Digitaler Serviceassistent

ONLINE-HILFE

Guten Tag lieber Kunde,

gern unterstützen wir Sie zu Ihrem Telekom Anschluss.

Stellen Sie einfach Ihre Frage – möglichst in einem kurzen Satz. So kommen Sie am schnellsten zur richtigen Antwort.

Ich kann nicht telefonieren

Es besteht also eine Telefoniestörung.

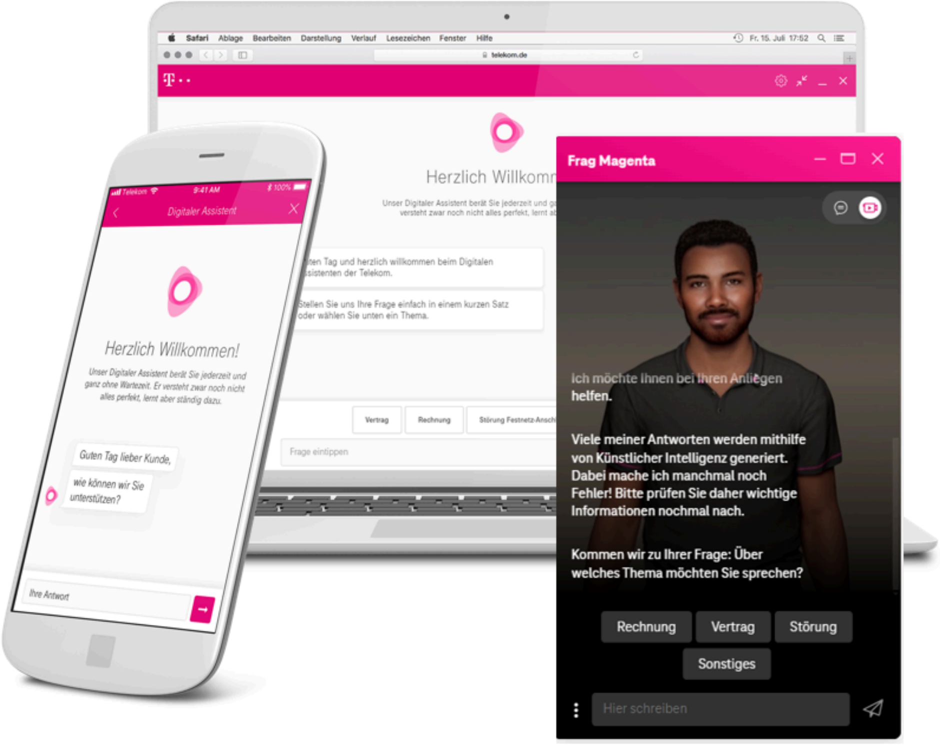
THE EVOLUTION OF LMOS

ROBERT WINKLER

LIFE IS FOR SHARING.

Frag Magenta

- Digital assistant of DT Customer Service
- Available for Web, OneApp, WhatsApp, Apple Business Chat and IVR in Germany



Herzlich Willkomm

Unser Digitaler Assistent berät Sie jederzeit und g
versteht zwar noch nicht alles perfekt, lernt ab

ten Tag und herzlich willkommen beim Digitalen
ssistenten der Telekom.

Stellen Sie uns Ihre Frage einfach in einem kurzen Satz
oder wählen Sie unten ein Thema.

Herzlich Willkommen!

Unser Digitaler Assistent berät Sie jederzeit und ganz ohne Wartezeit. Er versteht zwar noch nicht alles perfekt, lernt aber ständig dazu.

Guten Tag lieber Kunde,

wie können wir Sie unterstützen?

Ich möchte Ihnen bei Ihren Anliegen helfen.

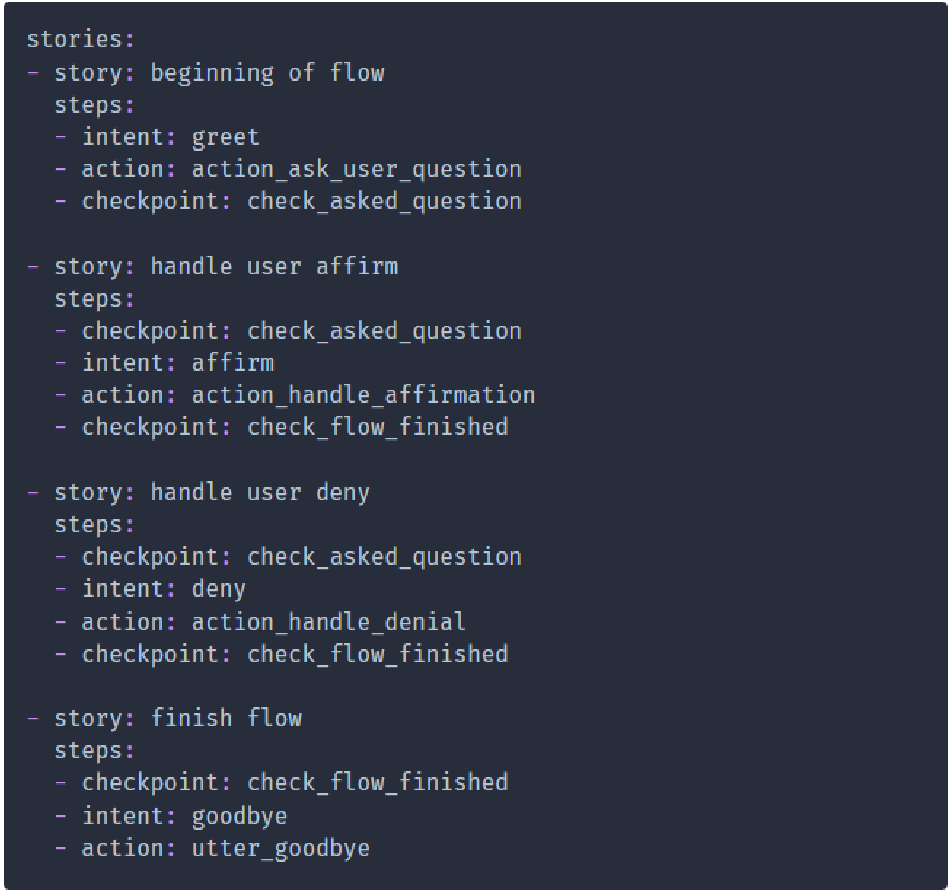
Viele meiner Antworten werden mithilfe von Künstlicher Intelligenz generiert. Dabei mache ich manchmal noch Fehler! Bitte prüfen Sie daher wichtige Informationen nochmal nach.

Kommen wir zu Ihrer Frage: Über welches Thema möchten Sie sprechen?

Facing limitations

RASA’s Design:

- **Scripted Dialogue Flow:** Predefined YAML-based scripts to steer dialogues.
- **Predictive Planning Required:** Customer interactions must be anticipated in advance.
- **Intent Classification Challenges:** Natural Language Understanding (NLU) struggles with accurately classifying user intents.
- **Manual Effort:** Continuous effort is required to update intents and train NLU.
- **Knowledge gap:** No FAQ knowledge base was available.
- **Scripting:** YAML is no scripting/programming language.



```
stories:
- story: beginning of flow
  steps:
  - intent: greet
  - action: action_ask_user_question
  - checkpoint: check_asked_question

- story: handle user affirm
  steps:
  - checkpoint: check_asked_question
  - intent: affirm
  - action: action_handle_affirmation
  - checkpoint: check_flow_finished

- story: handle user deny
  steps:
  - checkpoint: check_asked_question
  - intent: deny
  - action: action_handle_denial
  - checkpoint: check_flow_finished

- story: finish flow
  steps:
  - checkpoint: check_flow_finished
  - intent: goodbye
  - action: utter_goodbye
```

Overcoming limitations

Key Pain Points:

- **High Maintenance Costs:** Updating scripted dialogues is time-consuming and labor-intensive.
- **Customer Frustration:** Unresolved inquiries due to lack of knowledge led to negative customer experiences.
- **Call Center Costs:** Increased volume of customer queries at the call center, driving up costs.

Need for Change:

- **Improve NLU/NLP:** The existing NLU/NLP, static scripts and knowledge sources were not sufficient for many customer queries.
- **Objectives:** Increase solution rate, reduce call center volumes, lower maintenance costs, and enhance customer satisfaction through improved NLU/NLP, increased knowledge and less manual scripting efforts.

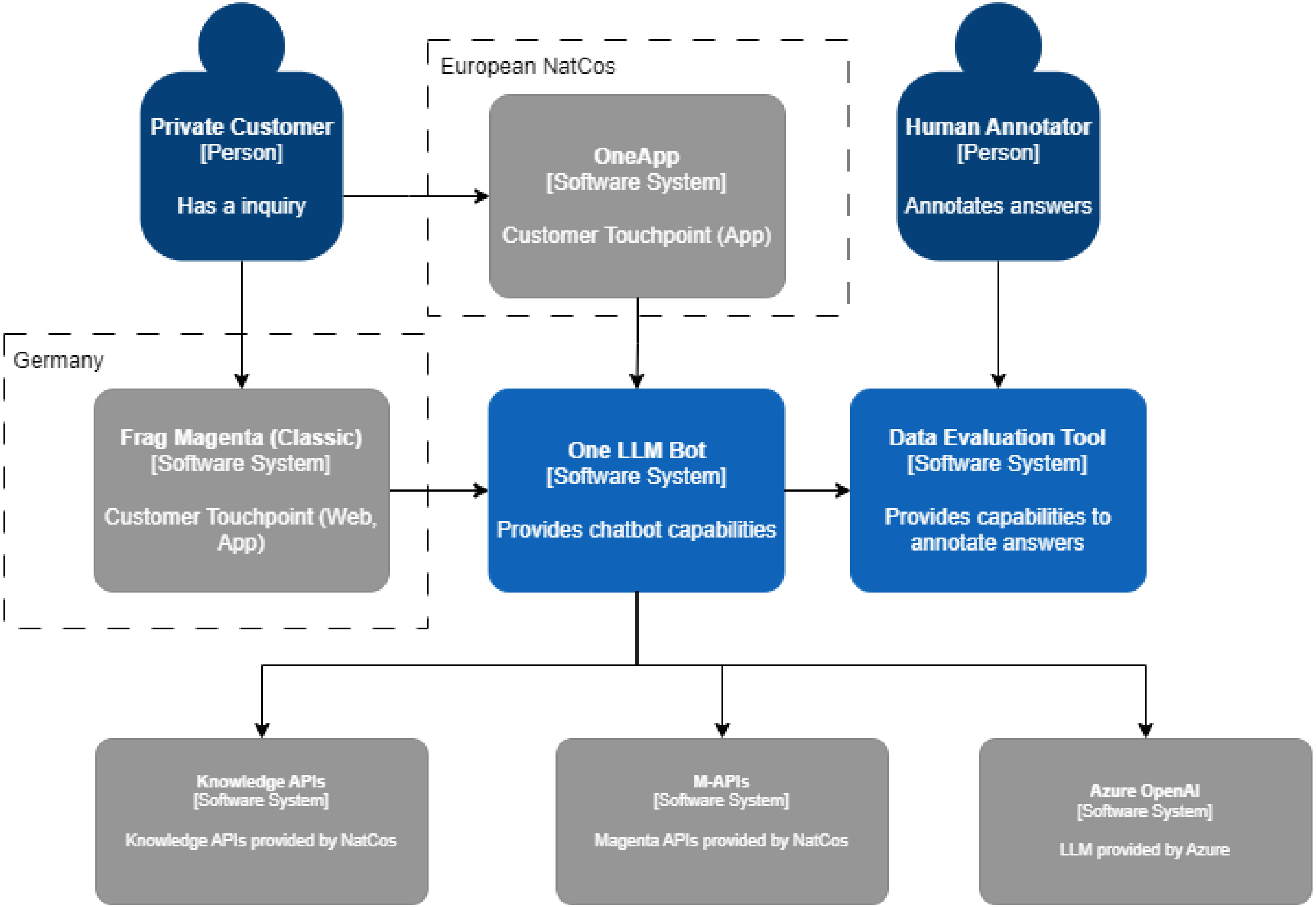
Taking a bold decision

Advancements in Generative AI:

- **Large Language Models (LLMs):** Models showed potential for better language understanding and processing.
- **Opportunity:** Use LLMs and Retrieval Augmented Generation (RAG) to improve NLU/NLP and reasoning/planning.

Decision to Innovate:

- **Multi-tenant and omni-channel system:** A strategic move to develop a single system for multiple NatCos and channels by making use of M-APIs.
- **Multi-agent system (MAS):** Multiple LLM-based agents, each focused on a specific business domain, working together to solve customer inquiries.
- **Highly configurable:** Every tenant can have a unique set of Agents, capabilities and knowledge sources.
- **Objective:** Improve speed to rollout the digital assistant to multiple NatCos. Starting with Germany, Austria and Croatia.


Private Customer
[Person]
Has a inquiry
European NatCos
OneApp
[Software System]
Customer Touchpoint (App)
Human Annotator
[Person]
Annotates answers
Germany
Frag Magenta (Classic)
[Software System]
Customer Touchpoint (Web, App)
One LLM Bot
[Software System]
Provides chatbot capabilities
Data Evaluation Tool
[Software System]
Provides capabilities to annotate answers
Knowledge APIs
[Software System]
Knowledge APIs provided by NatCos
M-APIs
[Software System]
Magenta APIs provided by NatCos
Azure OpenAI
[Software System]
LLM provided by Azure

Facing new territory and doubts

Industry Landscape:

- **Focus on Single-Agent Systems:** Most companies/frameworks were focused on single-agent RAG solutions.
- **Lack of Frameworks:** No established, production-ready multi-agent frameworks were available.

Team Concerns:

- **Distributed Team:** Team is distributed about multiple countries: Germany, India, Greece, …
- **Expertise Gap:** Dev team's background in Java/Kotlin rather than AI-specific technologies/languages.
- **Complexity:** Concerns about handling the complexity of a multi-tenant, multi-agent system and reaching the efficiency needed for Deutsche Telekom.

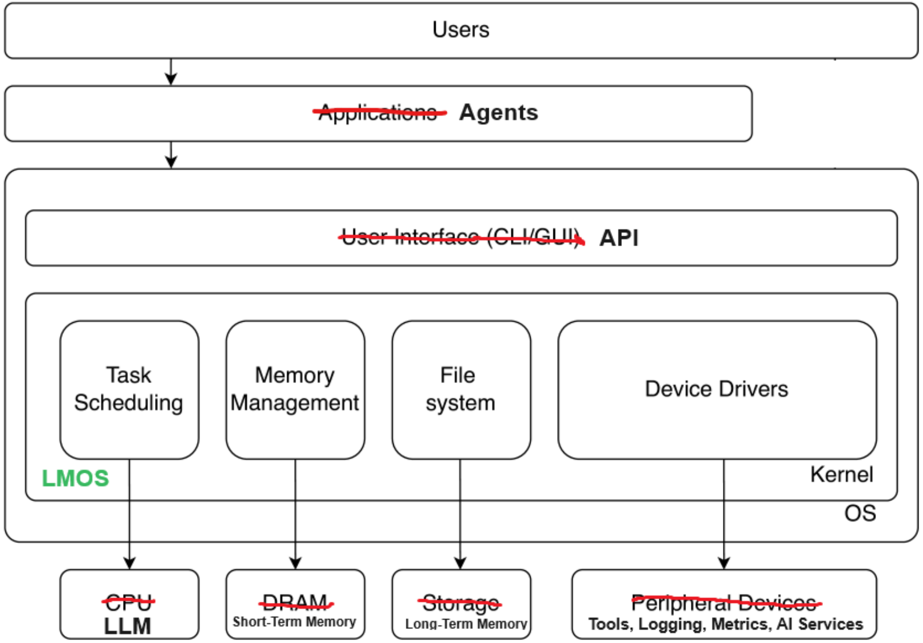
Internal Doubts in LT:

- **Building vs. Adapting:** Debate over whether to create a new system or adapt/buy existing solutions.
- **Feasibility Concerns:** Doubts about the team's ability to deliver a new scalable and efficient solution in time.

Finding Inspiration (1/3)

Operating System Design:

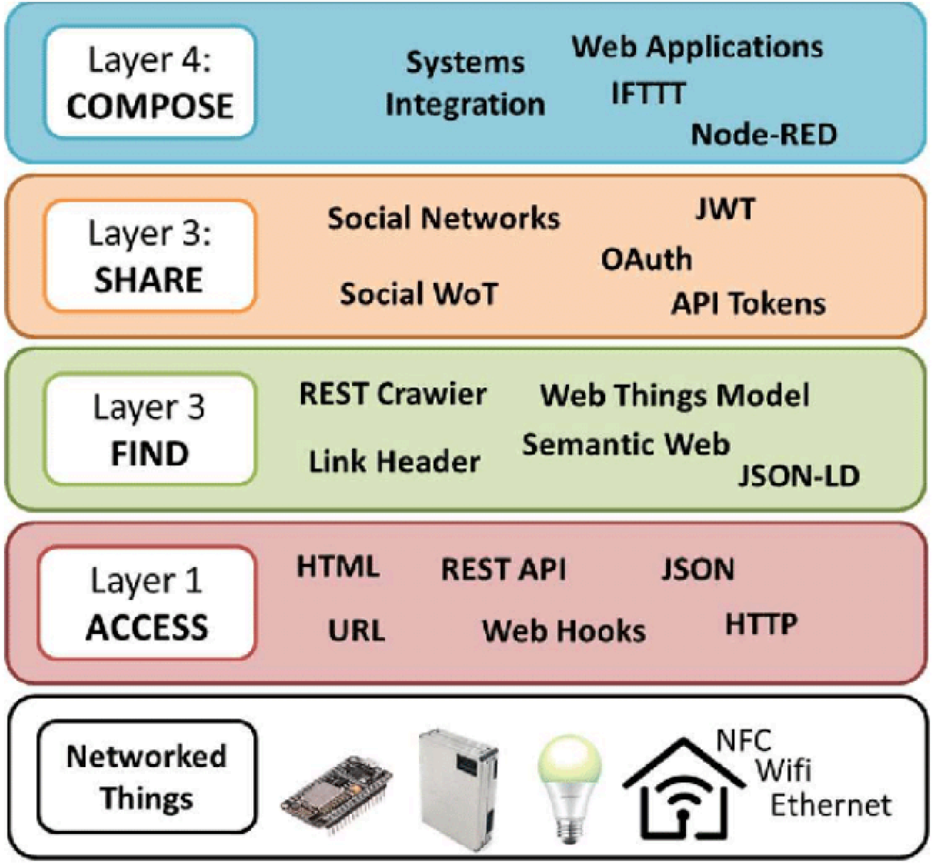
- **OS/Kernel Analogy:** LMOS acts as a Kernel/OS between Agents and Infrastructure providing an abstraction layer.
- **Application Developemt:** An OS provides system calls --> LMOS provides an API to simplify agent development.
- **Abstractions:** An OS abstracts hardware --> LMOS abstracts the complexities of working with LLMs, memory and tools.
- **Application Management:** An OS manages applications --> LMOS manages Agents and is doing execution planning.


Users
~~Applications~~ Agents
~~User Interface (CLI/GUI)~~ API
Task Scheduling
Memory Management
File system
Device Drivers
LMOS
Kernel
OS
~~CPU~~ LLM
~~DRAM~~ Short-Term Memory
~~Storage~~ Long-Term Memory
~~Peripheral Devices~~ Tools, Logging, Metrics, AI Services

Finding Inspiration (2/3)

Web of Things -> Web of Agents:

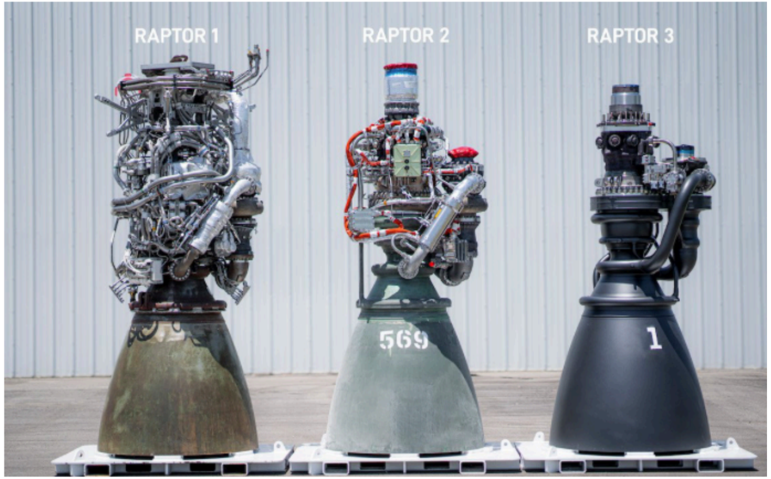
- **Standardized Communication (Access):** Built on top of open web standards and data models.
- **Discoverability and extendability (Find):** Dynamic discoverability of self-describing agents, allowing the multi-agent system to evolve.
- **Reusability (Share):** A single agent can be reused by multiple tenants concurrently.
- **Cross-Platform Compatibility (Compose):** Freedom to compose agents without being locked into a single platform.



Finding Inspiration (3/3)

SpaceX Engineering Model:

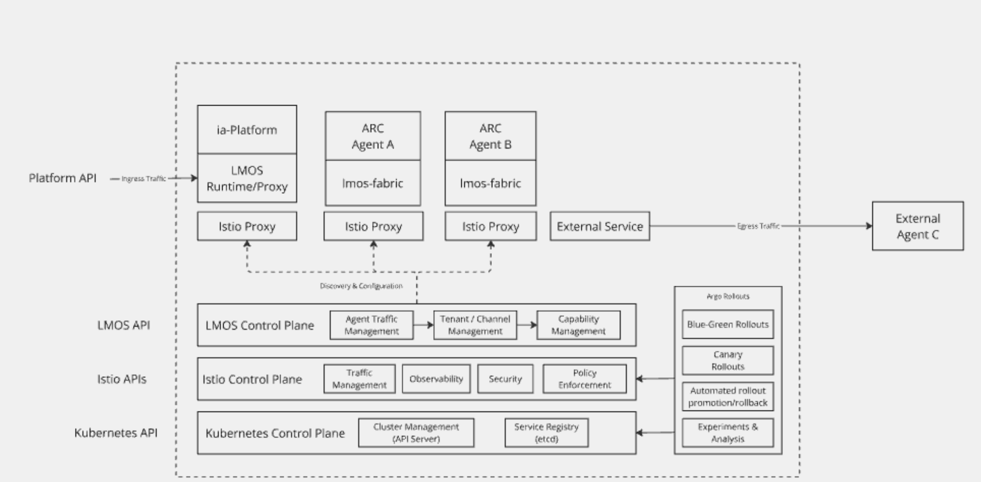
- **Frequent Deployments:** Frequent Agent releases with real-world feedback for quick improvement.
- **Parallelism**: Reusable launch system --> Develop and test agents simultaneously for faster innovation.
- **Reuseability**: Reusable rocket system --> LMOS provides reusable modules, e.g. Flows and Steps.
- **Reuseability**: Reusable rockets --> Reuse agents across multiple tenants.



The foundation

Technical Foundation:

- **Built on Kubernetes:** Leverage Kubernetes for orchestration and scalability.
- **Istio Service Mesh:** Utilize Istio as the service mesh to enhance traffic management, security, and observability across agents.
- **Extend Kubernetes:** Extend Kubernetes capabilities by developing a custom control plane to manage agents.
- **ArgoCD & GitOps:** Implemented ArgoCD with a GitOps approach to automate deployments and perform canary releases.



The concepts

Key Concepts:

- **Agent Registry**: Agents register their meta-data in a central registry.
- **Agent Runtime**: A runtime responsible for orchestrating the collaboration between multiple AI agents.
- **Agent Discovery**: Runtime can discover installed Agents and their capabilities.
- **Dynamic Routing:** Dynamically route queries to the most suitable agent. LMOS uses advanced methods like language models and vector embeddings to intelligently match queries to the right agent.
- **Knowledge Sharing:** The Runtime ensures that agents share context, memory, and knowledge as needed to handle customer queries holistically.
- **Memory Management:** LMOS includes built-in memory management for agents, allowing them to store and retrieve data during interactions.
- **Operator:** Listens for new or modified Channels and Agents and dynamically resolves capabilities.

Overcoming hurdles (1/2)

Development challenges:

- **Technical challenges:** Encountering various technical obstacles such as model performance and hallucinations.
- **Channel-Specific Requirements:** Addressing unique technical needs for different channels, such as voice, web or app.
- **High Testing Effort:** Demanding and resource-intensive testing processes, requiring extensive manual annotations and validations to guarantee model reliability.
- **Multilingual Data Anonymization:** Developing robust Named Entity Recognition (NER) models capable of handling data anonymization across diverse languages and linguistic structures.
- **Multilingual Language Support:** Ensuring language and embedding models possess the capability to process, comprehend, and maintain accuracy across multiple languages.

Overcoming hurdles (2/2)

International Collaboration:

- **Team Efforts:** Collaboration between teams in Germany and India.
- **Coordinated Work:** Overcame time time zone and way of working differences to ensure effective teamwork.

Pressure to Prove Concept:

- **Stakeholder Engagement:** Continuously communicate and showcase the platform’s value and potential to gain the trust and support of stakeholders.

The critical test

Deployment Requirements:

- **Scalability:** Ability to handle large-scale deployment across multiple use cases and NatCos.
- **Security and Efficiency:** Ensuring platform security and operational efficiency.

Real-World Testing:

- **Performance Evaluation:** Assessment of the platform’s performance and accuracy in live settings.
- **Competitiveness:** Comparison with leading industry products and solutions, such as Rasa CALM or Sprinklr.

The outcome

Live Implementation:

- **Handling Interactions:** Efficiently handled thousands of customer interactions accross multiple channels and diverse use cases.

Performance Metrics:

- **Solution Rate:** Achieved an 85% solution rate.
- **Hallucinations:** Less than 5% incorrect or irrelevant responses, demonstrating strong model reliability.
- **Development Efficiency:** Developed 14 use cases within a month, averaging 2.5 days per use case, highlighting rapid iteration and implementation.

Impact:

- **Higher solution rate:** Enhanced solution rate by leveraging additional context information, such as knowledge sources and M-APIs.
- **Accelerated Processes:** Accelerated development and deployment timelines, leading to quicker delivery of new use cases.

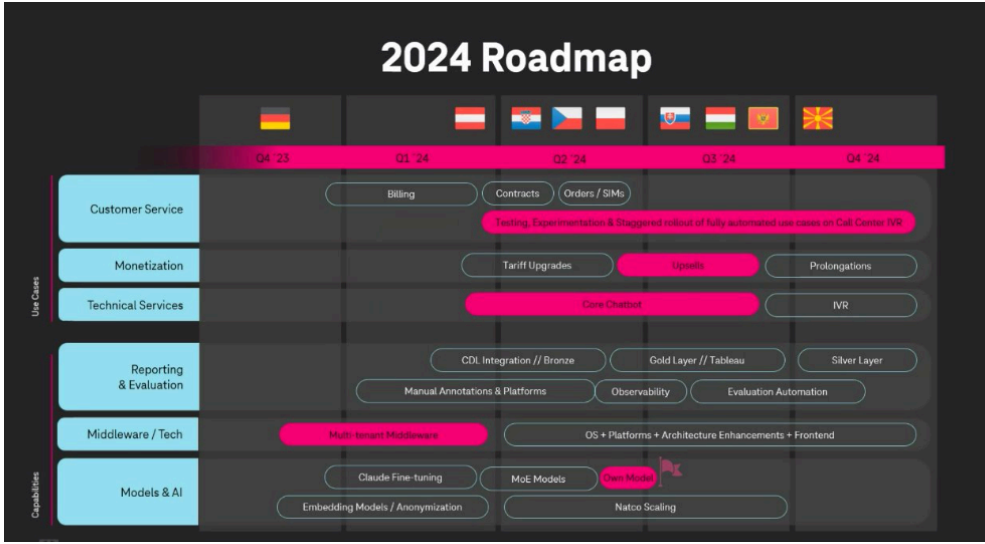
Moving forward

Business Value:

- **Return on Investment:** Delivered measurable business value - reduced call volumes and improved customer experience.

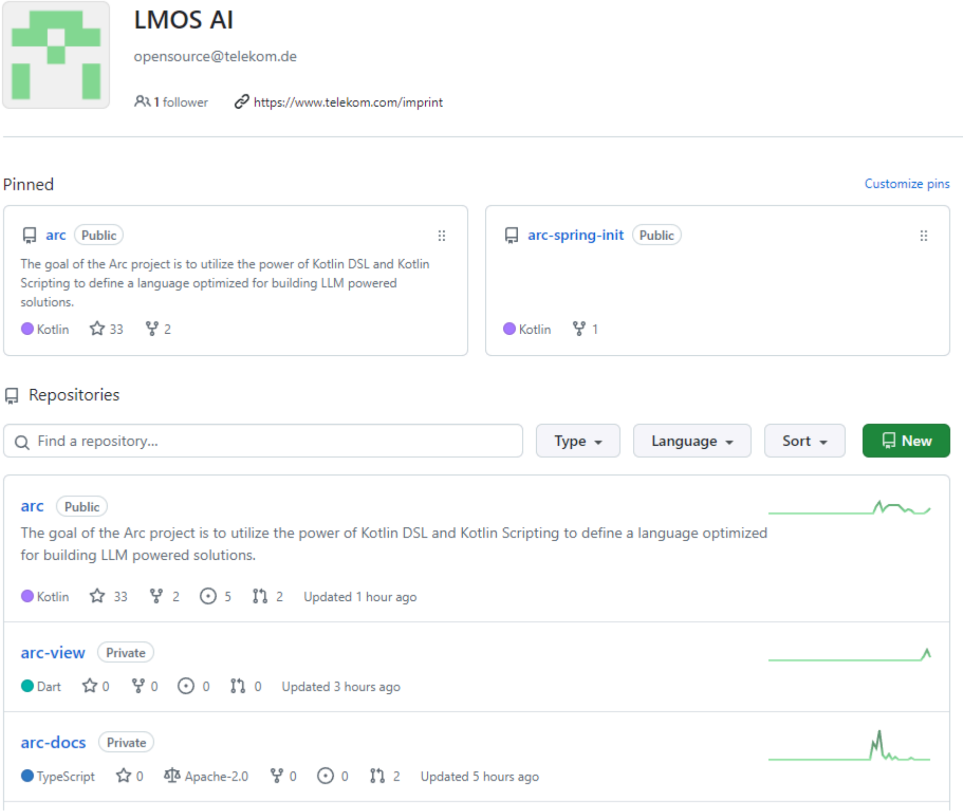
Impact on Deutsche Telekom:

- **Strategic Foundation:** LMOS became the cornerstone for Deutsche Telekom’s chatbot strategy in Europe.
- **NatCo Expansion**: Successfully rolled out in Austria, with Croatia as the next country on the roadmap.
- **International Teamwork:** A success of an international team pushing beyond traditional



Going Open Source

- **Community Contribution:** Release of LMOS to the open-source community on GitHub.
- **Sharing Innovations:** Providing a valuable solution for building an enterprise-ready, multi-tenant, multi-channel and multi-agent system.



LMOS AI

opensource@telekom.de

1 follower https://www.telekom.com/imprint

Pinned

arc Public

The goal of the Arc project is to utilize the power of Kotlin DSL and Kotlin Scripting to define a language optimized for building LLM powered solutions.

Kotlin 33 2

arc-spring-init Public

Kotlin 1

Repositories

arc Public

The goal of the Arc project is to utilize the power of Kotlin DSL and Kotlin Scripting to define a language optimized for building LLM powered solutions.

Kotlin 33 2 5 2 Updated 1 hour ago

arc-view Private

Dart 0 0 0 0 Updated 3 hours ago

arc-docs Private

TypeScript 0 Apache-2.0 0 0 2 Updated 5 hours ago

Questions?