Kore.ai

Experience Optimization (XO) Platform

Guide

Platform

7380 W Sand Lake Rd, Orlando, FL 32819

Table of contents

Introduction

The Kore.ai Experience Optimization (XO) Platform gives enterprises every necessary component to clear the hurdle of "how" when it comes to designing, building, testing, and deploying AI-powered virtual assistants & process applications. This one-stop solution provides a scalable, secure, and repeatable foundation to create experience optimization solutions that automate business interactions, perform a wide range of intelligent tasks, communicate with people, systems, and things via award-winning NLP, and leverage data to drive actionable employee and customer engagements across channels.

The Platform includes the following major components:

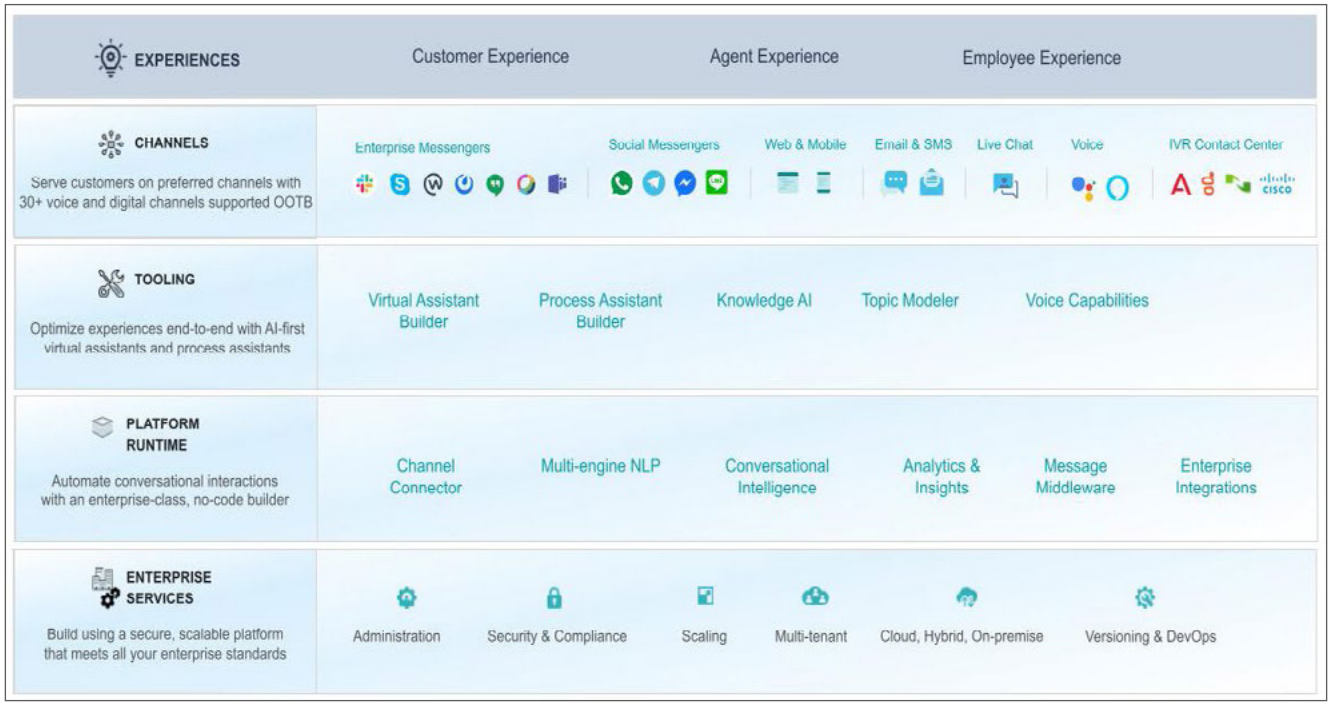
- Virtual Assistants
- Process Assistants
- Data Tables

The Kore.ai XO Platform's robust and flexible architecture can be deployed on cloud or on-premise, ensuring it addresses the most complex IT and business requirements, without the need for new infrastructure or sweeping changes to existing systems, policies, or procedures. This document provides a more detailed look at the architecture and major components of the platform, and how enterprises can utilize each piece to elevate and extend the functionality of current systems, reach more customers, make work easier for their employees, switch from GUI to conversational UX, and shorten time to value.

The Kore.ai Platform Guide is primarily written for citizen developers, solution architects and other technology enthusiasts who are looking for information on the frameworks and components used in the Kore.ai Experience Optimization (XO) Platform. This guide has details of every component needed to design, build, train, deploy and manage AI-rich virtual assistants and develop process assistants to automate critical business processes. This guide can benefit any technologist who wishes to explicitly understand the building blocks of the robust platform.

XO Platform Architecture

The architecture diagram below provides a visual representation of the major components and tools of the Kore.ai Experience Optimization (XO) Platform:



The enterprise-grade no-code platform blends Conversational AI and Digital UX. The primary components of the platform are the Virtual Assistant Builder and Process Assistant Builder.

- The virtual assistant builder caters to the various phases of the virtual assistant development life cycle. It is used to design, build, train, test, deploy and manage virtual assistants very rapidly. Knowledge AI capability extracts structured and unstructured data automatically from various documents thereby saving the effort of data ingestion.
- The process assistant builder lets you build multi-user, multi-step workflows to automate your critical business processes. The drag and drop interface with advanced tools simplifies the development. It offers multiple trigger options to use at your convenience. Accommodate complex business logic, multi-level approval flows, and build dynamic digital forms using the comprehensive tools offered by the platform.

The platform connectors can be integrated to the backend system for exchanging information about customers, products, transactions, or any other relevant information. This information shall be used by the virtual assistant and process assistants to accomplish user requests. Message middleware takes care of the end-to-end encryption of data.

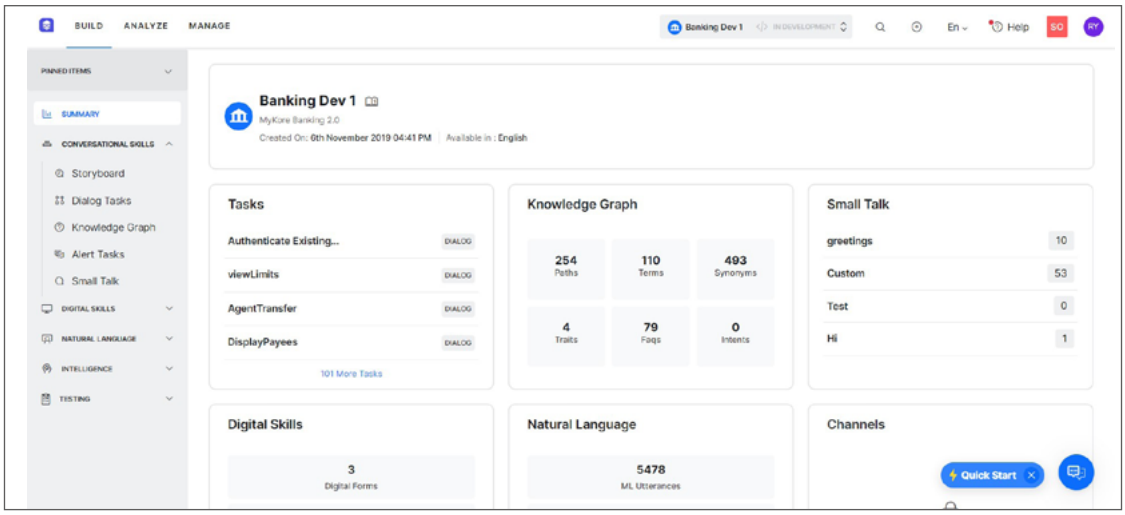
The NLP capability of the platform enables it to process the user conversation and identify the intent, extract the entity and process the user's request. The platform also has an intelligence framework to understand the user's context to handle complex conversations involving multiple topics and to personalize the interactions.

The enterprise integration framework is used to handle orchestration, authentication, transportation, and mediation for applications that may consist of multiple communication protocols or technologies. Bot SDKs can be used to embed Kore.ai's widget into customer applications to enable end-users to interact with their application. Also, the API-enabled service provides the interfaces that are defined and leveraged in the interactions between platform components.

The platform also provides other required additional components to make the virtual assistants scalable and secure. These components can be used for provisioning and restricting access to virtual assistants, managing deployments, and taking advantage of real-time actionable insights.

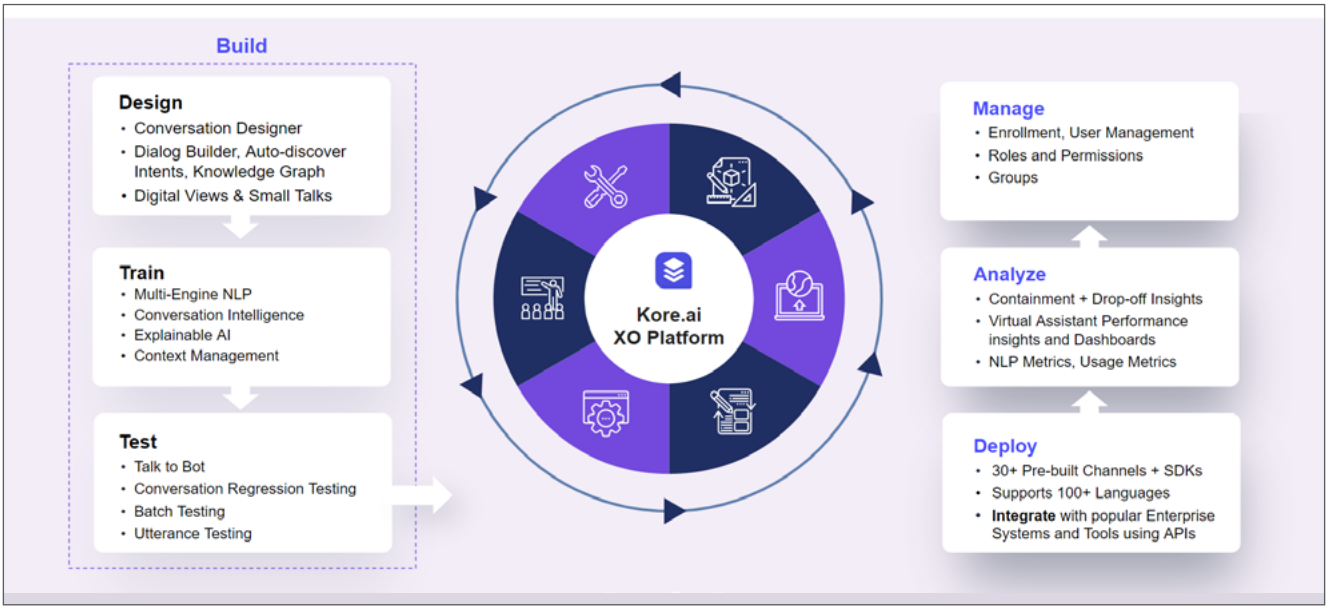
Virtual Assistant Builder

The Kore.ai Virtual Assistant Builder tool provides designers with everything needed to rapidly design, build, test, train, and deploy intelligent virtual assistants that seamlessly integrate with existing business systems. The Kore.ai VA Builder offers flexibility to design, agility to make changes, and functionality to scale quickly to achieve speed to market without the need for code, custom software, significant server space, or major changes to your infrastructure.



SDLC Support

The Kore.ai XO Platform provides full SDLC support, including support for development lifecycles across multiple teams with multiple release dates. Within the platform, there are modules defined to address the whole lifecycle from requirement gathering, conversation design, dialog building, natural language training, publishing, versioning, deployment, approvals and more. The platform gives clients all the tools needed to efficiently maintain virtual assistants before rolling them out to employees or customers and to deal with version control and updates over time. The platform also provides the ability to manage a development lifecycle across various Dev, Test, and Production environments.



Universal Virtual Assistants

Kore.ai's Universal VA facilitate a scalable, modular approach to VA building by helping you link several VAs into one.

Universal VA is a container bot that can be linked with one or more Standard VAs. When a user interacts with the Universal VAs, it routes the user utterance to the appropriately linked VA for intent detection or task fulfillment. Clustering. Users can then export the required topics, intents and utterances to generate flows. Users can select and use these utterances for ML training within the platform.

Kore.ai XO Platform Components

Design

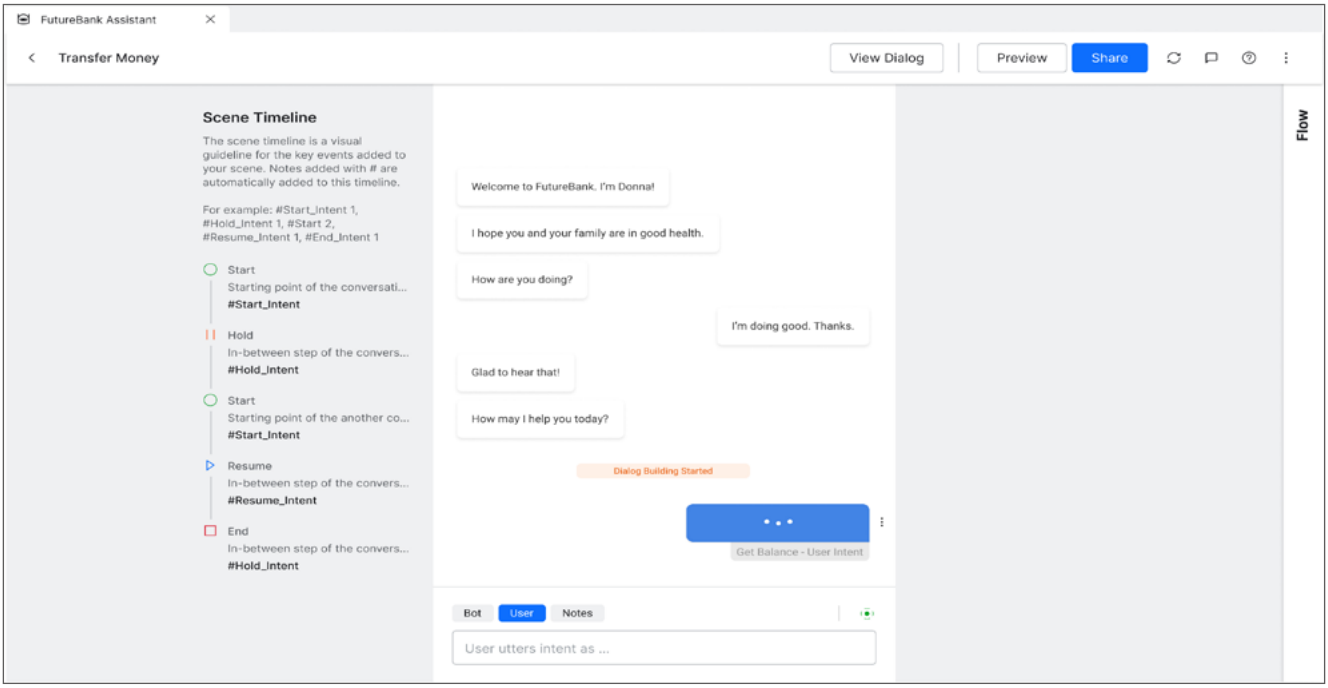
Topic Modeler

The virtual assistant modeling uses transcripts from past chat and voice conversations to uncover customer interaction patterns. Enterprises leverage this data to model virtual assistants. This approach offers enterprises significant advantages compared to the traditional method of developing virtual assistants.

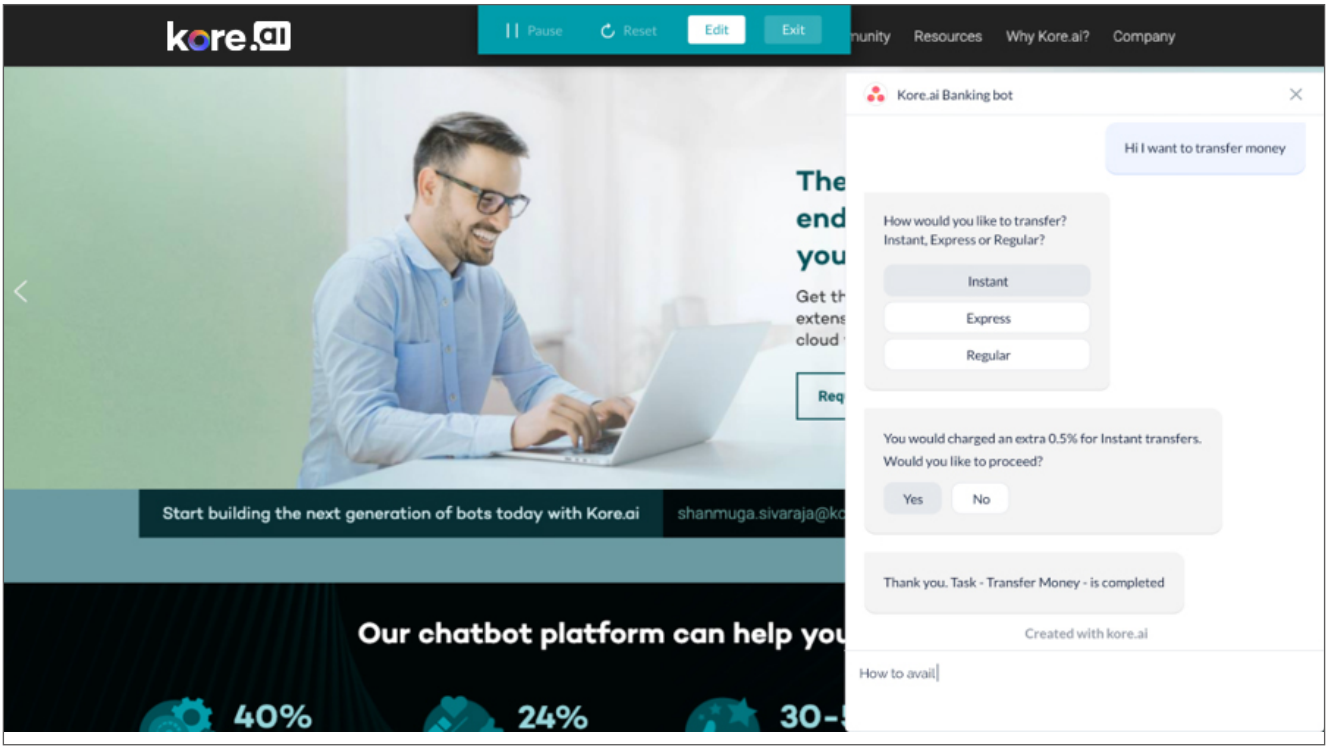
Ingest the data into the Kore.ai XO Platform; it analyzes and cleanup and executes pre-processing activities. Then the intelligent ML toolbox recognizes the topics, intents and entities through multiple methods - Named Entity Recognition and Unsupervised Text Clustering. Users can then export the required topics, intents and utterances to generate flows. Users can select and use these utterances for ML training within the platform.

Conversation Designer

Conversation Designer is the simplest and most natural way to design human-like virtual assistants. Kore.ai XO Platform's Storyboard is an intuitive conversational designer that simplifies the virtual assistant development process.



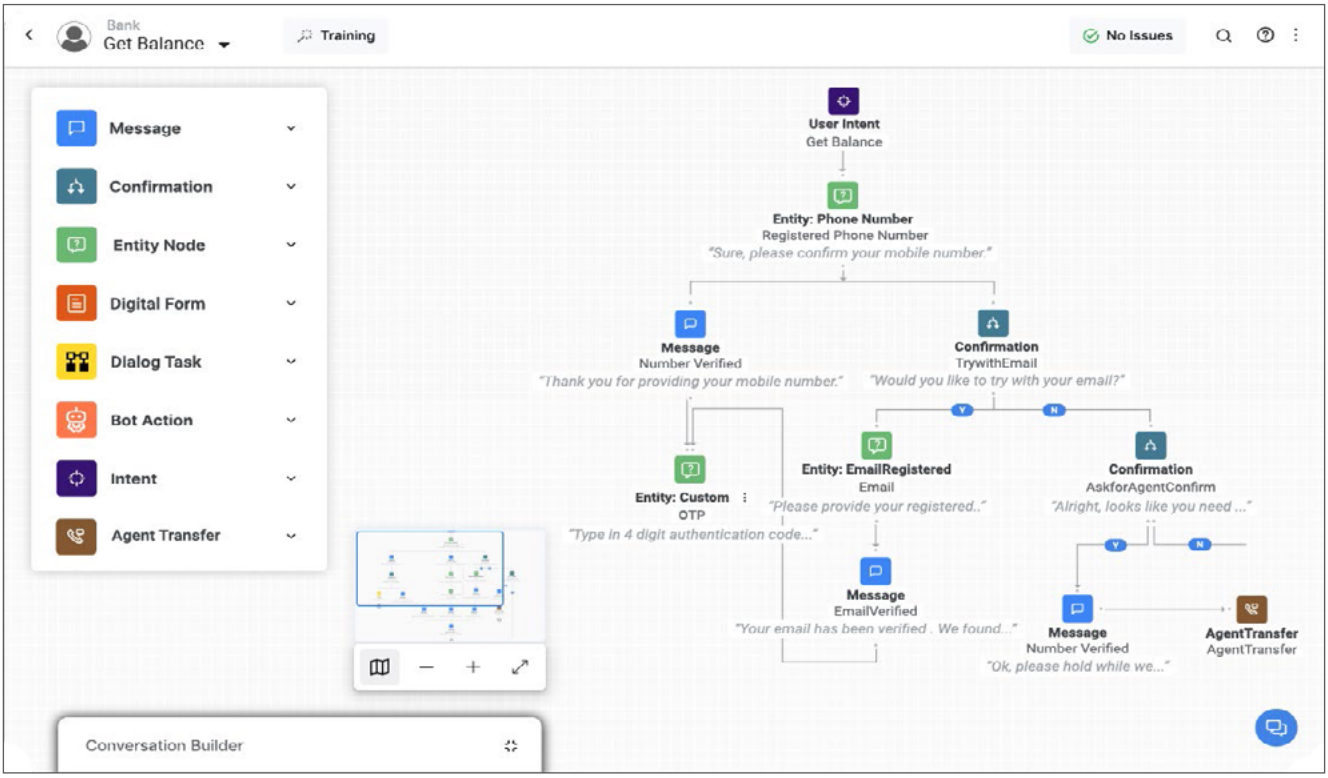
You can capture complex business requirements, build mock scenes and conversation flows. Improve the look and feel of virtual assistant responses and enhance user engagement with templates and customization options. The flow view lets you examine the non-linear conversational paths. Further, preview to see how the conversation flows on your website.



Conversation-driven Dialog Builder

Creating complex virtual assistants can be made easier with a visual tool. The Kore.ai Conversation-driven Dialog Builder gives designers and citizen developers a simple visual interface to create complex conversational flows and decision trees between users and virtual assistants. This section describes how dialogs can be constructed within the tool to support robust and complex conversations and workflows.

The platform adopts a conversation-driven approach. Conversation designers or business analysts feed the user-assistant conversations; the platform automatically converts the conversations into executable Dialog Task flows. The 2-way synchronization reflects the changes in conversation to the flow and vice-versa. Citizen developers, designers and linguistic experts can collaborate to work on a common goal.



Dialog Tasks

Overview

Dialog tasks are advanced tasks that are designed with logic-driven business processes and pre-established business workflows. Virtual assistants key off of the primary request intent to accomplish the task at hand, but can also understand and consider sub-intents to proactively suggest additional value-added steps and workflows. Dialog tasks represent a full conversational flow between a user and a virtual assistant with a network of nodes that are connected using conditions. The Dialog builder provides multiple node types to create fully customizable conversational flows:

- User Intent
- Entity
- Forms
- Logic
- Message
- Confirmation
- Bot Action
- Service
- Script
- Agent Transfer
- WebHook

The platform also provides the ability to code JavaScript nodes based on business logic when access to services via APIs is unavailable. Learn more about the nodes **here.**

User Intent

User intent is a typical task that you create to resolve one primary user request. You configure the different nodes and add them logically to identify and execute user intent efficiently.

Entity

Entities are the fields, data, or words you designate, these are necessary for a virtual assistant to execute the user's request. An entity could be a date, a time, a location, a description or any number of things.

Out-of-the-box the Kore.ai XO Platform supports variety of entities, here is the list -

- Address
- Airport
- Attachment
- Email
- City
- Country
- Company
- Color
- Custom
- Currency
- Composite
- Date
- Date Period
- Date Time
- Description
- List of Items (enumerated)
- List of items (lookup)
- Location
- Number
- Person Name
- Percentage
- Phone Number
- Quantity
- String
- Time
- Time Zone
- URL
- Zip Code

Confirmation

Confirmation Node allows you to prompt the user for a 'yes' or 'no' answer. It helps verify or allow users to accept or decline a choice.

The confirmation node's conditional transitions go beyond the If-Else If-Else expressions. The transition depends on user reply: assertion (yes) or negation (no). The Else condition comes into play if their answer isn't both.

Message

The Message Node delivers a message to the user. Message nodes commonly follow an API, web service call, or webhook event to define their results as a formatted response.

You can format virtual assistant responses as below:

- **Plain text:** Type a message in plain text. You can use stored context values with {{variable brackets}}.
  - For example: 'Hello {{context.session.UserContext.firstName}}. How can I help you?'
- **Javascript:** Compose JavaScript responses either:
  - For dynamically constructing messages using context and session variables.
  - To display templates or widgets that support various channels.

Service Node

The Service Node is a component type in a dialog task that you can add an API service to make REST or SOAP requests to third-party web services. You can use this node when you have the parameters needed for the API request using an entity or other nodes to capture the user input.

You can define the Service Type as:

- **Custom Service –** Define an API request to a third-party web service. It is the default setting.
- **HTML to Image –** Define HTML to render as an image using JavaScript.
- **URL to Image –** Define a web page URL to load to render an image.
- **Custom Authentication Service –** Define a URL to a third-party application that provides the authentication services necessary for the task flow.
- **Alert Subscription Service –** Define contextually relevant alerts to be sent proactively to the user as a part of the dialog journey.
- **Data Table Service –** Define CRUD operations to query and manipulate the data for any given data table/table view assigned to the bot.

Webhook

The WebHook node lets you subscribe to a message and WebHook events to capture and show real-time data from the events. Also, use it for server-side validation, execute business logic, or make backend server API calls.

Agent Transfer

The **Agent Transfer Node** transfers the communications from the virtual assistant to a live agent. You can use the Agent Transfer node in a conditional dialog flow to move a user to a conversation with a live agent.

When a user conversation transfers to an agent using the Agent Transfer node, the dialog task ends and controls the remaining conversation. The Kore.ai XO Platform supports integration with popular softwares using the Agent Transfer node.

Prompt Editor

After creating an entity, question, or message node in the Dialog Builder, Kore.ai lets you modify the default message displayed to the end-user and the prompt used to collect the values for the node, add new prompts or messages, and add channel-specific prompts or messages. It supports two types of user prompts:

- Standard: The prompt defined when adding a node in Dialog Builder is the standard, or the default prompt. When multiple standard prompts are defined for a node, the Platform chooses a random one to display to the end-user.
- Channel Specific: Optionally define user prompts for specific channels such as email, SMS, Twitter, and more. Different widgets can be used based on channels, such as a slider for numbers, date and time picker, and range selectors.

The prompt editor uses three tabs to offer the following features:

- Simple: Basic editor controls and HTML to define and format the user prompt. Enter text, and then format it using buttons for bold, italics, header styles, hyperlinks, ordered and unordered lists, and inserting lines.
- Advanced: User JavaScript code to define, format, and render the user prompt.
- Preview: View a sample of the rendered output message, with markup, that is displayed to the end-user.

Conditional Connectors for Component Transitions

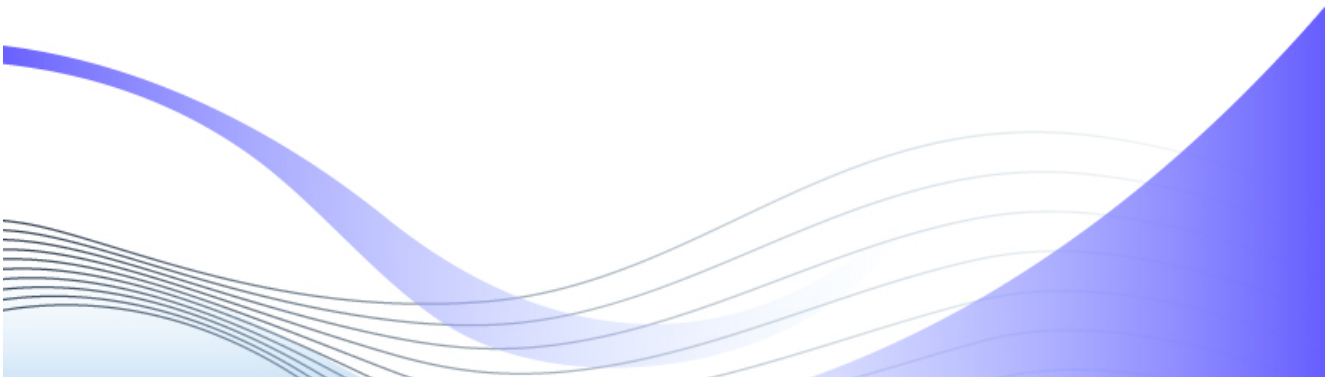
All components of a conversation flow are connected using color-coded arrows to signify conditions, or transitions from one component to another. When multiple conditions are defined with multiple connection options to other components, the colored arrows help you visualize the conversation flow. Instance connections help define the conditions for the transition between nodes and control the flow of the conversation. These conditions can be defined using simple IF-ELSE operators or more complex rule-based expressions, all with defined fallback conditions. Kore.ai supports two primary operators for building the default conditions:

Mathematical

- Greater than
- Less than
- Equal to
- Not equal to
- Greater than equal to
- Less than equal to

Logical

- Contains
- Does not contain
- Begins with
- Does not begin with
- Ends with
- Does not end with
- Exactly matches
- Not exact match



Sub-Intent Recognition

A dialog task starts with a root intent. Once the Platform identifies this root intent, the conversation flow starts. A dialog task multiple other associated intents (or sub-intent nodes) in addition to the root intent, such as:

- Intent only mode
- Run the complete dialog associated with the Intent node to return back to the current dialog

For example, in a dialog task with a root Intent node to find a branch of a bank, once the branch is located, the sub-intents could be to find a branch that sells personalized checkbooks or has an ATM.

Follow-up Intents

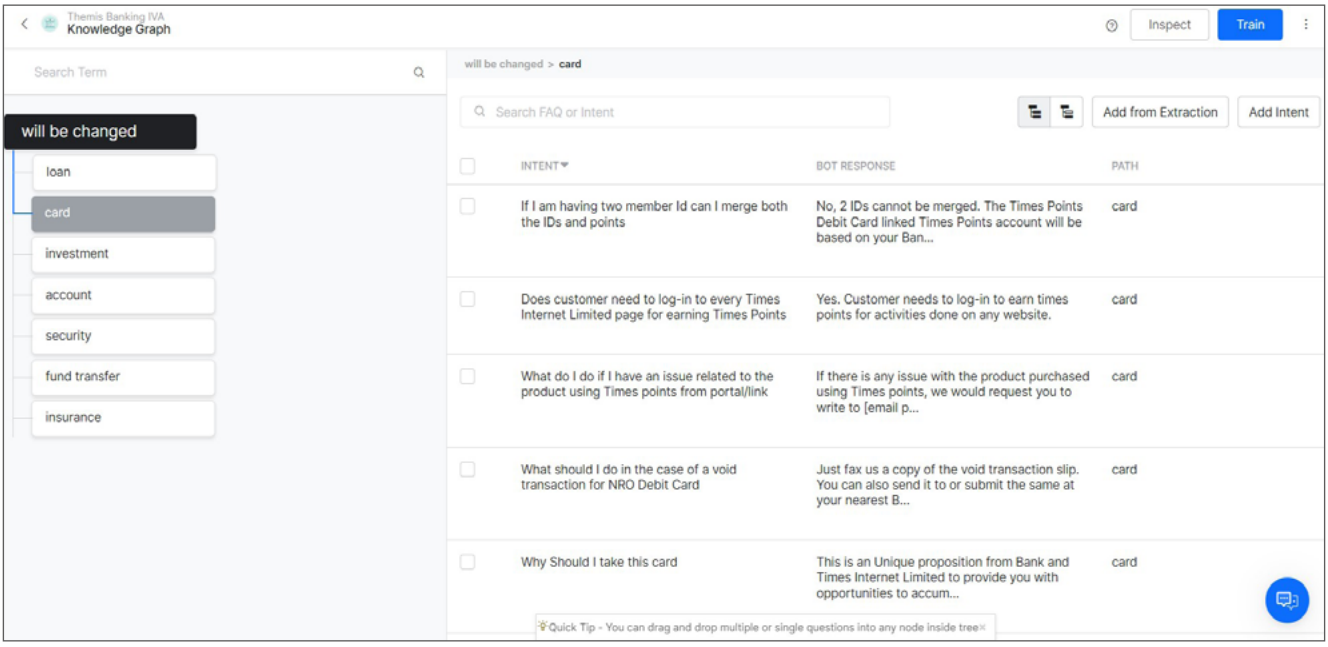
Using dialog tasks the user can be given an option to perform a task from the list of follow-up intents. You can also access this list to further train and refine the dialog flows.

Multiple Intents

Virtual assistants built on the Kore.ai XO Platform can recognize multiple intents or task commands in single-user input and process each without sacrificing the quality execution of the task.

Knowledge/FAQ Tasks

Knowledge tasks can take a user's question and query a predefined set of information to rapidly find the right answers to what they asked, without having to make a call to a web service. When a user utterance is registered with a specific virtual assistant, it searches knowledge tasks first to look for user intent and find an answer. It will only search for other tasks such as alert tasks if the user intent does not match any known knowledge tasks. Knowledge tasks can be designed to deliver answers from a variety of knowledge sources.

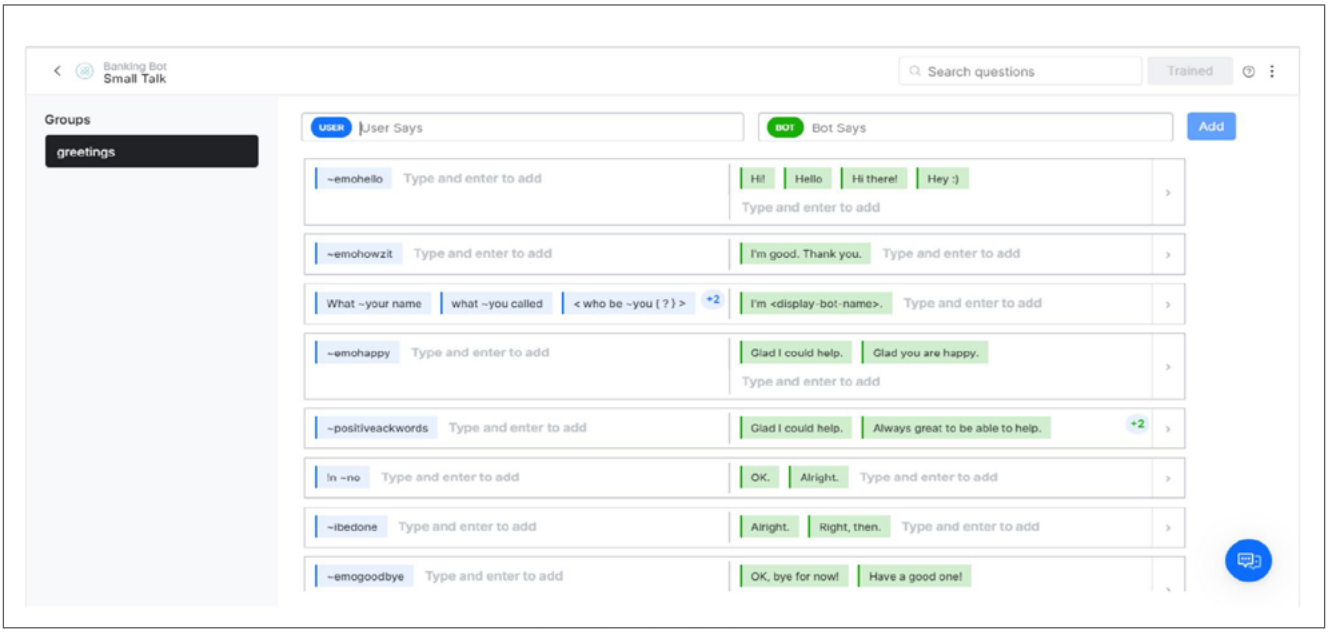


Knowledge AI

The Knowledge AI component enables data extraction from unstructured content such as web pages and PDF documents as well as from structured content such as CSV files. The extraction engine automatically detects TOC(table of contents) & bookmarks. It supports tables, pictures, and PDFs with multi-column or nested layouts, using the Knowledge AI component it can be ensured that images and tables are extracted and stored as virtual assistant responses (as images).

Small Talk

The custom small talk feature provides an easy-to-use interface for quickly generating multiple scenarios by simply providing user and virtual assistant utterances. The Custom Small Talk UI Editor can be used to build nested conversations to answer follow-up questions and context-specific responses.



Alert Tasks

Alert tasks deliver timely, relevant, and personalized information from enterprise systems to customers and employees by polling the relevant service in real-time or according to a user-defined schedule. These alerts contain detailed information, such as name, ID, notes, amount, and more, in a variety of formats for each channel.

- Scheduled Alerts - Users can schedule alerts to trigger according to the days, times, and frequencies they specify.
- Filtered Alerts - Users can configure alerts to trigger when certain thresholds are met such as amounts, dates, text matches, and more.

Digital Skills

The platform lets you build interactive digital experiences with Digital Views and Digital Forms.

Digital Views

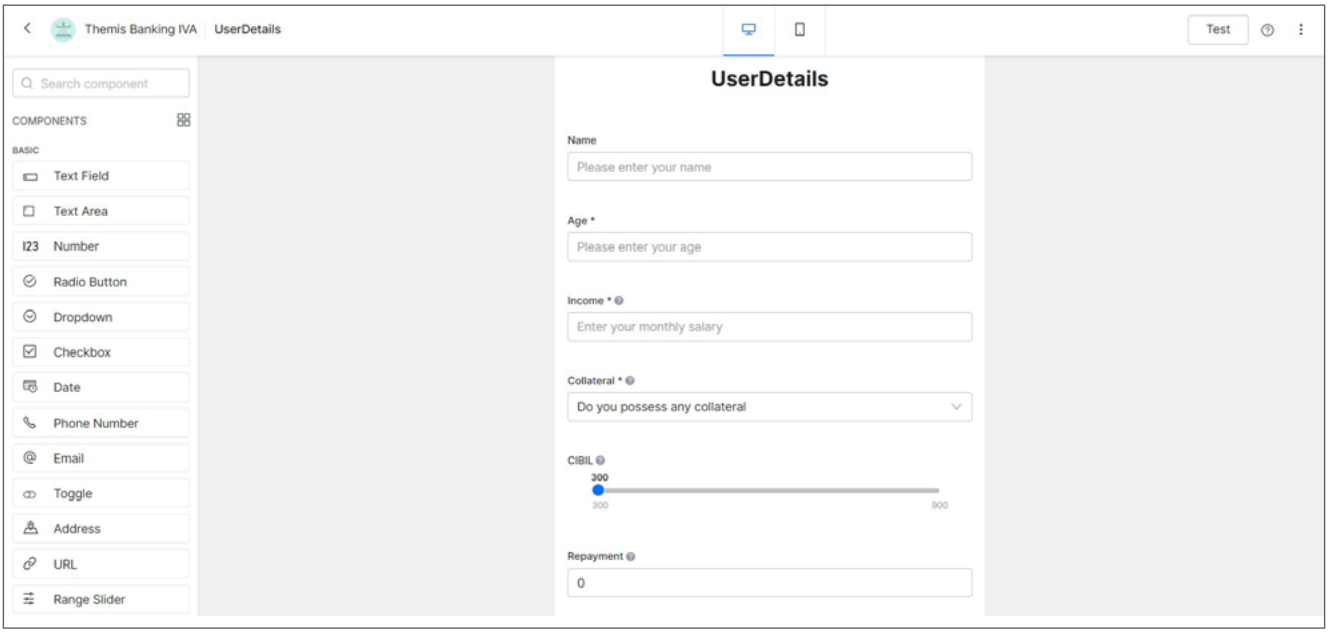
Digital Views are used to provide proactive updates and engaging interfaces to the end-users with minimal inputs. Using the Digital View of Kore.ai Experience Optimization (XO) Platform, you can design interactive components to capture the information efficiently from your customers and proactively present relevant information to them.

Widgets are individual components that can communicate with the virtual assistant for presenting information to the end-users. The source of information for widgets can be from a Dialog Task or a JSON using any of the pre-defined widget templates. The end users can interact with your virtual assistant either in the conversation mode or directly get the required information from the widgets. The digital interfaces can also trigger mobile/web apps from within the virtual assistant.

Digital Forms

Digital forms of the platform can be used to design and launch interactive forms for easy and efficient information capture. The digital form designer has all the required features and components needed to develop virtually any form specific to a business need.

Digital form designer has components such as text boxes, checkboxes, dropdowns, labels, tooltips, separators, date fields and more which are used to design a rich form. Digital forms can be used for layout management to check the responsiveness of your forms on various devices. In addition to this,



customization of themes can be done on the forms to suit your brand.

Information collected in Digital forms is automatically submitted to your virtual assistants so they can perform any necessary follow-up actions.

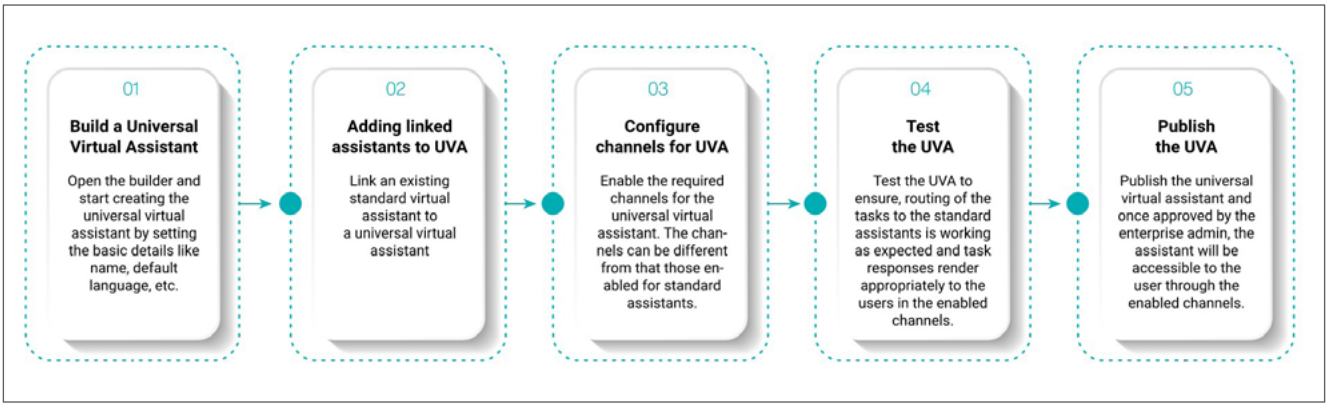
Universal Bot / Universal Virtual Assistant

Universal Virtual Assistant is a container assistant that can be linked with one or more Standard Virtual Assistants. When a user interacts with the Universal Virtual Assistant, it routes the user utterance to the appropriate linked assistants for intent detection or task fulfillment.

Kore.ai's Universal Virtual Assistant facilitate a scalable, modular approach to building virtual assistants by linking several standard individual virtual assistants into one.

- **Modularity:** Build separate assistants that address domain-specific issues and then integrate them to optimally interoperate.
- **Scalability:** Start building assistants that address key use cases and keep adding additional assistants to address more use cases.

Building a Universal Virtual Assistant



Training a Universal Virtual Assistant

You can train a Universal Virtual Assistant to guide it to the most relevant link assistant(s) from which the intents are to be identified. You can use a combination of Invocation Phrases and Training Utterances to define the Virtual Assistant Identification Training.

Training a Universal Virtual Assistant is essential for the following reasons:

- To ensure that the user utterances are routed to the relevant linked virtual assistants.
- Training Utterances or Invocation Names help the Universal Virtual Assistant to identify the linked assistants and route the user utterances to these identified linked assistants.
- In case no linked assistants are qualified, then the utterances are routed to the linked assistants as Fallback Virtual Assistants.
- It is essential that you review the linked virtual assistant identification flow from the Utterance Testing module.

Training a Universal Bot is a three-way process:

1. Train with Invocation Phrases to identify a specific intent in a specific linked virtual assistant;
2. Train with Invocation Names that would help identify a specific linked virtual assistant;
3. Train with user utterance for scoping linked virtual assistants

Enabling a Language

When you enable a new language for a standard virtual assistant, you need to upload language packs consisting of the entire virtual assistant definitions. Whereas to enable a new language for a universal virtual assistant, you just need to set up the corresponding default dialog.

Universal virtual assistants do not obtain the enabled languages of the linked virtual assistants. You must enable default and additional languages for the universal virtual assistant. When user utterances are made in an enabled language, the universal virtual assistant routes it to only those linked assistant tasks that have the same language enabled. For example, if German is enabled for the universal virtual assistant and a user utters in German, the NLP engine looks for all the linked virtual assistant tasks that support German and sends the utterance only to those tasks.

Intelligence

For virtual assistants to hold smart, human-like conversations – that ultimately foster the satisfying experiences for customers and employees that drive and convert more business – they must do more than simply process user commands. The Kore.ai Experience Optimization (XO) Platform includes a state-of-the-art Intelligence Engine that enables your virtual assistant to understand, remember and learn from the information gathered during each interaction, and provides you with the features needed to perfect your virtual assistant's natural language understanding.

This section describes the Intelligence Engine and the components that handle context, memory, sentiment analysis and learning.

Managing Interruptions

Twists and turns characterize human conversations, and no two directions are ever the same. Natural conversations often go beyond linear intent resolution paths defined in the bots. Often, users tend to ask a new question before the logical conclusion of the previous question or often ask multiple questions in a single utterance. The bots should be intelligent enough to understand the context and handle such scenarios efficiently

Dialog Management

Kore.ai provides you with granular control over hold and resume functionality at the bot, task, and node levels and allows you to control context switching rules and behaviors. You can handle such interruptions in intent flows by providing a whole range of options to define the task-switching experience.

You can set up the **Manage Interruptions** option at the bot, task, and node levels to ensure the configurations are layered to suit your various business requirements. You can also add conditional exceptions between tasks with the ability to pass contextual data between them.

Apart from defining the generic interruptions options at the bot level, you can customize these options at the dialog task or dialog node level. When you set up the Hold and Resume options in more than one levels, the order of precedence of the settings works as follows:

Sub-Intents & Follow-up Intents

Kore.ai bots platform offers **Sub Intents** defined at the task level and are the easiest way to shape dialog without managing context manually. It enables conversations to seamlessly branch into related intents as part of the primary intent and work only within this context.

Follow-up Intents come in handy during a dialog when you cannot execute user intent immediately. The platform stores these utterances in the FollowupIntents array. You can then ask users at the end of the dialog flow to select and perform tasks from a list of identified follow-up intents. You can also access this list to further train and refine their dialog flows.

Amend Entity

Use this feature to handle the scenarios when the users amend entity values in the later stage of the conversations. It enables you to control the post-amendment dialog behavior.

Multi-Intent Detection

Kore.ai NLP engine breaks conversations down to their essence and can identify and follow-up on multiple action items, or intents, from a single message. The bot can then execute tasks sequentially and logically.

The sequence in which these intents are executed is based on the standard phrases like before that, after that, and then, etc. in the user utterance. If no order is specified or identified by the NLP engine, then the intents are executed in the order present in the original utterance.

Component Reusability

Reuse previously developed nodes and entities in current or future dialogs, saving you time and money. The Kore.ai XO Platform Dialog Builder also supports importing and exporting dialog tasks

Context Management

Virtual assistants that understand and use contextual information throughout conversations will interact in a way that's easier, quicker, and more natural. However, information collected from users, companies, and an individual virtual assistant will vary in importance, utility, and lifespan. The platform supports multiple context variables for thoughtful design of how information gets used for distinct scenarios. Through the Dialog Builder tool, you can customize, categorize and apply contextual information that maximizes the virtual assistant's ability to make intelligent decisions and avoid user confusion or extra steps. The Intelligence engine can apply the following types of context:

- Enterprise Context - Information that represents company-wide rules and standards that apply to all users and virtual assistants.
- Session Context - Information specified by a user is the primary context for a virtual assistant to keep in mind during a session.
- User Context - Individual user information or preferences can be shared by all enterprise virtual assistants the user will interact with.
- Bot/Virtual Assistant Context - User or task information is dynamically captured at an assistant level that can be used in context with some or all of the users of that virtual assistant.
- Custom Code Logic - You can use custom code logic to manipulate API responses, promote additional data to the user context, and pull data from the user context.

Session and Context Variables in Tasks

When you create and define tasks, they can access session variables provided by the platform, custom variables that they define, and the context that defines the scope of the variable. For example, some API requests may require you to set session variables before the request is executed, or a Dialog task component may need to access a session variable to transition to the next node. Dialog tasks can also access the context object with additional system variables.

These sessions and context variables allow you to persist data and store, for example, a user's home address for commerce, transportation, and home delivery-focused services to be used by the virtual assistant when executing a task.

Sentiment Management

The platform's NL engine can parse user inputs for specific words and phrases to detect the sentiment of the utterance and the user's likely mood to provide an average tone score to steer the flow of the conversation.

Sentiment Types

The Kore.ai Experience Optimization (XO) Platform evaluates user inputs to find the following six possible emotions:

- Anger
- Disgust
- Fear
- Sadness
- Joy
- Positivity

This sentiment algorithm can score multiple emotions. For example, an input could yield a high score for joy, but a mild score for sadness.

Sentiment Scoring

The NL engine provides a sentiment score, on a scale of -3 to +3, that represents how much or how little the user input relates to the indicated emotion(s). Positive values represent expressed emotion, negative values represent repressed emotion, and 0 represents a neutral tone.

MESSAGE TONE		
TONE NAME	LEVEL	COUNT
positive	3	1
DIALOG TONE		
TONE NAME	LEVEL	COUNT
angry	0.5	1
joy	0.5	1
positive	2.5	2

The overall tone score is calculated and assessed for modifiers. Modifiers are usually adverbs or adjectives that supplement the emotion, and either increase or decrease the base score. For example, "I am extremely disappointed" would return a higher angry tone than, "I am disappointed."

You can store these tone scores as context and alter the flow of conversations using the Dialog Builder within the Kore.ai platform. Tone scores are returned as context object variables in the following ways:

Authorization Management

Kore.ai XO Platform's Enterprise Integration Framework supports multiple authentication models that can be configured at a task level for each virtual assistant to use authenticated and unauthenticated tasks in the same configuration.

Developers can build custom authentication profiles for virtual assistants by defining subdomains through tenancy URLs, adding new fields through IDP form fields, and more. They can also test and validate the authorization definition before moving on to the next steps in the building process. The Platform passes user identity information and authentication tokens to the virtual assistant context as the web and mobile SDK initializes, so your virtual assistant can leverage existing authentication and authorization settings for end-users.

Below are the supported authentication models -

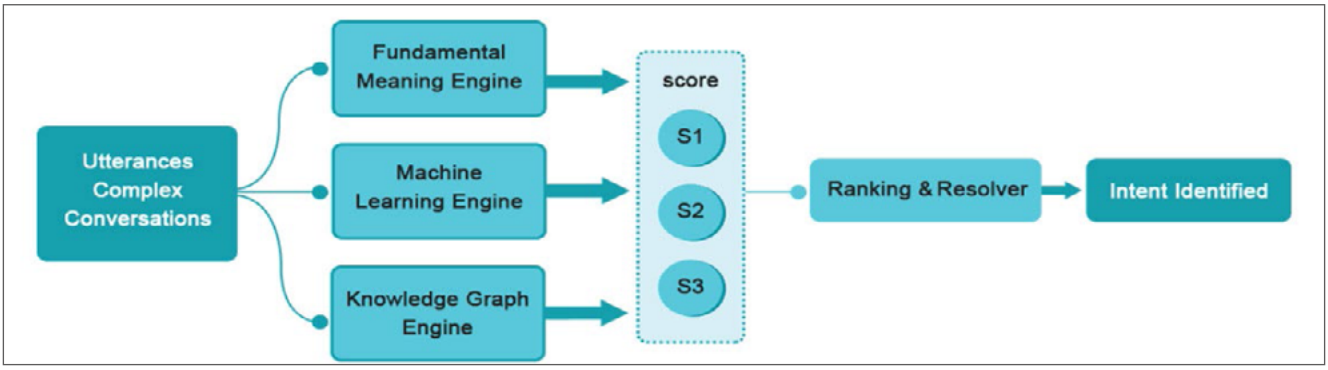
- Basic Auth
- OAuth v2 password grant type
- OAuth v1
- OAuth v2
- API Key

Training

Natural Language Processing

Kore.ai XO Platform uses a unique Natural Language Processing, i.e., **ML + 2 NLP Approach,** combining Machine Learning, Fundamental Meaning and Knowledge Graph engines for maximum conversation accuracy with little upfront training. It enables virtual assistants to understand and process multi-sentence messages, multiple intents, contextual references made by the user, patterns and idiomatic sentences, and more.

The NL engine includes recognition support for a wide range of entities and provides the tools needed to customize your bot's language understanding further using additional patterns.



Machine Learning Engine

The Platform provides an engine that uses ML algorithms and queries training data to determine the best match for a user's intent, and searches for patterns to train and tune the NL engine. The engine lets your large sets of training data leverage this information from the get-go. It also reviews user history to correct failed utterances and false positives. You can set a virtual assistant to be automatically trained via machine learning whenever a task is added, upgraded, approved, rejected, suspended, or deleted.

The latest model allows your virtual assistants to precisely identify the user intents by understanding the meaning of a user sentence and at the same time give importance to the domain-specific terminology.

Fundamental Meaning Engine

Fundamental meaning is powerful as it understands semantics, grammar, rules and language nuances. It allows a virtual assistant to understand meaning by parsing a user's input by meaning, position, conjugation, capitalization, plurality, and other factors to recognize the intent and extract key entities, or field data, needed to complete a task. If an utterance contains more than one possible intent, the virtual assistant presents both to minimize failures.

Knowledge Graph Engine

The KG engine of the platform provides an ontology, through domain terms and the relationships between them, provides a framework that limits false positives. Synonyms for each term easily extend the identification possibilities without the need to provide very large numbers of alternative training questions. Using the Ontology Generator, the key domain terms in the Q&As can be identified and an

Ontology can be automatically generated. Using the Knowledge Graph Analyzer, the platform performs an 11-step diagnosis of knowledge collection, identifies potential issues and makes recommendations for the corrective action. Taxonomy-based Knowledge Graph with built-in flows engages the user in a multi-turn conversation to disambiguate the path and find the right question.

Traits

Traits are specific entities, attributes or details that the users express in their conversations. In natural conversations, it is quite common for users to provide background information while describing a specific scenario. Traits as a feature use this background information to identify the intent and accordingly drive the conversational flow.

Ranking and Resolver Engine

The Kore.ai NLP engine uses Machine Learning, Fundamental Meaning, and Knowledge Graph (if any) models to match intents. All the three Kore.ai engines finally deliver their findings to the Kore.ai Ranking and Resolver component as either exact matches or probable matches. Ranking and Resolver determines the final winner of the entire NLP computation.

Other NL Capabilities

The Kore.ai platform's NLP approach recognizes simple yet critical nuances to a human's natural language to mitigate potential misinterpretation and prevents you from designing for every idiomatic variation. It also includes features that let you easily customize, extend and reuse vocabulary. These include:

- Capitalization: Recognizes proper nouns and removes capitalization from common nouns
- Numeric Words vs. Digits: Recognizes the communication of numeric values as words or
- digits
- Singular vs. Plural Nouns: Processes singular and plural nouns the same way
- Contractions: Expands contractions and removes apostrophes to simplify the task
- processing
- Tensed Verbs: Understands a single verb communicated in different tenses as
- synonymous
- Vocabulary Transfer: Transfer of developed vocabulary from one virtual assistant to the next

Intent Recognition

Virtual Assistant tasks can be broken down into a few words that describe what a user intends to do, usually, a verb and a noun: Find an ATM, Create an event, Search for an item, Send an alert, Transfer funds, etc.

Our NLP engine analyzes the structure of a user's command to identify each word by meaning, position, conjugation, capitalization, plurality, and other factors. This helps the virtual assistant correctly interpret and understand obvious and non-obvious synonyms for these common "action" words.

The goal of intent recognition isn't just to match an utterance with a task, it's to match an utterance with its correctly intended task. We do this by matching verbs and nouns with as many obvious and non-obvious synonyms as possible.

Entity Extraction

Entities are the fields, data, or words that you designate necessary for the virtual assistant to complete a task: a date, time, person, location, description of an item or a product, or any number of other designations.

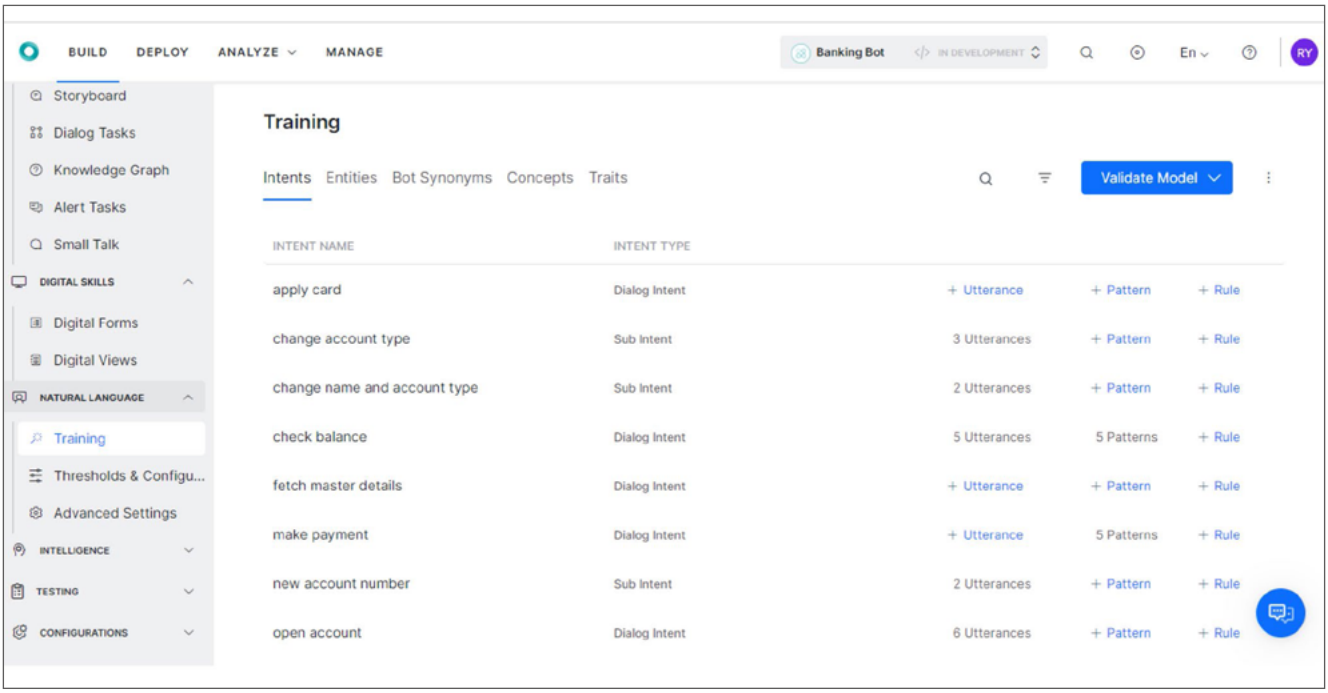
Through our NLP engine, the assistant identifies words from a user's utterance to ensure all available fields match the task at hand or collects additional field data if needed. Our platform also supports the use of machine learning for entity recognition.

The goal of entity extraction is to fill any holes needed to complete the task while ignoring unnecessary details. It's a subtractive process to get just the necessary info whether the user provides it all at once, or through a guided conversation with the virtual assistant.

Synonyms & Patterns

The platform includes a built-in synonym library for common terms. You can further optimize the accuracy of the NLP engine by easily adding synonyms for virtual assistant names, words used in the names of your tasks and task fields, and any words associated with your dialog task entity node.

The platform lets you take into account slang, metaphors, and other idiomatic expressions by adding patterns to tasks, task fields, dialog task intent nodes, and dialog task entity nodes. A pattern, in this case, can be defined as a sentence that represents the task but does not actually contain any of the words in the name of the task, task field, or associated dialog task. When the NLP engine matches a synonym to one task or field, and a pattern to a different task or field, the pattern match is prioritized and used for positive recognition over the synonym match.



Personalization

You can replace the default, universal responses and error messages with unique, configured messages via a dedicated virtual assistant response editor. These messages can vary by channel and be randomized when more than one message option is present in a virtual assistant configuration.

The Kore.ai Experience Optimization (XO) Platform supports the following standard response types:

- Statements
- Queries
- Errors & Warnings
- Choices
- Greetings
- Questions

NLP Rules

The platform provides custom entry and exit hooks, available as API endpoints, in the NLP engine to enable the conversation flow to be taken in different directions based on business processes and rules. You can control the behavior of the virtual assistant and the dialog and each of these points. **Key Benefits of Our NLP Approach:**

- Interprets accurately with fewer false positives
- Communicates comprehensively
- Resolves development gaps faster
- Requires less training data to be NL capable
- Repurposes training data
- Uses statistical modeling for conflict resolution and user input failures
- Simplifies correction of false positives

Supervised Learning for NL

Through the platform's Dialog Builder tool, you and your admins can evaluate all interaction logs, easily change NL settings for failed scenarios, and use the learning to retrain the virtual assistant for better conversations. You can also leverage chat logs to build predictive models and use the outcomes to further define additional proactive alerts, suggested actions, or automated workflows.

How it is implemented

1. Review chat logs to see NL successes and failures
2. Assess scenarios to:
    - Identify any incorrect matches within successful user utterances
    - Determine which failure scenarios should be fixed
3. Fix correctable scenarios by selecting the aligned virtual assistant task for the user utterance
4. Expand NL capabilities:
    - Add synonyms based on the scenario learning
    - Add patterns, or slang and other sentence variations, to associated tasks
    - Retrain the virtual assistant using ML to dynamically expand the language model

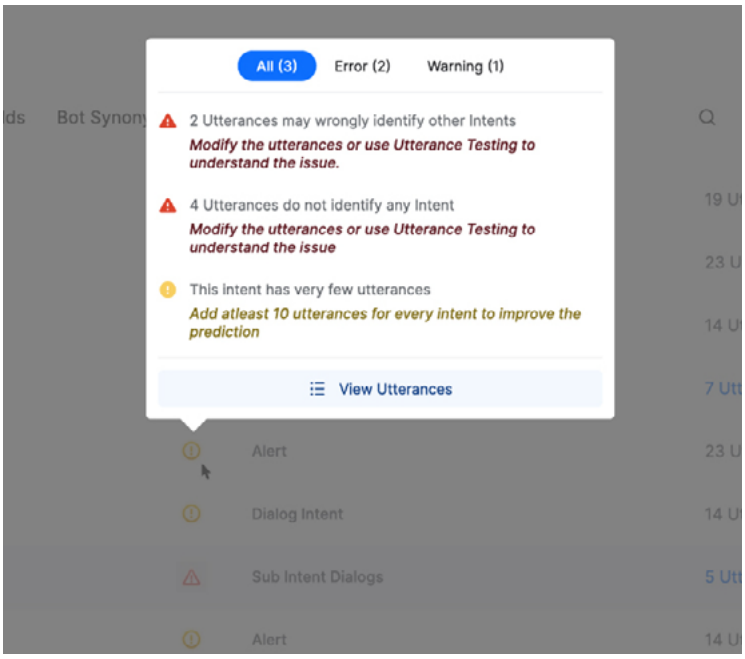
Unsupervised Learning for NL

Unsupervised learning for NL can be applied to expand the language capabilities of your virtual assistant – without human intervention. Unlike unsupervised models in which virtual assistants learn from any input – good or bad – the Kore.ai Experience Optimization (XO) Platform enables virtual assistants to automatically increase their vocabulary only when the assistant successfully recognizes the intent and extracts the entities of a human's request to complete a task.

Proactive NLU Validations

The Kore.ai XO Platform proactively guides you on how to improve the NLU performance. The performance of the virtual assistants relies on the quality and quantity of training provided, and this new capability guides you to better train virtual assistants for optimal performance. You can:

- **Easily identify** the intent that needs your attention such as intents with low confidence scores
- Take **corrective** action for the problems identified
- **Enhance** the NLU training and improve the overall performance of your VA



It focuses on validations related to the intent training with the ML & FM Engines for untrained intents, inadequate training, utterance does not qualify any intent, utterance predicts an incorrect intent, utterance predicts the expected intent with low confidence, incorrect intent patterns, short training utterances, incorrect entity annotations and more.

Multilingual

Communicate with your customers in their preferred language. The Kore.ai Experience Optimization (XO) Platform supports enabling multiple languages for an assistant without having to rebuild the definitions.

The platform lets you enable multilingual virtual assistants automatically. You only need to train the virtual assistant in one of your preferred languages, and the language-agnostic NLP models start engaging the users in over 100 languages. Further, you can leverage auto-translation of the responses via translation services from Google Cloud or Microsoft.

The platform also allows you to build language-specific models for 25+ popular languages. As part of enabling the language, the platform automatically translates all Standard Responses and you can then translate various other virtual assistants elements in the new language.

Testing

After you have defined your virtual assistant/bot and configured one or more tasks, it is advisable to test the bot's performance before publishing. The Kore.ai XO Platform provides a talk-to-bot functionality, where you can chat with the bot in real-time to test intent recognition rate, performance, and flow as if it were a live session.

The platform provides multiple methodologies to test the bots.

Explainable AI - Debugger & Tracer

The platform provides you with complete details on how the utterances are processed. Dialog Builder includes a testing and debugging tool that you can use to expose chat logs and outputs in real-time—as an actual virtual assistant conversation. Testing can be done in real-time with debugging and trace features accessible at each stage of virtual assistant development. This tool includes:

Real-Time Error Detection

The Dialog Builder displays any detected errors and warnings in real-time to you as they build or modify the conversation flow.

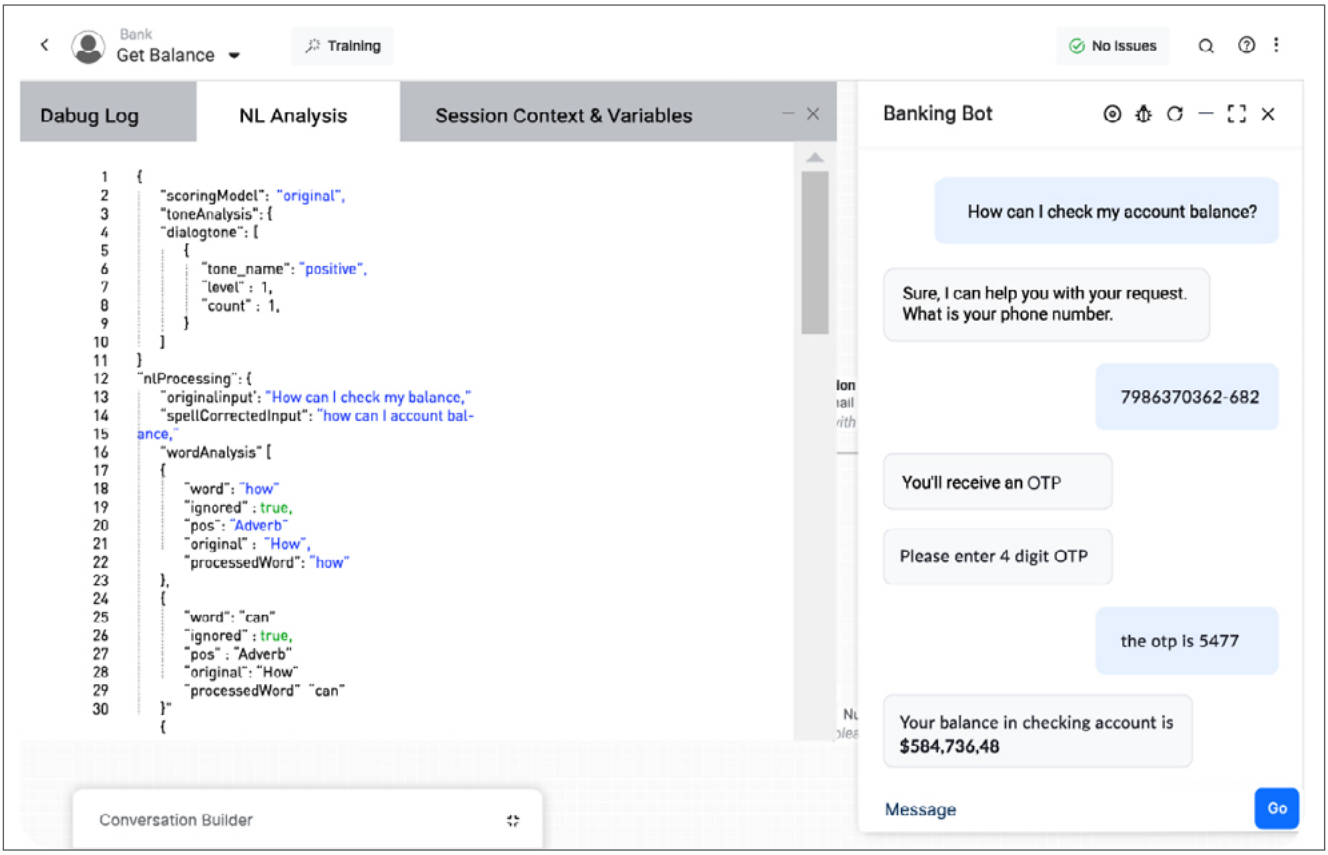
Dialog Node Tracking

You can view the flow of a dialog task from the start of the root intent to a selected component. This allows you to quickly find broken flows or conditional transition issues. The ability to track the various paths to a node allows you to easily retrace steps in the flow.

Debug Log

Provides the sequential progression of a dialog task and context and session variables captured at every node. The debug log supports the following statuses:

- Processing
- Processed
- WaitingForUserInput
- Pause
- Resume
- WaitingForServerResponse
- Error
- End

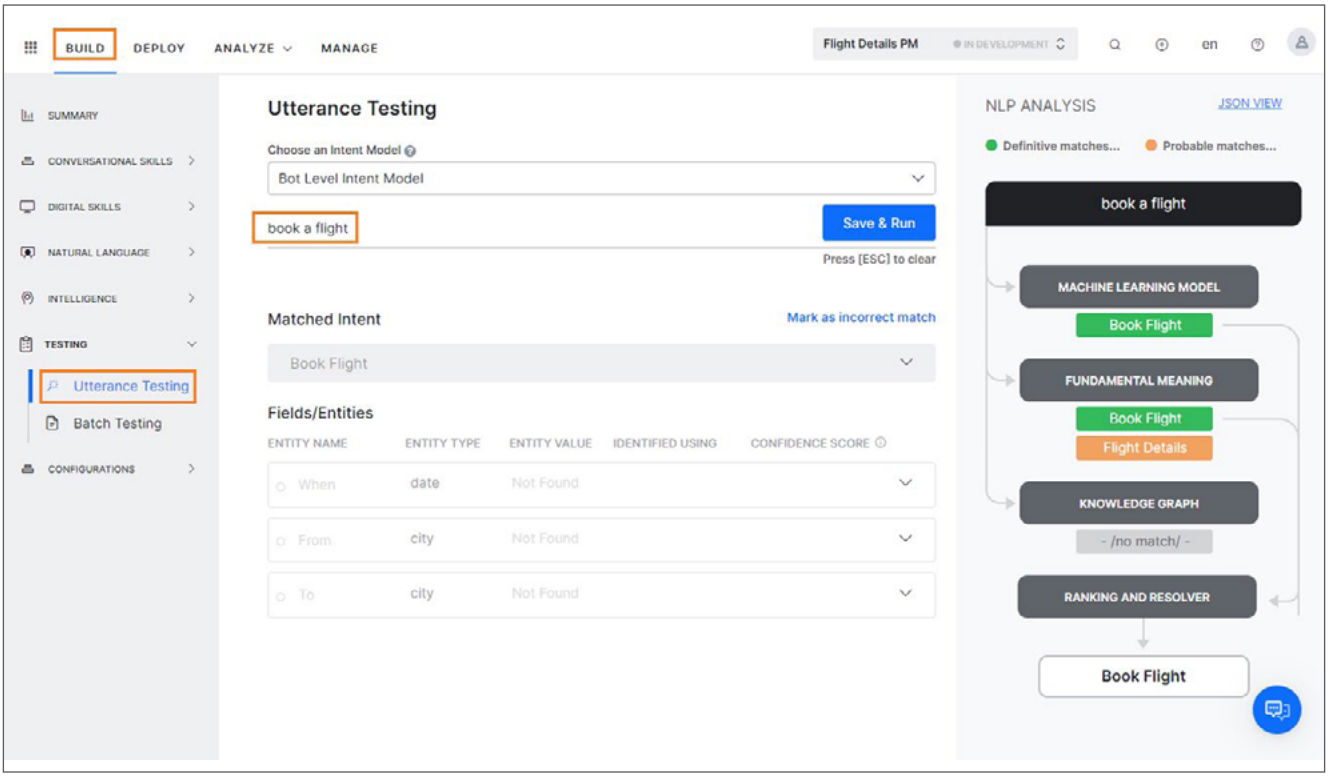


Utterance Testing:

This approach lets you test the virtual assistant's performance for the given utterance. When you feed the bot with an utterance, the NLP engine finds the bot tasks that match the intent. The NLP engine uses a hybrid approach using Machine Learning, Fundamental Meaning, and Knowledge Graph (if the bot has one) models to score the matching intents on relevance. The model classifies user utterances as either being *Possible Matches or Definitive Matches*.

Definitive Matches get high confidence scores and are assumed to be perfect matches for the user utterance. If the utterances match with multiple *Definitive Matches*, the bot sends options for the end-user to choose one.

Possible Matches are intents that score reasonably well against the user input but do not inspire enough confidence. In this case, the bot sends these matches as *"Did you mean?"* suggestions for the end-user. If the models shortlist more than one *Possible Match*, the Ranking and Resolver engine re-scores all the shortlisted intents to determine the final winner.



Batch Testing:

This feature helps you discern the ability of your bot to correctly recognize the expected intents and entities from a given set of utterances. It involves executing a series of tests to get a detailed statistical analysis and gauge the performance of your virtual assistant's ML model.

Out-of-the-box, the platform provides **Test Suites** to perform batch testing. **'Developer defined utterances'** and **'Successful user utterances'.** You can also create a **New Test Suite** in CSV or JSON format for testing a custom set of utterances. Each test run will create a test report record and display a summary of the test result.

Use the Platform's visualization tools, confusion matrix and K-fold testing to understand the assistant usage patterns, conversation flows and drop-off points.

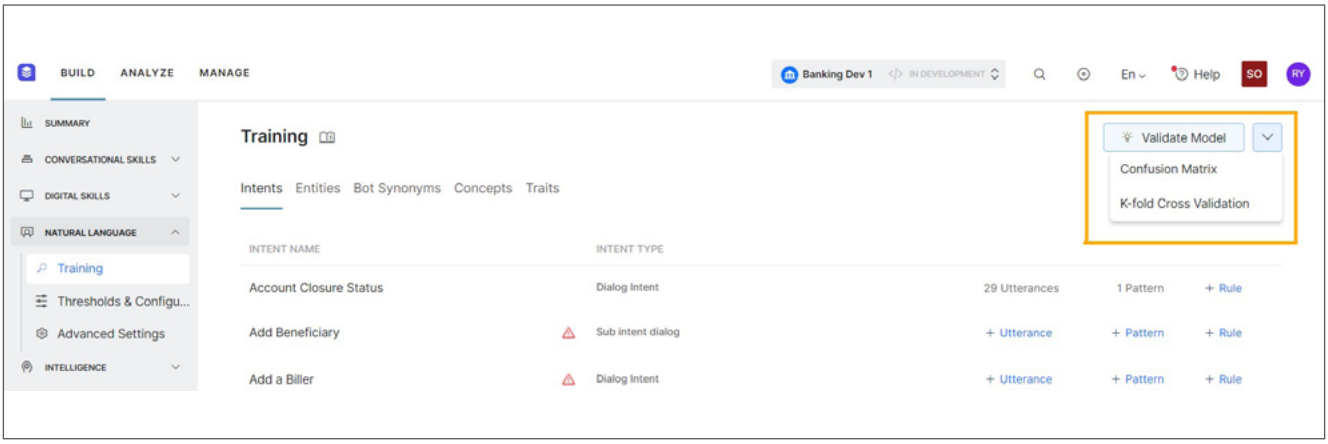
Further, drill down results to intent and test case level analysis. Tag specific test case results that need follow-up actions and collaborate with your team to improve the performance.

Conversation Testing

Conversation Testing enables you to simulate end-to-end conversational flows to evaluate the dialog task or perform regression. You can create Test Cases to capture various business scenarios and run them at a later point in time to validate the performance of the bot.

Model Validation

Once you have built your virtual assistant and trained it, the Kore.ai platform builds an ML model mapping user utterance with bot intents. Once created, it is recommended to validate the model to understand and estimate an unbiased generalization performance of the ML model.



Kore.ai XO Platform offers two validation methods:

Confusion Matrix or Error Matrix to visualize the performance of the machine learning model.

Confusion Matrix is useful in describing the performance of a classification model (or classifier) on a set of test data for which the true values are known. The graph generated by the confusion matrix presents an at-a-glance view of the performance of your trained utterances against the virtual assistant tasks. The name stems from the fact that it makes it easy to see if the model is confusing utterances.

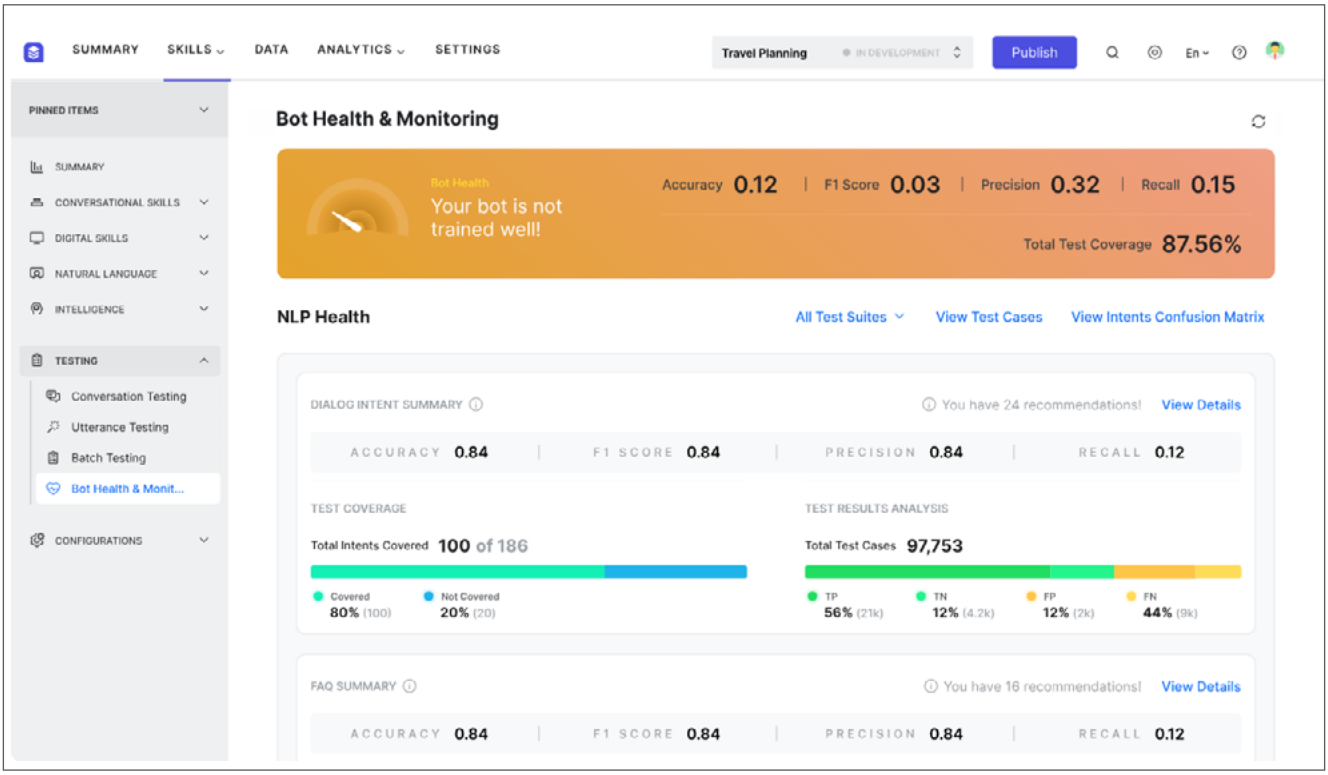
The ML Model graph evaluates all the training utterances against each task and plots them into one of these quadrants of the task: True Positive (True +ve), True Negative (True -ve), False Positive (False +ve), False Negative (False -ve). A quick look at the graph and you know which utterance-intent matches are accurate and can train further to produce better results.

K-fold Cross-Validation to estimate the skill of the machine learning model.

Cross-Validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The technique involves partitioning the data into subsets, training the data on a subset, and using the other subsets to evaluate the model's performance. Performing Cross-Validation gives a more generalized metric on model performance which is a better indicator of the ML model's performance.

Health & Monitoring

Get a 360-degree view of the NLP training provided to the virtual assistant by seeing the overall Accuracy, F1 score, Precision, and Recall. With an in-depth analysis of the test coverage and data results, achieve close to an error-free IVA training. Get extra help to enhance the training for incorrect patterns, short utterances, and incorrect entity annotations.



Deploy

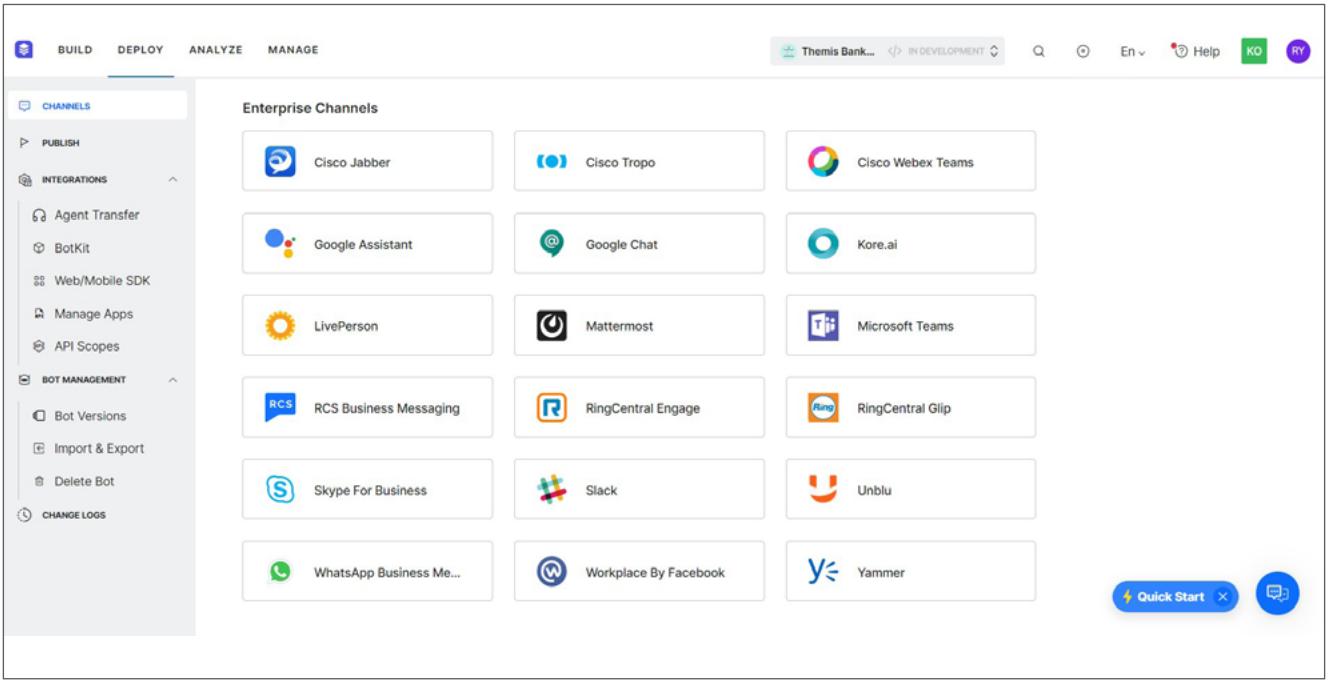
Omnichannel Approach

Virtual assistants built on the Kore.ai Experience Optimization (XO) Platform support an omnichannel approach, meaning customers and employees can communicate with digital systems in the same ways they already engage with a brand or other people. Unlike building a virtual assistant for a specific channel, the platform allows you to build a virtual assistant and its underlying tasks once, and adjust the channel-specific UI elements or virtual assistants' responses as desired.

The channels supported by the platform are

- Web and Mobile SDK - JavaScript, iOS and Android Java
- Email
- SMS - Twilio, Tropo, Syniverse
- Enterprise Channels - Skype, Cisco Jabber, Slack, Yammer, Glip, Mattermost, Microsoft Teams, Google Hangout, Workplace by Facebook, WeChat, Unblu, RingCentral Engage, Zendesk Sunshine
- Social Messenger - Facebook Messenger, Twitter, Line Messenger, Telegram, Whatsapp
- Voice Assistant - Amazon Alexa, Google Assistant
- IVR - Avaya, Genesys, Nuance, Cisco, Nice, Five9

The Kore.ai platform also has a RESTful Real Time Messaging (RTM) API that enables the communication of varying channels with the platform using web sockets. Kore.ai Experience Optimization (XO) Platform also offers an IVR Sandbox environment for its cloud version. This built-in IVR Sandbox experience enables you to instantly launch the virtual assistant for interactions over voice calls.



Integration Framework

The complexity, time and cost of building virtual assistants are directly impacted by the work required to integrate virtual assistants into your diverse and distributed enterprise systems. The platform provides a complete integration model for managing the movement of data between both cloud and on-premise systems and users via virtual assistants. The platform's Enterprise Integration Framework, which handles orchestration, authentication, transportation, and mediation for applications that may consist of multiple communication protocols or technologies, reduces operational complexity and provides a standardized integration methodology. Save time and money by freeing yourself from writing complex custom glue code and implementing third-party integration suites or ESBs for every virtual assistant.

This section describes the platform's Enterprise Integration Framework and the features that simplify and accelerate integration to any enterprise system.

Auth Configuration

The platform's Enterprise Integration Framework provides support for multiple authentication models that can be configured at a task-by-task level for each virtual assistant, for an assistant to use both authenticated and unauthenticated tasks in the same configuration.

You can build custom authentication profiles for virtual assistants with support for defining subdomains through tenancy URLs, adding new fields through IDP form fields, and more and test and validate the authorization definition before moving on to the next steps in the virtual assistant building process. The platform can also pass in or provide user identity information and authentication tokens to the virtual assistant context as the web and mobile SDK initializes, so your virtual assistant can leverage existing authentication and authorization settings for end-users.

Supported Authentication Models

- Basic Auth: SSL encryption and basic authentication. When used, the virtual assistant prompts the user for login credentials to access the web application or web service
- OAuth v1: Open protocol for web, mobile, and desktop applications or services to access protected resources using an API without end-users having to disclose their log-in credentials to Kore.ai
- OAuth v2: New version of the open protocol for web, mobile, and desktop applications or services to access protected resources using an API without end-users having to disclose their log-in credentials to Kore.ai
- OAuth v2 Password Grant: Customized authorization for non-standard web service
- authorization
- API Key: Prompts the user for an API key when configuring the virtual assistant instead of username and password. The API key acts as a unique identifier and secret token for identification as well as authentication to provide a set of access rights on the associated API.

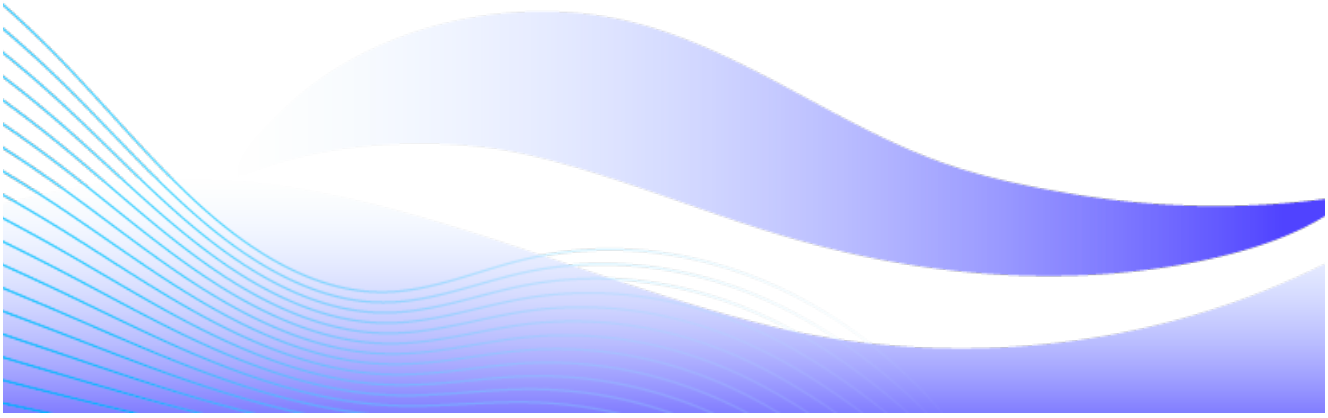
Alert Scheduler

You can define interval options for users to determine the days, times, and frequencies they wish to receive an alert and set the default for how often the platform will ping the appropriate service or system to check for predefined changes, and request data when needed, since the last update. The scheduler submits a request for the alert to be delivered at the appropriate time. The scheduler supports initial polling to display events immediately to end-users after they set up an alert task instead of waiting for the specified interval to elapse and the ability to set up alert reminders.

Service API Orchestration

The Enterprise Integration Framework brings virtual assistants and backend systems of record together using a simple, yet powerful API-based integration framework. This framework allows for:

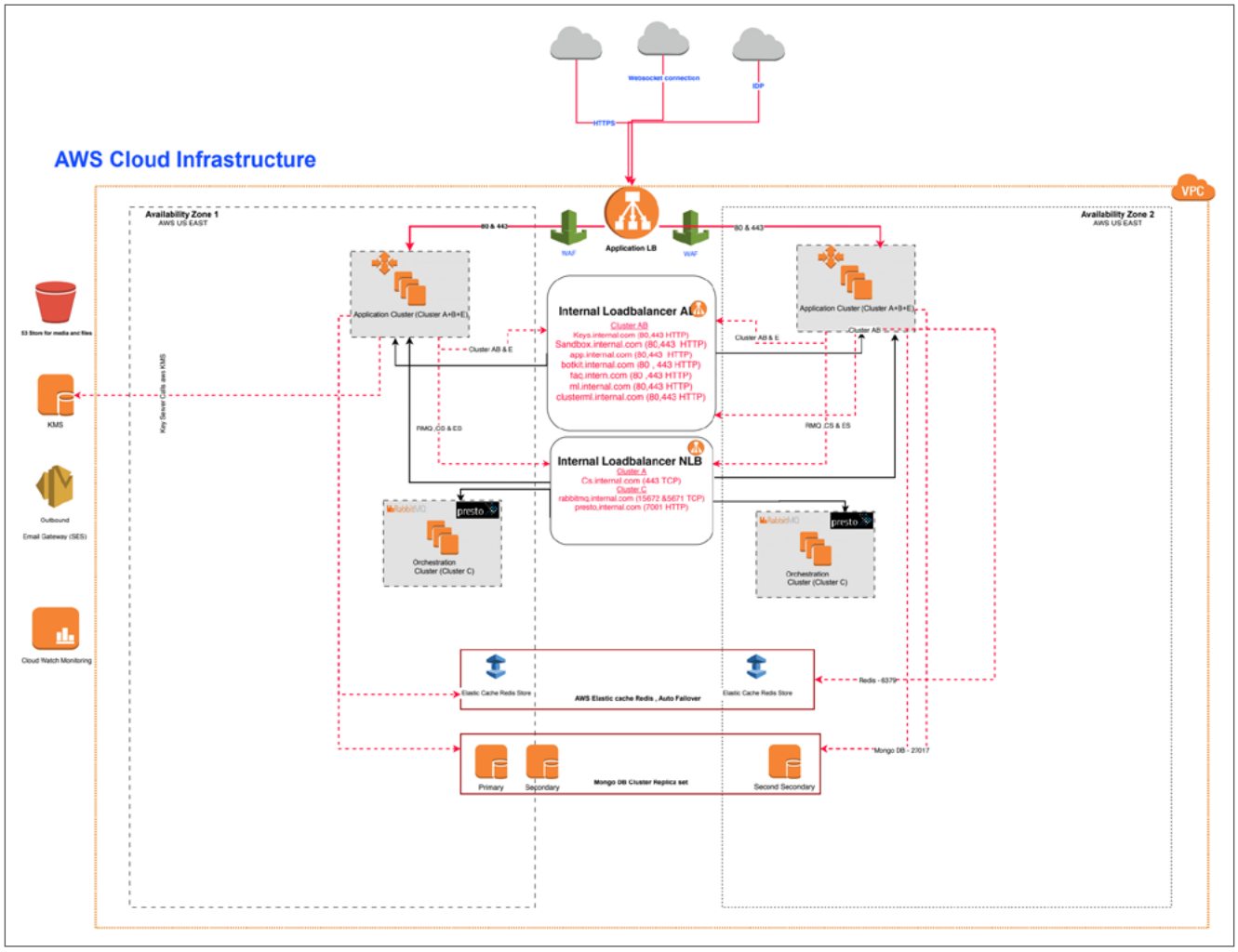
- Trigger-based alerts using Webhooks and a poll scheduler
- Strategies to identify new events or content and prevent duplicate alert messages
- None (default method) - Incoming task payload is hashed and the latest 500 task hash values are persisted. Each new task payload is compared against the list to prevent duplicate messages
- Chaining of API requests and aggregated responses
- Custom javascript processors for both requests and responses
- Parametrizing of headers, path fields, and request payloads using both user input and session data



Deployment Environment

Cloud

The Kore.ai platform cloud version of the platform meets high availability configuration needs across all components and is hosted across two Amazon Web Services availability zones. Server components across the architecture tiers have built-in redundancy, load balancing, autoscaling and replications to withstand failures. They are constantly monitored to provide High Availability.



On-Premise

The logical architecture diagram and chart presented below provide a visual and explanatory understanding of the major clusters and components involved in an on-premise deployment of the Kore.ai Platform.

Hybrid

The Kore.ai Experience Automation Platform also includes a Cloud Connector, an agent that runs behind your firewall within your company Intranet or perimeter network (DMZ) to provide secure data exchange between the Platform (when run in the cloud) and your on-premises applications.

Sandbox - The JavaScript execution engine which executes the scripts written in Script Node and the message templates. This can be deployed as part of Cluster A. Optionally this can be deployed in a DMZ environment to further protect the app from any malicious code the virtual assistant developers may write, and not required in case enterprise developers are writing scripts in a controlled manner.

Component Name	Description	Required software
Application Server	Exposes REST API endpoints to different subsystems of the application cluster including bot builder. REST API endpoints for storing retrieving bot/task definitions, get/store/ update context information, encryption service, endpoints for webhook, web and mobile channels. Required for enabling corresponding channels Twilio, Alexa, Workplace for Facebook and HTTP listener that handles requests for IVR. Expands the resource metadata in REST API response. Internal service used by the Application server. API Service that provides HTTP over WebSockets for persistent connections. This service provides the endpoint for web and mobile channels.	Nginx, Node.js
IDProxy Server	Provides SSO and OAuth integration. Exposes an endpoint idp. <domain>. <tld> for configuring as a callback URL for oAuth endpoints that are being integrated for bot authentication.	Node.js
RabbitMQ Server	AMQP Message broker. Internal subsystems of the Kore cluster uses AMQP messages as a way of communicating across services.	Rabbitmq
Bot Service Runtime	Provides Adaptor for Bots channels which process incoming messages and sends the outgoing messages Processes incoming and outgoing messages for web or mobile channel. Standalone timer service which handles scheduled tasks including user-configured alerts	Node.js

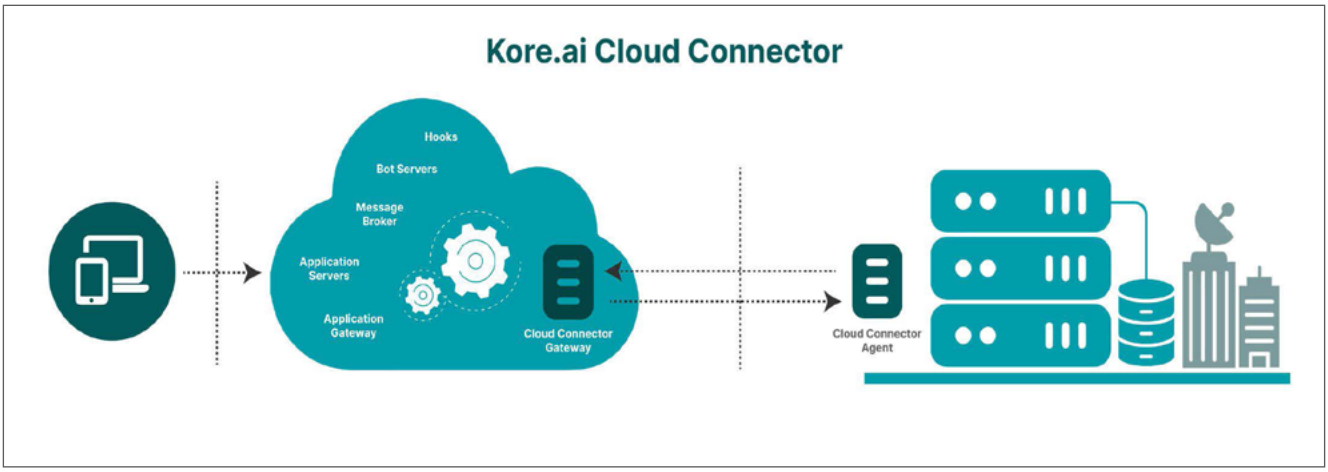
ChatScript	Natural language processing engine to process the incoming messages and identifies the intent	C
Bots Service Admin	Standalone service that pulls the enterprise user profiles from LDAP or Active Directory and updates them in Kore platform Executes the backend web services. Optionally this can be configured to use forward proxy (ex. squid) Executes the API requests for backend web services. Optionally this can be configured to use the forward proxy (ex. squid)	Node.js
Sandbox	Provides javascript execution environment to execute developer provided script while processing message templates and script nodes.	Node.js
Key Server	API server to manage key encryption/ decryption. This is also the server that interacts with HSM.	Node.js
Presto	Presto cluster is responsible for fetching the analytics data (dashboard and analyze screens in the UI), and audit logs.	Presto
Machine Learning	Machine Learning Service is a that is responsible for training the bot model and also the intent prediction based on the trained model	Python (3.5.3)
FAQ	Engine is a that is responsible for predicting the answers for knowledge collections	Python (2.7)
Haraka Server	Required for enabling email channel for a virtual assistant	Node.js

Listener Service	Required for enabling corresponding channels Slack, Yammer, Twitter, LivePerson Channel and Facebook channel	Node.js

Kore.ai provides hardware and software capacity-planning documents to help size the environment for the deployment and an installation guide for easy set-up.

Cloud Connector

The Cloud Connector runs behind your enterprise firewall and acts as a bridge to facilitate secure data exchanges between on-premise systems and The Kore.ai XO Platform's cloud-based infrastructure. This lets you rapidly connect legacy systems to the platform without disrupting existing business processes or risking the loss of your data via exposure or expensive system replacements.



How it works:

The platform contacts the platform's Connector Gateway with a virtual assistant request. The Connector Gateway relays the request to the Connector Agent, which is installed and securely configured in your on-premises Intranet or DMZ to interact with your virtual assistant's applications. Responses from your applications are sent back through the Connector Gateway to get to the Platform and processed.

Optionally, you can enable your on-premises applications to push data, for example, alert messages, to the platform using Webhooks. For virtual assistants that are being deployed in 3rd party channels like Facebook Messenger, the platform also leverages the maximum encryption protocols available when transmitting assistant-based data to and from other communication channels.

Other features of the Kore.ai Experience Optimization (XO) Platform approach to the cloud connector include:

- **Persistent TLS Connection**

  A Transport Layer Security (TLS) tunnel connection is established between the Kore.ai Gateway (integrated with the Kore.ai virtual assistant server) and the Kore.ai Connector agent. This persistent connection utilizes TLS to send payloads. HTTP/HTTPS requests and responses can be multiplexed to allow a client to make multiple requests for efficient use of the tunnel connection.

- **Automatic Re-Connectivity to the Kore.ai Gateway**

  The Kore.ai Connector agent maintains a persistent connection with the Kore.ai Gateway. If network connectivity fails, including application server resets, the Kore.ai Connector re-establishes a new connection automatically.

- **Configuration Channels Audit Logging**

  The agent writes an audit log entry whenever an administrator changes any critical configuration settings, such as virtual mappings or permitted resources.

- **Scalable Load-Request Handling**

  The agent starts in cluster mode with the number of workers equivalent to the number of CPUs.

- **Configuration Channels Audit Logging**

  The agent writes an audit log entry whenever an administrator changes any critical configuration settings, such as virtual mappings or permitted resources.

- **X.509-Based Authentication**

  TLS communication is established by the Kore.ai Gateway issuing X.509 certificates to the Kore.ai Connector agent.

- **Flexible Access Control Support**

  Administrators can specify a virtual hostname and port for on-premise systems accessed by the Kore.ai Connector Agent without publicly exposing internal physical host information. They can also use a whitelist to specify resources or paths to be invoked on that host.

- **Secure Access With No Firewall Changes**

  The agent can be installed in the DMZ or in an inner network zone without changes to an external firewall provided port 80 and 443 are open.

Virtual Assistant Runtime

Virtual Assistant Connectors

To build, deploy, and manage virtual assistants using a scalable model, they need to be built in a way that balances rapid integration with channel-specific customization. The platform channel connectors simplify, standardize, and largely automate the process of deploying virtual assistants to multiple communication channels, with no code required. The connectors eliminate the time-consuming burden of tweaking, testing, and debugging every aspect of a bot's configuration to best fit the different formats and standards required by each channel, while still allowing for easy, channel-specific UI or response customization.

This section describes the Channel Connector features that make building an omnichannel virtual assistant faster and more repeatable using the platform.

Authorization Control

The Connectors allow virtual assistants to systematically adhere to channel-specific authorization and authentication nuances. This saves you from having to determine, for every channel within a single virtual assistant configuration, how a virtual assistant will compare a user's credentials to the information contained within that channel's authentication server in order to determine appropriate access to the channel and any relevant gated resources.

Channel API Handling

The Kore.ai XO Platform's Channel Connectors use APIs to automatically handle the distinct connectivity requirements of each available communication channel, such as linking to Facebook, SMS to a brokerage gateway, or email to email servers. It grants access to deploy virtual assistants across channels by simply checking a box.

Auto Formatting

The Connectors automatically adapt messages to match the unique formatting used by each communication channel. Virtual assistants can then be deployed to multiple channels from a single configuration. You can also leverage the specific user interface and user experience capabilities of each channel, and maintain consistency with the native channel experience. Additionally, virtual assistant responses can be customized on a channel-by-channel basis from a single configuration.

Live Agent Handoff

The Kore.ai platform allows a virtual assistant to hand off a conversation, based on context, to your human agents via the live chat systems they use. Context, in this instance, is used to route the existing dialog to the appropriate agent, including a complete history of the conversation. The live agent receives these messages, whether written or spoken, directly within their live chat software. The user receives these messages in the same thread and the channel they use to chat with the virtual assistant, ensuring a seamless transition from virtual to a human agent. The Kore.ai Platform allows for live agent handoff to be designed in two distinct ways:

- Rules/ Logic - You design a dialog that includes a node to hand off the conversation to a live agent when specific questions are asked or other specified actions are performed by the user. This is based on your predefined rules and business logic. For example, you may want the human agent to take over the conversation for certain types of questions or issues or when a customer's profile or behavior matches a specific customer segment or level.
- Sentiment - Leveraging the Platform's sentiment analysis tool, you can define sentiment thresholds for conversations, such as if the user sentiment is negative and below a score of 3.0, which automatically triggers the handoff process.

Since the Platform leverages the APIs provided by the live agent software, your team can continue to use the software and tools they already use without sacrificing functionality, including the call routing configuration and business logic that may already be in place in your contact center support software. The platform has proven integrations with LiveChat, LivePerson, Genesys, and Freshdesk. It can also easily integrate with other popular call center solutions as a custom feature.

SDKs

The platform offers Bot SDKs as a set of platform-specific client libraries that provide a quick and convenient way to get Kore.ai Experience Optimization (XO) Platform capability into custom applications and to provide you with more control for the virtual assistants you build on the platform.

Using our BotKit SDK that resides on your servers and interacts between the bot user and the Kore.ai Platform, you can intercept and customize messages and influence the flow of the conversation dialog.

Real-Time Messaging (RTM)

Kore.ai SDK libraries use a web socket or a webhook channel to establish communication between your client and the Kore.ai Experience Automation Platform. The platform exposes an RTM API for Server Event JSON responses to be sent to and from the client app.

- Bots Web SDK – The Bots Web SDK contains HTML5 and JavaScript libraries that you can integrate with web applications and hybrid mobile apps. The SDK is available in our Github repository at https://github.com/Koredotcom/web-kore-sdk.

- Bots Native iOS SDK – The Bots Native iOS SDK contains native iOS libraries that you can integrate with native iOS mobile applications. The SDK is available in our Github repository at https://github.com/Koredotcom/iOS-kore-sdk
- Bots Native Android SDK – The Bots Native Android SDK contains native Android libraries that you can integrate with native Android mobile applications. The SDK is available in our Github repository at https://github.com/Koredotcom/android-kore-sdk

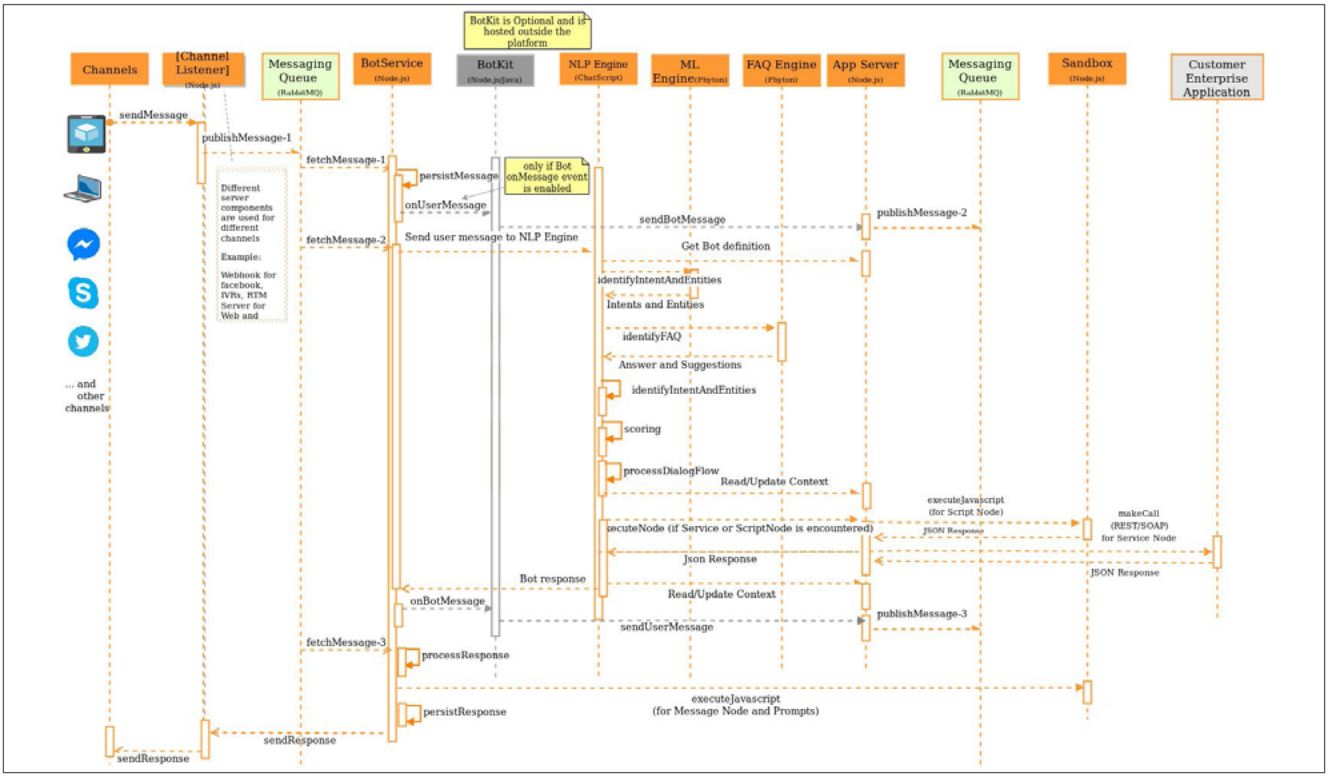
BotKit SDK

The Kore.ai Platform also includes a Node.js+express SDK consisting of a set of libraries that give you more control over the virtual assistants. This SDK lets you write custom extensions using template code to host callback services. Trust is established using auth tokens generated on the Platform. When building a dialog task using the Dialog Builder, you can subscribe to the message and webhook events by adding event handlers and then controlling them using the SDK.

The following callback events for tasks are supported by the SDK:

- onMessage: Triggered when a user sends a message to the virtual assistant and it is received by the channel adapter. The user message is wrapped in this event and sent to the SDK
- onWebhook: Triggered when the NL Engine processes the SDKWebHook node in the dialog flow. This event is sent to the SDK with component Id and context. The SDK can execute business logic and send the updated context back to the Platform.
- onAgentTransfer: Triggered when the Kore.ai NL Engine processes an Agent Transfer node in the dialog flow. This event is sent to the SDK with requestId and context.
- onEventNode: Triggered when Dialog Task or FAQ ends in the virtual assistant and sends request ID and context to the SDK.
- onAlertNode: Triggered when a virtual assistant user receives an alert.
- onVariableUpdate: Triggered on any of the following cases related to Virtual Assistant Variables

The diagram below provides a visual understanding of the user message flow involving Dialog SDK Webhook execution:



Analyze

Administrators have real-time visibility from a single dashboard and can get actionable insights into virtual assistant and account usage.

Overview Dashboard

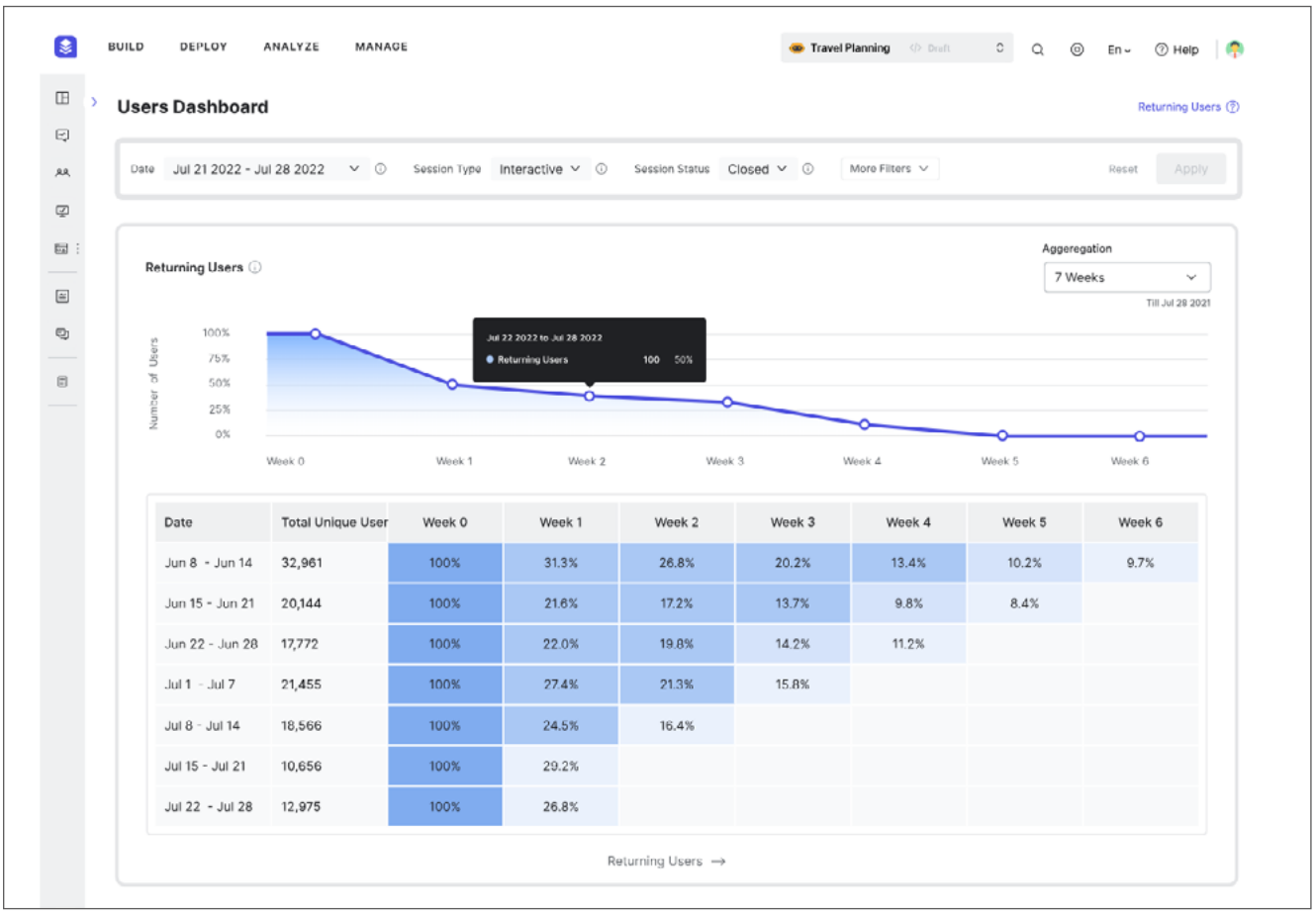
Once a virtual assistant is published, it is essential to understand the performance, users trend, usage and containment of the conversation session. The Overview dashboard gives a snapshot of the Virtual Assistant's conversations, user analytics, and performance over the period of time. The summary information on these metrics can be filtered by date, conversation type (interactive and non-interactive), conversation status and more.

Conversation Dashboard

The Conversations Dashboard is one such dashboard under analytics that provides insights into the usage and containment of the conversation sessions. It focuses on whether the calls were successfully answered by the VA, landed with agents or were incomplete. You can view many other metrics like conversation sessions trend, conversation path analysis, session distribution by channel, VA's engagement, etc with the breakup of self-service, drop-off, and agent transfer sessions.

Users Dashboard

Allows you to analyze the trends of the users interacting with your virtual assistant. The dashboard also provides metrics of new and returning users. You can also analyze user retention with cohort analysis.



Performance Dashboard

To understand the performance of a virtual assistant, a VA designer needs insights of how many intents were identified, how many tasks were successfully completed using the virtual assistant, how many services or scripts failed during the interactions etc. You can use the Performance dashboard to get all the insights related to performance of a virtual assistant. These insights will help a VA designer to improve the performance of the virtual assistant by making necessary changes to it.

Custom Dashboards

Custom Dashboards allow you to design your own reports and dashboards to meet business needs using a combination of built-in metrics as well as custom KPIs based on user demographic or context information. Custom Dashboards are in addition to the out-of-the-box dashboard reports available in the Bot Builder

NLP Insights

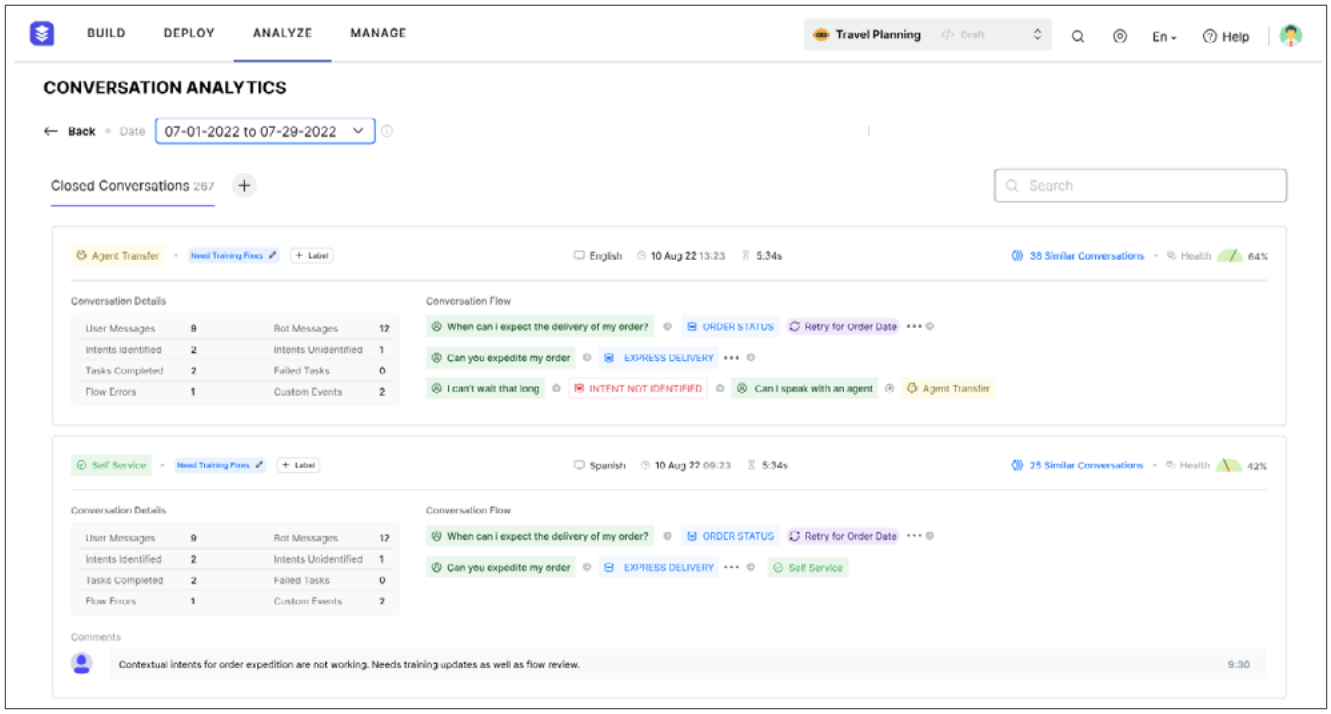
The NLP Insights give developers in-depth insights into their bot's performance at identifying and executing tasks. It lets you view necessary user utterances that matched and didn't match with intents. You will know Intents Found, Intents Not Found, Failed Tasks, Performance and Debug log.

Conversations History

The Conversations History module feature helps you analyze the user interactions by summarizing the key events identified during the conversation. Intent identifications, entity or confirmation retries, task execution failures, etc., are some of the key events tracked. Along with the summary, you can also see the complete conversation transcript to make data-driven decisions. You can also tag conversations that need follow-up actions and collaborate with your team to resolve them.

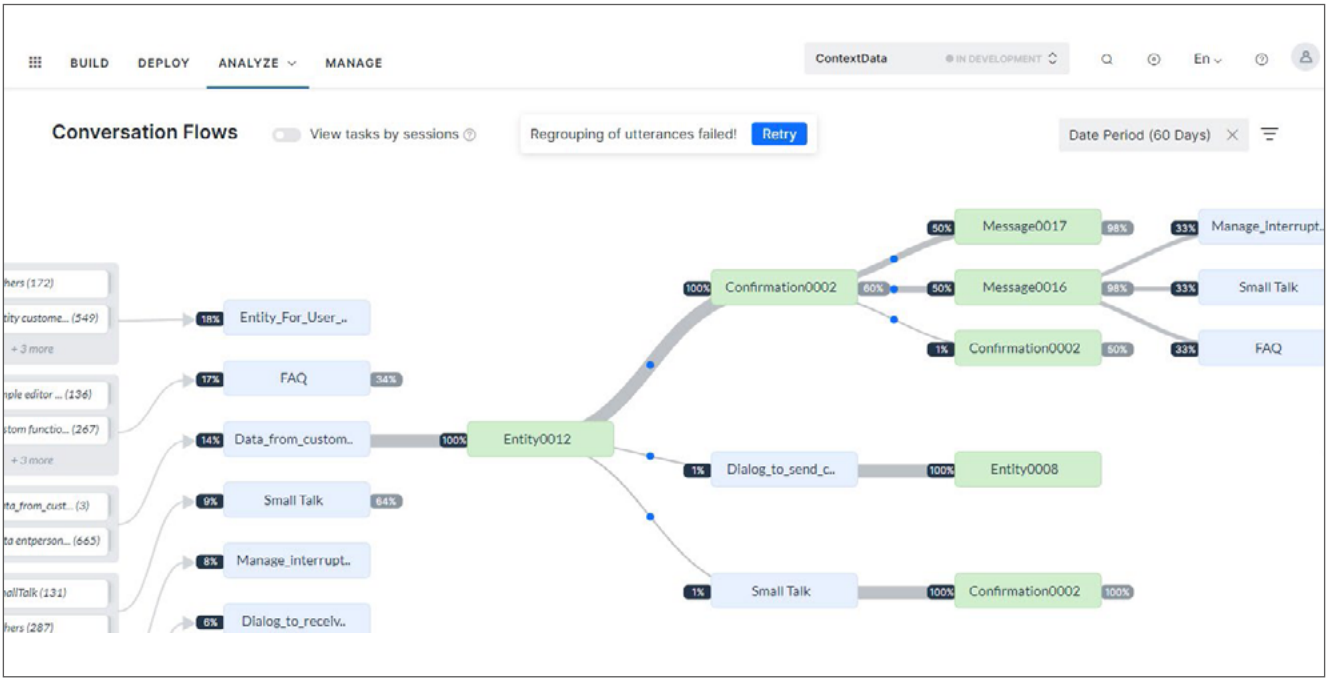
Conversation Flows

You can visualize the journey of various bot user interactions across conversation sessions in a given period. It helps developers understand the frequently used tasks of their bot, how users are invoking these tasks, popular task execution paths, dropout points, etc.



The Audit module provides the history of all administrative tasks with attributes such as User, Date/ Time, and Description for the event. The Audit module captures and displays events such as user logon failure, user enrollment, user and role management changes, changes to security settings, eDiscovery/ audit report access and much more.

As all virtual assistant conversations and interactions are logged and persisted in MongoDB, the schema definition of our logs can be made available and logs can be extracted from MongoDB and archived



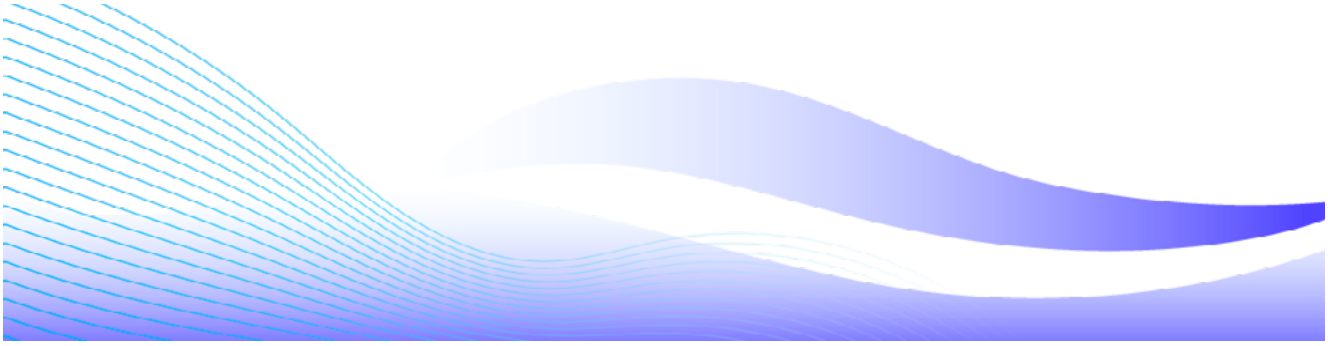
into your business records system. We support SMTP and XML/ JSON file-based integration into archive systems. The platform also provides the JSON schema definition of virtual assistant interaction and event logs so they can be exported and consolidated into common third-party analytics engines to generate additional insights.

Process Apps

The Kore.ai Experience Optimization (XO) Platform allows you to automate enterprise business processes. Develop workflow rules using the Kore.ai Process App Builder where human tasks, data, or files are routed between relevant stakeholders through a pre-defined path to execute the processes. Design multi-step, multi-user workflows using a rich set of building blocks like triggers, integrations, business logic evaluation, conditional flows and approval flows.

Conversational Experience

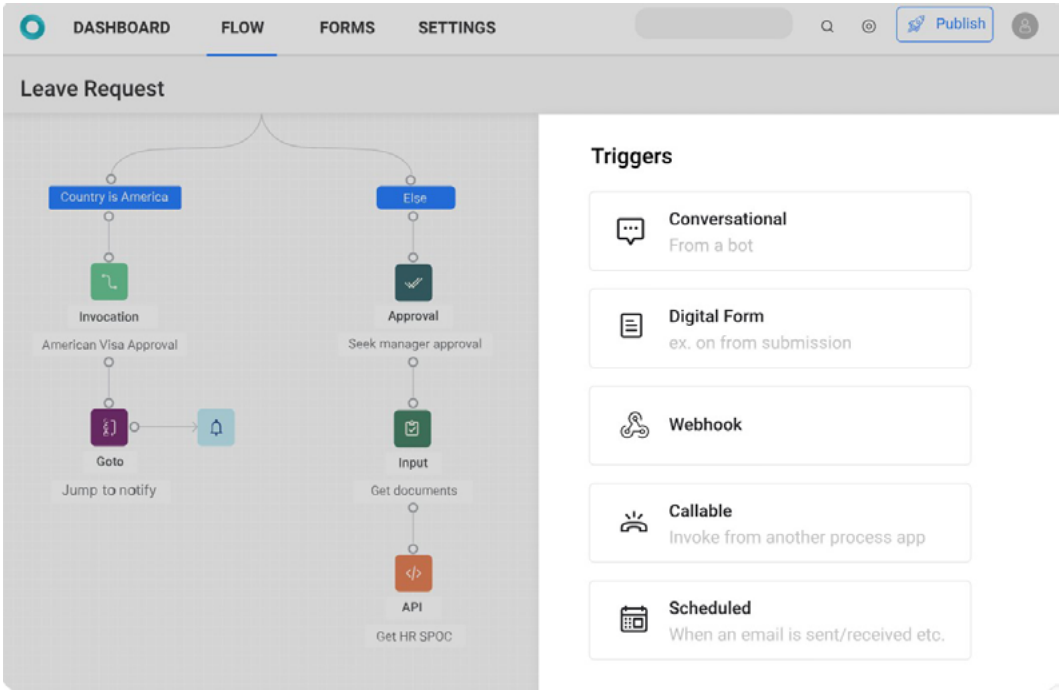
The process apps developed on the Kore.ai XO Platform are unique as they offer a conversational experience. It means you can seamlessly trigger a process by interacting with a virtual assistant.



Design and Build

Process App Builder

The no-code flow builder provides drag-n-drop tools to accommodate complex business workflows. The process builder acts as a designer that helps you to define the process flows. The following are the basic components of the process builder:



Triggers

Triggers are specific events that drive a specific process application based on conditions defined. There are different triggers that you can use to trigger a process flow based on the requirement. The following are the types of triggers:

- **Conversational -** Conversational Triggers refer to the triggers that occur through a conversational bot/virtual assistant.
- **Digital Form -** Digital Form triggers refer to the triggers that occur through the submission of a digital form.
- **Webhook -** Webhook Trigger allows you to trigger the Process App from a third-party application.
- **Callable -** Callable Triggers refer to the triggers invoked from another process app.
- **Scheduled -** Scheduled Trigger allows you to trigger the process app automatically at a scheduled time without any human intervention.

Logic

Logic refers to the method that deals with rules and processes used in reasoning. A Logic in a Process Assistant is used to direct the workflow to different branches with or without conditions in the workflow. It also helps you repeat actions or unify the branches created in the workflow. There are four types of logic involved in building a process flow:

- **Split -** Split logic helps you to split the process into two or more branches and define the branch execution.

- **Merge -** Merge logic helps you merge individual process flows into a common flow by defining business logic.
- **GoTo -** GoTo logic helps you to conditionally switch the flow to a different step in the process flow.
- **End -** End logic helps you to end a process flow.

Tasks

Tasks are the key components of a workflow. It allows workflows to interact with users using human tasks or modify the data in the workflow using system tasks to provide an outcome for a business process. In a Process Assistant tasks are further classified into:

- **Approval Task -** It allows you to perform an approval action.
- **Form Task -** A task where a digital form is sent to the recipient to gather necessary inputs.
- **Script Task -** It allows you to add or modify the data to use during the execution of a flow.

Integrations

Integrations allow you to send or receive data from your process assistant to other internal flows or third-party apps. In Process Assistant, integration is subdivided into two:

- **Connect to Data Table -** It allows you to store, read, and modify the data in the data tables on the platform.
- **Connect to API -** It is a service request made to a third-party application to help both applications communicate with each other to share the necessary information.

Events

Events are used to pause and resume actions in the process flow. It also helps you to notify the user of actions performed in the business process. There are two types of events involved in building a process flow:

- Call a Subflow - This event allows you to invoke an independent process assistant.
- Notify - This event helps you notify users through the channels.

Forms

The Kore.ai platform's Digital Form provides a range of input fields required to capture the details from end-users. After the user completes the form, the input is submitted to the virtual agent to proceed with the task at hand.

Test (Simulate)

Simulate the Process App by selecting a trigger. The primary purpose of the simulator is to test the workflow. It displays all triggers applicable to the process app and you can choose a trigger to simulate.

- If the trigger type is a **Form**, the system opens up the form to submit.
- If the trigger type is **Callable**, the system displays all the process apps that have to Call a Subflow task and are associated with the current process app.
- If the trigger type is **Scheduled**, the system allows you to schedule the process app execution.

- If the trigger type is **Conversational**, you can select the virtual assistant to trigger the process app. Then, open the chat window in the process app to trigger it.
- If the trigger type is **Webhook**, the system displays a text message against each trigger as shown below:
  - » Webhook Trigger – The system displays a text stating Please do a post-call to the endpoint URL, and if posted successfully, check the execution status under simulator instances.
- If the trigger type is Email, the system displays the below details to trigger a process app:
  - » From Address
  - » To Address
  - » Subject
  - » Compose message
  - » Trigger

Deploy

Self-service Portal

The platform offers a dedicated user portal to monitor all process apps in one place. In addition to the third-party apps, users can also access the process applications through the built-in user portal. They can create new requests, view the pending tasks, perform the required actions, or track progress.

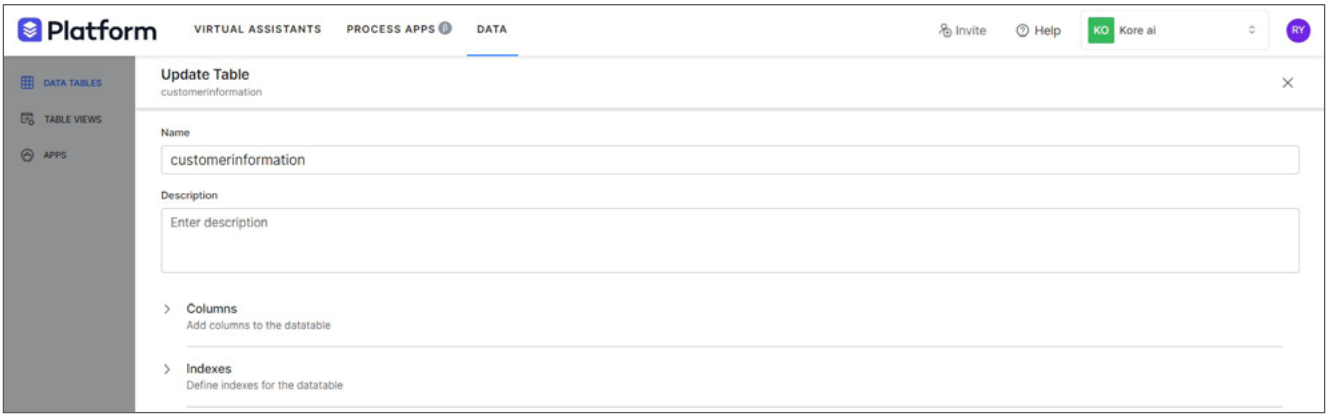
Dashboard

The Dashboard gives an overview of your Process Assistants. It allows you to track the real-time status of your application. You can find the details of the instances, tasks, usage metrics, and task trends on the dashboard. You can also filter the data based on a period (24 hours, 7 days, custom date). The dashboard refreshes every 10 seconds and you can also change the refresh time from the drop-down list.

Data Tables

The Kore.ai Experience Optimization Platform provides API-based integrations with all the popular data systems. In addition, it also allows you to build your own Data Tables for storing and managing information related to virtual assistants. You can build it from scratch or import table definition using valid JSON files. Associate tables to multiple virtual assistants and process apps to capture the information.

Table Views let you create custom datasets by joining one or more Data Tables. These definitions get converted to queries to extract data from one or more data tables using the required joins.



Enterprise Admin Console

The platform was designed from the ground up to make managing and protecting users and virtual assistants easy and scalable. This section describes the Enterprise Admin Console features for provisioning and restricting access to your virtual assistants, managing omnichannel deployment, and taking advantage of real-time actionable insights.

Enrollment

The platform makes it easy to invite and enroll new users in your domain and to review a complete history of bulk, file-based, and AD Sync user imports. The enrollment module offers the following options:

- Direct: Invite users manually via email invitation
- Bulk: Create a basic .txt invite file that contains domain member email addresses
- Import: Add new users and update existing users based on a list that contains additional name and profile information like department, phone number, and job title
- AD Sync: Configure the platform to synchronize with your organization's LDAP Active Directory, define connection information, create exclusion rules to prevent unwanted updates to existing users and groups, define distribution lists to import, define organizational units to import, and set up automated scheduling for future syncing
- The platform can either connect directly to your AD server and request data using standard authentication, or you can configure a Kore.ai Directory Agent in your company Intranet to push data to the Kore.ai platform as needed.

User Management

The platform makes it easy to manage your newly enrolled and existing users with tools for updating user information, adding or deleting users, creating groups, and more. The Kore.ai platform's enterprise administration console provides the following user management capabilities:

- Review and modify user profile information
- View group membership
- Reset user passwords
- End user sessions
- Convert managed users to unmanaged users
- Activate and deactivate user profiles

Groups

Administrators can create, modify, and delete custom groups of managed users manually via the user management module or automatically via AD sync. Custom groups, such as those created across department lines or geographic locations, make it easy to manage large numbers or users.

Distribution Lists

If Active Directory sync is configured for your domain, The platform provides a module so you can view distribution lists that have been imported for your domain. These lists can consist of both managed and unmanaged users.

Admin Roles & Permissions

Assign, define, manage, and transfer built-in and custom administrative privileges to managed users and groups. The custom roles dashboard lets you provide or deny access to one or more specific sections of the admin console, such as enrollment, user management, virtual assistant management, roles and permissions, security and control, billing, analytics, and compliance.

Message Middleware

Middleware is required for virtual assistants to exchange and transport information between distributed users and systems seamlessly and in real-time. Kore.ai's Message Middleware specifically manages the movement and processing of information, including reliably formatting, passing, and securing messages between users, virtual assistants, and enterprise cloud and on-premise systems. – regardless of programming languages, protocols or Operating Systems.

This section describes the components of Kore.ai's Message Middleware and the role each plays to make conversational interactions function flawlessly.

Message Broker

The Message Broker consumes user inputs from any communication channel and system outputs from the cloud and on-premise systems. It redirects (i.e. routes and forwards) the message through the platform, such as via the NLP and intelligence engines, and then on to the appropriate endpoint channel using virtual assistants as the common user interface, or to the appropriate system via an API call.

Message Store

The Message Store captures and saves all messages sent to-and-from identified and unknown users, virtual assistants, and systems, both before and after the Broker deliver the message to the appropriate destination. These logs are persistent, segregated by the user, and maintained inside a secure MongoDB.

You can access and review the complete logs of user-to-bot interactions contained in the Message Store via the Kore.ai Virtual Assistant Builder tool. Every message gets categorized as a success or failure by the NLP engine, so you can assess why an interaction failed and easily address the failure by optimizing the virtual assistant configuration. Administrators also use the Message Store logs to perform compliance-related checks via the Platform's enterprise Administration Console.

Built-in Encryption

As messages flow through the Broker and get stored on designated servers, the Platform automatically encrypts all enterprise and user data with an AES 256 cipher.

The encryption key for this data is reissued every 60 days, or whenever manually refreshed by an enterprise administrator. The platform also automatically leverages the maximum encryption available when transmitting virtual assistant-based data to destination deployment channels.

Key Benefits of Middleware

- Increases efficiency and lowers the cost of ownership by eliminating the need to design, build, and host your own message servers/hubs to organize user to system and system to system communication
- Simplifies complex operational environments consisting of different programming languages,

communication protocols, dependencies, and operating systems by providing a common messaging paradigm for distributed systems

- Improves information integrity between multiple systems with built-in routing logic, the transformation of messages based on source and destination format, and automatic storing of all message types

Security & Compliance

The Kore.ai Experience Optimization (XO) Platform follows the strictest guidelines for security and compliance regardless of industry. From personally identifiable customer information to proprietary business data, the Platform allows you to meet strict requirements passed down from local, state, and federal regulatory and governing bodies.

This section describes the security and compliance methods the platform uses to protect your data in accordance with the highest industry standards. It also describes the features, contained within the Enterprise Admin Console, to take advantage of built-in data retention policies such as eDiscovery, audit logging, and more.

Encryption

Method

The platform uses the latest AES-256-CBC (cipher block chaining) implementation with 16-byte random initialization vectors (IVs). The platform provides full encryption of virtual assistant messages on the server. All application data at rest in the database, Search Indexes and SAN storage is encrypted. The platform also provides full encryption of all virtual assistant messages during transit. The platform uses HTTPS over Transit Layer Security (TLS) using AES 256 standard.

Encryption Key Management

The platform uses two sets of keys: a master key and enterprise data keys. Each enterprise is assigned a data key. Enterprise-specific data keys are kept encrypted using the master key. The master key is provisioned and stored using the HSM-backed Key Management Service.

Enterprise data keys are rotated periodically or on-demand.

Enterprise administrators can manage data keys from the Admin Console. The administrator also can view the key meta information and rotate the key on demand if required. The newly generated data key will be used to encrypt the data from that point of time onwards.

Authentication Layers

SSO

Industry standard SSO solutions are leveraged to grant managed users secure access to an organization's enterprise app store and application using existing identity providers.

Supported SSO Configurations

- SAML
  - » Okta
  - » OneLogin
  - » Bitium
  - » Other (generic SAML identity provider configuration)

- WS-Federation
- Ping Identity
- OpenID Connect

Integrated System Authentication

Use basic auth, OAuth, or API keys for user authentication before virtual assistants can deliver alerts or take actions against integrated systems.

Configurable Password Policy

Define and enable default password strength when Single Sign-On (SSO) is not used. The platform includes support for dual-factor authentication.

Administrator can create policies with the following properties:

- Minimum length of characters and letters
- Inclusion of at least one numeric value
- Inclusion of at least one letter
- Enforce the use of at least one uppercase letter
- Enforce the use of at least one lowercase letters
- Inclusion of at least one special character
- Password expiration (between 1 and 365 days)
- Policy applies to existing users or only new users

Domain Verification

For additional security, the platform requires that you verify your domain and register your Kore.ai account to your domain before virtual assistants can be used. Kore.ai provides a unique. TXT record to add to your domain settings or <meta> tag for your website home page. These tokens let you initiate a domain verification check in the Virtual Assistants Admin console. After verification, the platform provides detailed information related to your domain, such as the verification method used, date, and any associated comments.

Compliance

Data Retention Policies

The platform automatically retains user and virtual assistant messages for 365 days with built-in data retention policies. The Enterprise Admin Console lets you create custom retention policies for each registered domain based on your organization's unique requirements for data persistence and compliance.

Archiving/ Exporting

Kore.ai lets you extract user-virtual assistant conversations from the platform for archival purposes. The schema definition of the logs can be extracted from MongoDB and exported into your business records system using SMTP or XML/JSON file-based integrations.

eDiscovery

Kore.ai provides a platform that facilitates compliance with federal regulations for HIPAA, PCI, FINRA, and others. The Enterprise Admin Console lets you perform eDiscovery searches for a user and virtual assistant content, across all deployments, usually for civil or government investigations. Evidence surfaced by these searches can be retained for any specified duration. Administrators can also provide secure custodian access to on-hold content for internal or external inquiries.

Profanity Filters

The platform supports integration with third-party services that provide profanity filters; user utterances can be sent to the third-party service to be tagged and filtered in real-time (if supported). The Kore.ai platform NLP engine also has the ability to identify profanities.

Audit Logs

The platform provides full auditing capabilities of all system events, such as eDiscovery searches performed and error messages, and user-generated events, such as the addition of a new managed user. Each logged event provides detailed information such as the date and time stamp of the event, category, description, user that initiated the event (when applicable), and the IP address of the device or channel used to access the assistant. The Enterprise Admin Console also lets you create and filter custom reports by event category, time period, and more.

Security Protocols & Compliance Reporting

Rigorous Penetration Testing

Kore.ai servers are regularly scanned for vulnerability and monitored by intrusion detection and file integrity agents. We combine exhaustive internal security practices with third-party penetration audits, run on a periodic basis, to ensure data integrity and security and that industry best practices are consistently followed.

Data Center

Kore.ai's services are hosted within an Amazon Web Service environment. This environment maintains multiple compliance certifications, including ISO2001, PCI, and SOC. Please visit https://aws.amazon.com/compliance/ for more information.

Service Organization Control Report (SOC)

Kore.ai is SOC2 Type 1 and Type 2 compliant. A report of our most recent audit can be shared upon request by help@Kore.ai.com.

The Kore.ai Advantage

At Kore.ai, we believe the future of business is digital, and that the winners will be those who blend convenience and speed with a personal and human touch. And that customers, partners, and employees should be able to communicate with companies, systems, smart machines, and more in the same way they talk to their friends and colleagues. Our mission then is to help you build and deploy conversational AI solutions that make interactions simple and effective and that help automate and streamline routine tasks and processes.

That's why we built our conversational AI-powered Experience Optimization Platform - to provide you with everything you need to create, perfect, deploy, and manage virtual assistants and automation solutions in a manner that is consistent, scalable, customizable, and secure – without months of development time and expenses.

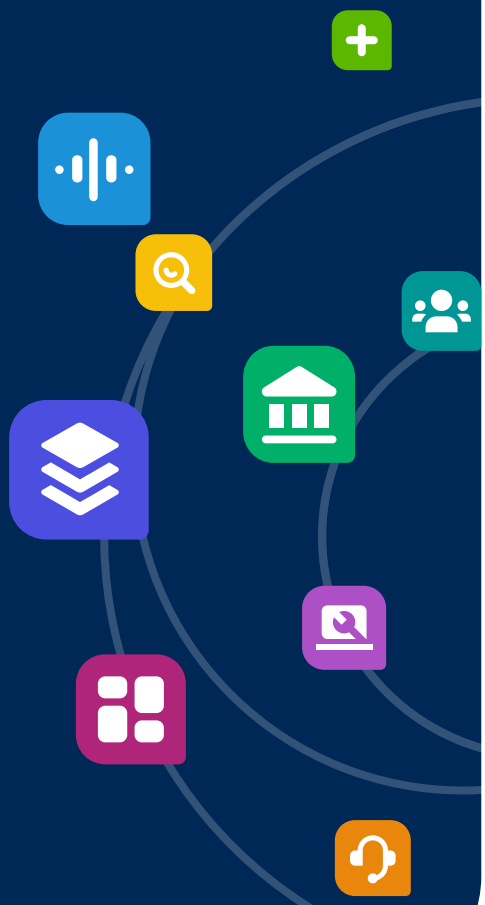
Here are a few reasons why you should choose Kore.ai

- Complete, end-to-end platform designed specifically to exceed the enterprise requirements
- Intelligent dialog turn management that allows your virtual assistants to handle virtually all nuances related to human conversations
- Zero to minimal coding required and easy to use virtual assistant and dialog builder tools, which enables business users to create bots instead of just IT
- The lowest total cost of ownership of any Virtual Assistant Platform on the market today with a flexible SaaS business model
- Powerful multi-pronged NLP engine that allows for better conversations - without tons of training data
- Intuitive consoles with rich analytics and comprehensive enterprise administration, security, and compliance functionality
- An open and extensible platform that enables seamless integration with 3rd party functions and engines, including NLP, speech, logic, and more
- Fastest time to market, empowering enterprises to go live in as little as 30 to 60 days - even for complex use cases

Globally recognized:

We have been recognized leader in

- Gartner® Magic Quadrant™ for Enterprise Conversational AI Platforms 2022
- Everest Peak Matrix Conversational ai Technology, Sep'21
- IDC MarketScape: Worldwide Conversational AI Software Platforms for Customer Service, Oct'21
- ISG Provider Lens - Intelligent Automation Solution & Services, Mar'21
- Opus Research's Decision Makers' Guide to Enterprise Intelligent Assistants, Feb'21



Kore.ai Named a Leader in **2022 Gartner® Magic Quadrant™** for Enterprise Conversational AI Platforms

Select customers

GoDaddy · AT&T · yahoo! · ABInBev · citi · PNC · McDonald's · Pfizer · Coca-Cola · AIRBUS · ManpowerGroup · Roche · Cigna · Florida Blue

Kore.ai is a global leader in the conversational AI platform and solutions helping enterprises automate front and back office business interactions to deliver extraordinary experiences for their customers, agents, and employees. More than 200 Fortune 2000 companies trust Kore.ai's experience optimization (XO) platform and technology to automate their business interactions for millions of users worldwide to achieve extraordinary business outcomes.

7380, West Sand Lake Road, Suite 390, Orlando, FL 32819.
+1-321-420-0990 | info.kore.ai | www.kore.com

kore.ai