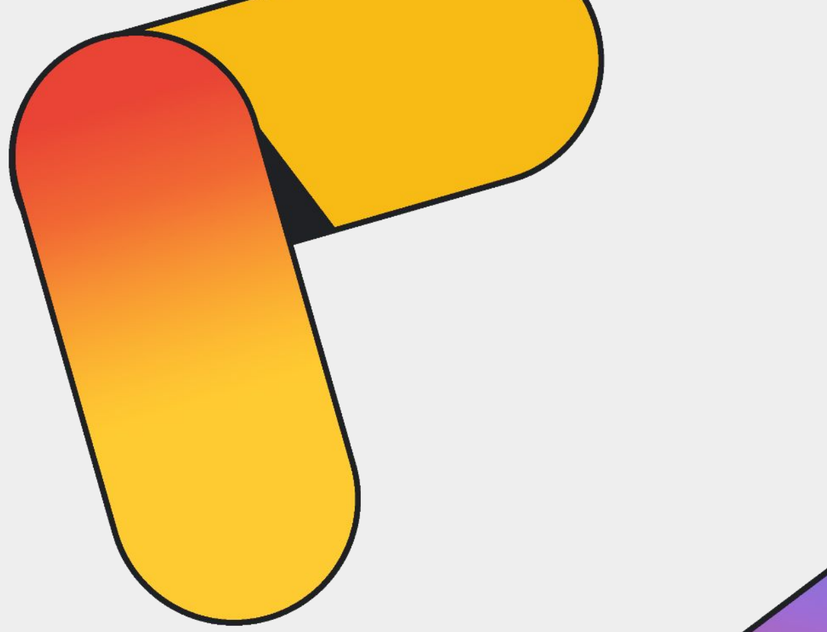
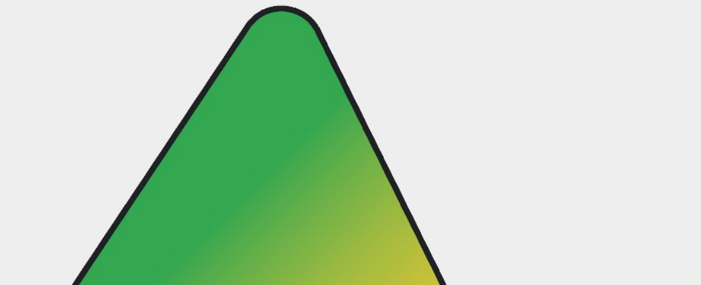
Google I/O Extended

Java in AI: Implementation of Google Vertex AI using Spring AI and Hilla





Google I/O Extended

Agenda

01 Introduction
02 Spring Boot Concepts
03 Spring AI Concepts
04 Google Gemini in Spring Boot
05 Hilla Framework Concepts
06 Demonstration (Chatbot)
- Architecture
- Live-coding

Google I/O Extended

Tristan Mahinay

JUG PH Leader, Senior Technical Specialist, watsonx Innovations Lead @ IBM

rjtmahinay

Proprietary & Confidential

Why use Java?

Offers faster execution times!

	Time
(c) C	1.00
(c) Rust	1.04
(c) C++	1.56
(c) Ada	1.85
(v) Java	1.89
(c) Chapel	2.14
(c) Go	2.83
(i) Python	71.90

From "Energy Efficiency across Programming Languages"
by Rui Pereira et al., Universidade do Minho, Portugal.

Why use Java?

It‘s Eco-Friendly!

	Energy
(c) C	1.00
(c) Rust	1.03
(c) C++	1.34
(c) Ada	1.70
(v) Java	1.98
(c) Pascal	2.14
(i) Python	75.88

From "Energy Efficiency across Programming Languages"
by Rui Pereira et al., Universidade do Minho, Portugal.

Google I/O Extended

Spring Boot

Spring Boot

Simplify Java applications through auto-configuration

Spring Boot

Overview of Spring Boot

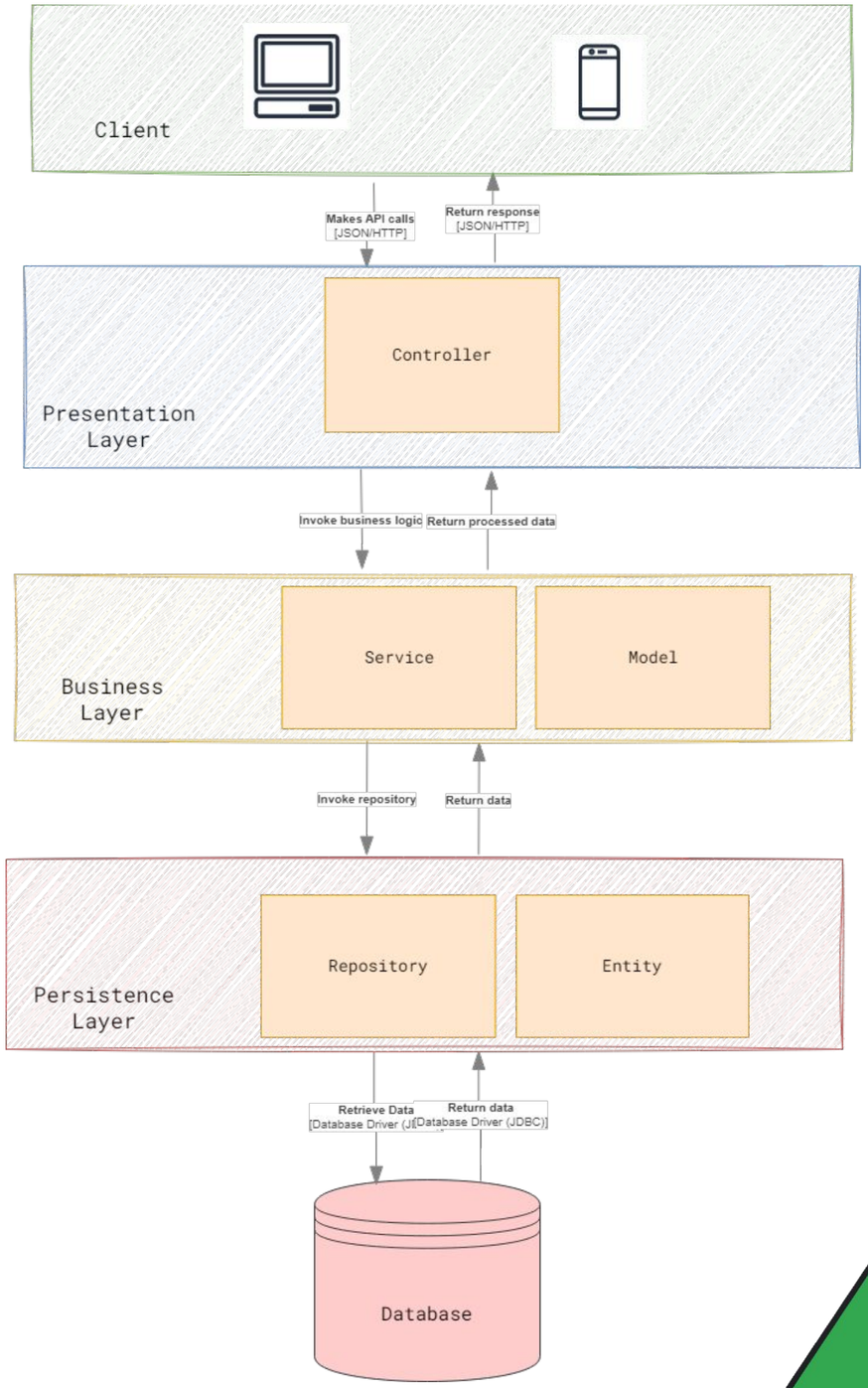
Simplifies Spring Framework configuration via annotations.

- **Annotations** - A declaration to describe a class, method, interface or field (variables).
- **Auto-configuration** - Spring Framework automatic resolution of dependencies.

```java
@SpringBootApplication
public class SpringAiHillaApplication {

  public static void main(String[] args) {
    SpringApplication.run(SpringAiHillaApplication.class, args);
  }

}
```

Google I/O Extended

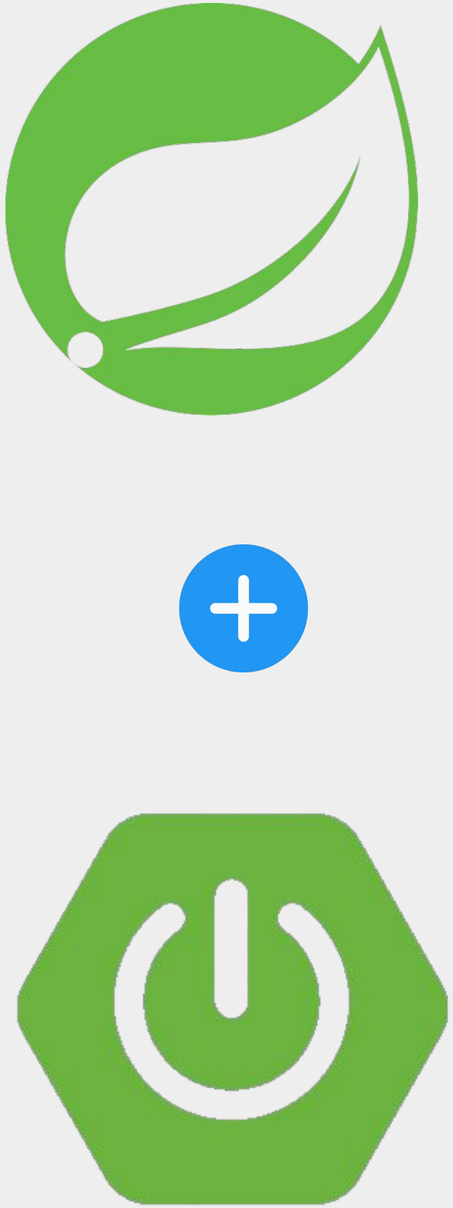
Spring Boot Layered Architecture



Spring Boot

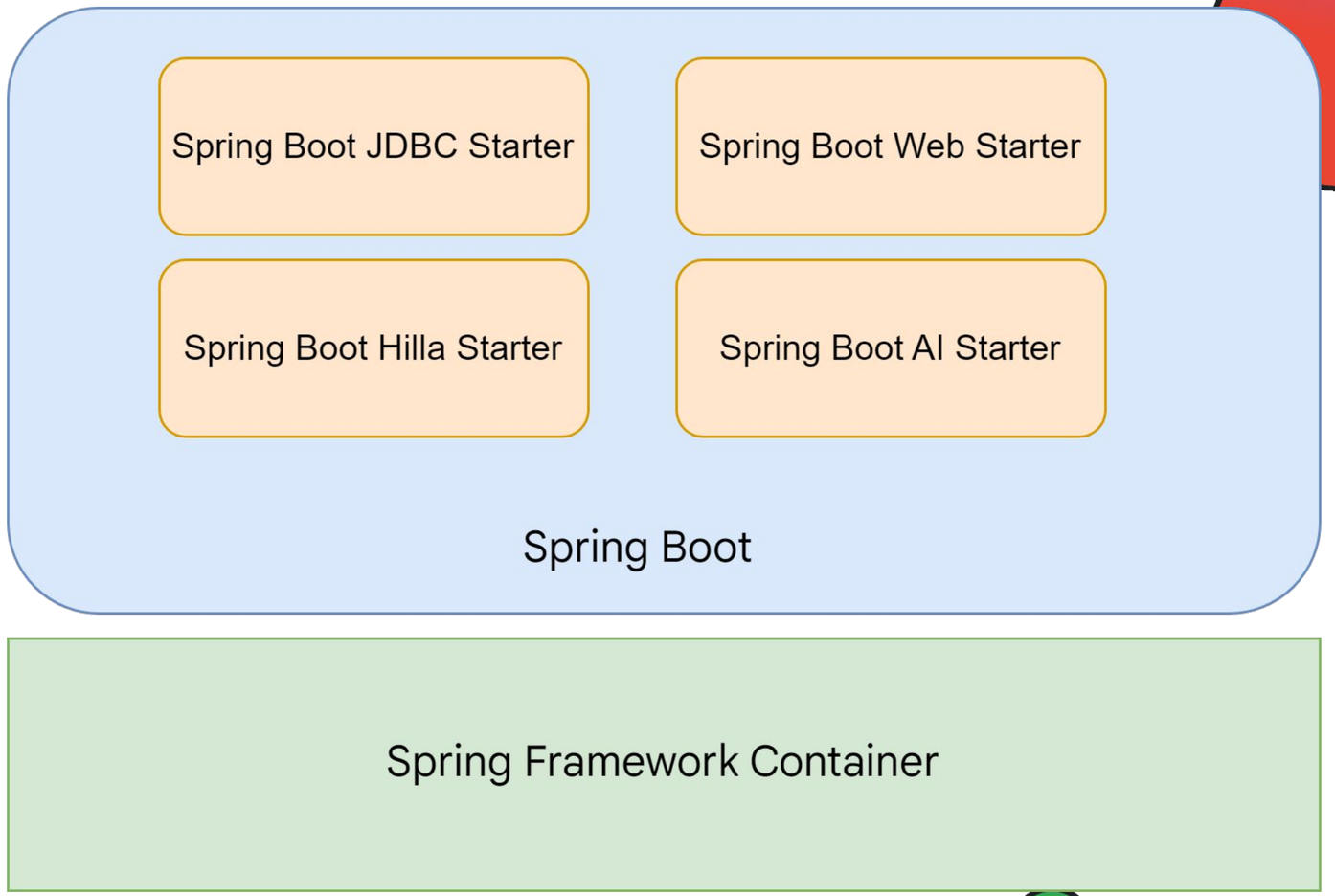
What are Spring Boot starters?

Simplifies Spring Framework dependency management. This includes autoconfiguration and dependency versioning resolution.



Google I/O Extended

Spring Boot Starters



```
dependencyManagement {
  imports {
    mavenBom "dev.hilla:hilla-bom:${hillaVersion}"
    mavenBom "org.springframework.ai:spring-ai-bom:${springAiVersion}"
  }
}

dependencies {
  implementation 'dev.hilla:hilla-react-spring-boot-starter'
}
```

Google I/O Extended

Spring AI

Spring AI

Simplify AI application development through abstraction

Spring AI

What is Spring AI?

An approach to streamline the development of artificial intelligence using Java, reducing the inherent complexities.

Inspired by LangChain and LlamaIndex. A Java version is made for LangChain which is LangChain4j.

Google I/O Extended

Use of abstractions with minimal coding changes

Goal

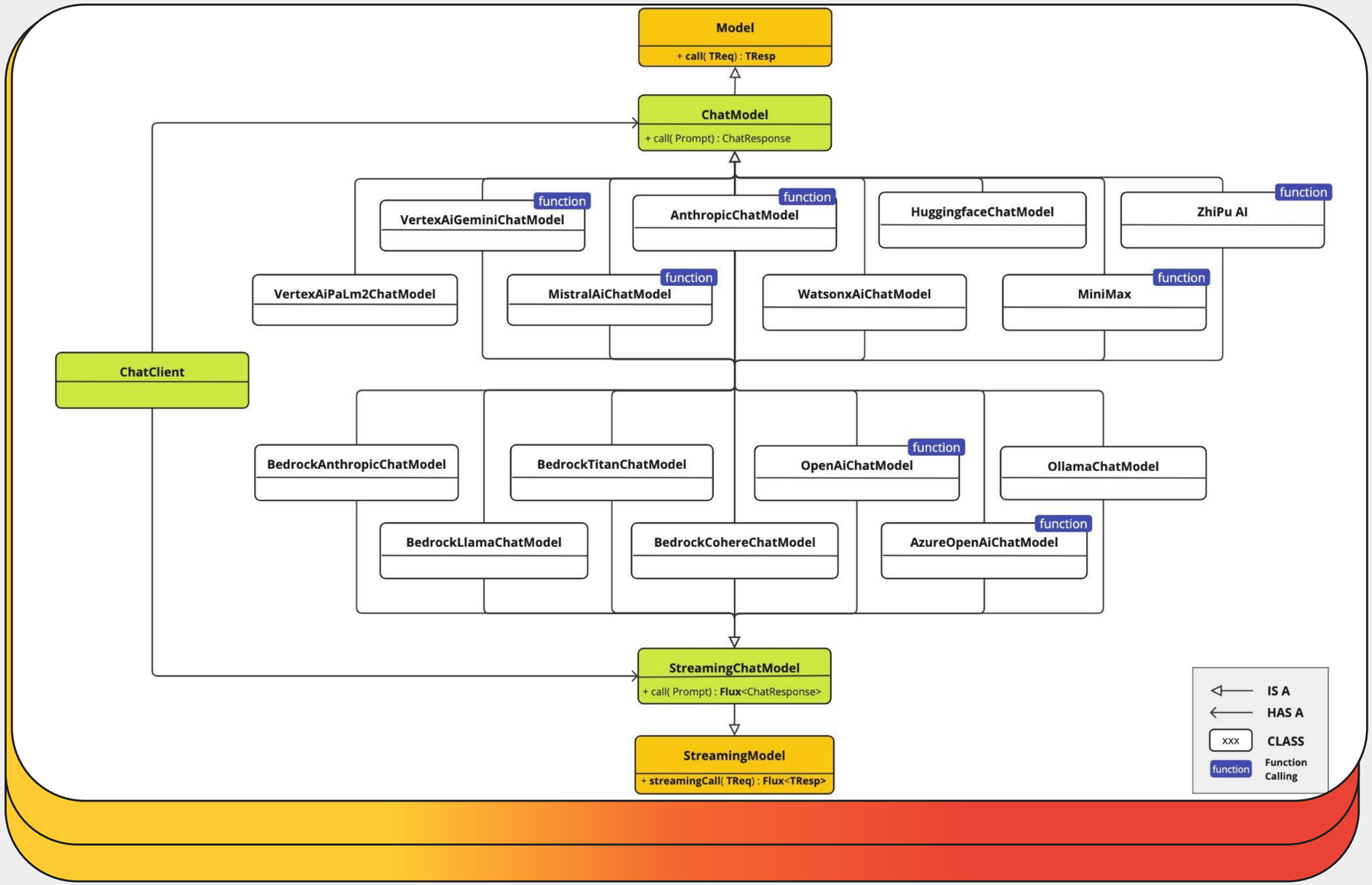
Spring AI

AI Model

Spring AI abstraction of Chat Models for different providers.

The Spring Framework managed all dependencies from the providers for minimal coding changes.

Source: Spring AI Documentation



Spring AI

Support for Major Model Providers

- Azure OpenAI
- Amazon Bedrock
- Google Vertex AI Gemini
- Hugging Face
- OpenAI

Support for different Model Types

- Audio
- Chat
- Image
- Embeddings

Vector Databases

Support for major vector databases to store documents into chunks for indexing and similarity search.

- Apache Cassandra
- Elastic Search
- Open Search
- Oracle

Spring Boot Starters

Utilizes Spring Boot's auto-configuration for different models and vector databases.

Portable API

- Call a function synchronously or asynchronously (streaming)
- SQL-like filter for Vector stores

Custom Output

Mapps the AI Model Output to a custom POJO of a developer.

Google I/O Extended

Google Gemini

Google Gemini in Spring Boot

Google Gemini Spring Boot Auto-configuration

Google Gemini

What are Gemini Models?

A multimodal model from the Gemini Model Family of Vertex AI. It accepts text, video and audio as an input in the prompt requests.

A non-multimodal model only accepts text.

Google I/O Extended

Gemini Models	Input	Output
Gemini 1.5 Flash	Text, code, images, audio, video, video with audio, PDF	Text
Gemini 1.5 Pro	Text, code, images, audio, video, video with audio, PDF	Text
Gemini 1.0 Pro	Text	Text
Gemini 1.0 Pro Vision	Text, images, audio, video, video with audio, PDF	Text
Gemini 1.0 Ultra	Text	Text
Gemini 1.0 Ultra Vision	Text, code, images, audio, video, video with audio, PDF	Text

```
dependencies {
  implementation 'org.springframework.ai:spring-ai-vertex-ai-gemini-spring-boot-starter'
  implementation
'org.springframework.ai:spring-ai-vertex-ai-embedding-spring-boot-starter'
}
```

Google I/O Extended

Hilla

Hilla Framework

Modern User-interface Framework for Java Engineers

Hilla

What is Hilla?

Spring Boot

Hilla uses Spring Boot as a back-end service in creating applications. It utilizes Spring and Hilla custom annotations and autoconfiguration.

ReactJS

Hilla integrates with ReactJS libraries to create user-interfaces. Additionally, Java classes like Controller and Models are converted to a TypeScript code.

Full-stack Development

A Java Engineer can develop enterprise grade applications with expertise with Spring Boot and ReactJS in one go.

Vaadin Components

Hilla is built on top of Vaadin. Vaadin UI Components can be integrated inside Hilla.

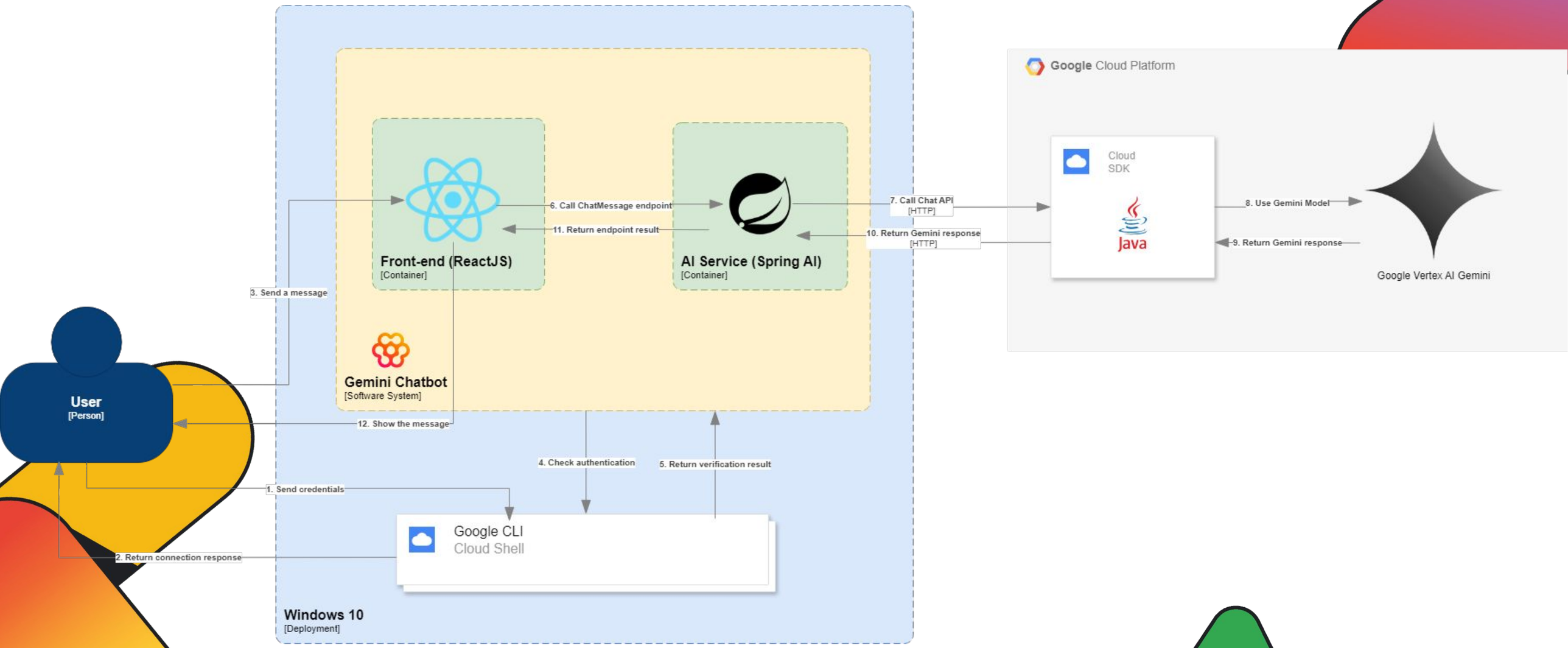
```java
@Endpoint
@AnonymousAllowed
@RequiredArgsConstructor
public class ChatEndpoint {

    private final ChatClient chatClient;

    public String generateChatMessage(String message) {
        return chatClient.prompt().user(message).call().content();
    }
}
```

```
import { EndpointRequestInit as EndpointRequestInit_1 } from "@hilla/frontend";

import client_1 from "./connect-client.default.js";

async function generateChatMessage_1(
      message: string | undefined, init?: EndpointRequestInit_1): Promise<string
    | undefined> {
 return client_1.call("ChatEndpoint", "generateChatMessage", { message }, init);
 }

export { generateChatMessage_1 as generateChatMessage };
```

Google I/O Extended

Demonstration

Google I/O Extended



Gemini Chatbot Dynamic Diagram

Google I/O Extended

Q & A

Google I/O Extended

Thank You

Tristan Mahinay he/him

Senior Technical Specialist