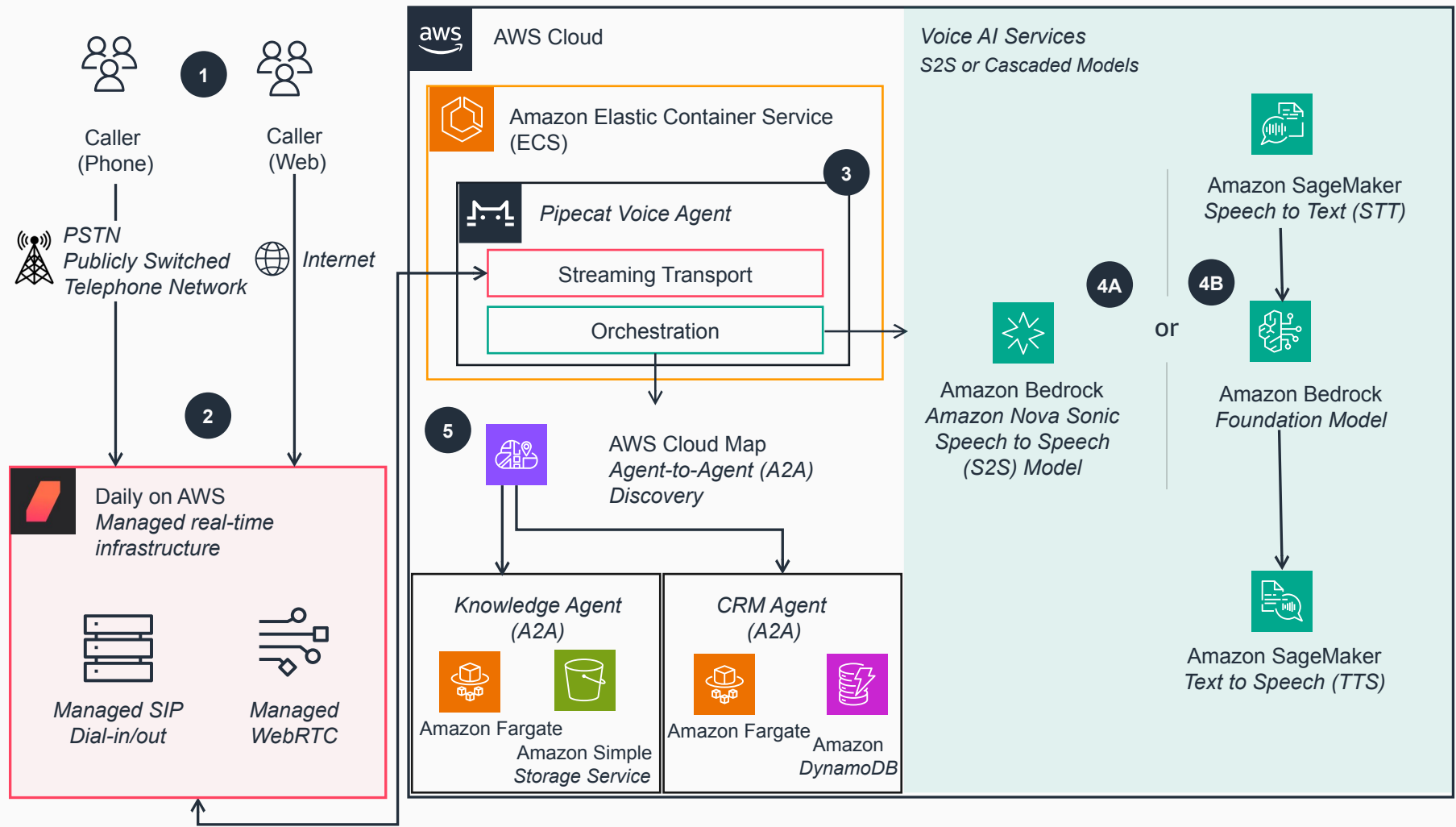


Guidance for Building Voice Agents on AWS

This architecture illustrates deployment of voice agents on AWS using speech-to-speech and cascaded models, with agent-to-agent capabilities for multi-agent coordination. Users speak naturally via phone or web/mobile, receiving natural voice responses while maintaining session context throughout.



- 1 Caller interact with the voice agent solution via a phone line or web or mobile application.
- 2 Daily managed real-time infrastructure on AWS supports *both* telephony and webRTC streaming.

Alternatively, the **Pipecat voice agent** can connect via WebSockets or other telephony providers using another transport service.
- 3 **Pipecat voice agent** is a Python-based voice agent framework that orchestrates real-time voice pipelines with ultra-low latency, with two key components:
1) streaming network transport, and;
2) orchestration of AI services and sub-agents
Pipecat is served on **Amazon Elastic Container Service (ECS)**. Alternatively, you consider **AgentCore** which provides enterprise-grade services for deploying and operating AI agents at scale, or **Elastic Kubernetes Service (EKS)**.
- 4 The solution supports voice agents across two common approaches:

A –**Speech-to-speech (S2S)** model such as **Amazon Nova Sonic**, which delivers real-time, voice conversations with low latency, or;

B - **Cascaded models** approach that orchestrates speech-to-text (STT) and text-to-speech (TTS) models with a foundation model on **Amazon Bedrock**, with the option to integrate a reasoning agent on **Amazon Bedrock AgentCore**.

You can configure the solution to use speech models with APIs for prototyping, and **Amazon SageMaker** with bidirectional streaming in production.
- 5 The main voice agent delegates to sub-agents via the A2A protocol, using **AWS Cloud Map** for service discovery. For example, the **Knowledge Base Agent** retrieves context from **Amazon S3** vectors, while the **CRM Agent** queries **Amazon DynamoDB** for customer-specific data.