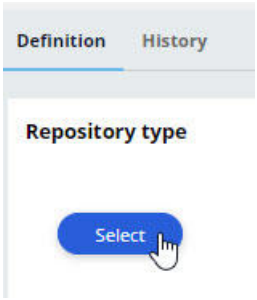
In this case, you will create a new repository.



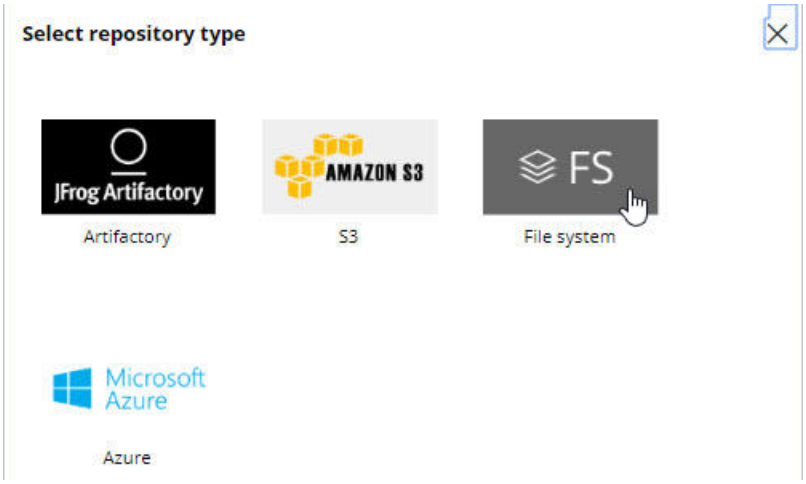
Provide a short description and a name for the repository to create one.



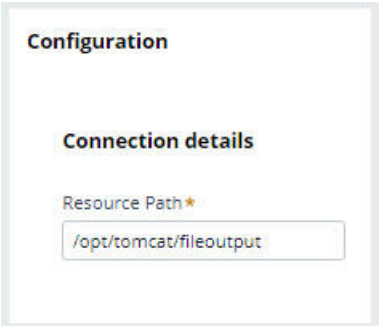
Select a **Repository Type** where you would like to store the file.



Pega Customer Decision Hub supports various cloud-based destinations as repository types such as **JFrog Artifactory**, **Amazon S3**, and **Microsoft Azure** in additional to the local file system.



When you select the **File system** repository type, in the **Resource Path**, provide a directory path in the local file system to where you want to store the file.



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.

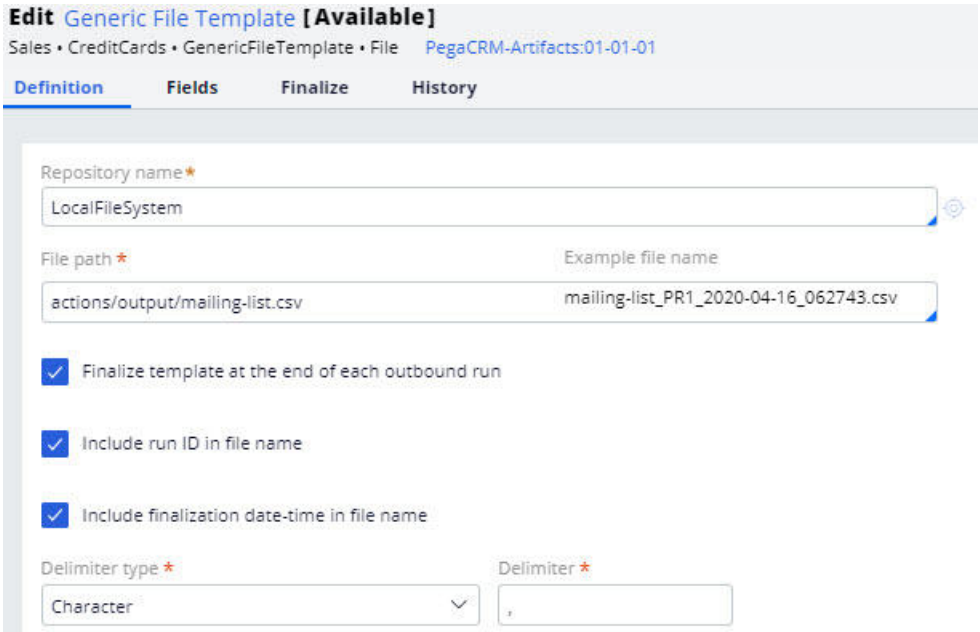
For example, if the repository path is */opt/tomcat/fileoutput* and the file path is *actions/output/mailing-list.csv,* then the file will be created under */opt/tomcat/fileoutput/actions/output.*

Output template finalization

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. Finalization can be triggered manually or automatically by the system based on a schedule. Select the **Finalize template at the end of each outbound run 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 have to configure a separate finalization schedule in the **Finalize** tab or finalize the file manually in the**Treatments** > **File** landing page.

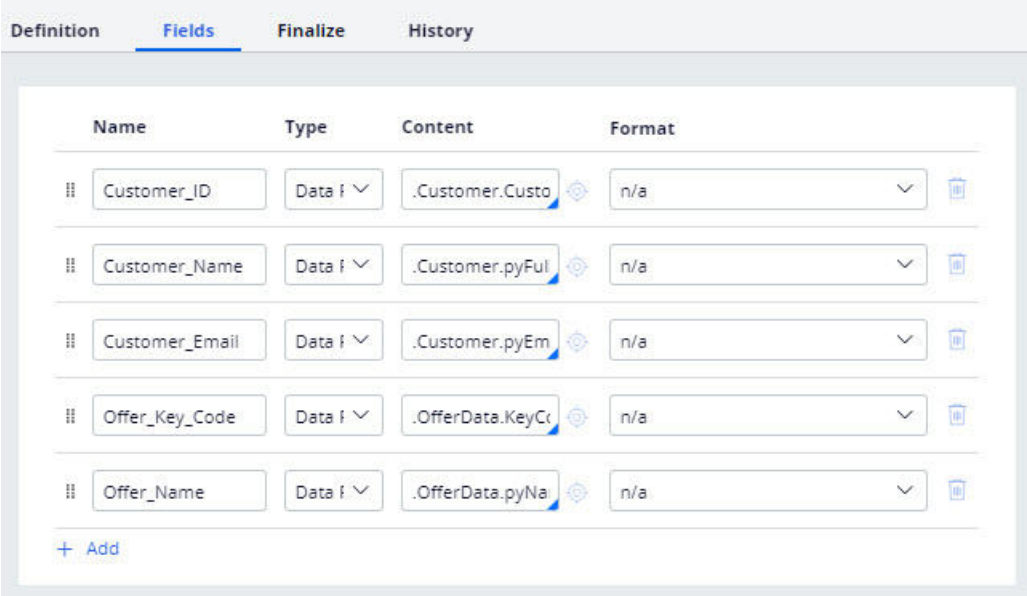
If you chose to finalize at the end of each run, select**Include run ID in the file name** to include the run ID for easier identification of the files generated in each run.



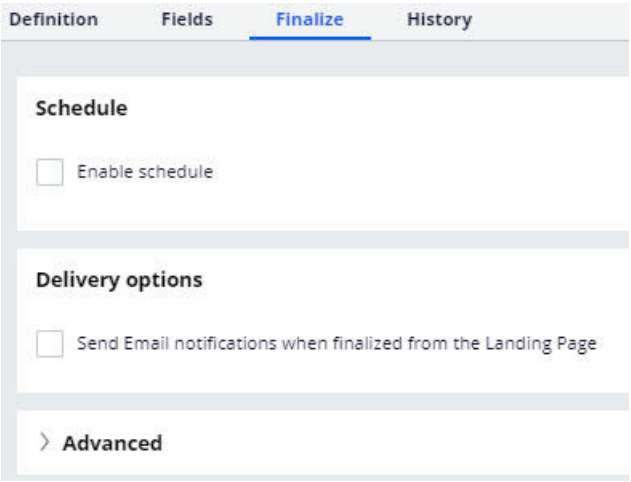
In the **Fields** tab, you can configure the information elements that you want to output.

That is, you can configure the customer and action details that you want to share with the third-party email distributor. 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.



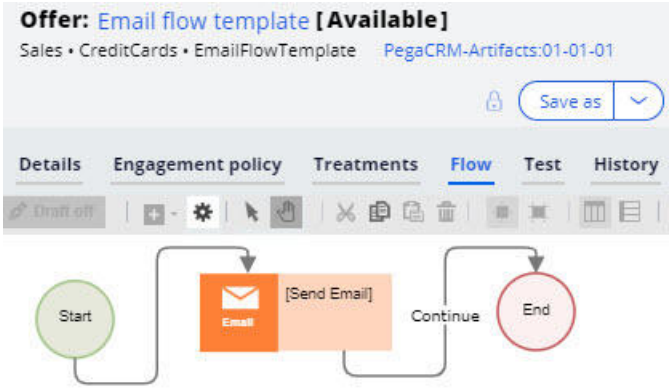
In the **Finalize** tab, you can configure further finalization options.



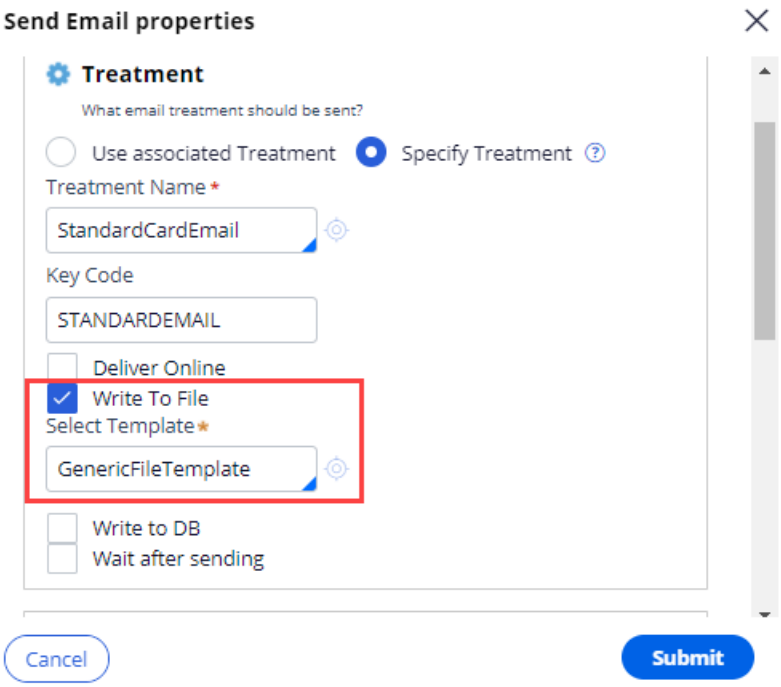
The **Advanced** section can be used for additional processing during the finalization process. For example, you can use it to initiate the file transfer to a particular location.

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**Specify Treatment** option. Now, select a treatment and reference the file template in the **Write To File** option.



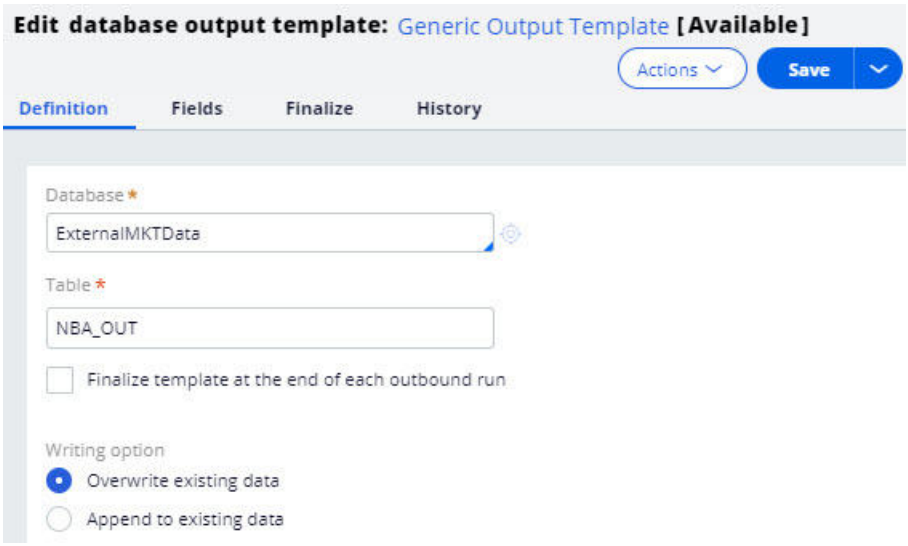
To write action details to a file, run an outbound schedule. The file with action details will be stored in the specified repository.

```
[root@ip-172-31-20-32 output]# wc -l mailing-list_PR12_2020-04-16_122456.csv
42 mailing-list_PR12_2020-04-16_122456.csv
[root@ip-172-31-20-32 output]# more mailing-list_PR12_2020-04-16_122456.csv
Customer_ID,Customer_Name,Customer_Email,Offer_Key_Code,Offer_Name
CON-894,Adrienne Marks,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-752,Angus Schultz,customer1@enablement.com,STANDARDCARD,StandardCard
CON-539,Antwon Graham,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-220,Freeman Schinner,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-306,Deondre Predovic,customer1@enablement.com,STANDARDCARD,StandardCard
CON-245,Jacey Hilpert,customer1@enablement.com,STANDARDCARD,StandardCard
CON-880,Lourdes Rohan,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-613,Roscoe Skiles,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-123,Mose Macejkovic,customer1@enablement.com,STANDARDCARD,StandardCard
CON-986,Delta Toy,customer1@enablement.com,STANDARDCARD,StandardCard
CON-79,Madge Renner,customer1@enablement.com,STANDARDCARD,StandardCard
CON-347,Pearlie Collier,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-766,Nelle Davis,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-367,Eleanora Larson,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-140,Ali Emmerich,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-338,Melvina Ledner,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-107,Pete Kerluke,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-916,Haskell Cummerata,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-141,Reva Koelpin,customer1@enablement.com,STANDARDCARD,StandardCard
CON-186,Clay Leffler,customer1@enablement.com,STANDARDCARD,StandardCard
CON-610,Claude Runolfsdottir,customer1@enablement.com,STANDARDCARD,StandardCard
CON-719,Aliza Doyle,customer1@enablement.com,STANDARDCARD,StandardCard
CON-688,Gwen Mertz,customer1@enablement.com,STANDARDCARD,StandardCard
CON-830,George Boyle,customer1@enablement.com,STANDARDCARD,StandardCard
CON-336,Kale Franecki,customer1@enablement.com,STANDARDCARD,StandardCard
CON-479,Elna Hessel,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-405,Glenna Emard,customer1@enablement.com,STANDARDCARD,StandardCard
CON-111,Dorcas Schumm,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-508,Madelyn Carter,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-804,Ahmad Gleason,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-939,Garrison Brekke,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-118,Joanny Pfeffer,customer1@enablement.com,STANDARDCARD,StandardCard
CON-519,Destini Vandervort,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-587,Caden Dibbert,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-528,Kaleb Aufderhar,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-163,Oran Beer,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-493,Trace Zboncak,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-654,Antonio Walsh,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-518,Germaine Schamberger,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-571,Charlie Hoppe,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
CON-938,Cristina Satterfield,customer1@enablement.com,REWARDSPLUSCARD,RewardsPlusCard
[root@ip-172-31-20-32 output]#
```

The file contains the fields as configured in the file template.

Writing action details to a database

Similar to writing details to a file, you can make the outbound run write details to a database table using a database template. 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.



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**. Under 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.

Here, you can configure only database templates that have the Writing option set to Append to existing data.