Aalto University
School of Science

# Robustness for Building Trustworthy Hybrid Intelligence Software (HIS) with LLM in Multi-Continuum Computing: Design and Analytics

Korawit Rupanya
korawit.rupanya@aalto.fi

01.10.2025

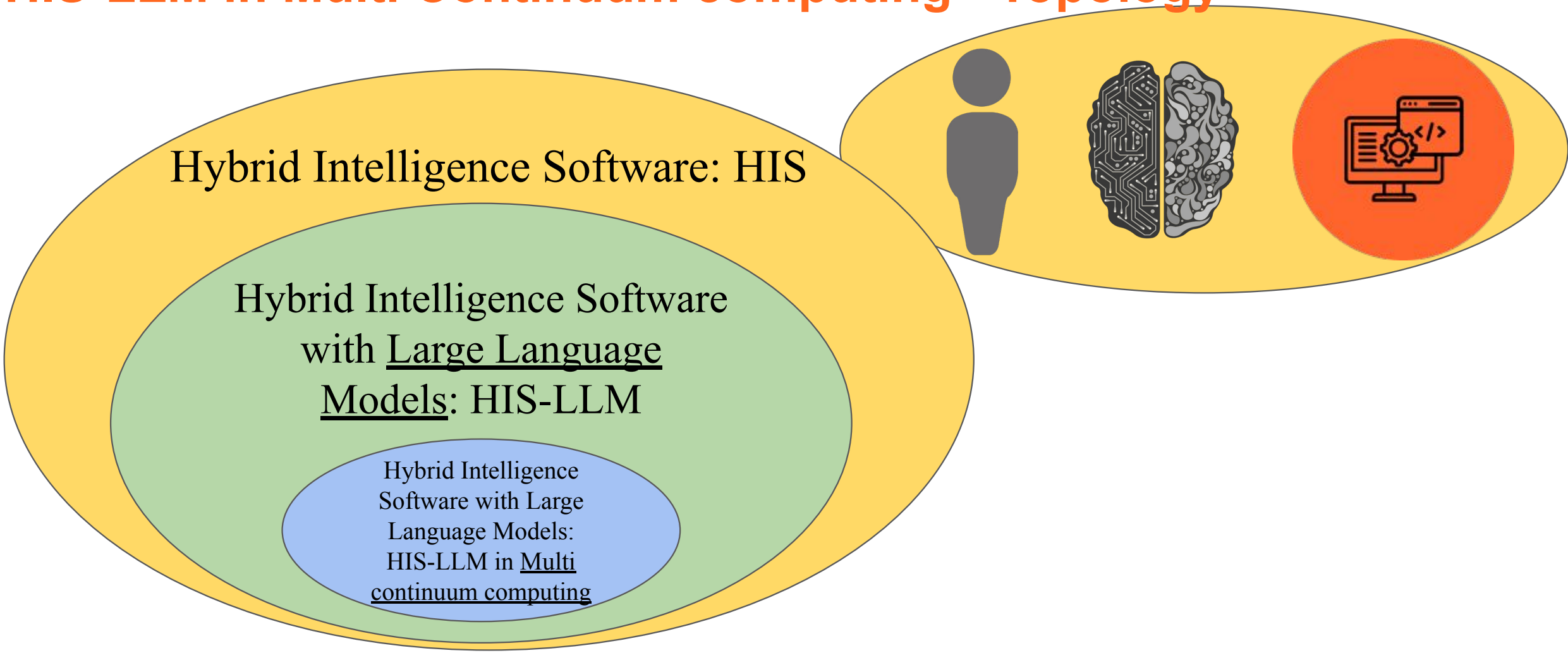
# Learning Objectives

- Be able to understand **Robustness** in HIS-LLM especially in Multi continuum computing
- Be able to understand elements of **Trustworthiness** for HIS-LLM
- Be able to construct **Observability** for analyzing **Robustness** as part of **Trustworthiness**.
- Be able to evaluate **Robustness** metrics for understanding HIS-LLM

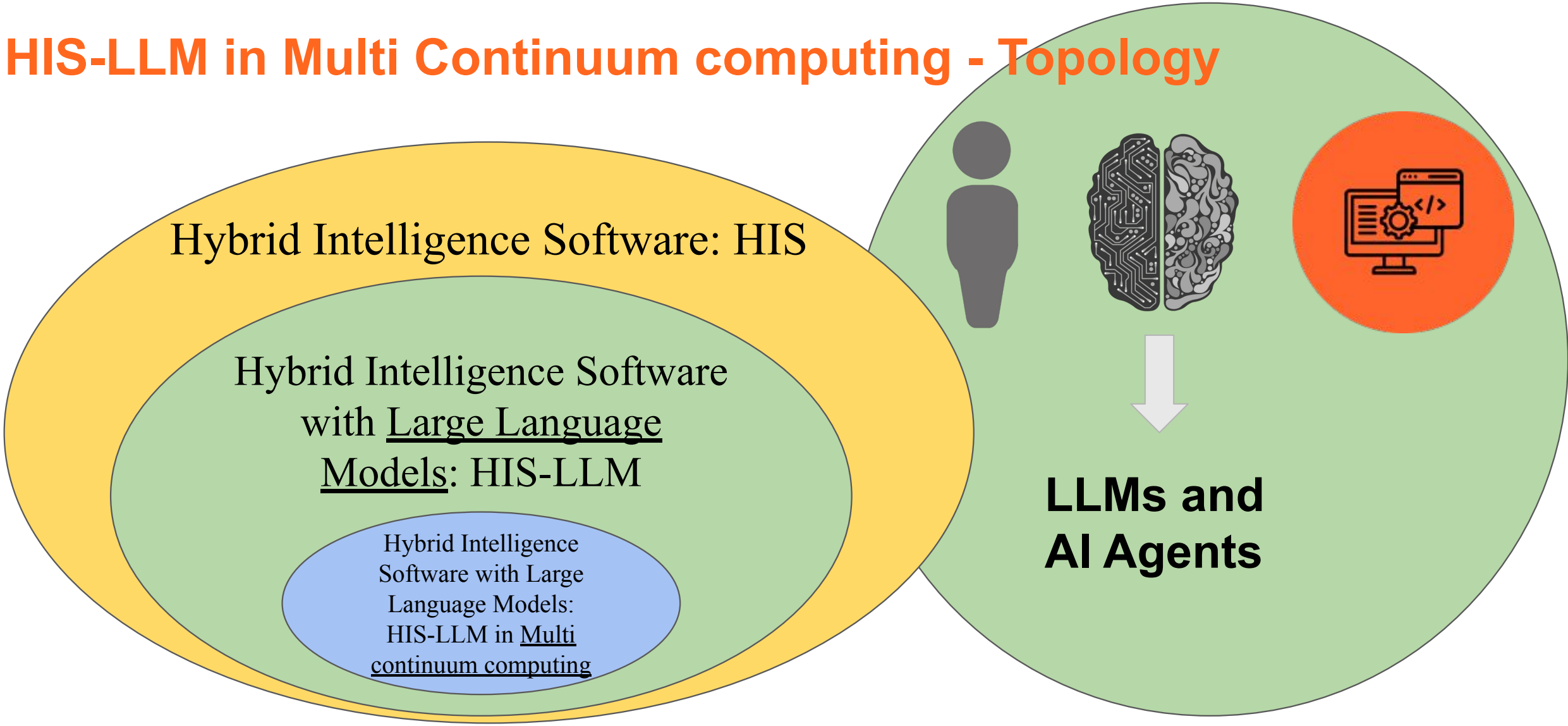
# Contents

- Hybrid Intelligence Software System(HIS)-LLM in Multi continuum
- Problems in building trust with HIS-LLM
- Components that support in building trustworthiness for HIS-LLM
- How to satisfy/employed these components in HIS-LLM?
- Simple hands-on tutorial

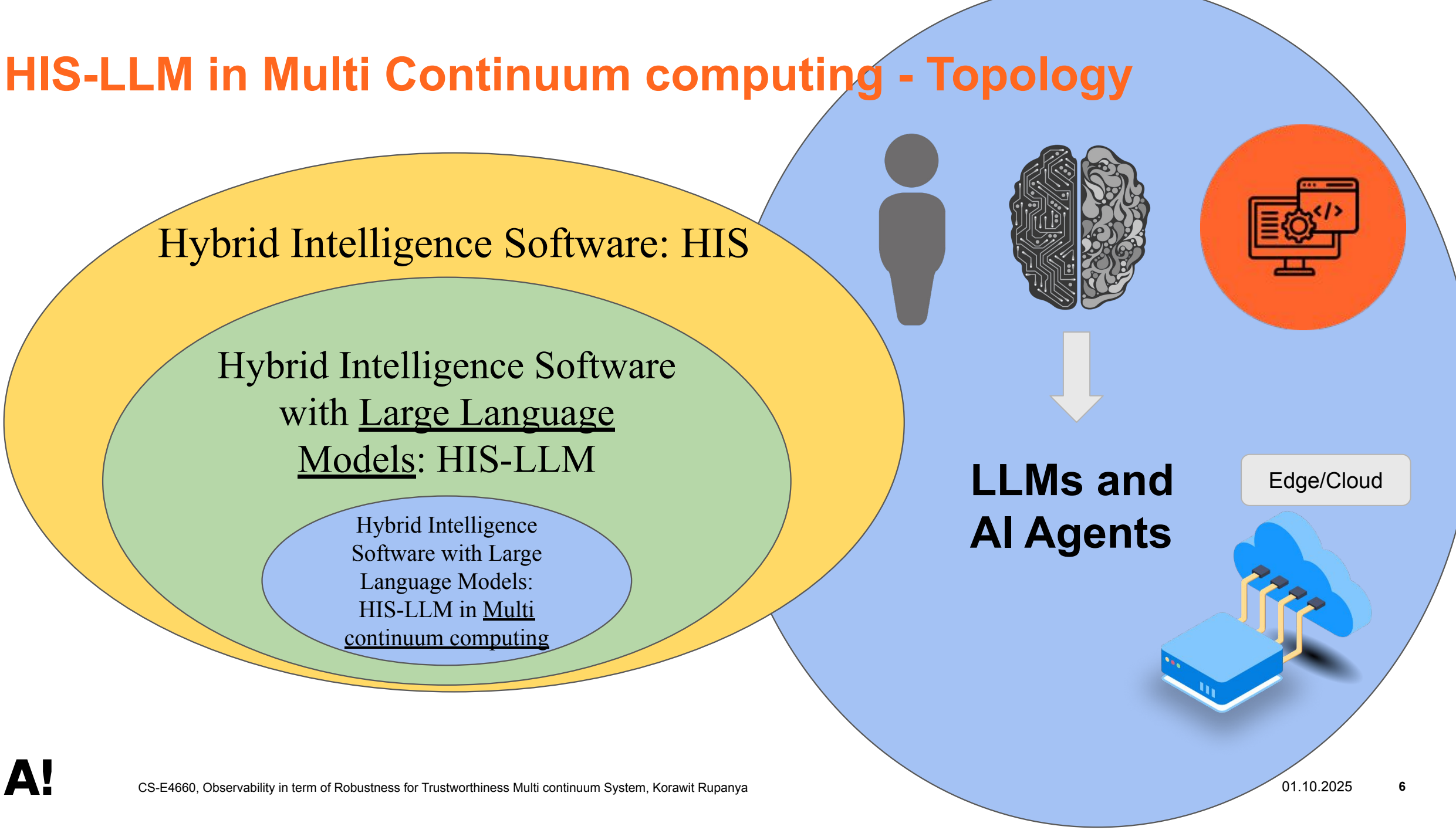
# HIS-LLM in Multi Continuum computing - Topology



# HIS-LLM in Multi Continuum computing - Topology



# HIS-LLM in Multi Continuum computing - Topology



# Application area(Domain) of HIS-LLM

- Search and Rescue (SAR)
- Security Orchestration
- Data Analytics & Recommendation Systems
- Predictive Maintenance in Industry
- Healthcare Decision Support
- Smart Mobility & Transportation

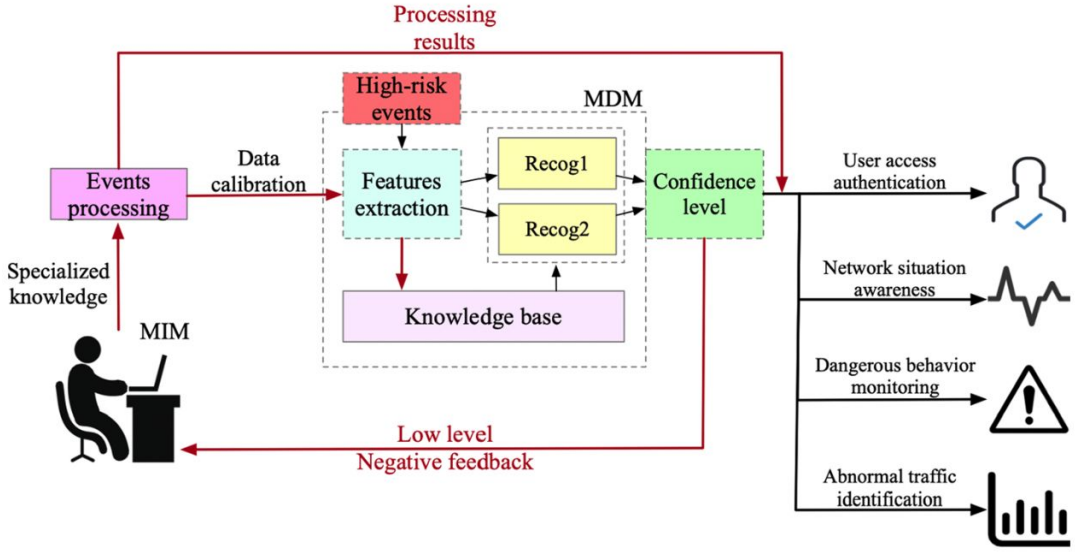



**Figure source:** "Artificial intelligence in cyber security: research advances, challenges, and opportunities", Home Artificial Intelligence Review 2022, 10.1007/s10462-021-09976-0.

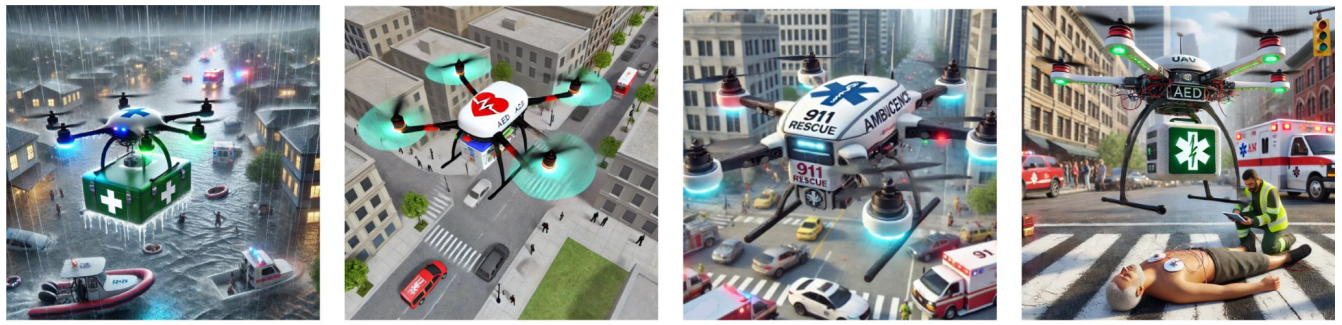



**Figure source:** "Unmanned aerial systems in search and rescue: A global perspective on current challenges and future applications", *International Journal of Disaster Risk Reduction 2025, 10.1016/j.ijdrr.2025.105199.*

# DESIGN - BASIC ELEMENTS AND INTERACTIONS IN HIS-LLM

# Scenario of HIS-LLM

## Paragliding planner system

A software application that helps a user plan a paragliding trip.

- What tasks should the system support (weather checking, location suggestion, safety alerts)?
  - Do we need AI capabilities?
- How reliable and accurate must the predictions be?
  - Do we need Human also?
- How should the software architecture style look (edge-cloud, mobile app, software service)?
- Can we do it in multi continuum? Is it possible to be multi-continuum?

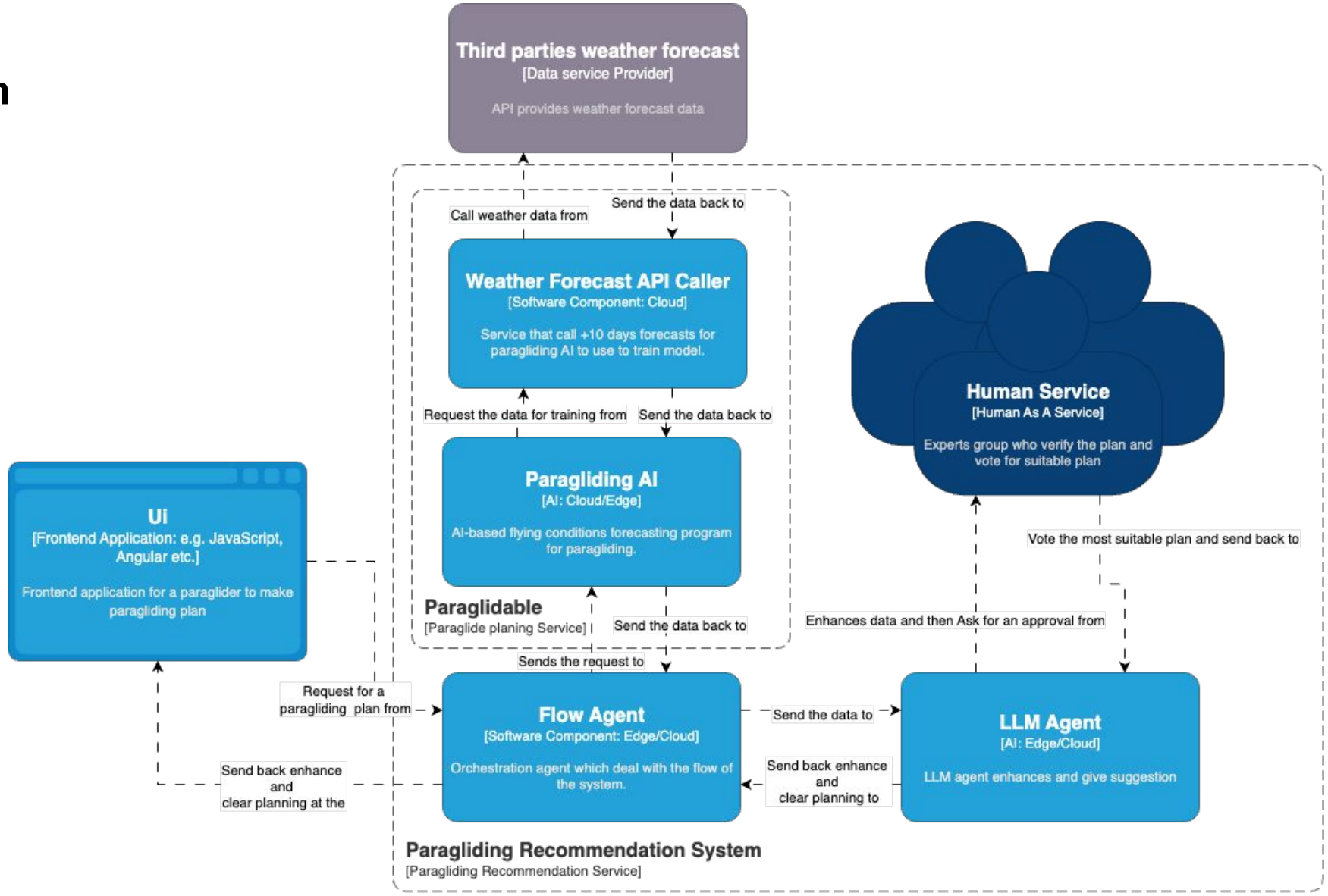
**Figure source:** Quang Nguyen Vinh - https://www.pexels.com/photo/person-riding-on-parachute-2162689/

**How do we ensure trustworthiness of recommendations as the output of paragliding HIS-LLM?**

# HIS-LLM version

**Paragliding planner system**

- Is this system reasonable? – fulfill all required functionality
- What are pros/cons?
- How to do we ensure the trustworthiness of HIS?




**Project source:** https://github.com/rdsea/sys4bigml/tree/master/tutorials/r4hisllm

# Basic elements of HIS-LLM

- **Human as a Service**: Human users or experts who provide input, feedback, decision-making, or oversight as integral components of the HIS-LLM.
- **LLM/Agent as a Service**: Large Language Model–based agents that perform reasoning, generation, summarization, and coordination tasks, acting as autonomous or semi-autonomous services.
- **Data/Computing as a Service**: Traditional software modules (e.g., APIs, databases, monitoring tools) that provide computational, analytical, or infrastructural functionalities.

**The key distinction between each element arises from its inherent characteristics, which may relate to functionality, reliability, or the degree of uncertainty it entails.**

# Basic interactions of HIS-LLM

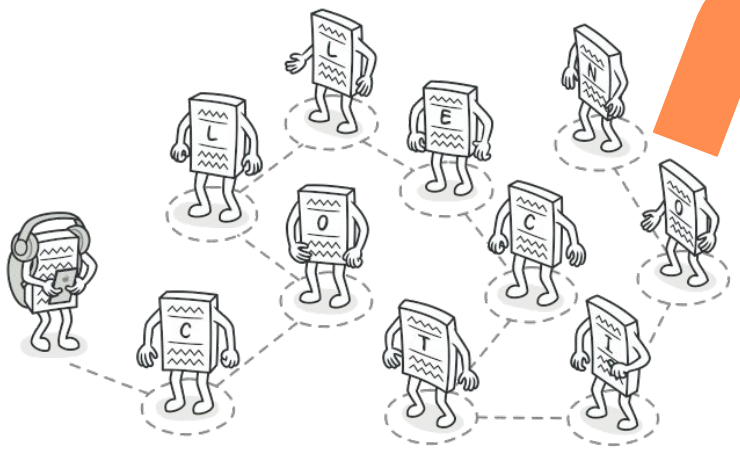
We classify interactions into three kinds by taking a holistic view of the interactions among components in HIS-LLM from a service-oriented perspective.

1. **Software Service to Software Service(S2S)** – Interactions between software, LLM-agents, and other computational services.
   a. **AI-to-AI / LLM-to-LLM interaction:** coordination between LLM-based agents or AI models (e.g., planning agent requesting domain-specific predictions).
   b. **AI-to-service interaction:** Calls from an LLM-agent to supporting software modules (e.g., retrieval services, knowledge graphs, or simulation engines).
   c. **Service orchestration and choreography:** Workflow-driven communication between services (e.g., API calls, event-driven triggers in microservices).
2. **Human service to Software Service(H2S)** – Interactions between humans and services (e.g., prompting, monitoring, or feedback).
3. **Human service to Human service(H2H)** – Interactions between humans, mediated or facilitated by services.

# Basic patterns of HIS-LLM

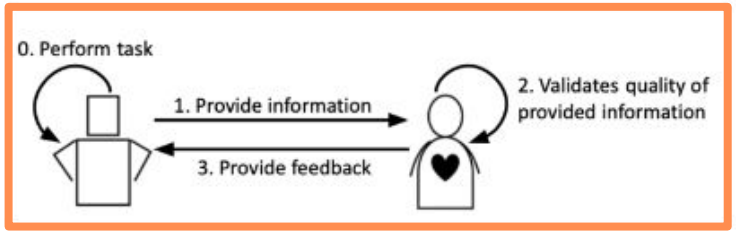
- **Patterns:** There are several patterns that represent HIS-LLM. Most of these patterns are derived from agent representation patterns. Below are example patterns that characterize HIS-LLM
  - **Hierarchical/Composite** – Layered structure: higher-level agents or humans delegate tasks to lower-level agents/services, typically resembling a pyramid with levels of increasing or decreasing authority, importance, or complexity
  - **Concurrent/Parallelization** – Multiple agents/services divide a task in to into tree structures and then operate in parallel on their independent subtasks .
  - **Sequential/Iterator** – Tasks flow step-by-step; output of one agent/service becomes input for the next.
  - **Rearrange/Command** – Roles or responsibilities adapt dynamically based on context or task needs.
  - **Router/Mediator** – A meta-component routes tasks to the most appropriate agent/service based on capability, load, or context.

# Example - apply to Paragliding planning HIS-LLM



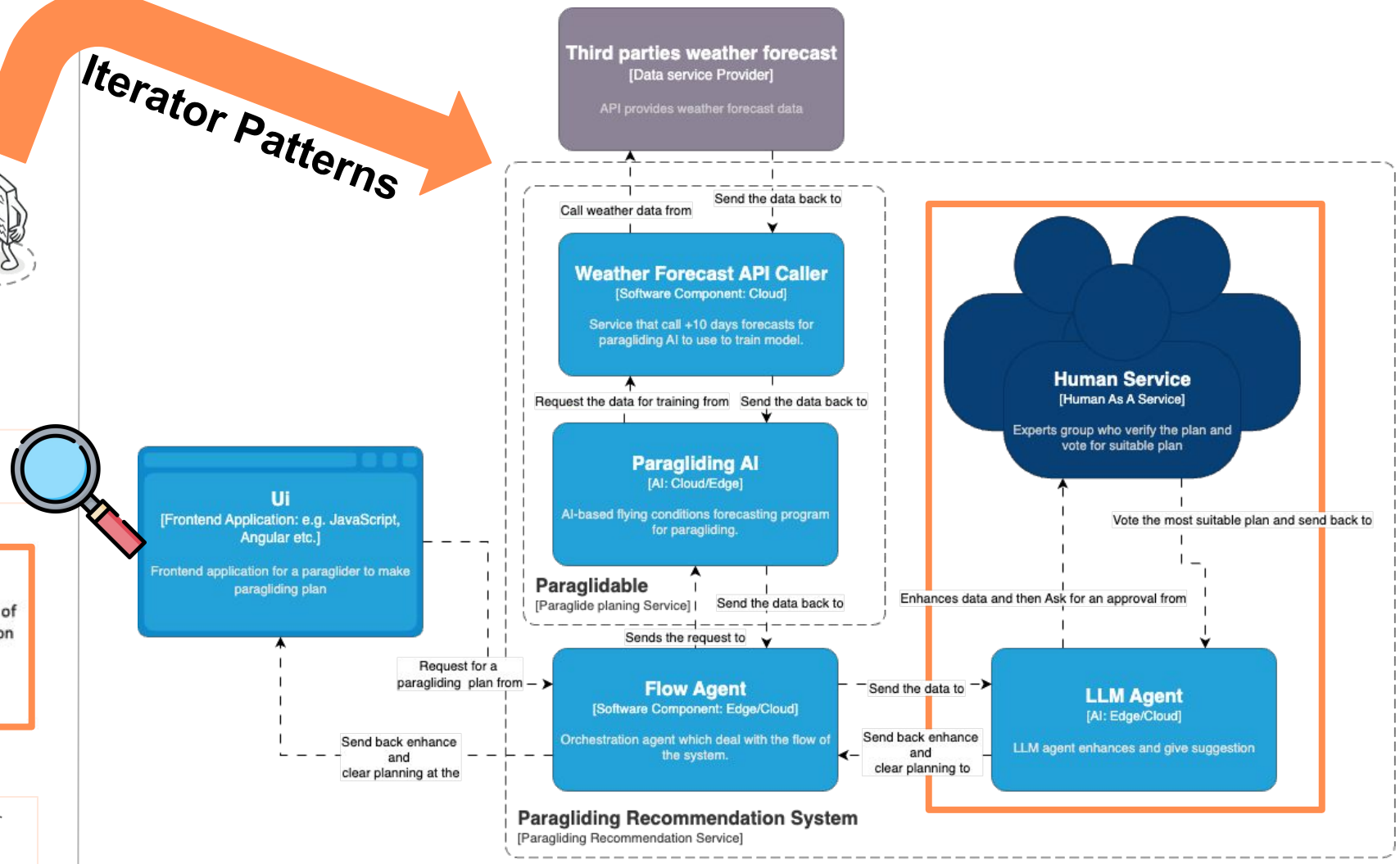
Iterator Patterns

**Figure source:**
https://refactoring.guru/design-patterns/iterator




**AI Performer and Human Validator**

**Figure source:** "Developing Team Design Patterns for Hybrid Intelligence Systems", *HHAI 2023,* 10.3233/FAIA230071.




**Project source**: https://github.com/rdsea/sys4bigml/tree/master/tutorials/r4hisllm

# To build / understand HIS-LLM: what is the main concern?

**System Behavior**

- How do hybrid interactions (human–AI–service) produce unexpected outcomes?
- Are these behaviors predictable, reliable, and explainable?

**Human Role & Cognitive Fit**

- What is the optimal role of humans in the loop (controller, collaborator, overseer)?
- How does HIS-LLM support human cognition instead of overwhelming it?

**Evaluation Frameworks**

- What metrics capture HIS-LLM performance beyond accuracy (e.g., robustness, resilience, adaptability)?
- How to evaluate hybrid workflows, not just AI components?

**Patterns of Interaction**

- Which interaction structures (hierarchical, concurrent, etc.) are most effective under different conditions?
- How do these patterns affect trust and efficiency?

# ROBUSTNESS FOR HIS-LLM

# Trustworthiness - Robustness in AI/ML system

| Dimension | Definition | Section |
|---|---|---|
| Truthfulness | The accurate representation of information, facts, and results by an AI system. | §6 |
| Safety | The outputs from LLMs should only engage users in a safe and healthy conversation [72]. | §7 |
| Fairness | The quality or state of being fair, especially fair or impartial treatment [208]. | §8 |
| Robustness | The ability of a system to maintain its performance level under various circumstances [83]. | §9 |
| Privacy | The norms and practices that help to safeguard human and data autonomy, identity, and dignity [83]. | §10 |
| Machine ethics | Ensuring moral behaviors of man-made machines that use artificial intelligence, otherwise known as artificial intelligent agents [85, 86]. | §11 |
| Transparency | The extent to which information about an AI system and its outputs is available to individuals interacting with such a system [83]. | §12 |
| Accountability | An obligation to inform and justify one's conduct to an authority [209, 210, 211, 212, 213]. | §13 |

**Figure source:** "The definitions of the eight identified dimensions.", *arxiv 2024,* doi: 10.48550/arXiv.2401.05561.

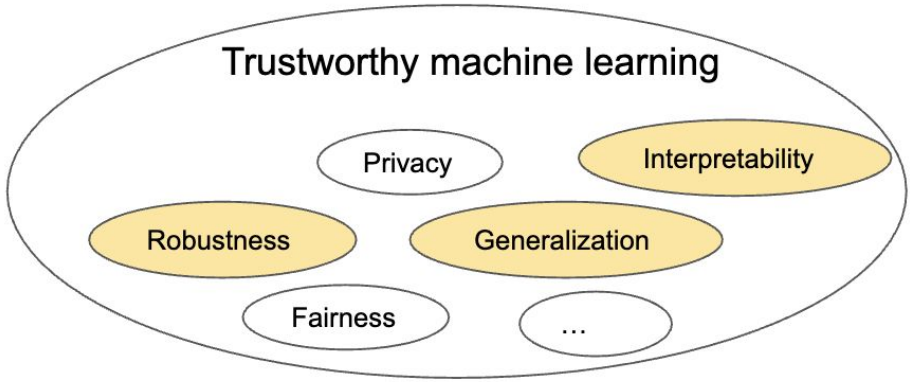



**Figure source:** "Trustworthy Machine Learning: Robustness, Generalization, and Interpretability", *KDD 2023,* doi: 10.1145/3580305.3599574.

**The most crucial dimensions in achieving trustworthy AI:**

1. Safety & **Robustness**
2. Nondiscrimination & Fairness
3. Explainability,
4. Privacy
5. Accountability & Auditability,
6. Environmental Well-being.

**Quote source:** "Trustworthy AI: A Computational Perspective", *ACM Transactions on Intelligent Systems and Technology 2022,* doi: 10.1145/354687.

# Robustness in HIS-LLM

Robustness can be classified into two categories:

- **Natural Robustness** - a system's ability to perform reliably when encountering variations similar to real-world, everyday situations, such as paraphrased inputs or minor errors, rather than being tested with synthetic or artificial disruptions
- **Adversarial robustness** – resisting malicious manipulations.

**Definition source:** A.I. Robustness, ACM Comput. Surv 2025. Doi: https://doi.org/10.1145/3665926

**Robustness in HIS-LLM:**

Ensure reliable performance, fairness, and trust across human–AI–service interactions. Key difference from AI/ML is that AI/ML robustness is model-centric. In HIS-LLM, robustness is system- and interaction-centric (spanning humans, AI components, and infrastructure).

# Examples of key HIS-LLM characteristics and their associated risks - paragliding recommendation system

1. **Data-related issues: (Concerns about the correctness, timeliness, and consistency of information used in decision-making.)**
   - **Data accuracy deviation** – Faulty wind sensors or noise in turbulence data produce misleading situational awareness.
   - **Data timeliness issue** – Delays in transmitting weather updates cause forecasts to diverge from real-time conditions.
2. **Service-quality issues: (Concerns about availability, reliability, and correctness of computational services/algorithms.)**
   - **Service reliability degradation** – Intermittent unavailability of turbulence prediction or route-optimization service disrupts continuity.
   - **Function execution error** – Misclassification of turbulence intensity by anomaly detection or LLM reduces reliability of warnings.
   - **Adaptation failure** – Inability to cope with changes in connectivity or resource availability prevents timely updates.

# Examples of key HIS-LLM characteristics and their associated risks - paragliding recommendation system

**3. Interaction-related issues: (Concerns about human–AI/system communication, usability, and trust.)**

- **Interaction breakdown –** Unclear or misinterpreted recommendations degrade situational awareness.
- **Explainability gap –** Lack of transparent reasoning behind suggestions undermines trust and acceptance.

# Some important Robustness Metrics

- **Performance Drop Rate (PDR)**: “PDR quantifies the relative performance decline following a prompt attack, offering a contextually normalized measure to compare different attacks, datasets, and models.”

**Metric source:** "PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts", arxiv 2024, arXiv:2306.04528 .

- **Cohen’s h Effect Size:** In statistics, Cohen's h, popularized by Jacob Cohen, is a measure of distance between two proportions or probabilities. Cohen's h has several related uses:
  - It can be used to describe the difference between two proportions as "small", "medium", or "large".
  - It can be used to determine if the difference between two proportions is "meaningful".
  - It can be used in calculating the sample size for a future study

**Definition source:** https://en.wikipedia.org/wiki/Cohen%27s_h

- **Consistency or stability Rate:** Rating of the prediction and predicted probabilities under natural perturbation of the input.

**Metric source:** "SCORE: Systematic COnsistency and Robustness Evaluation for Large Language Models", arxiv 2025, arXiv:2503.00137v

# Observability Tools for measuring and monitoring HIS-LLM

- **General Observability tools:**
  - **Prometheus**: Instrument, collect, store, and query your metrics for alerting, dashboarding, and other use cases.
  - **Logstash**: open-source data ingestion tool that allows you to collect data from various sources, transform it, and send it to your desired destination.
- **LLM Observability tools:**
  - **Langfuse:** Open Source LLM Engineering Platform; Traces, evals, prompt management and metrics to debug and improve LLM application.
  - **Langsmith:** unified observability & evals platform where teams can debug, test, and monitor AI app performance — whether building with LangChain or not.
  - **DeepEval**: Enables evaluation with 14+ metrics, including summarization and hallucination tests, via Pytest integration.
  - **Opik by Comet**: Tracks, tests, and monitors LLMs with feedback and scoring tools for debugging and optimization.
  - **Arize Phoenix**: Facilitates AI observability, experimentation, and debugging with integrations and runtime monitoring.

# How to use the Robustness metrics?

1. **Quantify Stability**; Measure how LLM outputs and service workflows handle noisy sensor data, ambiguous prompts, and fluctuating edge–cloud resources.
2. **Expose Weak Points**; Identify brittle interactions (e.g., misalignment between human input and LLM reasoning, or failure propagation between services).
3. **Guide Design**; Provide feedback for resilient hybrid workflows, such as robust prompt strategies, fallback mechanisms, or adversarial stress tests on edge–AI pipelines..
4. **Ensure Trust & Reliability**; Reduce risks of misleading recommendations, unsafe automation, or inconsistent human–AI decisions.
5. **Support Compliance**; Demonstrate explainability, fairness, and robustness metrics for HIS-specific standards (e.g., in safety-critical or regulated domains).
6. **Enable Monitoring**; Continuously track robustness across human inputs, LLM components, and service-to-service interactions throughout the system lifecycle.

# Conclusion

- Robustness is one of the main component to building trustworthy HIS-LLM in multi-continuum systems.
- Trustworthiness emerges from Robustness, Resilience, Reliability, Elasticity, Explainability, Fairness etc.
- Metrics such as Performance Drop Rate, Cohen's h, and Consistency Rate enable systematic evaluation of Robustness.
- Observability tools (OpenTelemetry, Jaeger, DeepEval, etc.) provide continuous monitoring and feedback.
- Robustness analytics help identify weak points, guide design improvements, and ensure trustworthiness of the system.
- Ultimately, robust HIS-LLM enhances user trust, safety, and compliance across domains (SAR, healthcare, mobility, industry).

# Challenges

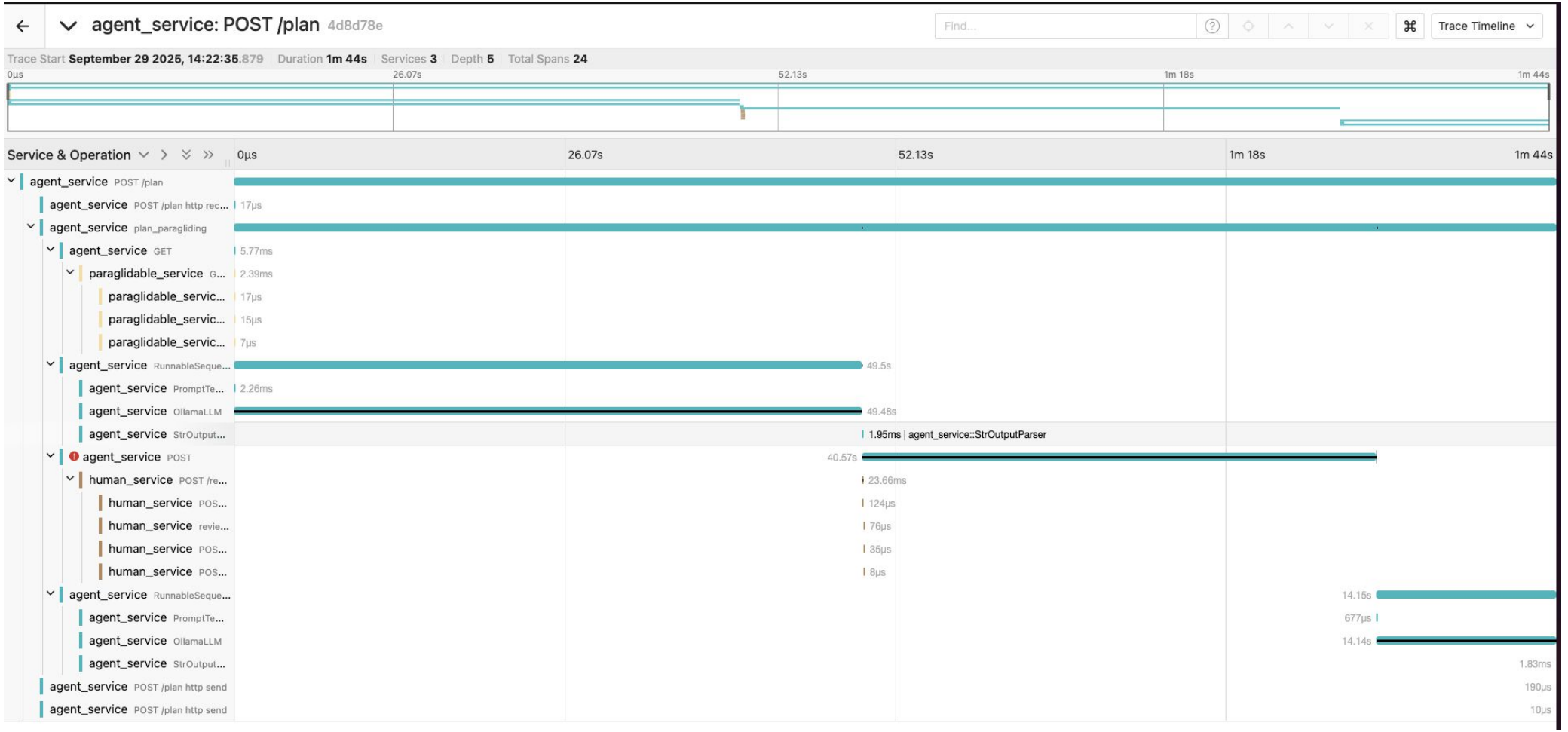
- "If you are a developer, what would you want to improve, develop, or design in HIS-LLM? How would you approach it, and when should you consider HIS-LLM as a solution for your problem?
- If you are not a developer and simply want to understand HIS-LLM, how would you prefer it to be explained or addressed?
- Most importantly, how can all this information be linked back to trustworthiness—specifically resilience, reliability, robustness, and elasticity?"

# Hands-on tutorials

Follow hands-on tutorial for this topic here:

https://github.com/rdsea/sys4bigml/tree/master/tutorials/r4hisllm

# Trace overview provided by Jaeger



# Map the traces to interactions using the HIS-LLM analytics tool.

```
🐱 ❯ python hisllm_analytics.py --services=agent_service,human_service,paraglidable_service --jaeger-api=htt
p://localhost:16686/api/traces --feature=summary_interactions

Aggregated interaction summary:
+--------------------+---------+----------------------+
| Interaction Type   |   Count |   Avg Duration (ms)  |
+====================+=========+======================+
| S2S                |      17 |               11.564 |
+--------------------+---------+----------------------+
| H2S                |      19 |               10.301 |
+--------------------+---------+----------------------+
| H2H                |       0 |                0     |
+--------------------+---------+----------------------+
```

# Study log

- In what ways should adversarial testing differ for hybrid human–AI workflows compared to standalone AI models?

- How would you address the other R3E dimensions, beyond Robustness, in HIS-LLM?

Choose one and write short thought and sent it to MyCourses.

Reading;

- A.I. Robustness: a Human-Centered Perspective on Technological Challenges and Opportunities
- TrustLLM: Trustworthiness in Large Language Models
- Design Principles and Guidelines for LLM Observability: Insights from Developers

Kiitos
**aalto.fi**