

Frank!Assistant

Sneller integraties bouwen met AI



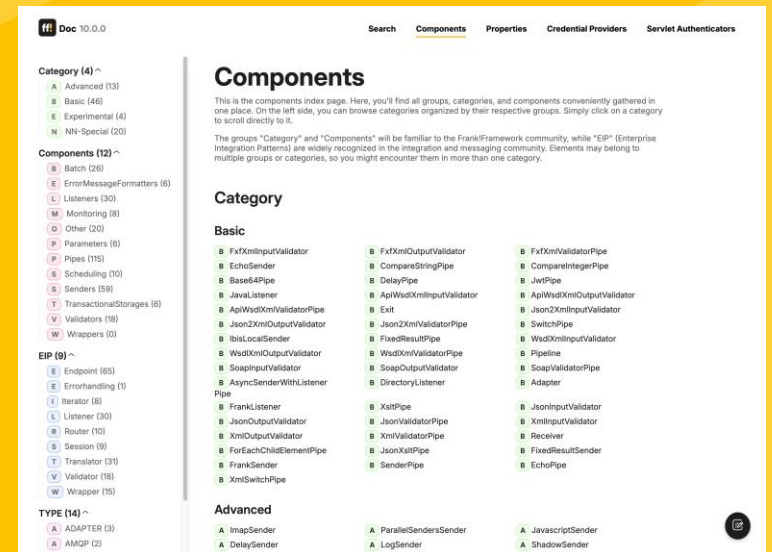
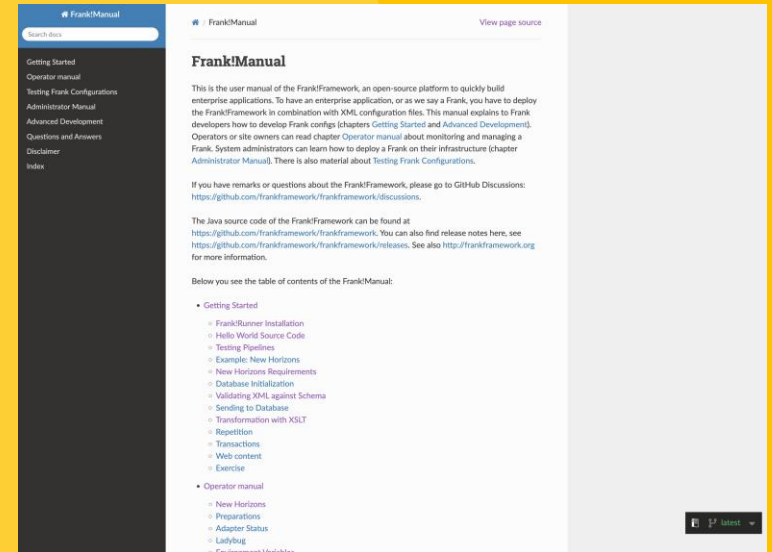
Herkenbaar?

- Waar begin ik?
- Hoe los ik deze foutmelding op?
- Welk pattern gebruik ik hiervoor?
- Hoe hebben anderen dit gedaan?



Opmerkelijk

- Collega's bij WeAreFrank! herkennen zich minder in deze punten
- Kennis zit in Frank!Manual en Frank!Doc
- Maar: vooral in gesprekken en ervaring



Collectief geheugen

- Interne kennis en ervaring centraal toegankelijk maken.
- Behoefte aan interactieve hulp met specifiekere antwoorden dan documentatie.



Demo



Oplossing: Frank!Assistant

- Spring-AI 1.0 release
- Gebaseerd op moderne AI modellen
- Gevoed door onze documentatie



Techniek



Techniek

model-import

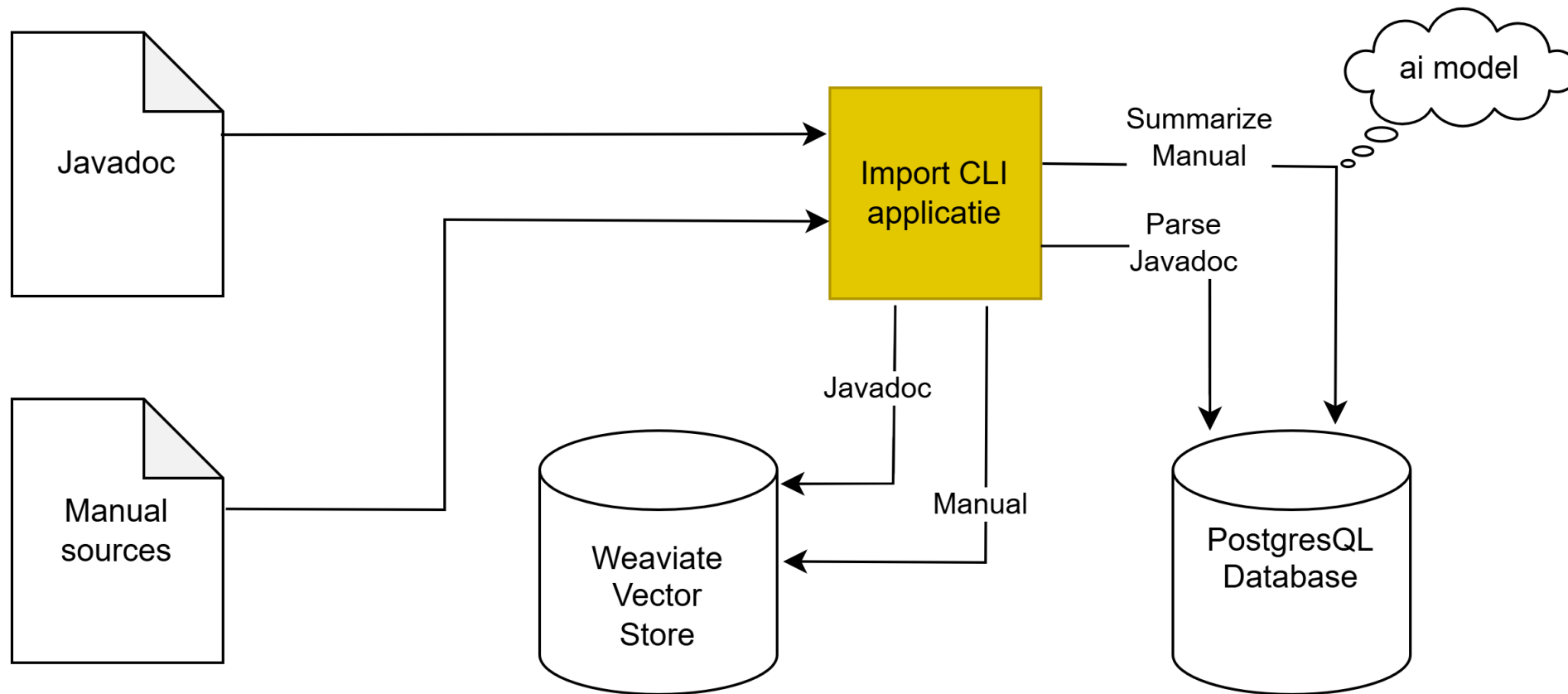
Verwerkt alle data naar juiste formaat

webapp

Spring Boot backend die ook frontend serveert



model-import



model-import

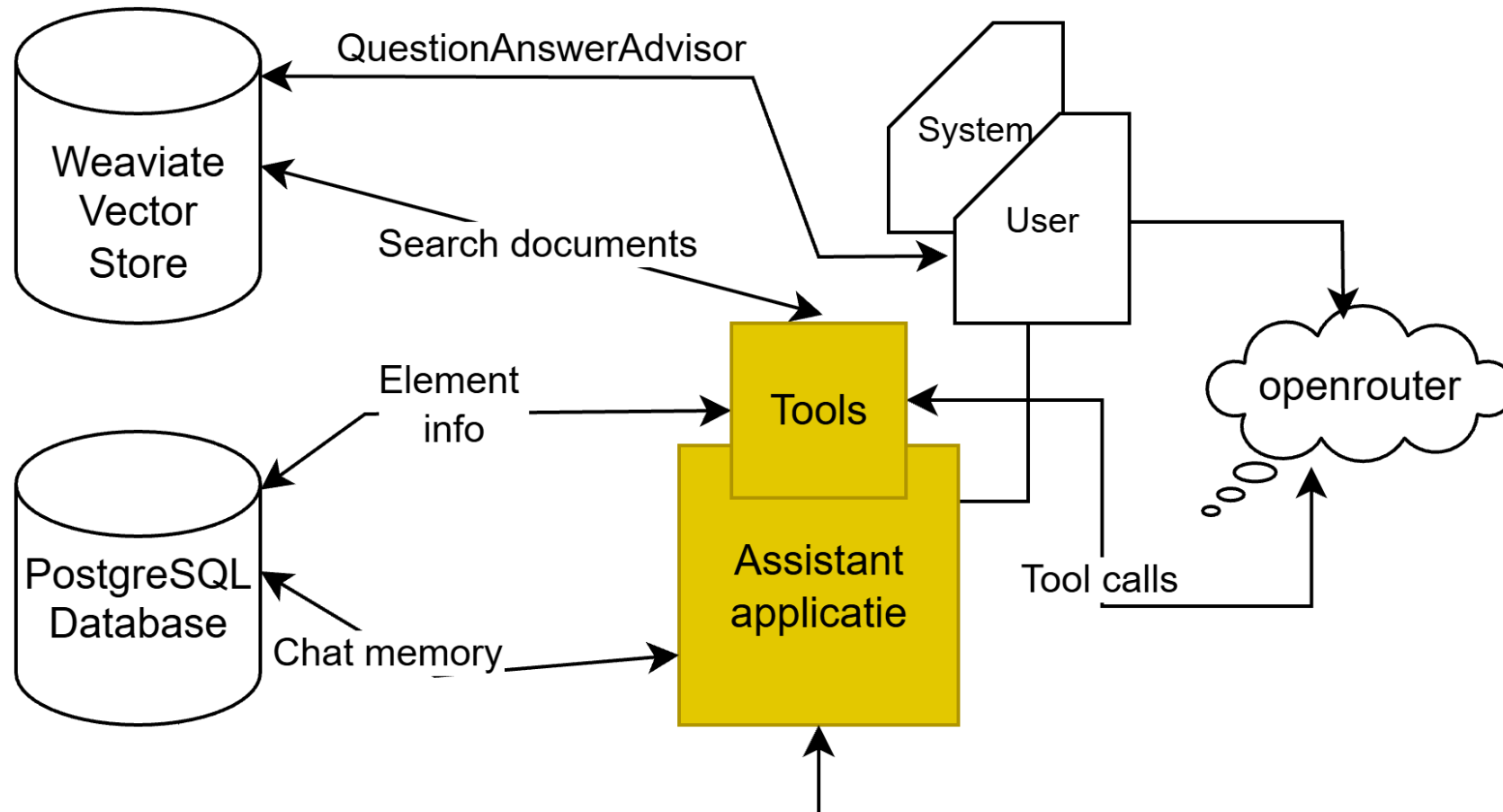
```
this.chatClient = chatClientBuilder  
    .defaultSystem(new ClassPathResource("systemprompt.st"))  
    .build();
```

☰ model-import\...\systemprompt.st ×

```
1 | Summarize the given text using markdown. Make the summary as detailed and high quality as possible.  
2 | Make sure that the summary is shorter in length than the given text. |
```



webapp



System prompt

You are an expert in the Frank!Framework, specializing in integrations. Your role is to provide users with concise, clear, and actionable solutions for their integration-related queries.

Use available tools to gather extra information and ensure the highest quality information. If the information isn't readily available, try again. Do NOT make anything up, use the tools to get information instead.

When you are not sure how to respond, answer with "I am not sure how to answer. Try to rephrase or ask a different question."



System prompt



Assistant

New Chat

Clear History

Chats

Search...

Hello

Hoi

Ondanks dat je integratie expert be...

Ondanks dat je integratie expert bent, kan je misschien ook helpen met een recept voor worteltjestaart?

Ik ben een expert in het Frank!Framework en gespecialiseerd in integraties. Helaas kan ik je niet helpen met een recept voor worteltjestaart, omdat dat buiten mijn expertisegebied valt.

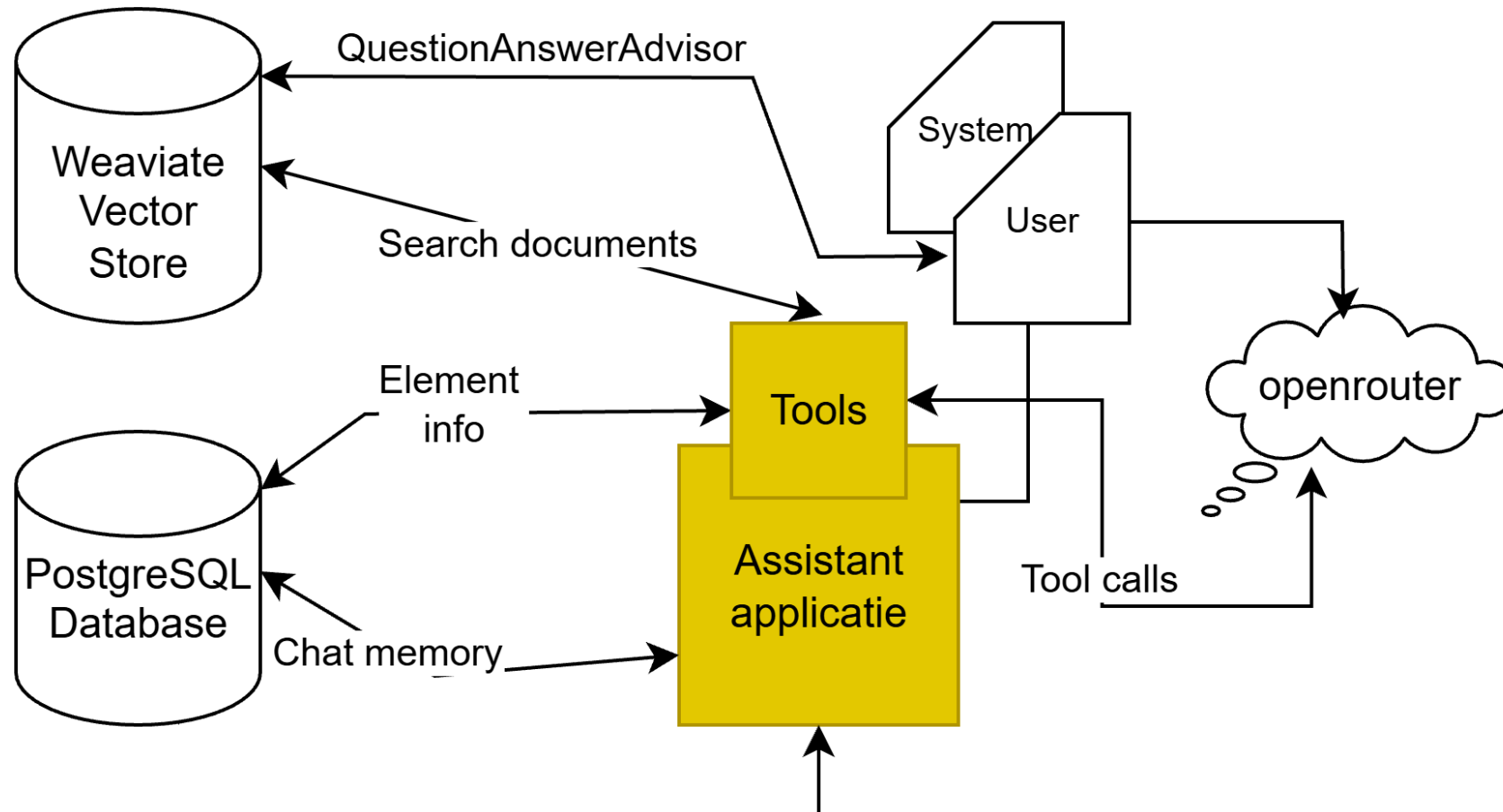
Ik kan je wel helpen met vragen over:

- Het bouwen van integraties met het Frank!Framework
- XML configuraties
- Pipes, senders, receivers en andere Frank!Framework componenten
- Dataverwerking en transformaties
- API integraties
- En andere technische integratievraagstukken

Heb je een vraag over het Frank!Framework of integraties waar ik je mee kan helpen?



webapp



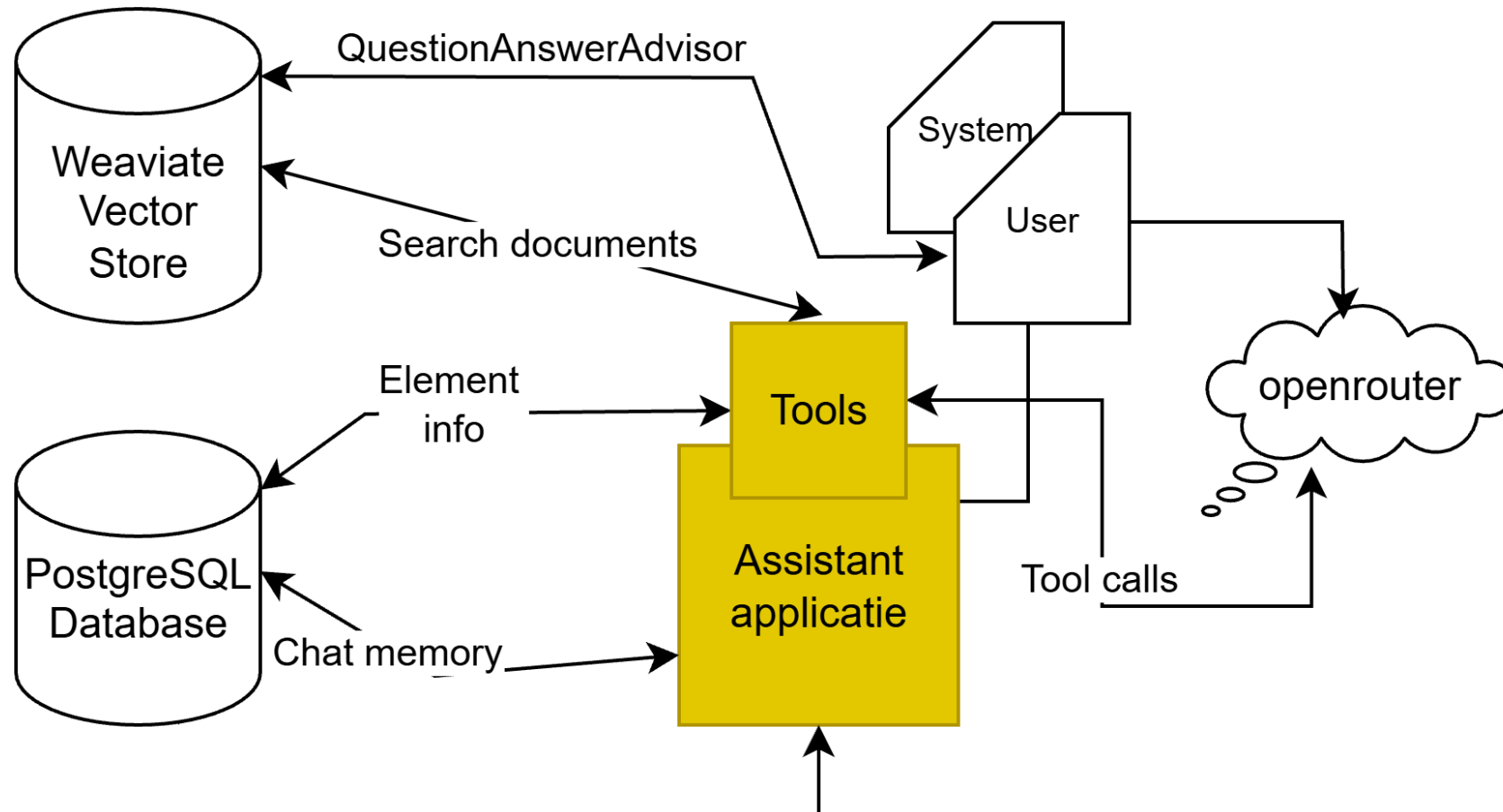
Tools

- Wat zijn dan Tools?
- Tools zijn API calls voor een AI model
- Spring-ai abstractielaag

```
@Tool(name = "get_all_elements", description = "Returns a list of all elements, " + no
    "which can be used in the XML.")
public String getAllElements() {
    return frankDocElementNameRepository.findAll().stream() Stream<FrankDocElementName>
        .map(FrankDocElementName::getName) Stream<String>
        .collect(Collectors.joining( delimiter: ", "));
}
```



webapp



webapp

- Spring-ai maakt opzet eenvoudig met advisors en ChatMemory

```
@Bean  👤 evandongen +1
ChatClient chatClient(ChatClient.Builder builder, VectorStore vectorStore, ChatMemory chatMemory) {
    return builder.defaultAdvisors(
        QuestionAnswerAdvisor.builder(vectorStore).build(),
        MessageChatMemoryAdvisor.builder(chatMemory).build()
    ).build();
}
```

- Model configuratie

```
@Bean  👤 Erik *
public EmbeddingModel embeddingModel(@Value("${OPENAI_API_KEY}") String apiKey) {
    return new OpenAiEmbeddingModel(OpenAiApi.builder()
        .apiKey(apiKey)
        .build());
}
```



Webapp: Chat starten of voortzetten

```
String response = chatClient.prompt() ChatClientRequestSpec
    .system(new ClassPathResource("systemprompt.st"))
    .user(new UserMessage)
    .options(getChatOptions(smart))
    .advisors( AdvisorSpec advisorSpec -> advisorSpec.param(ChatMemory.CONVERSATION_ID, conversationId))
    .call() CallResponseSpec
    .content();
```

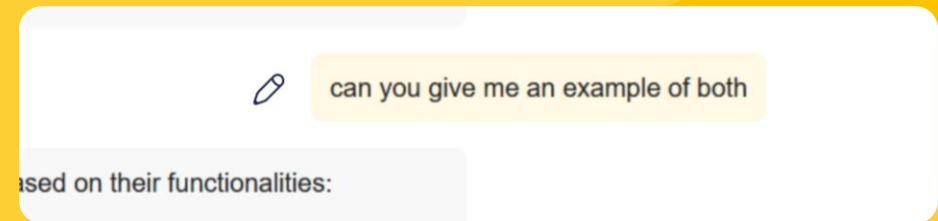
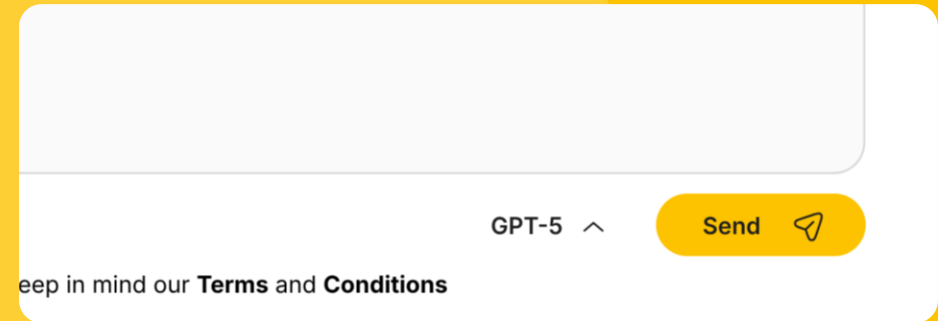
```
private ChatOptions getChatOptions() { 1 usage @Erik *
    ToolCallback[] frameworkTools = ToolCallbacks.from(tools);

    return ToolCallingChatOptions.builder()
        .toolCallbacks(frameworkTools)
        .model(this.smartModel)
        .build();
}
```



Toekomst: UX/UI

- Model keuze
- Eenvoudig opnieuw proberen van vorige vraag
- Copy code



Toekomst: Agent skills

- Vanuit Claude zijn 'agent skills' bekend geworden



Skills



Toekomst: Agent skills

```
my-skill/  
├── SKILL.md          # Required: instructions + metadata  
├── scripts/         # Optional: executable code  
├── references/      # Optional: documentation  
└── assets/          # Optional: templates, resources
```

Zie ook: <https://agentskills.io/specification>



Toekomst: Agent skills

- Specifieke skills toepassen waar nodig
- Uitbreidbaar voor de klant



Toekomst: bring your own model

- Configuratie uitbreidbaar
- Koppelen via openrouter mogelijk
- Koppelen van eigen OpenAI compatibel model



Bring your
own model





frank!
assistant



Bring your
own model

support@wearefrank.nl

erik@wearefrank.nl



Vragen?

