Omni-channel Predictive Routing Blueprint

Reference Architecture

Authors Gordon Bell

Version: 1.0

Status: Publish

Published: 5/3/2018

GENESYS™

Table of Contents

Table of Figures

Table of Tables

Introduction

The purpose of the Omni-channel Predictive Routing Solution Blueprint document is to provide a set of design practices and guidance to ensure consistent architecture approaches are used for all deployments of the Predictive Routing. It provides a prescriptive list of components (both Genesys and 3rd party) that should be included in the solution. It also provides deployment guidance, including sizing considerations, and addresses several system concerns such as security, high availability, disaster recovery and serviceability.

The Genesys Omni-channel Predictive Routing Solution consists of the following core Genesys components:

- AI Core Services
- Agent State Connector
- URS/Orchestration and Framework components

The solution provides predictive Routing for all channels including voice and digital. The components for those channels are implicitly part of the solution.

Document Overview

The document contains the following sections:

- Chapter 2: Definitions and Acronyms
- Chapter 3: Overall Architecture
- Chapter 4: Deployment View
- Chapter 5: Interaction View
- Chapter 6: Implementation View

Intended Audience

The Blueprint Architectures are intended to provide Genesys Solution Consultants, Professional Services and partners with information on the general architecture design and considerations for the solution. The information provided in this document should meet the needs of pre-sales and provide appropriate general guidance for professional services. This document is not intended to provide configuration level information for professional services.

Describing system and solution architectures can be difficult as there are multiple audiences each with different expectations. This document is intended for multiple audiences with various chapters being more interesting to some readers than others. It is expected that readers will already have knowledge and training on Genesys products. This document provides high-level information for completeness.

The Overall Architecture and Deployment View are likely meaningful to most audiences. However, the Interaction View and the Implementation View may be of more interest to those configuring the network and components.

Definitions, Acronyms, and Document Standards

Definitions

This document uses various abbreviations and acronyms that are commonly used in Genesys product documentation and the telecommunications and contact center industries. The following table defines terms that will be referenced subsequently in this document.

Glossary

ASC	Agent State Connector
DB	Database
DBMS	Database Management System
FTP	*File Transfer Protocol*
GA	*Genesys Administrator*
GAX	*Genesys Administrator Extension*
GIM	*Genesys Info Mart*
GIR	*Genesys Interaction Recording*
GI2	*Genesys Interactive Insights*
GUI	*Graphical User Interface*
HA	High Availability
HTTP	Hypertext Transfer Protocol
ICON	Interaction Concentrator
IM	Instant Messaging
IP	Internet Protocol
ISCC	Inter-Server Call Control
IVR	Interactive Voice Response
JDBC	Java Database Connectivity
LAN	Local Area Network
NMS	Network Management System
ODBC	Open Database Connectivity
ORS	Orchestration Server

RDBMS	Relational Database Management System
REST	Representational State Transfer
SDK	Software Development Kit
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLib	TServer Library
UI	User Interface
URS	Universal Routing Server
VM	Virtual Machine
WAN	Wide Area Network
WDE	Workspace Desktop Edition

Document Conventions

The following documentation and naming conventions are used throughout the document:

- Code and `configuration property names & values` will appear in `console font`.
- References to other documents are bracketed ([]).

Overall Architecture

Omni-channel Predictive Routing solution draws on accumulated agent and interaction data to analyze and generate models to predict outcomes that can then be used to provide the best possible match between interactions and agents.

Solution Overview

The Omni-channel Predictive Routing solution builds on top of the URS and Framework components. The solution enhances Genesys Routing with Machine Learning to predict the best match for the incoming interaction.

This solution can be used with both Voice and Digital channels.

In addition, standard reports and dashboards are provided. These make use of the Reporting Common Components (GII, GCXI and Pulse).

Logical Architecture Model

The following diagram depicts the logical architecture for the Omni-channel Predictive Routing solution.

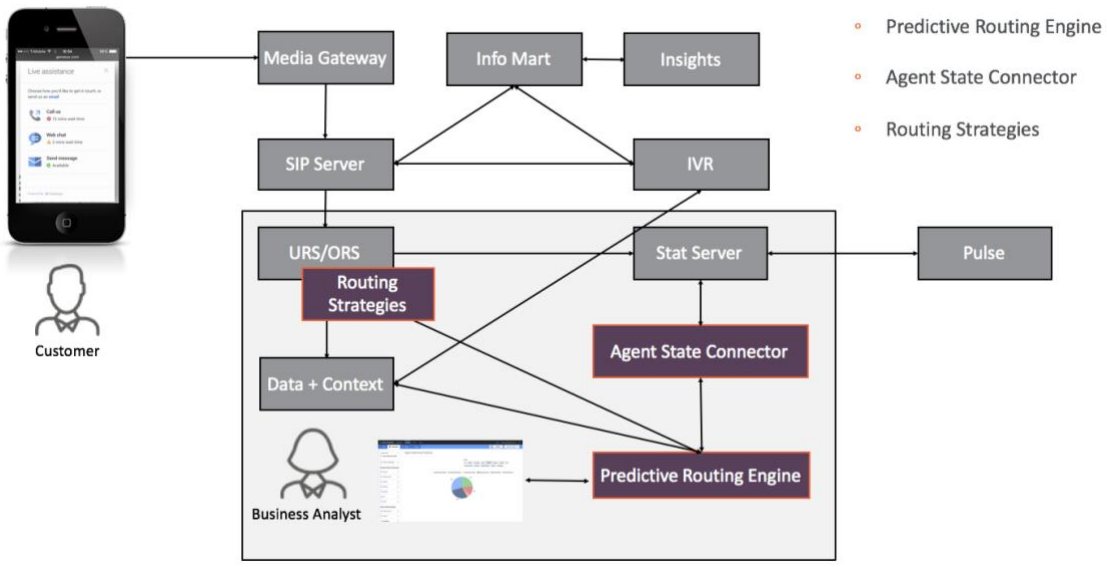

Figure 1: Logical Architecture Model

The solution is based on standard Voice and Digital deployments. It adds the following:

- AI Core Services component that hosts the Predictive Routing application and training modules,
- Agent State Connector to synchronize state with StatServer
- Predictive Routing Strategies for URS

The solution integrates with Routing and Reporting.

Functional View

Omni-channel Predictive Routing uses machine learning to enhance and improve Genesys routing to get each interaction to the best possible agent.

SMART Use Cases

BO06 – Improve routing accuracy using machine learning to route to the best agent

SL06 – Improve sales conversion rates using machine learning to route to the best agent.

Component View

The following components make up the overall Predictive Routing solution.

Component	Description
AI Core Services	Core component of the predictive routing solutions. It handles all analytics and administrative services. Provides a REST-API for flexible integration and serving end user web applications for model development, data analysis, and reporting.
Agent State Connector	Bridge between Pure Engage and Predictive Routing. It monitors agent state information from ConfigServer and StatServer and provides updates

	to the AI Core Services
Common Components	See the Common Component Blueprint for information on Management Framework, Orchestration/Routing and Reporting components used by this solution. Note that a set of subroutines are added(?) to URS for predictive routing. These include: ● **ActivatePredictiveRouting** fetches scored agent lists for interactions, storing results in URS ● **ScoreIdealAgent** looks up the ideal agent for an interaction based on the scoring algorithm ● **IsAgentScoreGood** evaluates if the scoring is sufficient to use that match for an interaction
Digital Channels	See the [Digital Solution Blueprint] for details on configuring the various digital channels that predictive routing can be applied to.
SIP Voice	See the [SIP Voice Solution Blueprint] for details on configuring SIP VoIP infrastructure that predictive routing can be applied to.

Table 1: Solution Components

Digital Channels and SIP Voice are included in the overall solution as potential channels that predictive routing can be used to target the best agent.

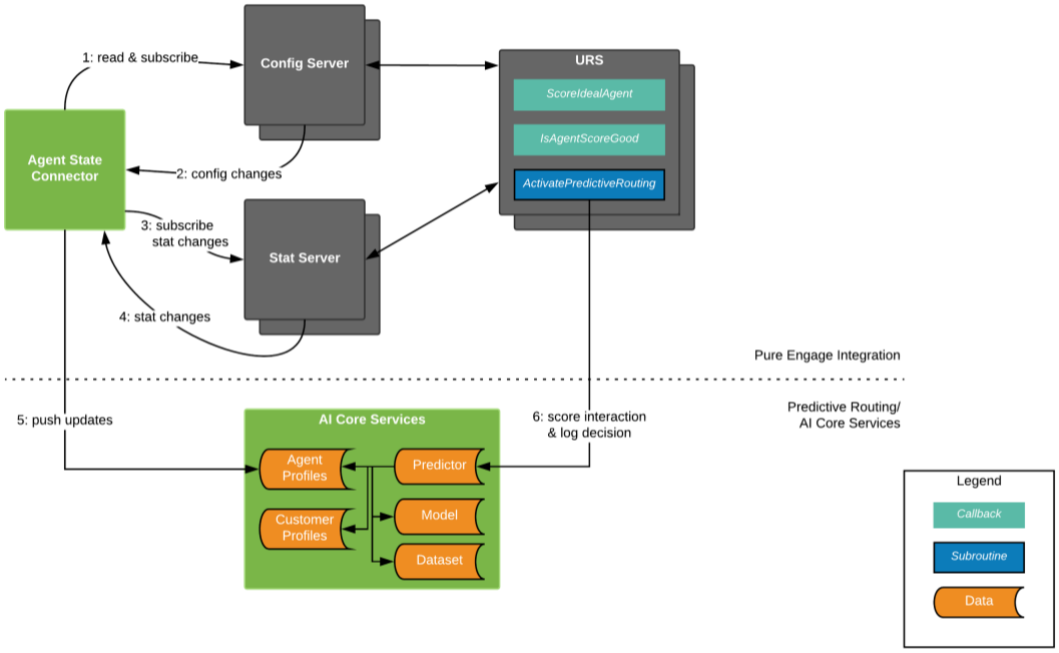

Figure 2: Logical Runtime Architecture

The logical runtime architecture diagram depicts how the various components function within the solution.

- The Agent State Connector reads configuration information from Config Server at startup
- Agent State Connector monitors and handles any agent config changes
- Agent State Connector also subscribes to StatServer and handles any agent state changes
- Updates on the agent and config changes are pushed to the AI Core Services' Agent Profile module
- Once the `ActivatePredictiveRouting` subrouting in URS is called, it will score the interaction based on outcome and log any predictive decision made using the `Predictor` module within the AI Core Services.

Third Party Components

Predictive Routing overlays most standard Genesys deployments. Each of those deployments may have numerous

3rd party components described in the relevant blueprints. In addition to those, the main third party components required for Predictive Routing are:

- Load Balancers (NGINX)
- Mongo DB

Note that NGINX and Mongo DB are currently deployed as part of the product.

Limits and Constraints

For a complete list of known limitation please see the Deployment and Operations Guide.

Deployment View

Genesys Deployment Options

For production, Omni-channel Predictive Routing supports a dual data center HA model. A single server configuration is also supported for pilots.

Single Server Deployment

The single server deployment consists of a single VM containing the AI Core Services and the Agent State Controller. The reusable Predictive Routing Strategies would be imported into the routing components.

The following diagram depicts the single server components, including the components within the AI Core Services.

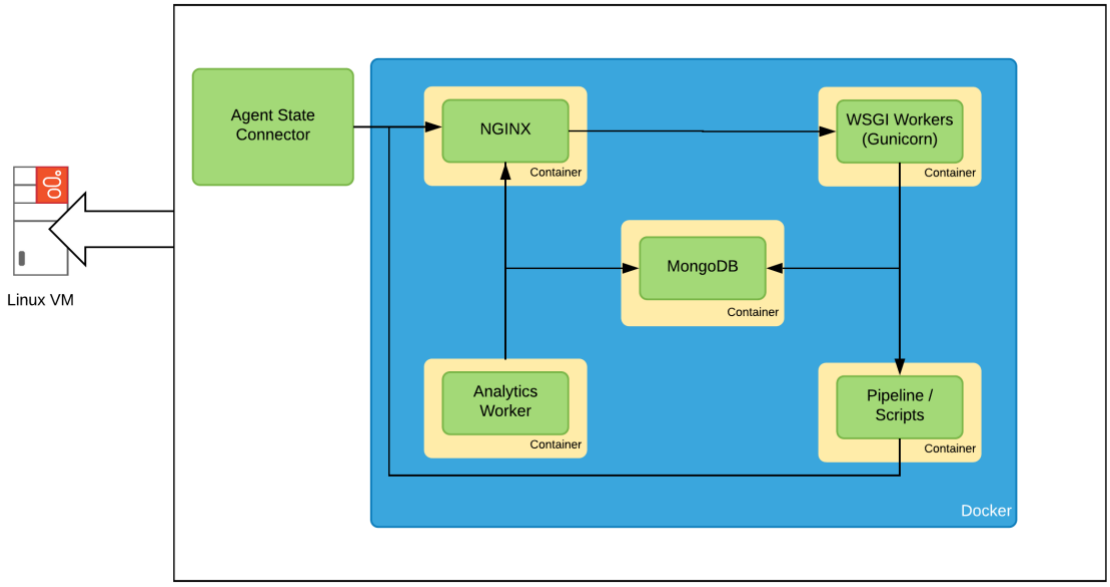

Figure 3: Single Server Deployment

The AI Core Services uses containerization to deploy and run the various services. In a single server deployment,

this will also include the 3rd party load balancer (NGINX) and MongoDB.

AI Core Services consist of the following:

- WSGI Workers (Gunicorn)
- Analytics Workers
- Pipeline / Scripts

Dual Data Center Deployment

A dual data center deployment is depicted below. It is a minimal configuration that can support data center failover.

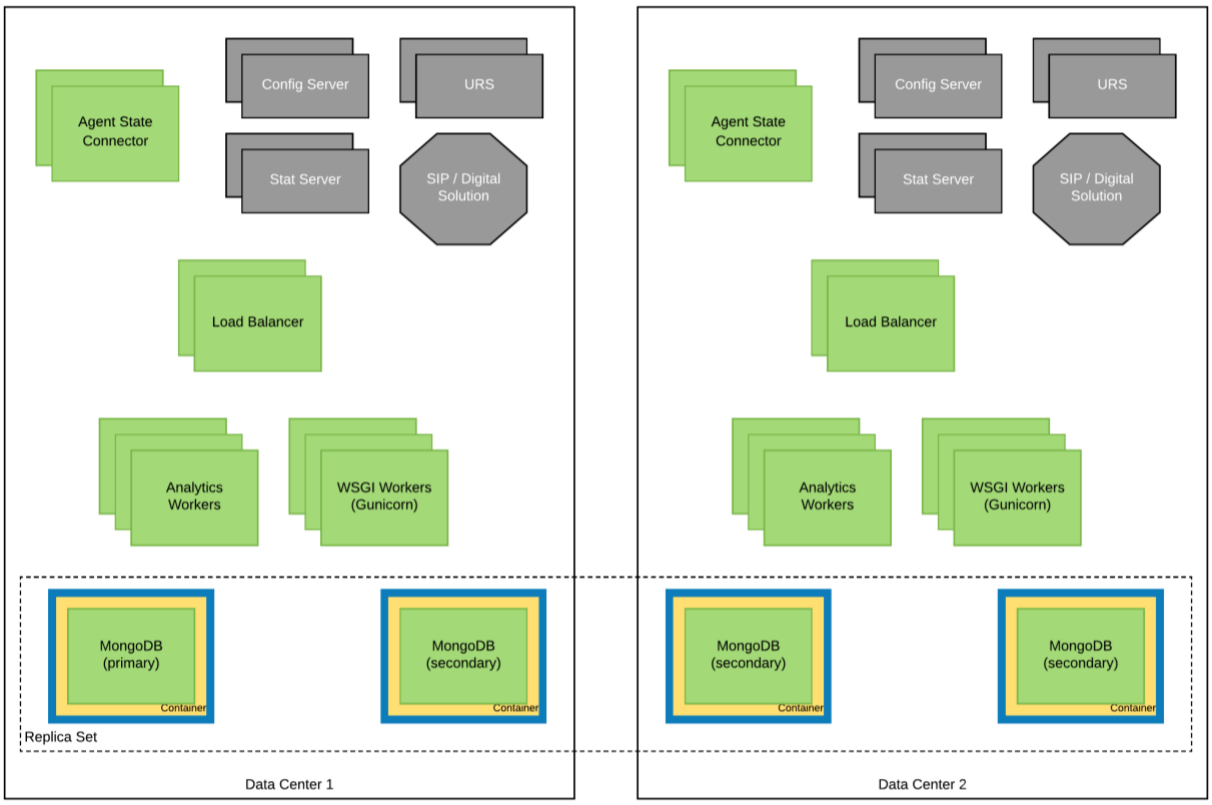

Figure 4: Dual Data Center Deployment

The Mongo DB uses Replica Sets which are split across the two data centers. A primary and secondary Replica Set are located in data center 1 (DC1) while a secondary Replica Set is located in DC2. All writes go to the Primary replica set. Reads cam ne directed to secondary sets and can be configured to prefer reads from local secondary sets. There will be cross-site data traffic due to all writes being directed to the primary in DC1 and the replication across sites.

Agent State Connectors (ASC) should be located in each data center, providing an active-active solution.

The Application Servers (Gunicorn) supporting the solution must be deployed in HA mode within each data center.

Analytics Worker Servers analyze and generate models. These nodes can be deployed in both data centers.

A load balancer is required in each data center and should be deployed in an HA fashion.

High Availability and Disaster Recovery

High Availability is built into the various components of the solution. For detailed information on HA and Disaster Recovery for Predictive Matching, please consult [Predictive Matching/HA] on the Genesys documentation site.

MongoDB

MongoDB uses replica sets to distribute data to secondary servers, providing high availability. A primary instance of MongoDB is the target server for write operations and will replicate the data to the secondary servers. If the primary server fails, read operations can still continue. Another server can be elected as the primary server to continue write operations.

For disaster recovery, MongoDB replica sets can be established in another data center. Note that sufficient bandwidth will be required for the data replication traffic between data centers.

Ideally an Arbiter node should be setup in a third data center or availability zone. This will facilitate the detection of a failed primary node when a data center becomes inaccessible and the proper election of a primary in the other data center. If there are only 2 data centers, manual intervention will be required to force one of the secondary replicants to become the primary replicant.

AI Core Services

The AI Core Services uses docker containerization and docker swarm technology to setup a cluster of nodes. Typically, an odd number of nodes should be used (3,5,7) to support HA.

The cluster can be spread evenly across multiple data centers to handle disaster recovery.

A load balancer is used to propagate traffic to the cluster instances. NGINX is provided for lab setups but in production environments, an enterprise-level HTTP load balancer that supports HA failover should be used.

Failover Scenarios

The following lists the failure modes that may occur:

- **Primary DB Server goes down**. Secondary is elected as new primary. Can preferentially elect a secondary in DC1 as primary via configuration.
- **DC1 Goes Down**. DC2 secondary cannot become primary since it does not have a quorum, so it becomes read only. Writes are rejected. Agent State Connector writes will fail, causing drift in synchronization of Agent State, and logging of scoring results will be rejected. No analytics jobs can be done.
- **DC2 Goes Down**. DC1 remains fully operational. Can still tolerate an additional secondary DB server failure if necessary.
- **Split-Brain**. DC1 will continue with primary. DC2 will now be read only since it will lack a quorum to promote to master. Writes from DC2 will now fail.
- **Agent State Connector goes down**. No operational impact since Agent State synchronized through shared database.
- **Application Server (GUnicorn) server fails**. All requests routed to alternate in the cluster.
- **Analytics Worker fails**. Jobs in progress die, incomplete. Should be canceled (currently manual). Backup worker takes over all pending jobs.

Database

Data for Omni-channel Predictive Routing is stored within MongoDB. As discussed earlier, MongoDB replica sets are distributed across data centers for additional data resiliency. For more information, please see the following links:

- https://eladnava.com/deploy-a-highly-available-mongodb-replica-set-on-aws/

- https://docs.mongodb.com/manual/core/replica-set-architecture-geographically-distributed/
- http://s3.amazonaws.com/info-mongodb-com/MongoDB_Multi_Data_Center.pdf
- https://stackoverflow.com/questions/43083246/requires-simple-explanation-on-arbiters-role-in-a-given-mongodb-replica-set
- https://docs.mongodb.com/manual/reference/method/Mongo.setReadPref/

Interaction View

Call Flows

The following diagram depicts a standard use case for the Omni-channel Predictive Routing Solution.

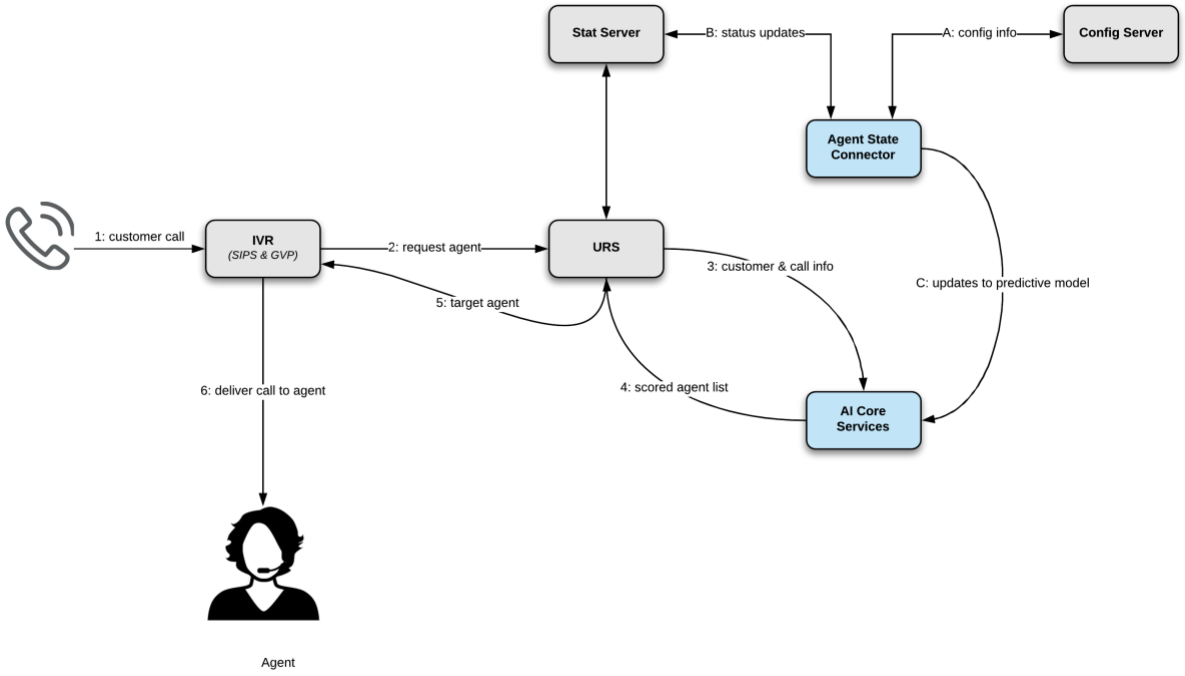

Figure 5: Predictive Routing Call Flow

During startup and normal operations, the Agent State Connector reads configuration information and changes (A) from ConfigServer/ConfigServer Proxy as well as agent status information (B) from StatServer. The Agent State Connector then updates the AI Core Services (C) with the updates.

During normal interaction, the following flow occurs:

1. Customer call enters the system and is sent to an IVR.
2. The IVR requests URS to find a suitable agent to route the call to.

3. URS (using the supplied predictive routing subroutines) requests the AI Core Services to find the best match for this request based on the customer data and call information (profile and call intent).
4. Predictive Routing uses the internal model within the AI Core Services to find any available agents that match this interaction. The scored list of candidate agents is returned.
5. URS returns the target agent to the IVR.
6. The IVR can then forward the call to that target agent.

Note that once the call is complete, StatServer will be updated assuming the agent is configured against SIP Server or another supported TServer. That data will be read by the Agent State Connector and sent to the AI Core Services (C) to refine the predictive model's score for that agent.

The following call flow depicts a digital use case.

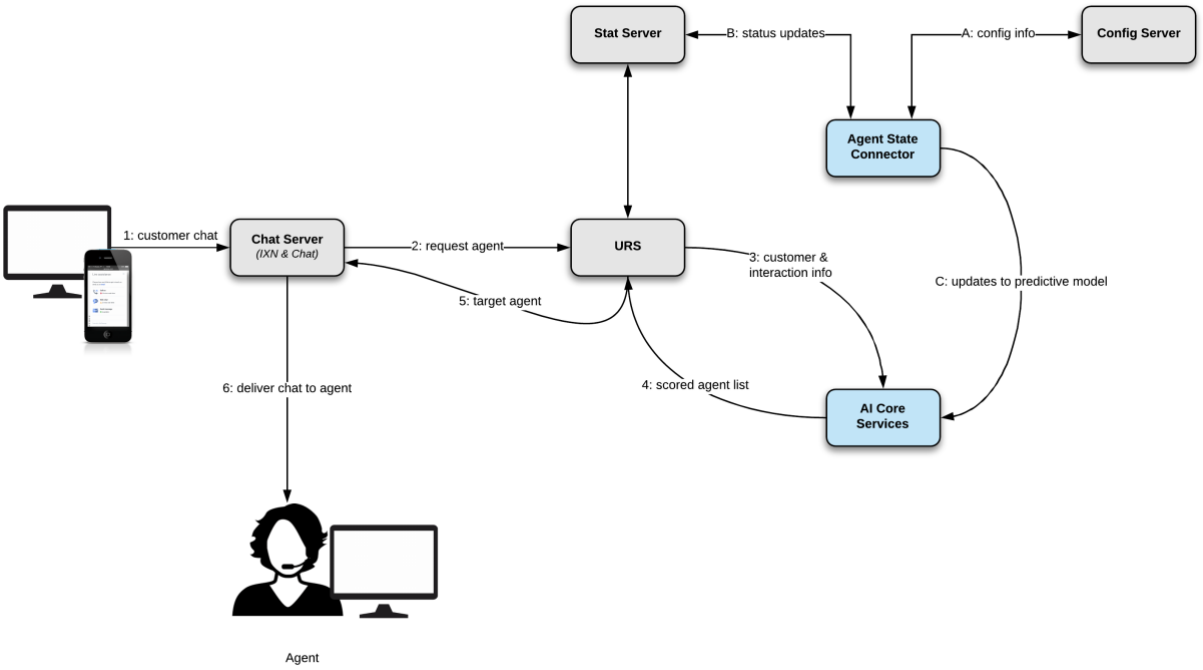

Figure 6: Digital Call Flow

During a typical digital interaction, the following steps would occur:

1. A chat session with the customer would start. This may be invoked directly by the customer or based on web engagement. The context of the chat (e.g. web page being viewed, customer id, etc.) might be sent as part of the chat. A chatbot could also be used to gather more context before routing to the agent.
2. Interaction Server requests URS to find a suitable agent.
3. URS (using the supplied predictive routing subroutines) requests the AI Core Services to find the best match for this request based on the customer data and call information (profile and chat context).
4. Predictive Routing uses the internal model within the AI Core Services to find any available agents that match this interaction. The scored list of candidate agents is returned.
5. URS returns the target agent to Chat/Interaction Server.
6. The chat is forwarded to that target agent.

External Interfaces

This solution is built on top of SIP Voice and Digital solutions which deal with most of the external interfaces that need to be configured. The following table depicts the additional optional external interfaces that may need to be integrated.

Interface	Protocol	Solution Components	Integration Tasks	Description
Load Balancer (Optional)	HTTPS	AI Core Services	Provision the cluster nodes into the load balancer	Load Balancer will distribute requests to the various AI Core Services Nodes.
Corporate Network	SNMP	Genesys SNMP Master Agent	Provision the network infrastructure (e.g. DNS) for the new traffic	This interface is used to integrate the solution with the

Management System (Optional)			Create and provision the security information (certificates, etc.)	Corporate network management system.

Table 2 - External Interfaces

Operational Management

Details on the operational management of this solution can be found on the docs site. Please see: https://docs.genesys.com/Documentation/GPM/latest/oneguide/Logging

Note that in addition to monitoring logs, there is a Pulse template for monitoring queue statistics.

In addition to the real time Pulse dashboard, there are also 5 standard reports available using GII or GCXI. These include:

- Predictive Routing AB Testing Report
- Agent Occupancy Report
- Predictive Routing Detail Report
- Predictive Routing Queue Statistic Report
- Predictive Routing Operational Report

Implementation View

Solution Sizing Guidelines

This section discusses the guidelines for sizing the Predictive Routing solution. The focus is on the components that make up this solution and not the surrounding SIP and Digital solutions. For details on the sizing of SIP and Digital components please consult the [SIP Voice Solution Blueprint] and the [Digital Solution Blueprint].

Sizing the Predictive Routing Solution requires several inputs and considerations. The following inputs are required:

- Total Agent Pool
- Active Agent Pool – number of logged in agents
- Target Agent Pool – typical number of agents targeted for scoring
- Customer Pool
- Interactions per second
- Model Retention Window – how many details to keep data model
- Scoring Requests per Interaction
- Concurrent Analytics Users
- Number of Predictors
- Predictor Feature Count – how many attribute values are considered for input
- Predictor Attribute Count – how many attributes used for a predictor

Details of how these and other inputs impact the sizing of the solution can be found on the Genesys documentation site:

- https://docs.genesys.com/Documentation/GPM/latest/oneguide/sizing

As a general guideline, the single server lab deployment can typically be handled by a base machine consisting of:

- 16 CPU core
- 64 GB of RAM and
- 512 GB of storage.

To support an HA deployment of the solution in a single data center, three (3) base machines would typically be used.

A dual data center deployment would typically require five (5) base machines.

For a more accurate and flexible sizing model, consult the link above. A few of the details to be considered are:

1. The Agent State Connector should run on a dual core 2GB RAM server. This will be sufficient for very large contact centers (10's of thousands of agents). It is not necessary to size it more fine grain than this.
2. Sizings for cores of Application Servers (Gunicorn) map directly to a multi-server deployment but cores are split across machines to provide HA. You can safely assume 2GB RAM per Gunicorn worker and 1 Gunicorn worker per core. This will be more than enough RAM for large agent pools and customer profiles, and multiple predictors per deployment. More details will be added to the sizing guide on how this number is derived 4.
3. Analytics Servers should be provisioned with 32GB RAM per worker, and 1 worker per core. The number of cores deteremines the number of concurrent analytics jobs are required. These are all initiated offline through the Web UI or Cron jobs. It will be entirely possible to operate with just a single core for analytics. As we evolve algorithms to exploit greater parallelism this guidance may change to achieve better performance.
4. DB Servers can be provisioned with 8GB of RAM. This is more than adequate for storing all necessary indexes (Agent Profile and Customer Profile) in RAM and will also provide

Configuration Guidelines

The configuration details can be found in the Configuration Options page of the Genesys Predictive Routing section on the Genesys documentation site:

- https://docs.genesys.com/Documentation/GPM/latest/oneguide/cfgOptions

In general, two objects need to be configured, the Agent State Connector (ASC) `Application` object and the AI Core Services which is provisioned as a Transaction List object under `Routing/eServices`.

Security

The solution uses HTTPS and TLS 1.2 for connections between components to ensure a high level of security.

Localization and Internationalization

Localization and Internationalization are topics for numerous Genesys components, especially user interfaces and reporting. Omni-Channel Predictive Routing is mainly focused on interaction routing. However, there are some reports that may require localization as part of deployment.

Genesys Standard Reporting Integration:

- https://docs.genesys.com/Documentation/GPM/latest/oneguide/GIMintegration

GII Predictive reports on:

- https://docs.genesys.com/Documentation/GI2/8.5.0/UNV/Supplement

GCXI reports in EAP stage