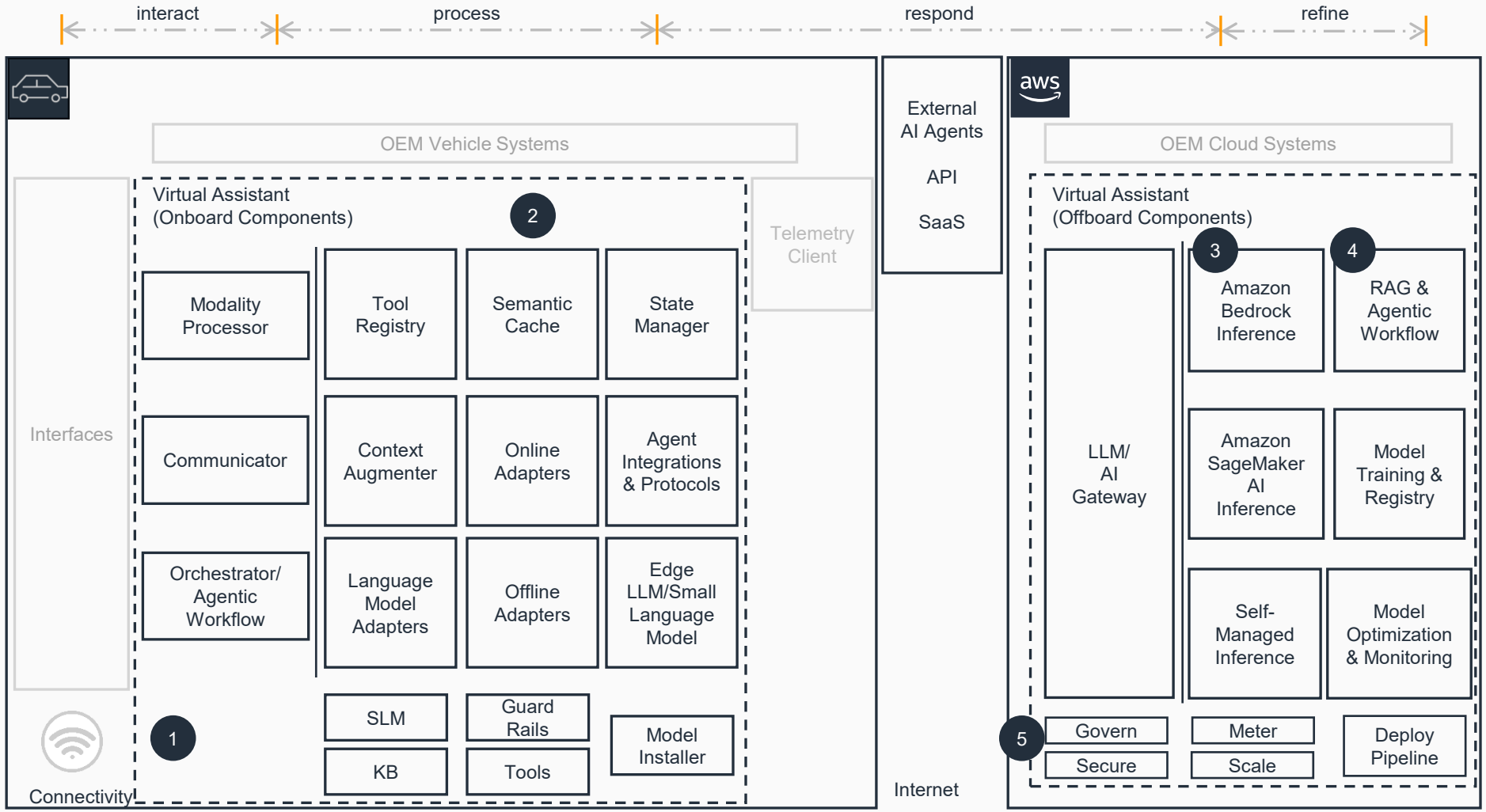
Guidance for In-Vehicle AI Assistant on AWS

Building Blocks

This architecture diagram illustrates the hybrid edge-cloud approach for implementing a in-vehicle AI Assistant on AWS. It shows the key components and their interactions, providing an overview of the architecture's structure and functionality.



AWS Reference Architecture

1. Virtual Assistant In-Vehicle Components establish a comprehensive on-board processing environment featuring a Tool Registry for managing available functions, Context Augmenter for enriching user queries, and State Manager for maintaining conversation context.

2. This layer includes Semantic Cache for quick response retrieval, Online and Offline Adapters for handling various connectivity scenarios, and Agent Integrations & Protocols for coordinating with external systems. The vehicle-based components, powered by Edge LLM/Small Language Models, supported by local Knowledge Base (KB), Guard Rails for safety compliance, and Model Installer for updates, ensure immediate responses even during connectivity disruptions.

3. When complex AI processing is required, the system seamlessly transitions to Virtual Assistant Cloud Components for AI Serve, which leverage **Amazon Bedrock** Inference for advanced natural language understanding, **Amazon SageMaker AI** Inference for custom machine learning models, and Self-Managed Inference capabilities for custom deployment scenarios.

4. Retrieval-Augmented Generation (RAG) and Agentic Workflow systems enable intelligent information retrieval and multi-step reasoning, and comprehensive Model Training & Registry services support continuous learning from user interactions.

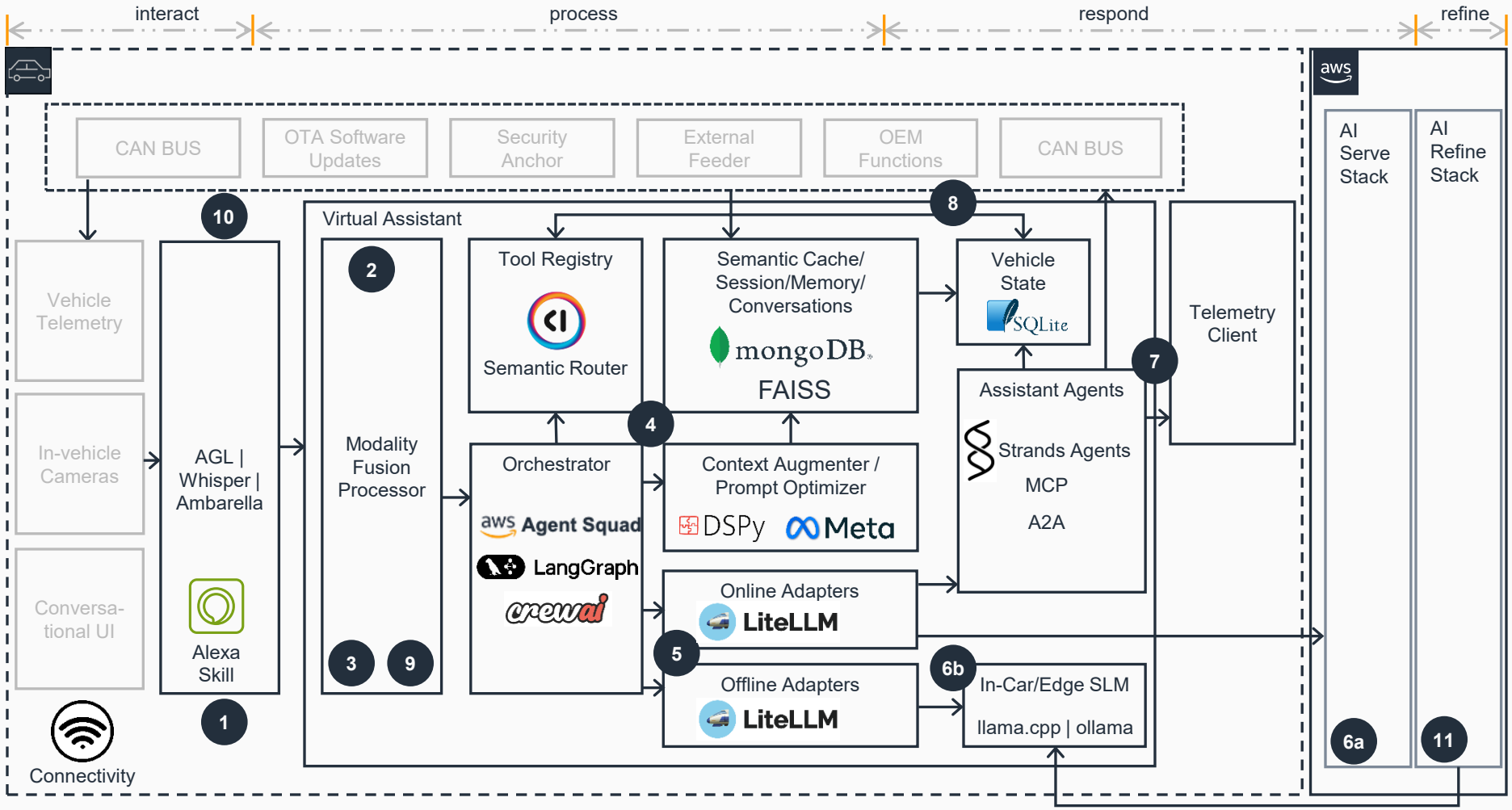
5. Virtual Assistant Cloud Components (AI Refine) Model Optimization and Monitoring systems continuously analyze performance metrics and user feedback. Automated Deploy Pipeline services push refined models back to vehicles. Enterprise-grade Govern, Secure, Meter, and Scale components ensure compliance, security, cost management, and scalability across the entire fleet.

Reviewed for technical accuracy October 8, 2025

Guidance for In-Vehicle AI Assistant on AWS

Virtual Assistant In-Vehicle Components

Virtual Assistant In-Vehicle Components provide local AI processing through edge language models and semantic caching, while orchestrating seamless integration with cloud services via online adapters and agent protocols.



Virtual Assistant Onboard & Offboard

OEM's existing systems

AWS Reference Architecture

1. The Input Layer processes sensor and **Alexa** Skills inputs including speech-to-text, computer vision for gestures and driver monitoring, haptic feedback, and vehicle data using Automotive Grade Linux (AGL), Whisper, and Ambarella for automotive-grade processing and accurate recognition capabilities.

2. The Multi-Modal Fusion Processor integrates speech, vision, and vehicle sensor inputs for error correction, environmental ambiguity resolution, and cross-modal redundancy to ensure accurate user intent interpretation.

3. The Orchestration Layer uses Generative AI workflow management to coordinate local and cloud models, handle state transitions, and securely invoke APIs through Agent Squad, LangGraph, and CrewAI. Agent Squad manages multiple AI agents, LangGraph orchestrates complex workflows, and CrewAI coordinates collaborative AI operations.

4. Context Augmenter Engine enriches interactions with vehicle state, driver conditions, and environmental factors using DSPy/Meta Prompt Ops for prompt optimization, Semantic Router for tool selection, FAISS for vector search, and MongoDB for data storage and rapid context retrieval.

5. Adaptive Processing The architecture includes two types of adapters: Offline Adapters using local models like Llama 3.2 for disconnected scenarios, and Online Adapters integrating advanced models from services like Amazon Bedrock or Amazon SageMaker Jumpstart through LiteLLM. LiteLLM provides unified access to multiple language model providers.

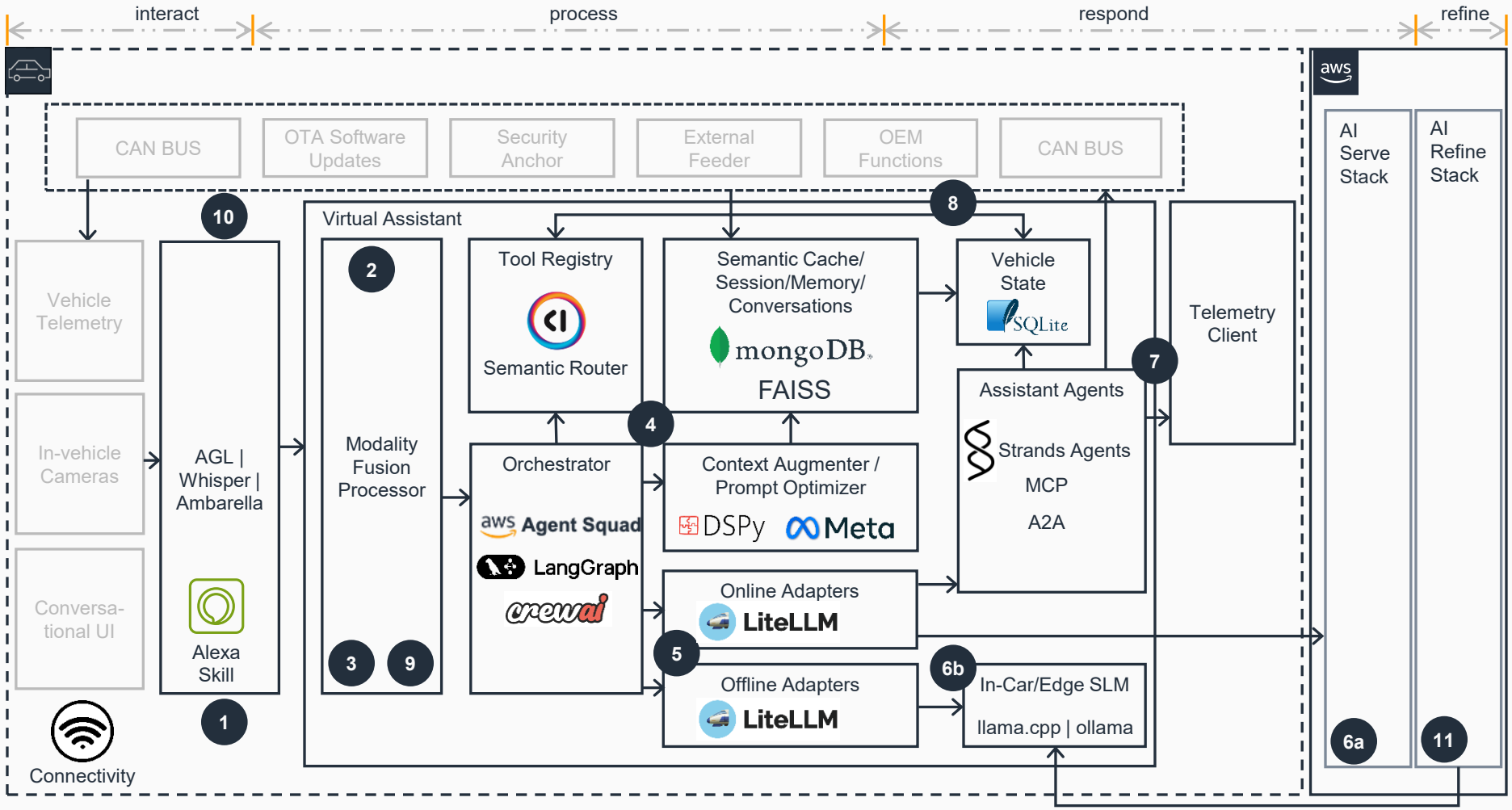
6a. The In-Car/Edge Small Language Models (SLM) operate through Llama.cpp for efficient C++ inference and Ollama for simplified model deployment, while the AI Serve Stack provides advanced online models from **Amazon Bedrock** and **Amazon SageMaker** Jumpstart when connected to the internet.

Reviewed for technical accuracy October 8, 2025

Guidance for In-Vehicle AI Assistant on AWS

Virtual Assistant In-Vehicle Components

Virtual Assistant In-Vehicle Components provide local AI processing through edge language models and semantic caching, while orchestrating seamless integration with cloud services via online adapters and agent protocols.



Virtual Assistant Onboard & Offboard

OEM's existing systems

AWS Reference Architecture

Reviewed for technical accuracy October 8, 2025

6b The offline functionality leverages Small Language Models (SLMs) like Llama 3.2 to enable reliable, low-latency interactions in disconnected scenarios, ensuring continuous AI assistance regardless of network availability.

7 The system implements **Model Context Protocol (MCP)**, **Agent-to-Agent (A2A)** communication built with **Strands Agents** to retrieve telemetry data for enhanced situational awareness.

MCP facilitates the exchange of contextual information and the execution of vehicle-specific actions, allowing the assistant to deeply integrate with the vehicle's internal systems.

A2A provides a standardized way for the assistant to fetch up-to-date telemetry data from external sources, enhancing the assistant's awareness of the vehicle's environment and state.

8 The architecture integrates with a Controller Area Network (CAN) bus for real-time vehicle state data, over-the-air (OTA) software updates for system maintenance, security trust anchors for cryptographic operations, external data sources for enriched context, and Original Equipment Manufacturer (OEM) vehicle infotainment functions.

9 The Orchestrator coordinates response generation across modalities, combining offline and online models while routing responses to appropriate output channels. This optimization creates a cohesive user experience by synchronizing all interaction interfaces.

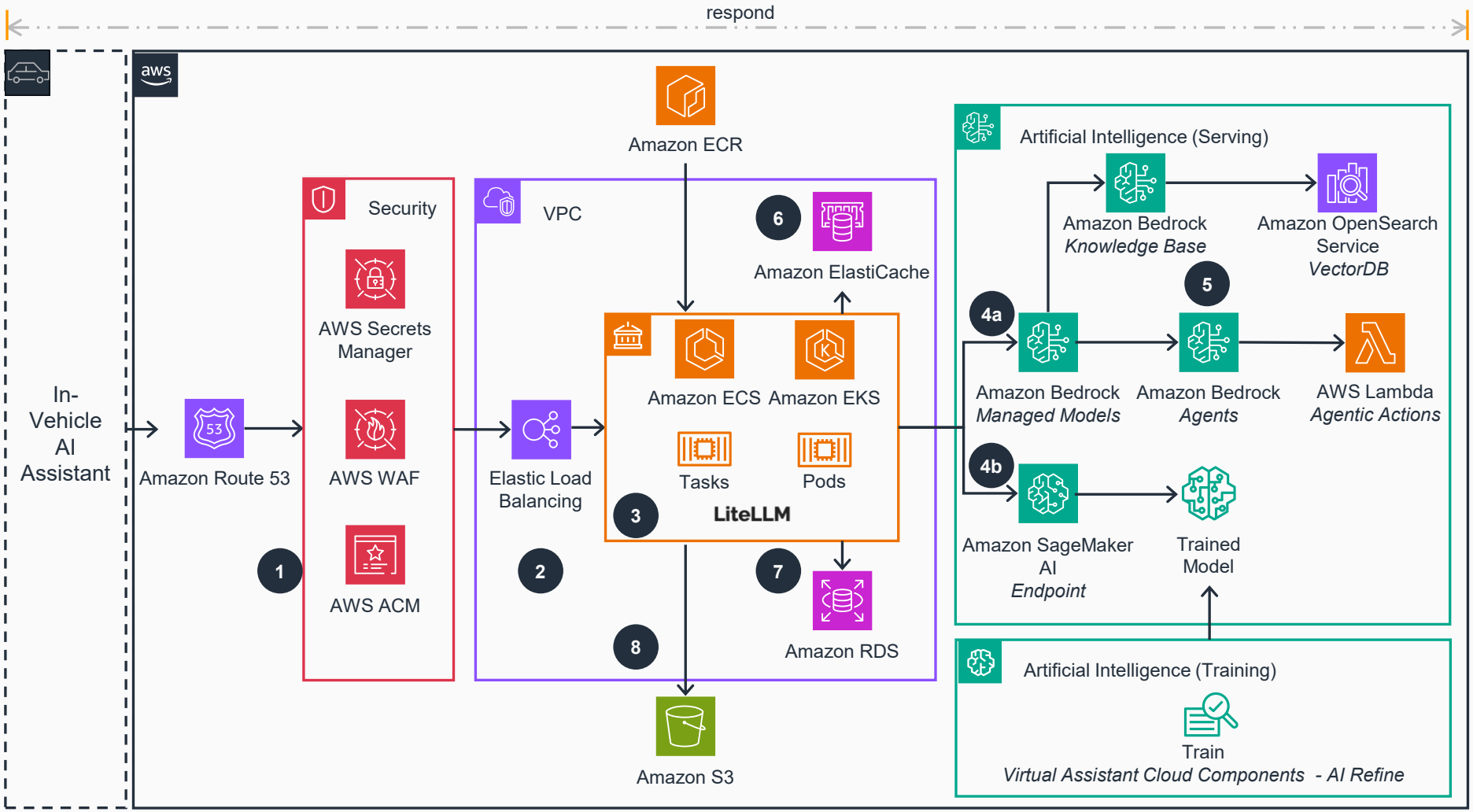
10 The system delivers responses through audio using high-quality text-to-speech engines for natural-sounding output, visual interfaces through on-screen displays, and haptic feedback for tactile interactions.

11 The assistant provides output through audio (text-to-speech), visual (on-screen displays), and haptic (tactile feedback) channels. Specialized components, such as high-quality text-to-speech engines enable natural-sounding audio output to enhance the user experience. The Orchestrator coordinates and optimizes these multi-modal output interfaces for a seamless, cohesive interaction.

Guidance for In-Vehicle AI Assistant on AWS

Virtual Assistant Cloud Components (AI Serve)

The Virtual Assistant Cloud Components for AI Serve deliver advanced AI inference capabilities through Amazon Bedrock, Amazon SageMaker, and Amazon EKS for self-managed serving, processing complex queries that exceed local vehicle processing capacity. These services provide sophisticated conversational AI responses.



1. The in-vehicle virtual assistant invokes the AI gateway through the Amazon Route 53 URL endpoint. This access is protected against common web exploits and bots using the AWS Web Application Firewall (AWS WAF). An AWS Certificate Manager (AWS ACM) certificate secures traffic via TLS/SSL.
2. **AWS WAF** forwards requests to the Application Load Balancer (ALB), which distributes traffic to Amazon Elastic Container Service (Amazon ECS) tasks or Amazon Elastic Kubernetes Service (Amazon EKS) pods running AI gateway containers.
3. Container images for the API/middleware and LiteLLM applications deploy in **Amazon ECS** on **AWS Fargate** or **Amazon EKS** clusters exposed by Elastic Load Balancing. These clusters run the applications as containers in **Amazon ECS** tasks or **Amazon EKS** pods, respectively. LiteLLM provides a unified application interface for configuration and interacting with **Amazon Bedrock** and **Amazon SageMaker AI**.
4a. Models hosted on **Amazon Bedrock** including **Amazon Nova** provide model access, guardrails, prompt caching, and routing to enhance the AI gateway and additional controls for the assistant through a unified API.
4b. The LiteLLM gateway supports integration with models hosted on **Amazon SageMaker**, in addition to **Amazon Bedrock**.
5. Integrate with Amazon Bedrock Knowledge Bases and Agents to enhance the capabilities of an in-vehicle virtual assistant backend. Deploy an **Amazon Bedrock** Knowledge Base with Amazon OpenSearch vector database, and **Amazon Bedrock** Agents, or Strands Agents integrated with **AWS Lambda** for architecture extensibility, scalability, and performance.
6. Amazon ElastiCache provides multi-tenant distribution of application settings and prompt caching.

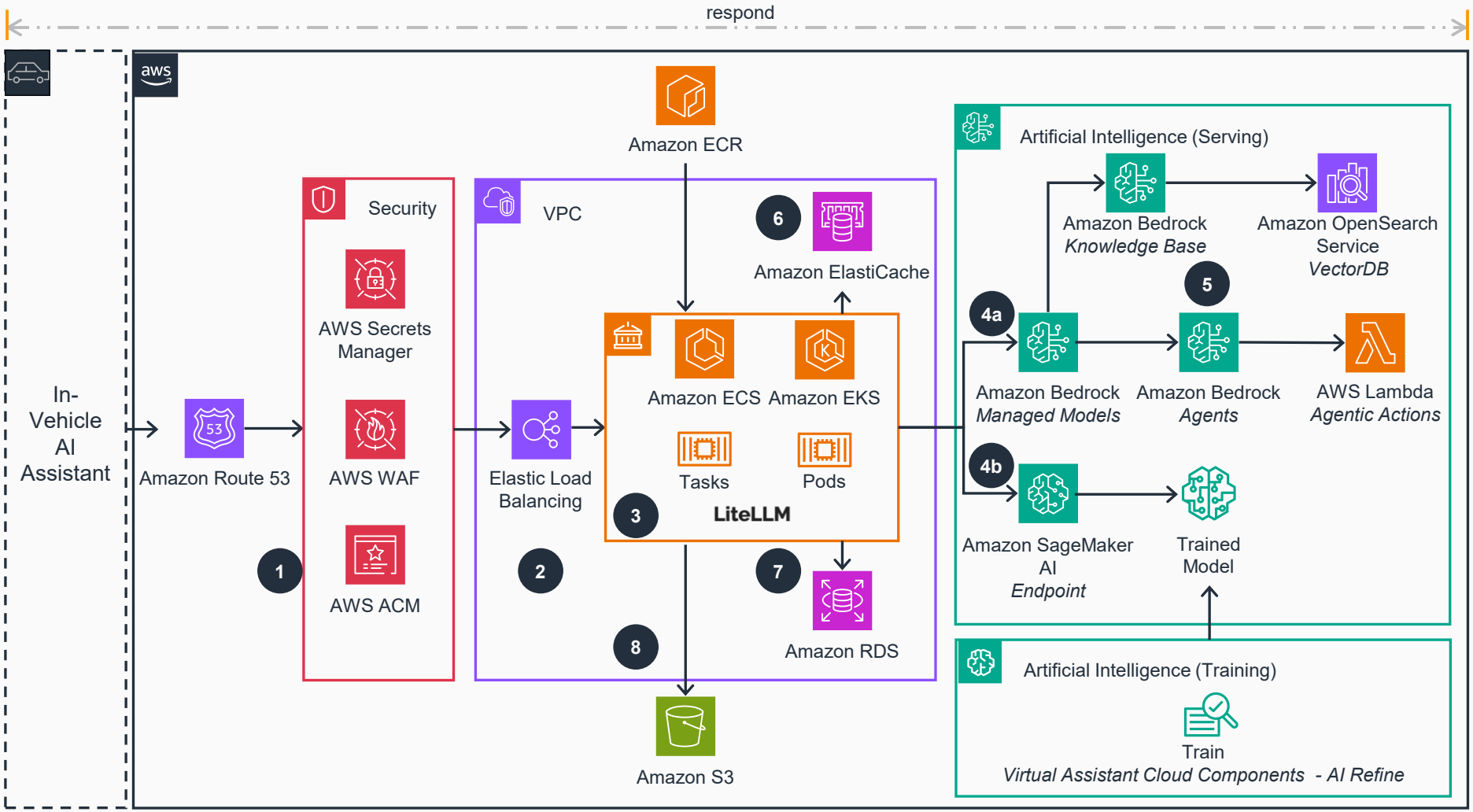
Reviewed for technical accuracy October 8, 2025

AWS Reference Architecture

Guidance for In-Vehicle AI Assistant on AWS

Virtual Assistant Cloud Components (AI Serve)

The Virtual Assistant Cloud Components for AI Serve deliver advanced AI inference capabilities through Amazon Bedrock, Amazon SageMaker, and Amazon EKS for self-managed serving, processing complex queries that exceed local vehicle processing capacity. These services provide sophisticated conversational AI responses.



7 The LiteLLM gateway utilizes Amazon Relational Database Service (Amazon RDS) to enable persistence of virtual API keys, organizations, teams, users, budgets, and per-request usage tracking.

8 The AI gateway and its associated services store application logs in a dedicated Amazon Simple Storage Service (Amazon S3) storage bucket.

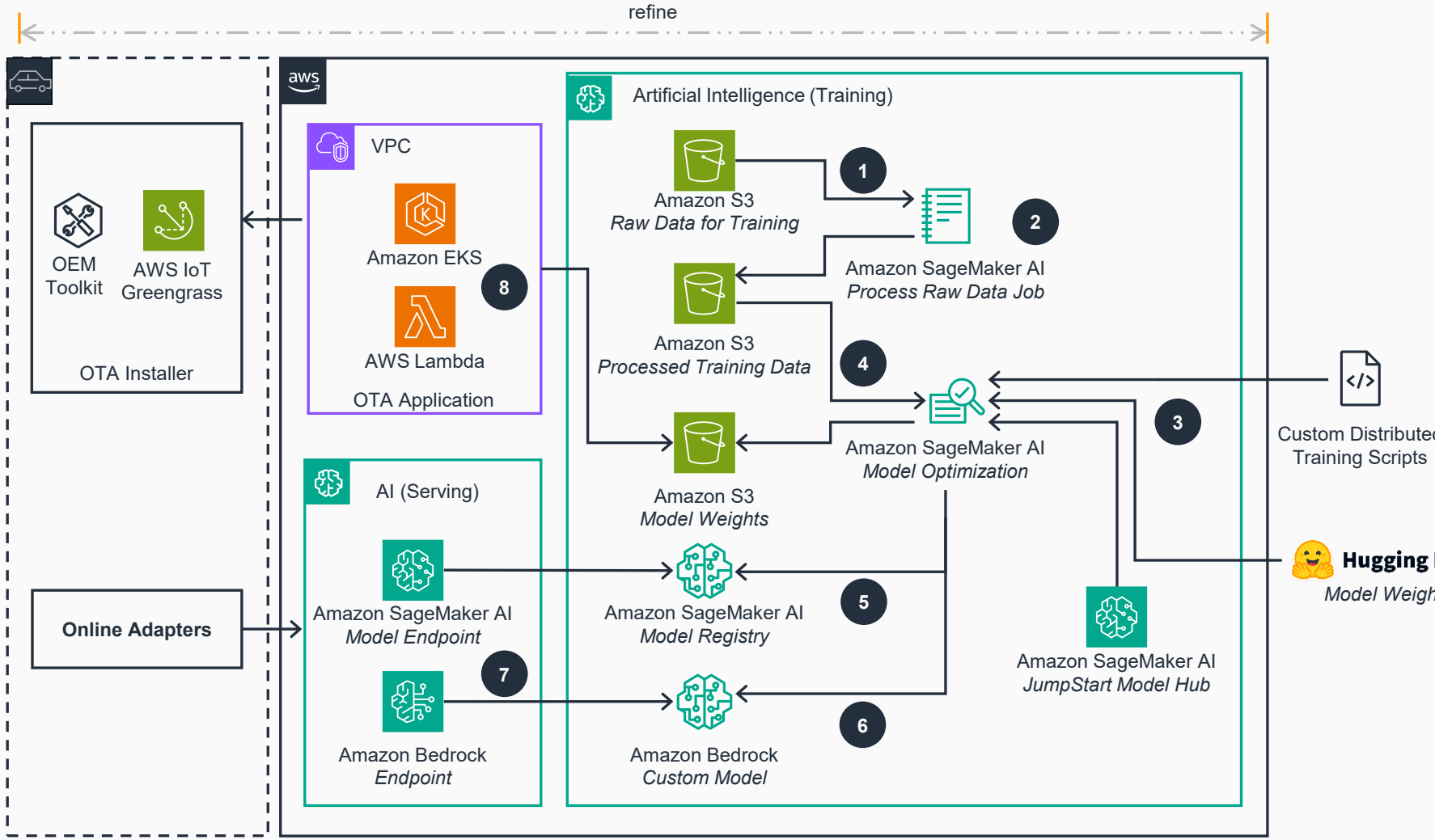
Reviewed for technical accuracy October 8, 2025

AWS Reference Architecture

Guidance for In-Vehicle AI Assistant on AWS

Virtual Assistant Cloud Components (AI Refine)

This architecture diagram illustrates the hybrid edge-cloud approach for implementing a In-vehicle AI Assistant on AWS. It shows the key components and their interactions, providing an overview of the architecture's structure and functionality.



1. Store multi-modal training data from vehicles including speech recordings, camera feeds, and sensor telemetry in **Amazon S3** buckets. Implement versioning and lifecycle policies for efficient data management during iterative model refinement cycles.
2. Process raw vehicle data using **Amazon SageMaker AI p**rocessing jobs to clean sensor noise, synchronize multi-modal inputs, and create labeled datasets. Transform raw telemetry into feature vectors optimized for automotive AI training.
3. Execute specialized training scripts designed for automotive AI workloads using **Amazon SageMaker AI** training jobs. Implement domain-specific loss functions for safety-critical scenarios and vehicle-context awareness. Discover and use pre-trained models from Amazon SageMaker JumpStart or Hugging Face.
4. Apply quantization techniques to reduce model size for edge deployment. Use knowledge distillation to create compact student models from larger teacher models while maintaining accuracy for in-vehicle constraints.
5. Register optimized models in **Amazon SageMaker** Model Registry with comprehensive metadata including latency benchmarks, accuracy metrics, and hardware compatibility. Track model lineage and approval status for automotive safety compliance.
6. Deploy fine-tuned foundation models to **Amazon Bedrock** for cloud-based inference supporting complex reasoning tasks. Configure custom model endpoints with automotive-specific prompt templates and safety guardrails.
7. Route to **Amazon SageMaker AI** endpoints for real-time edge inference or **Amazon Bedrock** endpoints for advanced cloud processing. Implement intelligent routing based on use case, network connectivity and computational requirements.

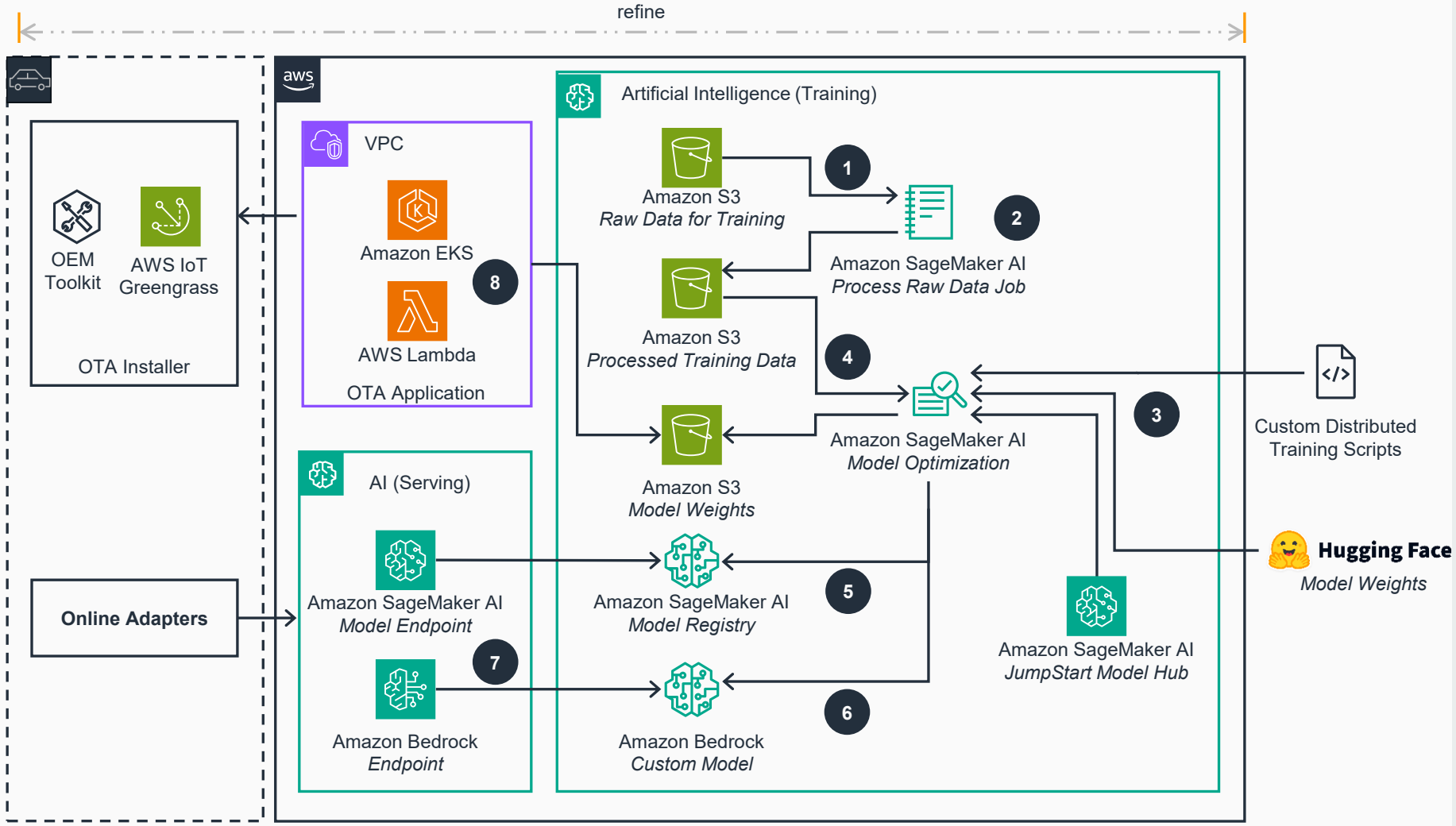
Reviewed for technical accuracy October 8, 2025

AWS Reference Architecture

Guidance for In-Vehicle AI Assistant on AWS

Virtual Assistant Cloud Components (AI Refine)

This architecture diagram illustrates the hybrid edge-cloud approach for implementing a In-vehicle AI Assistant on AWS. It shows the key components and their interactions, providing an overview of the architecture's structure and functionality.



8 The Over-The-Air (OTA) update mechanism delivers refined AI models and components from the cloud to vehicles remotely. This AWS Lambda function or Amazon EKS cluster enables secure deployment of updated AI capabilities, model weights, and application logic to the vehicle fleet without requiring physical service visits. The onboard OTA may be AWS IoT Greengrass or OEM's custom toolkit.

Reviewed for technical accuracy October 8, 2025

AWS Reference Architecture