**SmartAssist** is an automation-first, customer intent-driven contact center solution. It is available in the cloud and on-premises deployment modes and provides an intuitive, omnichannel solution for inbound customer service.

Based on customer intents/goals, you can tailor customer experiences to your business and customer needs. With SmartAssist, you can automate regular customer queries and seamlessly transfer them to a human agent as needed.

SmartAssist is built upon and integrated with the [Kore.ai XO Platform](https://kore.ai/platform/), an enterprise-grade VA development platform offered as a SaaS, by Kore.ai. The XO Platform enables enterprises to design, train, deploy, analyze, and manage AI-rich Virtual Assistants. It leverages both an enterprise-grade automation platform and a dedicated contact center solution that meets the needs of management, agents, and customers.

Smartassist offers different features for Experience Designers, Administrators, Customer agents, supervisors of agents, and business users (contact center clients)

Smartassist features include automating, redirecting or transferring to agents; design customer experience flows (like waiting experience), use cases creation to integrate KG into automations.

Smartassist for agents allows to handle omnichannel queries within one screen, which allows agents to seamlessly interact with customers no matter how they get in touch. AgentAssist features that support agents throughout their work with proactive answer suggestions.

To understand how smartassist works, follow <https://docs.kore.ai/smartassist/getting-started/introduction/#How_SmartAssist_Works>

## Lifecycle:

* **Design experience flows**: define the end customer experience in an easy-to-use builder. SmartAssist lets you create all experience flows from scratch using the **Experience Flow Designer**.
* **Configure Use cases**: configure the use cases for which customer experiences need to be defined.
* **Define Behaviour:** customize the messages for standard scenarios
* **Incoming channel setup**
* **Test**
* **Deploy**

## Experience Designers:

### Experience flow: An *experience flow*is a step-by-step process that helps you define the end-to-end customer experience at a contact center for each communication channel. Experience Flows can be built using the Experience Flow Designer.

There are different types of Experience Flows in SmartAssist:

1. Inbound Flow: Most flows are classified as inbound flows, applicable to both voice and chat channels.
2. Call Disconnect Flow: Used when a call disconnects and only applicable to voice channels.
3. In Queue Flow: This flow type can be attached to the Agent Transfer node.
4. Default out of operations flow: This flow type runs when a customer tries to reach an agent outside working hours either on a voice call or chat.
5. Default no agents available flow: This flow type runs if the skill combination needed by the customer does not match any agent but the conversation is within working hours.

### Node Types:

#### IVR Menu

An IVR menu node allows you to create a standard IVR menu that can be presented to customers during inbound calls. Adding this node helps route the customer to the relevant call experience based on the key they input. For example, in the case of a travel agency, pressing ‘1’ will route the customer to the booking update flow, pressing ‘2’ will execute the booking cancellation flow, and so on.

#### IVR Digit Input

The *IVR Digit Input* node records a non-IVR menu interaction. It is used during dynamic interactions when the caller needs to provide input such as a phone number or the last four digits of their credit card to proceed. You can configure the condition for receiving the input, the action if the condition is satisfied (success flow), the timeout value in seconds for the contact’s response/input, and the error-handling use cases. You can implement this node within your Experience Flows.

Input is stored as *context.steps.<<context\_node\_name>>*

#### Conversational Input

This node is an equivalent of a bot automation interaction for an IVR key input. The interaction flow is handled based on the open-ended input provided by the caller for a welcome message or prompt from a Bot and on the use cases defined for the Bot.