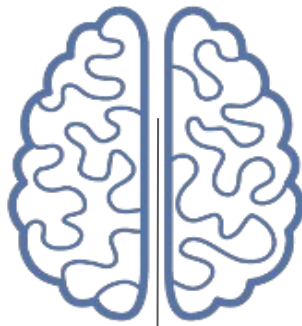




QUARKUS

AI for Application Developers

Max Rydahl Andersen, Distinguished Engineer, Red Hat



**Any sufficiently advanced
technology is
indistinguishable from
magic.**

**Magic is just science we
don't understand yet.**

Arthur C. Clarke, 1968

World Simplest AI
Infused Quarkus App

```
//DEPS io.quarkus.platform:quarkus-bom:3.11.1@pom
//DEPS io.quarkus:quarkus-picocli
//DEPS io.quarkiverse.langchain4j:quarkus-langchain4j-openai:0.15.1
```

```
import dev.langchain4j.model.chat.ChatLanguageModel;
import jakarta.inject.Inject;
import picocli.CommandLine.Command;
```

```
@Command
```

```
public class jokes implements Runnable {
```

```
    @Inject
```

```
    private ChatLanguageModel ai;
```

```
    @Override
```

```
    public void run() {
```

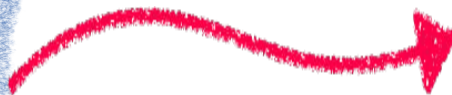
```
        System.out.println(ai.generate("tell me a joke"));
```

```
    }
```

```
}
```

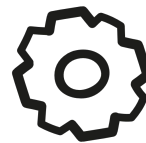
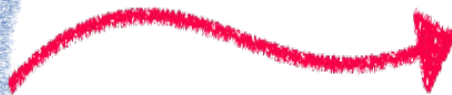
AI Models are just another Service

CRUD
application



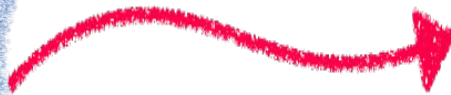
Database

Microservice



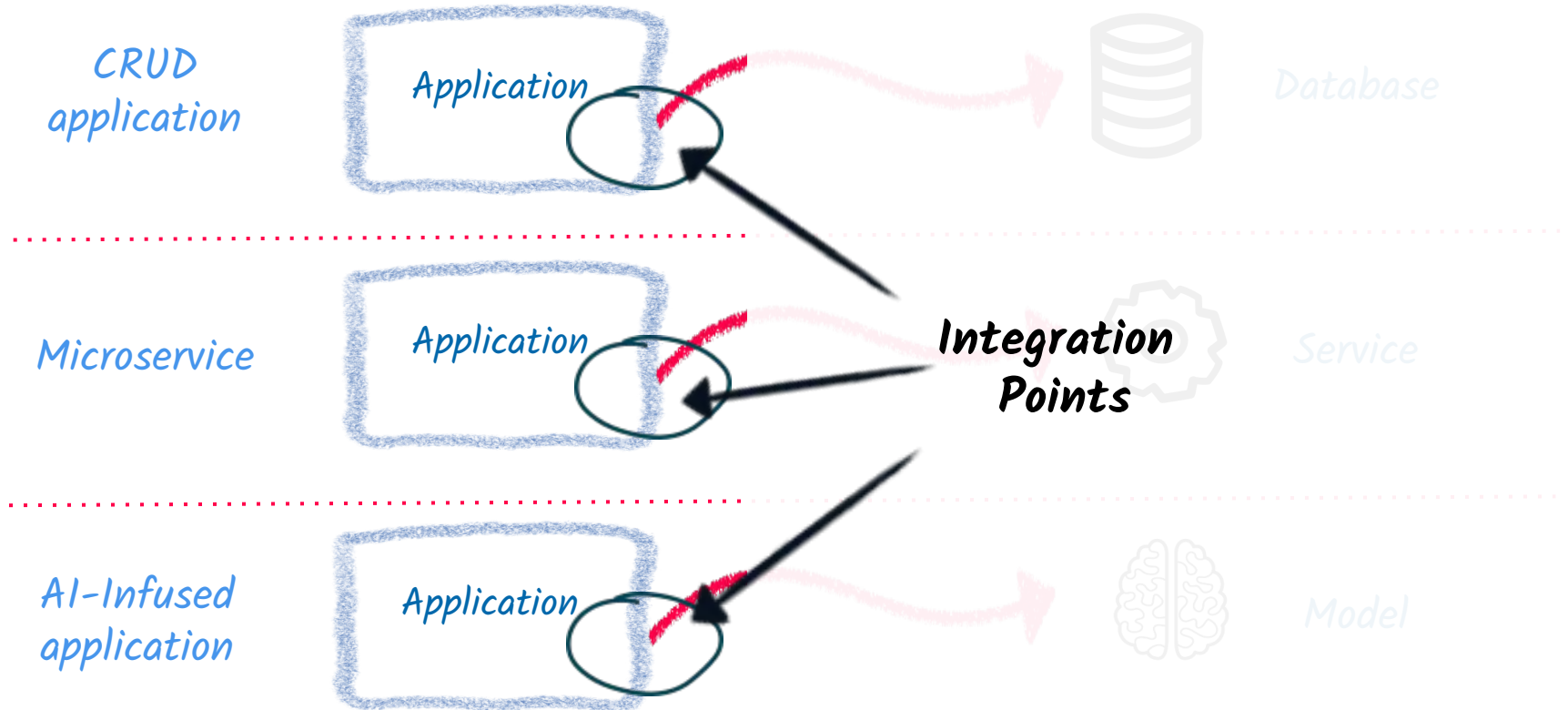
Service

AI-Infused
application

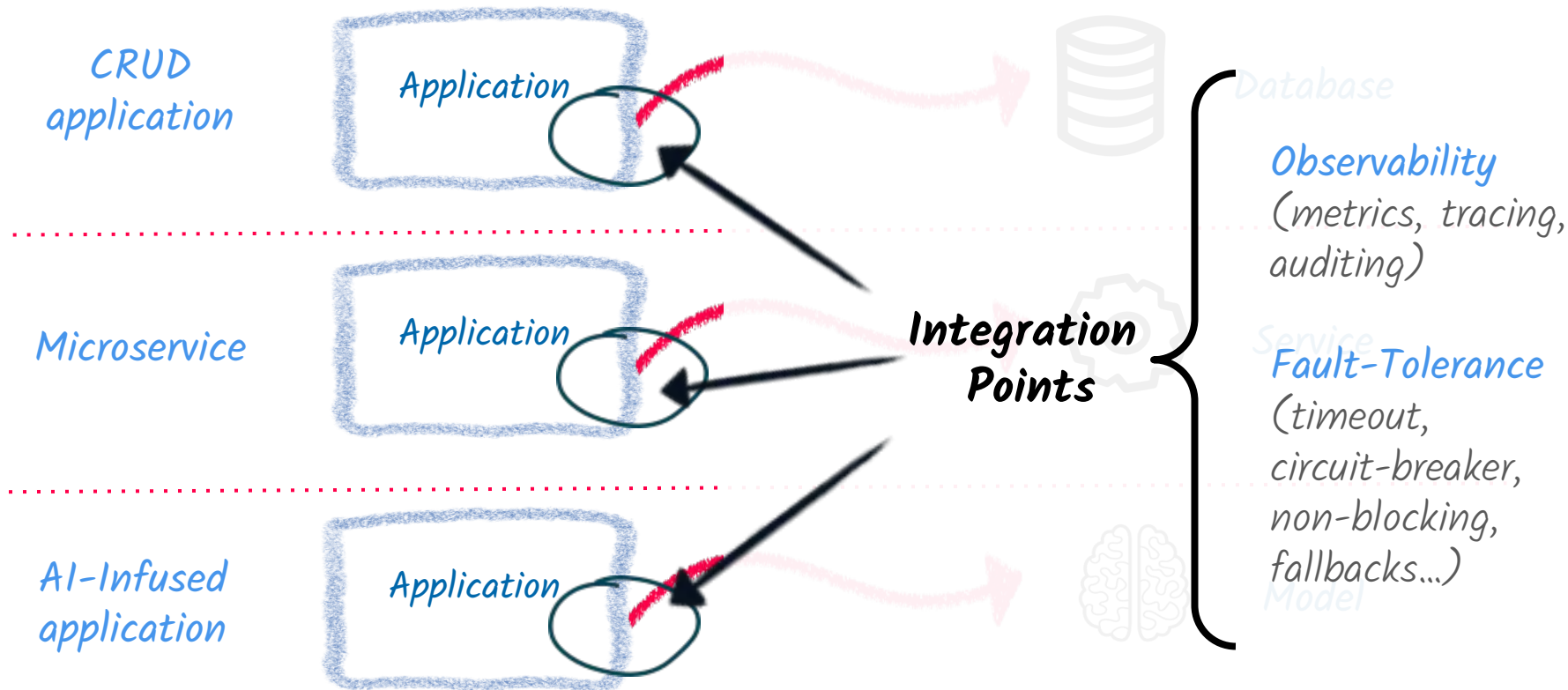


Model

What's the differences between these?



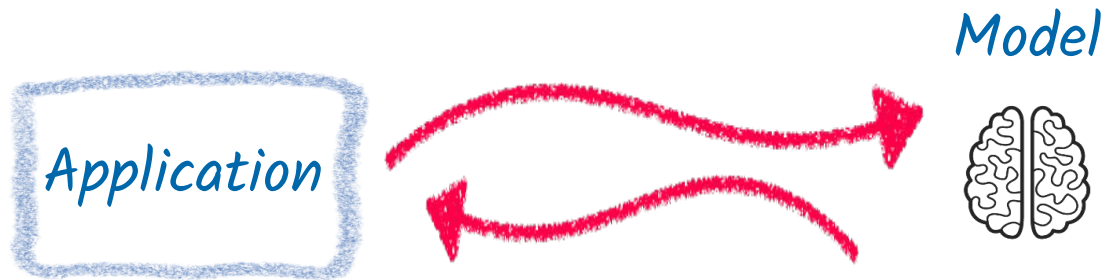
What's the difference between these?



AI-infused application |,eɪ,aɪ 'ɪn,fju:zd ,æplɪ'keɪʃən|

noun (Plural AI-Infused applications)

A software program enhanced with artificial intelligence capabilities, utilizing AI models to implement *intelligent* features and functionalities.



What are Large Language Models (LLMs)?

Neural Networks

- **Transformer** based
- Recognize, Predict, and Generate **text**
- **Trained** on a **VERY** large corpuses of text
- Deduce the **statistical** relationships between tokens
- Can be **fine-tuned**

**A LLM predicts the next token
based on its training data and statistical deduction**

The **L** of **LLM** means **Large**

LLama 3:

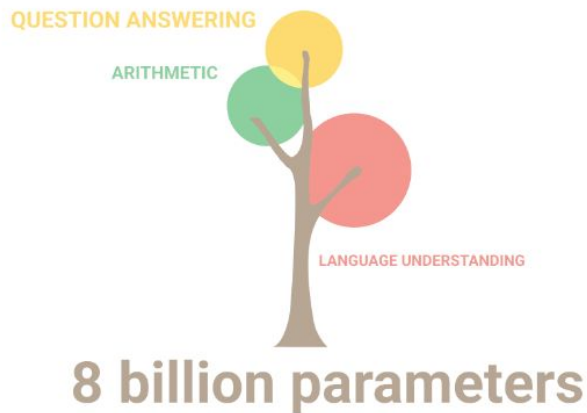
- 7B parameters
- Trained on 15000B of tokens
- 4.7Gb on disk

Granite:

- 34B parameters
- Trained on 3500B of tokens
- 3.8 Gb of RAM, 4.8Gb on disk

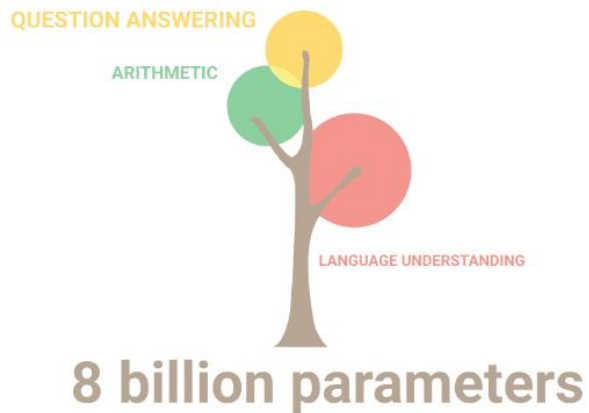
More on: [An idea of the size](#)

More parameters means more capabilities



<https://research.google/blog/pathways-language-model-palm-scaling-to-540-billion-parameters-for-breakthrough-performance/>

Less can be fine...



<https://research.google/blog/pathways-language-model-palm-scaling-to-540-billion-parameters-for-breakthrough-performance/>

How does it work?

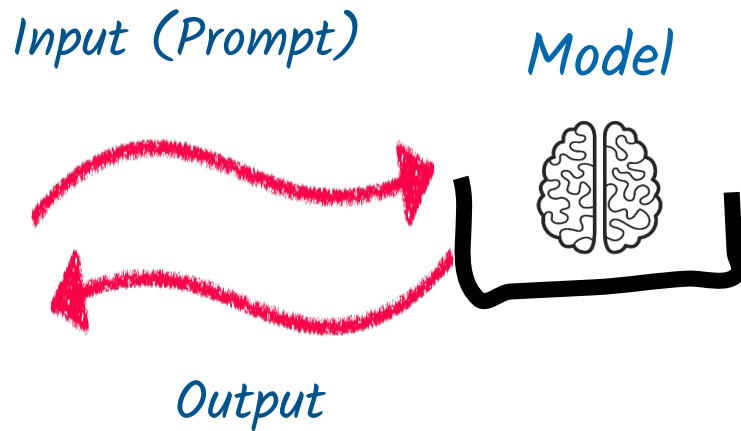
Prompt and Prompt Engineering

Input:

- Prompt (text)
- Instructions to give to the model
- **Taming a model is hard**

Output:

- Depends on the modality of the model



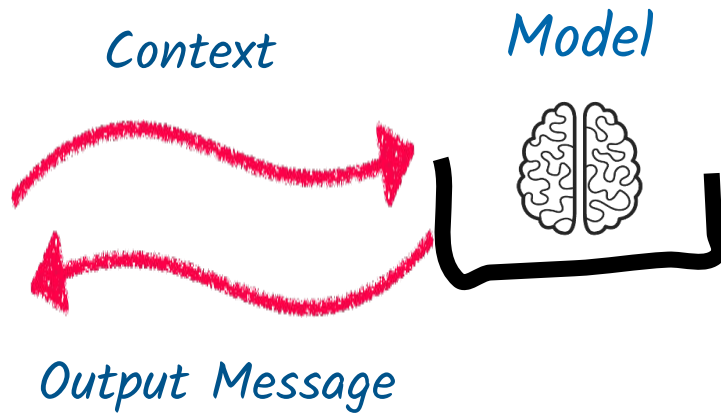


**Does LLM models
remember what you told
them?**

Messages and Memory

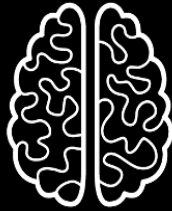
Models are stateless

- Pass a set of messages named **context**
- These messages are stored in a **memory**
- Context size is limited (eviction strategy)
- Quarkus manages this using CDI beans and scopes

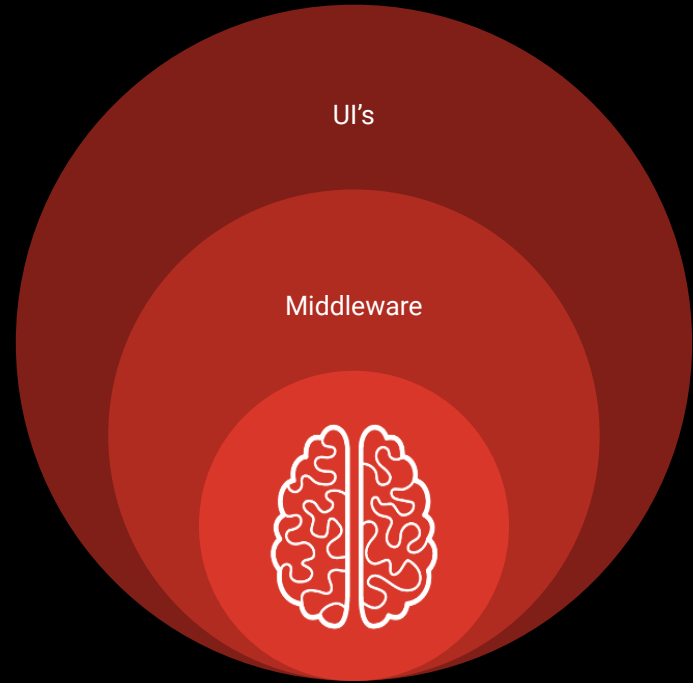


Context = (Stored input messages + Output messages) + New input

Model



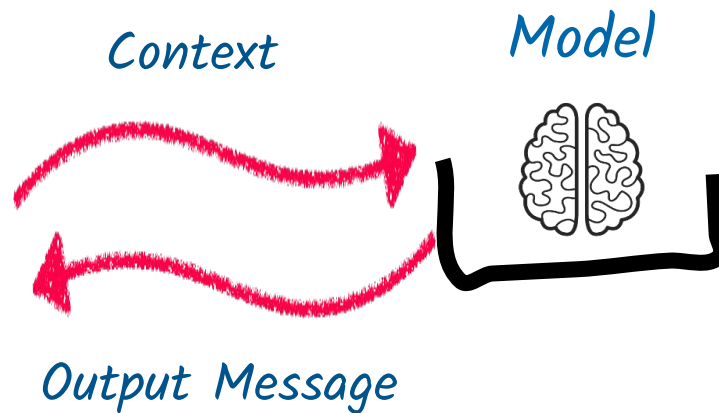
AI Infused Application



Context = Prompt

Prompts are king

Learn & Explore Patterns

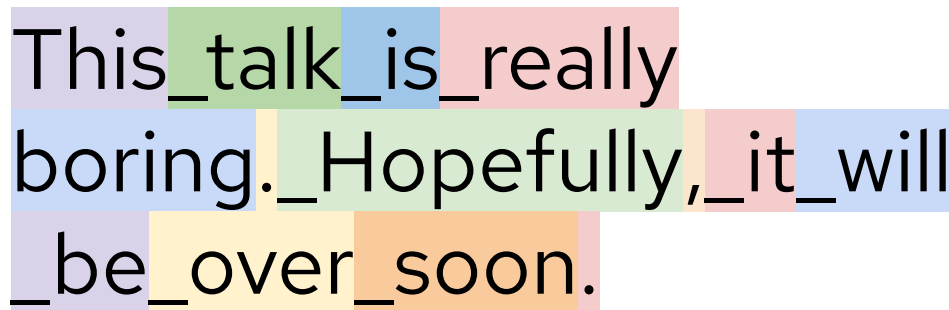


Context = (Stored input messages + Output messages) + New input

Context Limit & Pricing

Number of tokens

- Depends on the model and model serving (provider)
- Tokens are **not** words



This talk is really boring. Hopefully, it will be over soon.

The diagram illustrates how a sentence is broken down into tokens. Each token is represented by a colored rectangular block. The tokens are: 'This', '_', 'talk', '_', 'is', '_', 'really', '.', 'Hopefully', ',', 'it', '_', 'will', '_', 'be', '_', 'over', '_', 'soon', and '.'.

[2028, 3137, 374, 2216, 28859, 13, 38994, 11, 433, 690, 387, 927, 5246, 13, 220]

How many Rs in the word strawberry?

Tokens

8

Characters

35

How many Rs in the word
strawberry?

Tokens

8

Characters

35

[4438, 1690, 19766, 304, 279,
3492, 73700, 30]

Can it do Better?

how many r's in the word strawberry?



The word "strawberry" has **two** r's.

can you please check with a python script ?



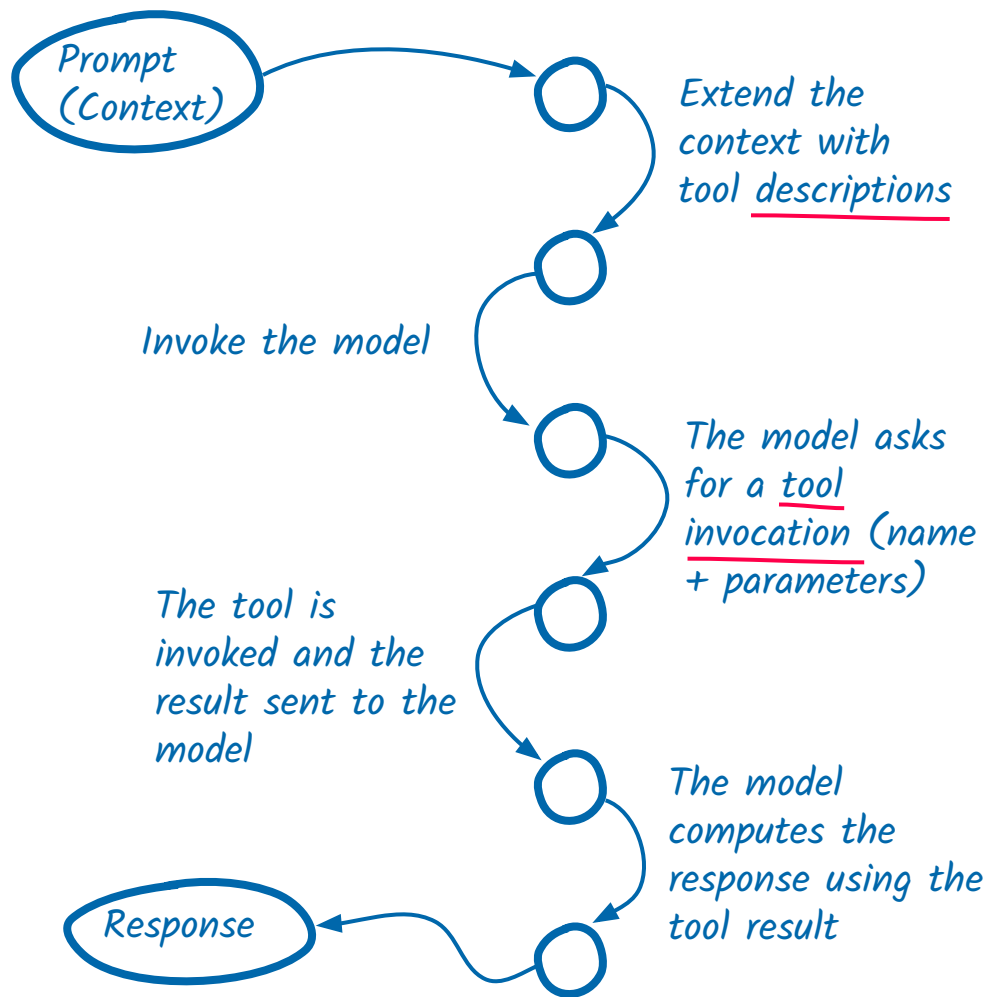
There are actually **three** r's in the word "strawberry." Thanks for asking to verify! [;-]



Agent and Tools

A **tool** is a function that the model can call:

- Tools are parts of CDI bean
- Tools are defined and described using the **@Tool**



Tools in Quarkus LangChain4j

```
@RegisterAiService(tools = EmailService.class)

public interface MyAiService {

    @SystemMessage("You are a professional poet")
    @UserMessage("""
        Write a single poem about {topic}. The poem should be {lines} lines long and your response should only include the poem itself, nothing
        Then send this poem by email. Your response should include the poem.
        """)
    String writeAPoem(String topic, int lines);

}

@ApplicationScoped
public class EmailService {

    @Inject
    Mailer mailer;

    @Tool("send the given content by email")
    public void sendAnEmail(String content) {
        Log.info("Sending an email: " + content);
        mailer.send(Mail.withText("sendMeALetter@quarkus.io", "A poem for you", content));
    }

}
```


Extensions

Configuration

Endpoints

Continuous Testing

Dev Services

Build Metrics

☒ Show tool-related messages☒ Enable Retrieval Augmented Generation (if a RetrievalAugmentor bean exists)[Start a new conversation](#)

System message:

Me

Write me a haiku about Quarkus and AI.

<~~ My prompt

Tools

Request to execute the following tools:

Request ID = call_gStAr1vpKRiMAsK5KIdGvoPt,

tool name = sendAnEmail,

arguments = {"content": "Quarkus speeds things up\nAI learns, predicts, evolves\nFuturistic world"}

Request ID = call_il2iOql3PktW7146w09YaAjy,

tool name = sendAnEmail,

arguments = {"content": "Quarkus and AI\nInnovation hand in hand\nBringing future near"}

<~~ Tool invocation request

Tools

Tool execution result for request ID = call_gStAr1vpKRiMAsK5KIdGvoPt,

tool name = sendAnEmail,

status = Success

<~~ Tool invocation response

Tools

Tool execution result for request ID = call_il2iOql3PktW7146w09YaAjy,

tool name = sendAnEmail,

status = Success

AI

I have written two haikus for you about Quarkus and AI. Here they are:

1. Quarkus speeds things up

AI learns, predicts, evolves

Futuristic world

2. Quarkus and AI

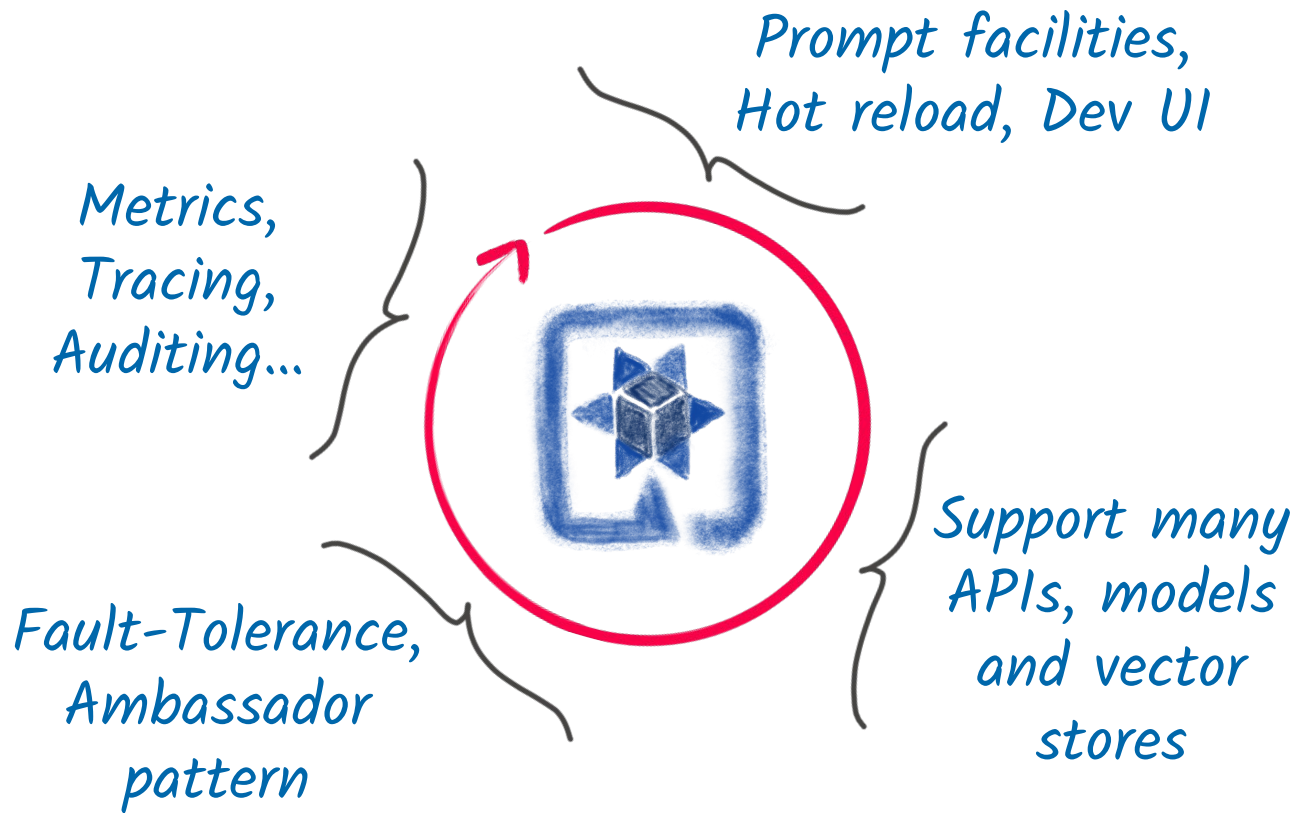
Innovation hand in hand

Bringing future near

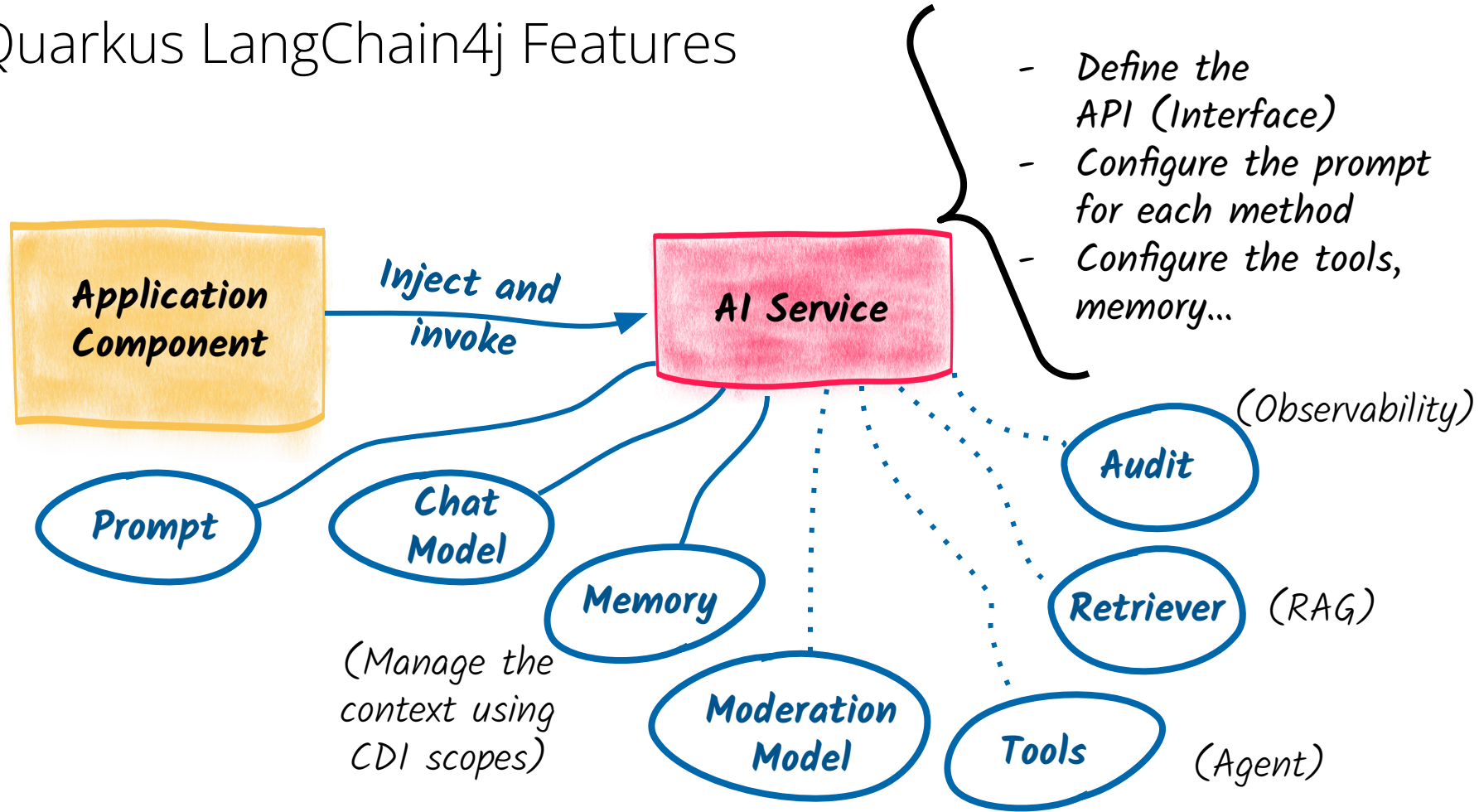
<~~ Model Response

I hope you enjoy them!

Why AI-infused applications with Quarkus?



Quarkus LangChain4j Features

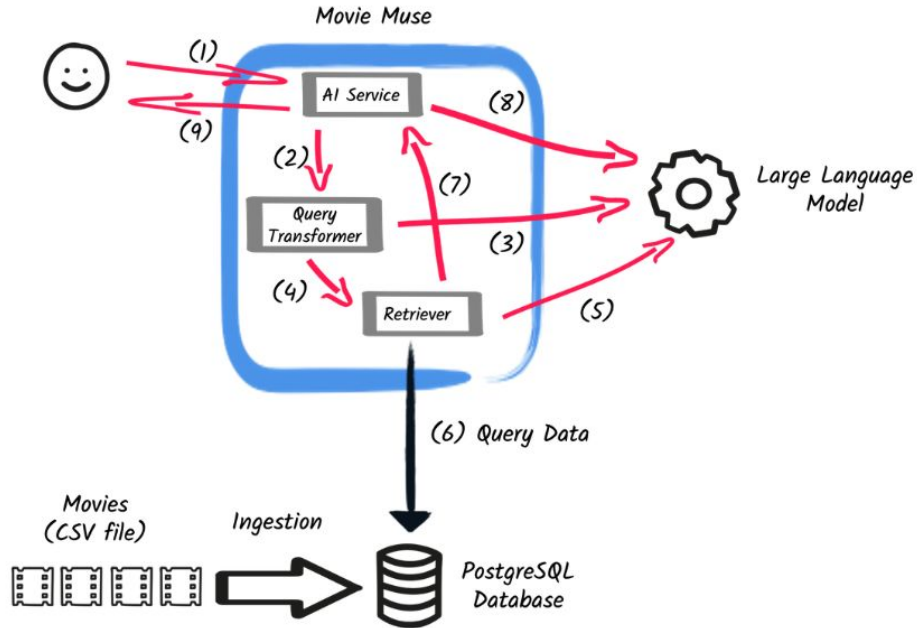


chat-easy-rag

Simplest way to do a RAG use case on set of docs

sql-chatbot

Using Tools to dynamically generate SQL queries to get user requested data.



What can *you* do?

LLM Bank Customer Usecases

“Pay 23.000 kr to Mr. Bean for robot mower”

“Did I ever pay that ~2500 bill in 2023?”

LLM Employee Usecases

“Evaluate today's messages and order them based on urgency”

“Check my message for grammar errors”

“Review my messages based on previous interactions with customer”

“Did I miss to follow up on any messages or task yesterday?”

LLM Acme Corp Developer Usecases

“Generate test data for this test run”

“Summarize results”

“Open issue based on CI failures”

“Automate calls to our infrastructure”

Conclusion

A stack to write Java apps



Cloud-Native



Microservices

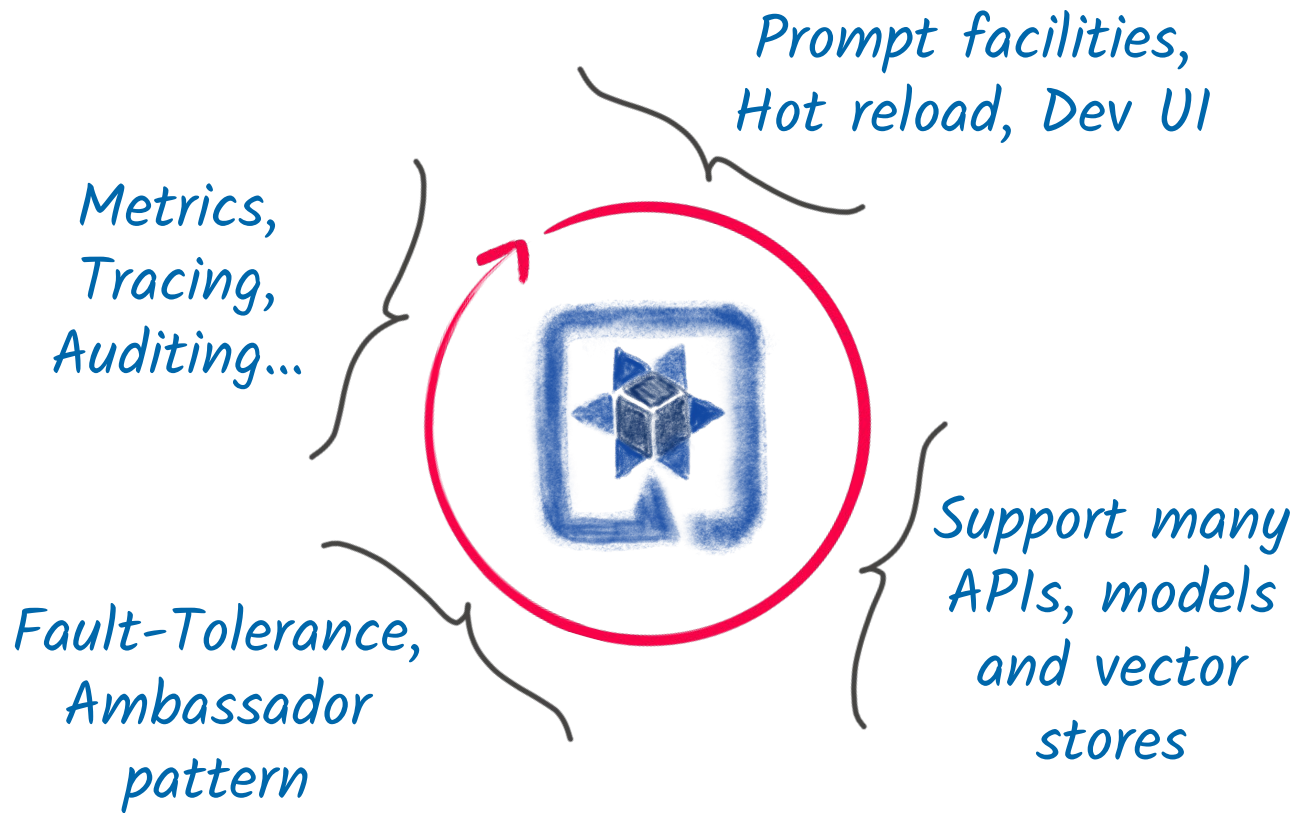


Serverless



AI-Infused

Why AI-infused applications with Quarkus?



It runs on and with all the things!

Podman Desktop / AI

OpenShift AI

RHEL AI

InstructLab

Granite

...and anything else you can find

Summary

GenAI is good old neural networks with bigger compute

Available today, just use it via service or locally

Learn and understand the powers and limits of LLM

Quarkus makes it easy to get started

Go Explore!

```
git clone https://github.com/quarkiverse/quarkus-langchain4j/
```

```
cd samples
```