Agent Pacing Service Deployment Guide

How Pacing works

2/25/2026

Contents

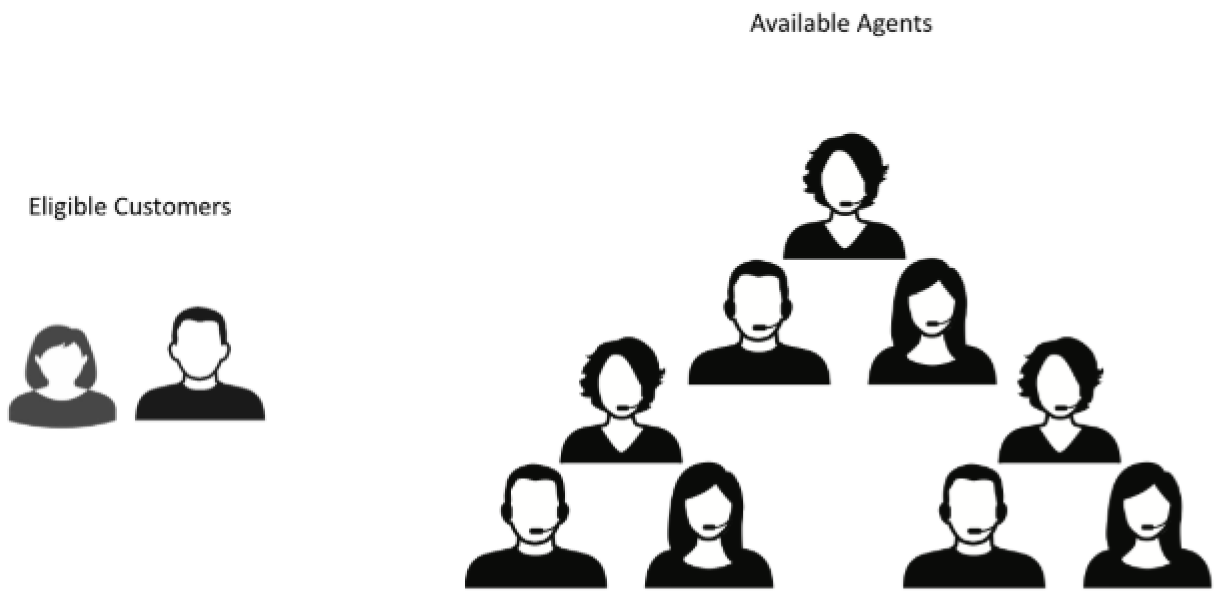
- 1 The pacing algorithm
    - 1.1 Virtual queue
    - 1.2 Agent Group
    - 1.3 Pacing target
- 2 Output parameter
- 3 High-level architecture

Understand the concept of pacing and learn how the Agent Pacing Service ensures agents are available to assist customers.

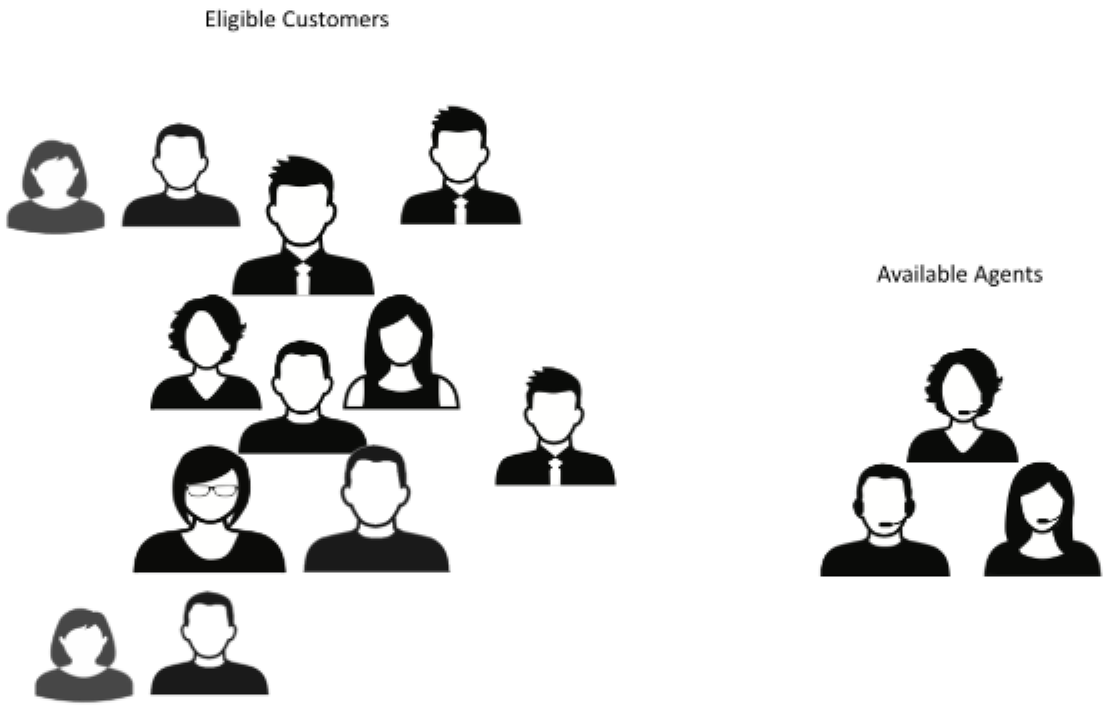
Many products allow you to focus on *reactive* engagement sessions, in which a customer initiates the connection with your contact center by responding to a *static* request, such as a **Chat Now** button. Genesys Predictive Engagement goes a step further, by predicting when it's a good time for your contact center to take the initiative—to reach out to a customer proactively. For example, Genesys Predictive Engagement might see that a customer is likely to abandon a purchase, so it proactively sends them a pop-up that says "Let's chat!" At that point, the customer can click an **Accept** button and your agents have an opportunity to help them complete their purchase.

Most of the information that Genesys Predictive Engagement works with is focused on your customers. This data helps you pinpoint the ones who are most likely to respond well to a proactive invitation.

This works great when you have enough agents:



But what if a customer accepts an engagement offer—and your agents are too busy to answer?



That's what pacing is for. A *pacing algorithm* helps you manage your agent load by comparing it to your traffic and predicting how many proactive invitations you can send out—without making your customers wait too long for a response from your agents.

The pacing algorithm

Agent Pacing Service uses a pacing algorithm to balance two things:

- The need for your agents to stay busy
- The need to minimize the abandonment rate for proactive interactions

The algorithm uses the following inputs as the basis of its calculations.

Virtual queue

Contact centers use queues of various kinds to hold interactions that are waiting to be processed. These queues can take the form of an automatic call distributor (ACD) on a hardware switch, a queue of digital media interactions supported by Genesys Interaction Server, or a router-based virtual queue. Virtual queues are great because you can easily use them with just about any media type—including voice, chat and email. Agent Pacing Service requires that you pass your interactions through virtual queues, and that you dedicate each queue to a single media type. It's important to keep this in mind when creating your routing strategies.

Agent Group

The Agent Group defines which agents the pacing algorithm should include in its calculations when it predicts whether it's safe for Genesys Predictive Engagement to generate a proactive request. You can use a regular Agent Group, which consists of a pre-configured list of agents that you can only change explicitly, or a virtual Agent Group, whose agent list is defined dynamically based on a skill expression, such as `Spanish > 5`.

Pacing target

The pacing target uses the output from the pacing algorithm (the number of proactive invitations that can be sent during a specific time interval) to decide whether to send an invitation to a specific visitor who has triggered a hot-lead event.

Important

Genesys Predictive Engagement automatically propagates Agent Group and Virtual Queue information to Genesys Widgets, making it available as UserData to the Routing Logic for the selected engagement channel.

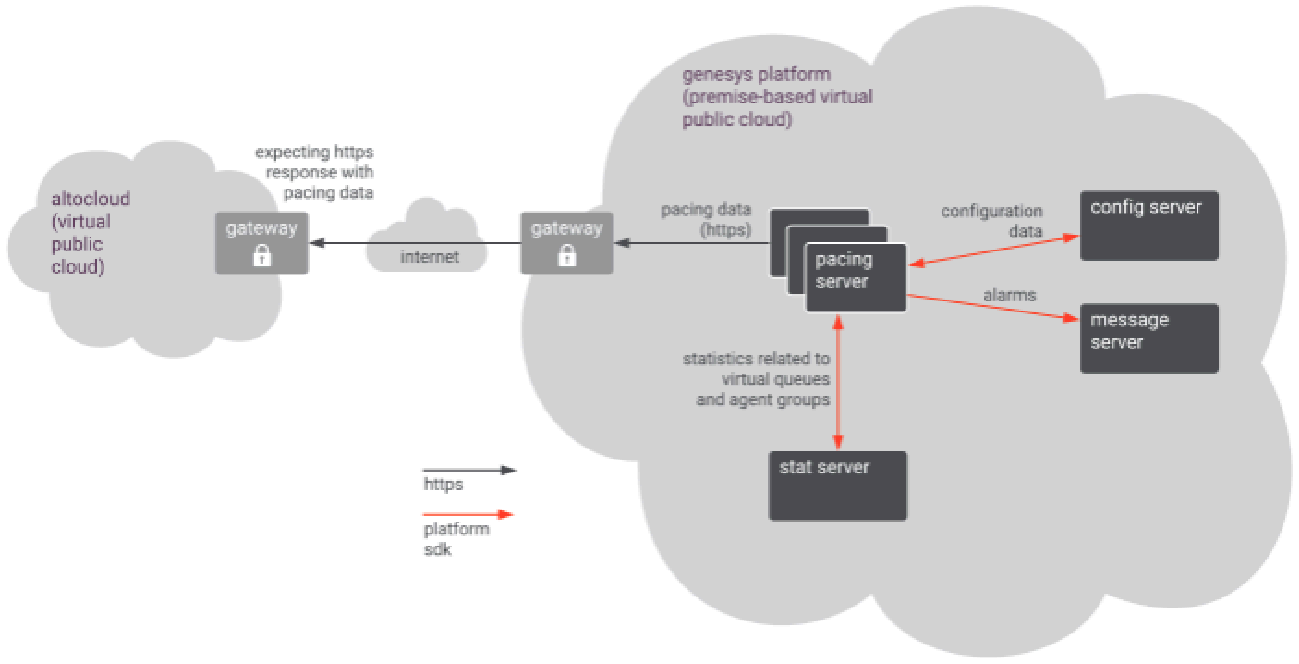
Output parameter

The pacing algorithm produces only one output parameter: the number of proactive invites that can be sent during a specific time interval.

High-level architecture

1. When the Pacing Server starts, it opens a connection to Config Server and discovers its pacing targets.
2. Pacing Server opens a connection to Stat Server.
3. Pacing Server starts performing its pacing calculations, using Stat Server as a primary source of information. For the initial calculations, the count of pending invitations is considered to be **0**.
4. Pacing Server uses the currently available data from Stat Server and Genesys Predictive Engagement to calculate the number of proactive invites that can be sent during the specified time interval.
5. Pacing Server sends the results of its calculations to Genesys Predictive Engagement.
6. Genesys Predictive Engagement processes the pacing information and returns the count of pending invitations.
7. A new pacing cycle is started from step 4, using the information sent by Genesys Predictive Engagement.

If something goes wrong—for example, if Pacing Server starts using too much memory, or if Genesys Predictive Engagement becomes unreachable—Pacing Server sends an alarm to Message Server.


altocloud (virtual public cloud)
gateway
expecting https response with pacing data
internet
gateway
genesys platform (premise-based virtual public cloud)
pacing data (https)
pacing server
configuration data
config server
alarms
message server
statistics related to virtual queues and agent groups
stat server
https
platform sdk