Composer Help

What is GVP and How Do Voice Apps Work

4/20/2025

Contents

What is GVP and How Do Voice Apps Work

The Genesys Voice Platform (GVP) is a VoiceXML-based media server for network service providers and enterprise customers.

What is GVP?

At the most basic level, Genesys Voice Platform (GVP) is Interactive Voice Response software (soft IVR). At a more complex level, GVP is a software suite that integrates a combination of call processing, reporting, management, and application servers with Voice over IP (VoIP) networks, to deliver web-driven dialog and call control services to callers.

Using features such as Automatic Speech Recognition (ASR) and Text-to-Speech (TTS), GVP provides a cost-effective way to implement automated voice interactions from customers calling your contact center. At the technology level, GVP is a collection of software components that complement and work with other Genesys products in order to provide a complete voice self-service solution.

Notes:

Whereas GVP is commonly used in enterprise self-service environments, many other applications of GVP —including those outside of the contact center—are possible.

A machine on which GVP components are installed is also referred to as a GVP Server in other places in this Help system.

Use Case

An example flow for a GVP voice self-service application is presented below:

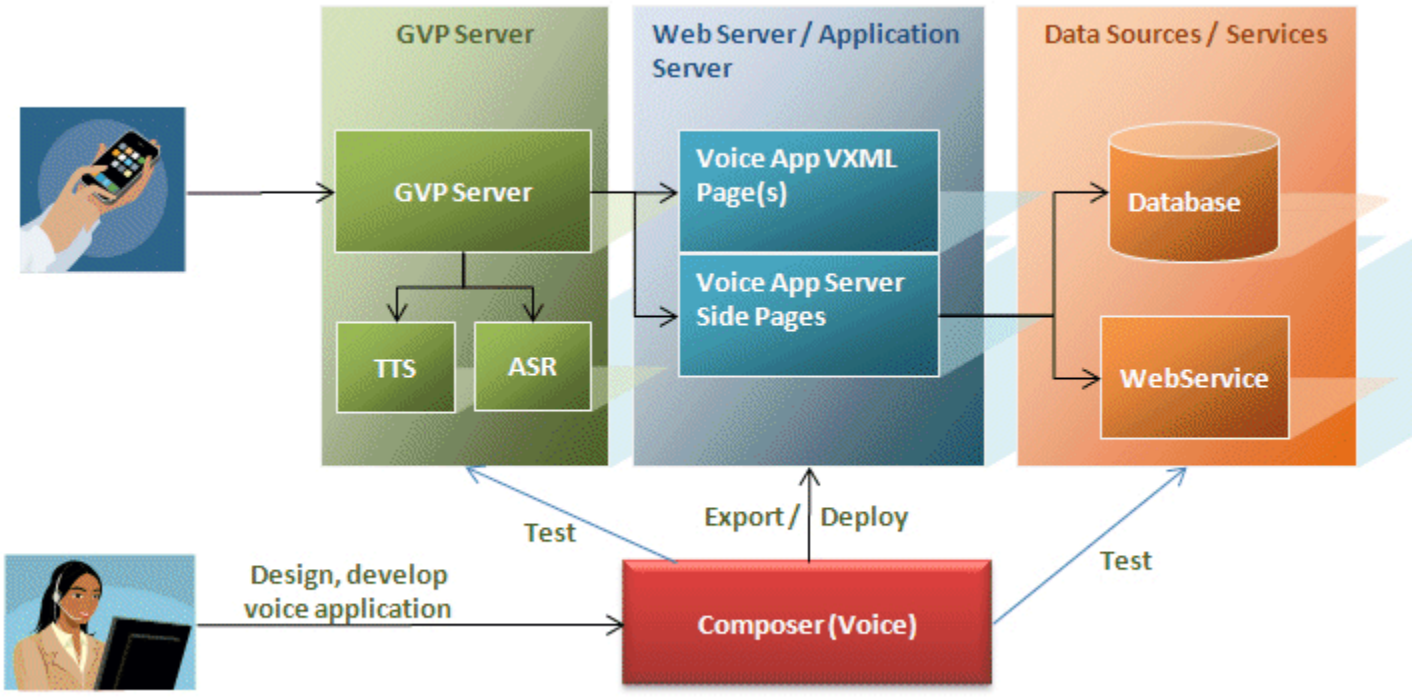
- A customer calling into an IVR would get Prompts / Announcements with a Welcome message. These prompts could be specific to a region based on incoming DNIS number or customized based on user options, such as a prompt to select a language.
- The applications could have business control logic allowing Business users to define open and close hours, set emergency announcements and flags, define special days, and so on. These options could be defined within Genesys Administrator or Genesys Rules System.
- The application can have the capability to collect user input with multiple options and repeatability to handle no input / no match capabilities.
- Once the application gets the user input, such as account numnber, this information can be verified against back office applications or can have the capability to pull data from a Conversation Manager Solution
- Customer information retrieved can be attached to the interaction as user data and could be used for Customer Segmentation and thereby determine how a particular customer would be managed within the self-service

- The application can provide self-service with Voice Recognition options and DTMF options in menus.
- The application can be configured to allow users to opt out of the self-service by default or within the predefined flow of the application.
- When users opt out of the IVR, they could be routed to an external source or to an Agent based on the environment as defined above. For example, customers could make a choice to get routed to “Last Called Agent” or be routed to any Agent by pressing repetitive “0”.
- Customers can have defined activities across the flow of the self-service and also be able to set milestones to define success/failure criteria within each segment of the flow.

How Do Voice Applications Work?

Just as one uses HTML to create visual applications, VoiceXML is a mark-up language one uses to create voice applications. With a traditional web page, a web browser will make a request to a web server, which in turn will send an HTML document to the browser to be displayed visually to the user. With a voice application, it's the VoiceXML interpreter that sends the request to the web server, which will return a VoiceXML document to be presented as a voice application via a telephone. What makes VoiceXML so powerful is that all of the most popular tools for making web pages are available for making voice applications. Developers can use technologies they are already familiar with such as JavaScript, JSP and ASP.NET/C# to generate exciting new voice applications.

The "Big Picture"



Composer is a fully featured VXML application development tool. Users can develop, debug and test their applications in its Integrated Development Environment (IDE) that provides developer-friendly features to test and debug VXML applications and server side web pages. Once the application is ready, it can be exported or manually deployed using an exported package onto an application server/web server like Tomcat or Microsoft IIS. Once deployed, GVP can access the voice application's VXML pages and any server side pages (JSP/ASP.NET) using HTTP.

When a call comes in to GVP, GVP determines the location of the VXML application through its provisioning data. It then fetches VXML page(s) and uses its VXML engine to execute them. The results are played back to the caller on his/her phone. Any server side pages that access databases or web services or other server side pages are executed on the application server/ web server through server side constructs implemented by Composer.

During development, Composer can use its bundled Tomcat or a local installation of Microsoft IIS as the web server and make test calls to the application right through GVP from within the IDE. This feature provides a quick way to test applications by removing the need to of deploy applications to another server and then point GVP to that location.

Once the application is deployed in production, Composer is no longer in the picture. The application is usually deployed on its own dedicated web server and application server from where it is accessed by GVP. The web/application server provides access to all pages and scripts that make up the application and executes any server side pages of the application.