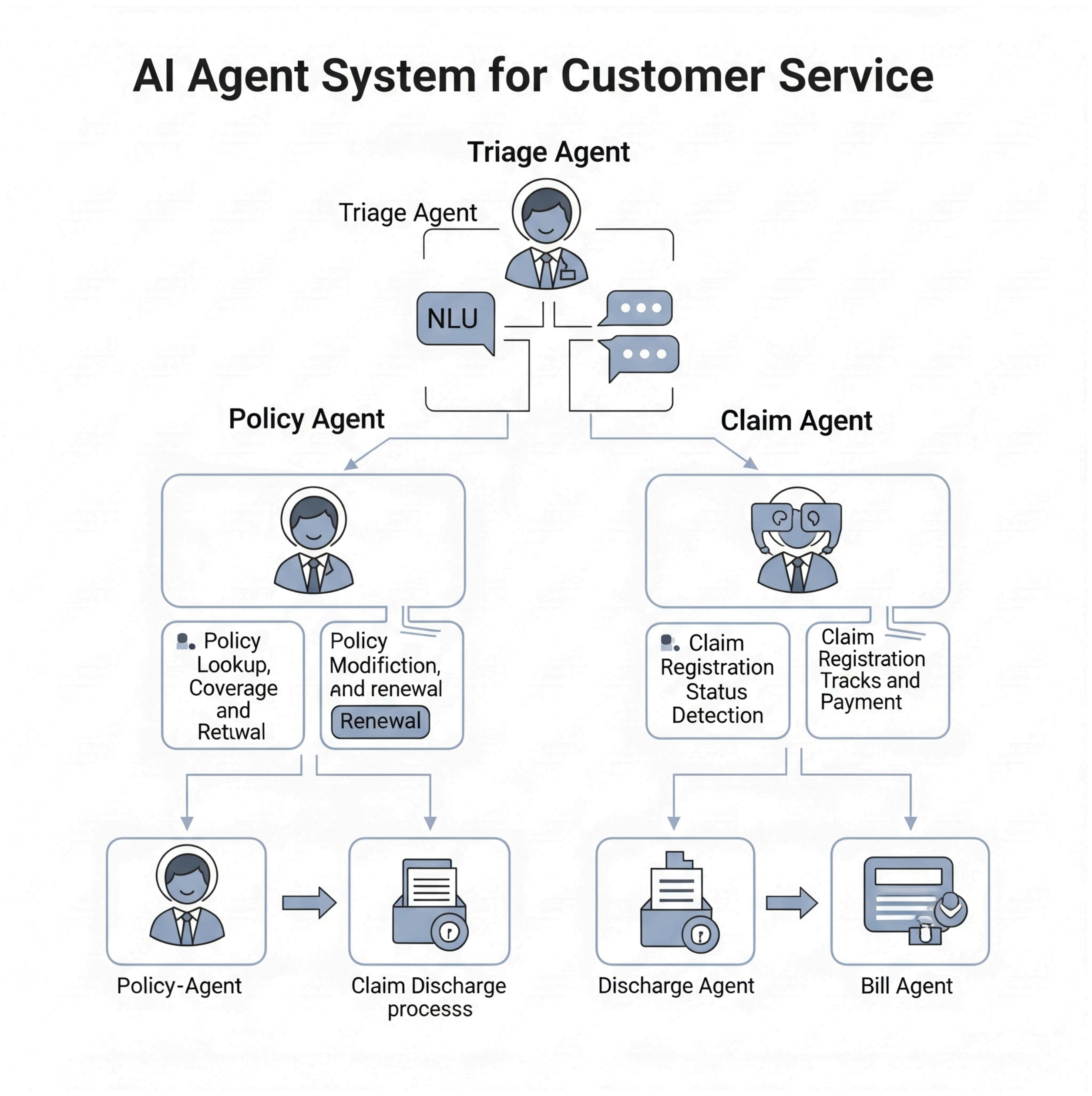

AI Agent System for Customer Service
Triage Agent
Triage Agent
NLU
Policy Agent
Claim Agent
Policy Lookup, Coverage and Retwal
Policy Modification, and renewal
Renewal
Claim Registration Status Detection
Claim Registration Tracks and Payment
Policy-Agent
Claim Discharge processs
Discharge Agent
Bill Agent

1. Overall Architecture Overview

The system is designed as a cloud-native, serverless, and highly scalable application. It follows a microservices-oriented approach where the core ADK agent logic is encapsulated within a containerized service deployed on Google Cloud Run. This service interacts with a managed PostgreSQL database for persistent data storage.

2. Component Breakdown

2.1. Google Cloud Run

Google Cloud Run serves as the primary hosting environment for the ADK agent.

- **Serverless Container Platform**: Cloud Run allows deploying containerized applications without managing the underlying infrastructure. It automatically scales up or down based on incoming request traffic, even scaling to zero instances when idle, which optimizes costs.
- **Request-Driven**: It's ideal for stateless, request-driven services like a web API, ensuring that the ADK agent is always ready to receive and process user requests.
- **Built-in Features**: Provides automatic HTTPS, custom domain mapping, traffic splitting, and integration with other Google Cloud services.

2.2. Google ADK + FastAPI Web Server

FastAPI is chosen as the web framework within the Cloud Run container.

- **High Performance**: Built on Starlette and Pydantic, FastAPI offers excellent performance comparable to Node.js and Go.
- **Asynchronous Support**: Its native support for `async`/`await` is crucial for handling concurrent requests efficiently, especially when the ADK agent might be waiting on external API calls or database operations.

2.3. Google ADK Agentic Framework

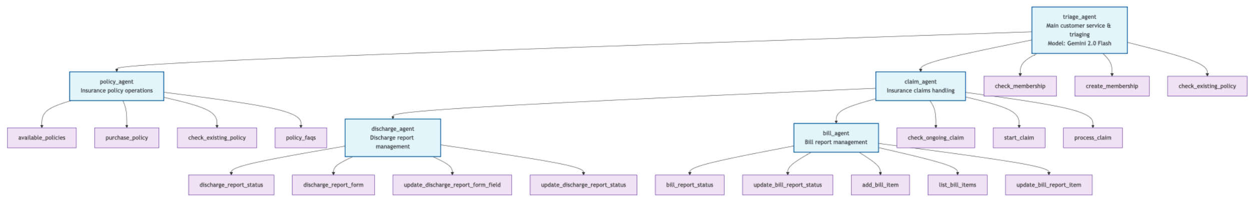
The Google ADK (Agent Development Kit) framework powers the core intelligence and orchestration of the agent.

- **Agentic Capabilities**: Provides the foundational components for building AI agents, including capabilities for understanding natural language, managing conversation state, reasoning, and tool utilization.
- **Tool Integration**: Facilitates the integration of various tools (e.g., `available_policies`, `purchase_policy`, `check_ongoing_claim`, etc.
- These tools use Fire Store for storing state along with session state provided by ADK.
- **Orchestration**: Manages the flow of interaction, deciding which tools to use and when, based on user input and prompt instructions.

3. Data Flow and Interactions

1. **User Request**: An end-user initiates an interaction (e.g., via a chat interface or an application) that sends an HTTP request to the ADK agent.
2. **Cloud Run Invocation**: The request triggers an instance of the ADK Agent service on Google Cloud Run.
3. **ADK's + FastAPI Server**: FastAPI receives the HTTP request and directs it to the appropriate endpoint.
4. **ADK Framework Execution**: The FastAPI endpoint invokes the Google ADK agentic framework. The framework processes the user's input, leveraging an LLM (e.g., Gemini 2.0 Flash) for natural language understanding and determining the appropriate action.
5. **Tool Execution**: Based on the agent's reasoning, it may invoke one or more internal "tools."
    - **Database Interaction**: If a tool requires data (e.g., `check_existing_policy`, `check_ongoing_claim`), the ADK framework, via FastAPI, connects to Google FireStore (Firebase) to query or update the necessary information.
    - **ADK Session Store:** Google ADK uses SQL data store for persistence powered by PostgreSQL.
6. **Response Generation**: After executing tools and processing data, the ADK framework generates a natural language response (if applicable) or a structured data response.
7. **User Receives Response**: The response is returned to the end-user and and rendered by Google's ADK web UI.

Google ADK Multi Agent Workflow Architecture


triage_agent
Main customer service & triaging
Model: Gemini 2.0 Flash
policy_agent
Insurance policy operations
claim_agent
Insurance claims handling
check_membership
create_membership
check_existing_policy
available_policies
purchase_policy
check_existing_policy
policy_faqs
discharge_agent
Discharge report management
bill_agent
Bill report management
check_ongoing_claim
start_claim
process_claim
discharge_report_status
discharge_report_form
update_discharge_report_form_field
update_discharge_report_status
bill_report_status
update_bill_report_status
add_bill_item
list_bill_items
update_bill_report_item