ELFMO

Engineering Large Foundational Models for Enterprise Integration

Deliverable D3.1
Research baseline for risk, quality and conformity assessment tools and procedures

Project title	Engineering Large Foundational Models for Enterprise Integration
Project acronym	ELFMo
Project number	23004
Work package	WP3
Deliverable	D3.1
Dissemination level	PU (public)
License	CC-BY 4.0
Version	1.0
Date	2025-06-30

Contributors

Editor(s)	Juhani Kivimäki (University of Helsinki)
Reviewer(s)	Mapi Aranda (Dextromedica), Diogo Martinho (ISEP)
Contributor(s) Alphabetical order	Afnan Baig (University of Helsinki), Marcos Cobo (CIC), Tuuli Lindroos (F-Secure), Juha Mylläri (University of Helsinki), Mikko Raatikainen (University of Helsinki), Isabel Ribeiro/André Rodrigues (FTP), Tomi Sarni (Nosto Solutions Oy), Davor Stjelja (Granlund)

Abstract

The purpose of this document is to establish a comprehensive baseline for identifying and managing risks associated with Large Foundation Models (LFMs) in commercial enterprise use. It aims to provide guidelines for quality assurance and regulatory compliance. It defines key concepts and terms and outlines the technological environment concerning state-of-the-art methods and tools relevant to the ELFMo project. Key areas covered include risk assessment, decision support frameworks, continuous monitoring, and AI governance, with a focus on privacy, data quality, security vulnerabilities, and transparency challenges. The document proposes hybrid approaches to balance efficiency, performance, and privacy, ensuring LFMs' effective deployment in consumer cybersecurity, e-commerce personalization, and enterprise resource planning scenarios.

Table of Contents

1. Introduction

The purpose of this document is to provide a baseline for the ELFMo project. The scope of this document is on acknowledging and managing LFM-related risks and how to provide quality assurance and compliance with the existing regulation when serving them. The document is complementary to the model-focused deliverable [D2.1]. This document describes or defines the key concepts and terms used within the project to form a shared understanding between all partners. It also outlines the technological environment with respect to the state of the art and practice of methods and tools that are of interest for the ELFMo project.

1.1 Intended Audience

The main intended audience of the present document is the ELFMo consortium with the purpose of capturing the baseline of the project that the project will advance. However, this document is public and can provide an overview of the current practices to any interested readers. This document describes technologies for the technically oriented audience rather than for the general public.

1.2 Definitions and Interpretations

The terms used in this document have the same meaning as in the contractual documents referred to in [FPP] with Annexes and [PCA] unless explicitly stated otherwise.

1.3 Applicable Documents

The following abbreviations are used to describe other documents related to the project and related to this deliverable.

- *[FPP] ELFMo – Full Project Proposal 23004* describes the full project proposal
- *[PCA] ELFMo Project Consortium Agreement* outlines the common agreement between project participants
- *[D2.1] Baseline methods and techniques for model training and benchmarking* is a parallel, more mode-focused baseline document
- *[D4.1] LFM ecosystem documentation and ELFMo methodology V1* defines the ELFMo lifecycle model providing a context for techniques used in WP3

2. Methods for Risk Assessment and Decision Support

In the rapidly evolving domain of Large Foundational Models (LFMs), risk assessment and decision support have become crucial in navigating their commercialization. Along with the transformative potential of LFMs come inherent challenges related to privacy, data quality, and security vulnerabilities. By exploring frameworks for risk mitigation and decision-making, this chapter aims to equip organizations with methods to utilize LFMs effectively, ensuring robust consumer protection while addressing critical operational and ethical considerations.

2.1 Large Foundation Models in Consumer Cybersecurity: Risk Assessment and Decision Support Framework

2.1.1 Introduction: LFMs in Consumer Cybersecurity Markets

In many business-to-consumer (B2C) domains, particularly the consumer cybersecurity sector, cost-effectiveness and scalability represent the primary requirements for Large Foundation Model (LFM) adoption. The consumer cybersecurity space has evolved dramatically in recent years, with scams and fraud targeting individual users becoming increasingly sophisticated and prevalent. This has led to recognizing that protecting everyday users from financial scams, phishing attempts, and social engineering attacks requires innovative approaches that can deliver enterprise-grade protection at consumer scale and price points.

The explosive growth of the Edge AI for Cybersecurity Market—projected to reach USD 643.2 billion by 2034 with a 35.6% CAGR (Compound Annual Growth Rate) according to recent market analyses—demonstrates the increasing demand for efficient AI solutions at the network edge (Market.us, 2025). A significant portion of this growth is being driven by the fraud detection segment, which accounted for 30.7% of the market in 2024, "reflecting the increasing need for AI-driven fraud prevention" technologies that can protect consumers across digital environments (Market.us, 2025).

Lightweight Foundation Models provide a promising approach by offering more compact, efficient alternatives to their larger counterparts. These models can operate with significantly reduced computational requirements, enabling deployment across a broader range of consumer devices. According to recent research by Intel (2023), Edge AI solutions bring "artificial intelligence to 'the edge,' meaning closer to where data is generated" creating substantial benefits including "near-real-time responsiveness and insights, increased efficiency, reduced operational costs, and the ability to deliver new types of customer experiences" (Intel, 2023). There are also privacy preserving benefits due to the reduced data transfer need between edge and cloud. This is particularly valuable for scam protection, where real-time detection at the moment a user encounters a suspicious email, message, or website can prevent financial loss and identity theft.

For consumer cybersecurity providers, lightweight models enable a multi-layered approach to scam protection that, for example, can:

1. Analyze communication patterns: Detect anomalies in messages, emails, and social media interactions that might indicate phishing or social engineering attempts
2. Evaluate website legitimacy: Assess in real-time whether a website is authentic or a sophisticated clone designed to steal credentials
3. Monitor transaction behavior: Identify unusual patterns in financial activities that could signal account takeover or fraud
4. Provide contextual warnings: Alert users to potential scams with specific, relevant guidance rather than generic warnings

These capabilities together, delivered through efficient models that can run directly on consumer devices, represent a significant advancement in protecting everyday users against cybersecurity risks from increasingly sophisticated scam attempts.

2.1.2 Risk Assessment for LFM Implementation in Cybersecurity

Despite their advantages, LFMs introduce several key risks that must be carefully evaluated before deployment in consumer cybersecurity environments:

2.1.2.1 Privacy Concerns

Privacy represents a significant risk when deploying LFMs in consumer environments. Privacy concerns stem from two primary sources:

1. Training Data Sensitivity: Consumer cybersecurity applications process highly sensitive personal information, including browsing habits, application usage patterns, and potentially identifying metadata. While lightweight LFMs can enhance detection capabilities, they must be designed with transparency and explainability (Rahmati, 2025).
2. Model Leakage Risks: Foundation models can inadvertently memorize training data, potentially exposing sensitive information during inference. A rigorous privacy-by-design approach is necessary.

To mitigate these risks, privacy-preserving techniques should be researched and implemented.

2.1.2.2 Data Quality Dependencies

Model performance in cybersecurity applications is heavily dependent on data quality. Poor or biased training data can lead to significant consequences, including:

1. False Positives: Incorrectly flagging legitimate activities as threats, causing disruption to users and potentially leading to alert fatigue among security professionals.
2. False Negatives: Failing to detect actual threats, leaving systems vulnerable to attacks that could have been prevented.

This highlights the importance of comprehensive testing across diverse datasets. To address data quality concerns, robust data validation pipelines should be implemented that include statistical anomaly detection, data cleaning procedures, and continuous monitoring of model performance metrics against benchmark datasets.

2.1.2.3 Security Vulnerabilities

LFMs themselves may introduce new attack vectors into cybersecurity systems, including:

1. Adversarial Attacks: Malicious inputs specifically designed to deceive the model.
2. Model Poisoning: Compromising the model during training by injecting malicious data samples or manipulating model weights.
3. Transfer Learning Attacks: Exploiting knowledge transfer between models to extract sensitive information or compromise model integrity.

2.1.2.4 Transparency and Explainability Challenges

The "black-box" nature of many foundation models poses significant challenges for cybersecurity applications, where understanding the rationale behind alerts and decisions is crucial. Limited explainability has several consequences:

1. Reduced Trust: Security professionals and end-users may be hesitant to rely on systems whose decisions cannot be verified or explained.
2. Compliance Issues: Regulations like GDPR include provisions for "right to explanation" for automated decisions, which may be difficult to satisfy with opaque models.
3. Incident Response Challenges: When threats are detected, understanding the underlying reasoning is critical for effective response and remediation.

2.1.3 Decision Support Methods

The key challenge in LFM implementation is finding the optimal balance between model size, computational efficiency, privacy, and detection accuracy. Recent research converges on a hybrid approach that combines lightweight edge models with more powerful cloud-based systems.

2.1.3.1 Hybrid Approach: Balancing Efficiency and Performance

This hybrid architecture offers several advantages:

1. Tiered Processing: Simple, frequent tasks can be handled by lightweight models at the edge, while complex analyses that require more computational power can be offloaded to larger models in the cloud.
2. Privacy-Preserving Design: Edge processing reduces the need to transmit sensitive data to cloud environments.
3. Adaptive Response: The system can dynamically adjust its operational mode based on threat levels, network conditions, and available resources.

2.1.3.2 Evaluation Framework for LFM Selection & Risk Mitigation

Given the rapid pace of model development in the field, a structured evaluation framework is essential for selecting appropriate LFMs for consumer cybersecurity applications. The framework should include:

- Performance Metrics: detection accuracy, false positive rate, response time, correctness, completeness, harmfulness, resource utilization
- Operational Considerations: deployment flexibility, update mechanisms, integration capabilities, scalability, costs
- Security and Privacy Evaluation: resilience to adversarial attacks, data handling practices, privacy guarantees

To address the identified risks, several mitigation strategies should be implemented such as privacy-preserving techniques, data quality management, security measures, and explainability enhancements.

For effective risk assessment and ongoing monitoring, organizations should implement a comprehensive evaluation and monitoring framework.

2.1.4 Conclusion

The adoption of Lightweight Foundation Models in consumer cybersecurity represents a significant opportunity to improve protection while addressing resource constraints and privacy concerns. By implementing a structured risk assessment and decision support framework, organizations can navigate the challenges associated with LFM deployment and realize their benefits.

Key directions for further assessment for or considering LFM implementation include:

1. Adopt a hybrid approach: Combine edge and cloud models to balance performance, efficiency, and privacy
2. Implement robust evaluation: Use standardized benchmarks and continuous monitoring to assess model effectiveness
3. Prioritize privacy by design: Incorporate privacy-preserving techniques from the outset
4. Ensure explainability: Integrate interpretable models and visualization techniques to build trust and support compliance
5. Plan for evolution: Establish mechanisms for continuous improvement and adaptation to emerging threats

By addressing these challenges and opportunities, the consumer cybersecurity industry can harness the potential of lightweight LFMs while ensuring they meet the stringent requirements of consumer protection in an increasingly complex threat landscape.

2.2 Requirements for built environment design and consultancy domain

In the built environment design and consultancy domain, adopting Generative AI (GenAI) and Large Foundation Models (LFMs) presents transformative opportunities but also significant technical, ethical, and human-centered challenges. Effectively leveraging these technologies requires a robust, human-centered methodology designed specifically for contexts where data is typically non-personal yet sensitive, often containing proprietary client information critical for competitive advantage. Human experts remain central to the process, ensuring that AI enhances rather than replaces professional expertise.

Core technical challenges include ensuring accuracy and reliability of AI outputs, integrating AI tools smoothly into complex workflows in the industry, and maintaining interoperability across specialized software platforms and data domains (Liang, 2024; Emaminejad, 2022; Zamora, 2025). Additionally, high costs of AI implementation and maintenance can disproportionately impact smaller firms, potentially limiting innovation and equitable access across the industry (Zamora, 2025).

Human factors, such as resistance to change, skills gaps, cognitive overload from managing AI-generated outputs, and potential over-reliance on technology, pose substantial hurdles to adoption (Choudhuri, 2024; Zhou, 2024; Zamora, 2025). Concerns around job displacement and loss of human creativity emphasize the need for targeted training programs and education initiatives to balance technological advancements with traditional architectural and engineering skills (Zamora, 2025).

Key risks specific to the built environment consultancy domain include privacy and confidentiality breaches, where sharing sensitive company and client data with AI systems risks exposing proprietary knowledge or violating confidentiality agreements. The rapid technological obsolescence inherent in AI advancements can lead to premature investment in solutions quickly overshadowed by more advanced models from major technology providers. Transparency and explainability gaps in AI-generated content raise accountability and liability concerns, particularly when recommendations influence critical professional decisions (Weidinger, 2025; Weisz, 2023; Zamora, 2025). Furthermore, variability in AI outputs without rigorous quality assurance can compromise deliverable reliability, while difficulties in monitoring and tracing AI model versions complicate issue resolution and ongoing improvement efforts.

In summary, applying LFMs within the built environment design and consultancy industry involves addressing critical issues such as privacy and confidentiality risks, technological obsolescence, transparency and explainability concerns, output variability, challenges in monitoring and traceability, and the education and training of specialists to effectively integrate AI into professional practice. These factors must be managed comprehensively to ensure responsible and effective use of LFMs.

2.3 Risk Assessment and Decision Support Methods for Telemarketing

In the telemarketing domain, a clear governance structure is needed to monitor project progress, manage risks, and ensure quality during the development phase. This structure consists of the parts described in the following sections.

2.3.1 Project Risk Management

A continuous risk management process ensures that potential challenges are identified, assessed, and addressed early to minimize disruptions caused by unexpected issues. This includes technical, ethical, and project-related risks, with plans developed and regularly reviewed to manage their impact effectively. This process consists of the following components:

- Identification: proactive identification of technical (data integration, real-time performance, interoperability, security), ethical (bias, privacy, and project (resources, timelines, scope changes) risks.
- Assessment: Analysis of the probability and impact of each identified risk.

- Response Planning: Development of mitigation, contingency or acceptance plans for key risks.
- Monitoring and Control: Periodic follow-up of risk status and effectiveness of response plans.

2.3.2 Development Quality and Compliance Management

Development quality and compliance management ensures that software meets defined standards through rigorous reviews, testing, documentation, and adherence to requirements. To achieve this, the following practices are suggested:

- Code and peer reviews.
- Multi-level testing strategies (unit, integration, system, system, user acceptance - UAT).
- AI specific testing (performance, robustness, bias).
- Comprehensive technical documentation of architecture, APIs, models and processes.
- Periodic conformity assessments with defined requirements (functional, non-functional, legal).

Furthermore, a formal process should be established to request, evaluate, approve, and implement changes to the project scope, requirements, or technology, ensuring that impact analysis on schedule, cost, risk, and compliance is conducted. This process entails a detailed communication plan with periodic progress reports to keep all stakeholders informed.

2.3.3 Security by Design and Default Principles

- Security by design and default ensures that protective measures are embedded from the outset, shaping both architecture and processes to minimize vulnerabilities. It emphasizes proactive planning and built-in safeguards rather than reactive fixes, and should address the following: Security is to be integrated into each phase of the software development life cycle (SDLC).
- The principle of minimum attack surface should be applied.
- Default configurations should be secure.
- Incident response will be planned from design.

2.3.4 Data Security and Privacy (GDPR)

Ensuring data security and privacy involves including regulatory compliance and protective measures in every stage of development. This consists of aligning with the following:

- GDPR Compliance: Design aligned with GDPR principles (lawfulness, fairness, transparency, purpose limitation, data minimization, accuracy, retention period limitation, integrity and confidentiality, proactive accountability).

- Anonymization/Pseudonymization Techniques: Implementation of robust techniques to protect personal data in training and operation.
- Differential Privacy and Federated Learning: Research and possible implementation of these advanced techniques to protect privacy in model analysis and training.
- Impact Assessments (DPIA): Conducting Data Protection Impact Assessments when required.

2.3.5 AI Model Specific Security (LFM)

Securing large foundational models (LFMs) requires addressing unique risks across the model lifecycle, from training data integrity to deployment safeguards. This includes ensuring trustworthy inputs, verifying model sources, and tightly controlling access to prevent misuse or compromise. At least the following should be addressed:

- Protection against Adversarial Attacks: Implementation of input sanitization and output guarding mechanisms to mitigate risks.
- AI Supply Chain Security: Verification of provenance and security of pre-trained models.
- Data Poisoning Detection: Monitoring of training data to detect anomalies that may indicate poisoning attempts.
- Model Access Control: Restricting access to hosted models and inference APIs.

2.4 AI Governance

Interest in AI governance is growing significantly. On one hand, the general public is increasingly vocal about the ethical implications of AI technologies. On the other hand, public authorities are setting new standards and implementing regulations to govern the operation of AI systems, with notable examples being the European Union’s AI Act (European Union, 2024). Similar actions are taking place globally, such as by the Biden administration’s Executive Order on AI in the U.S (White House, 2023). However, industry-specific practices and standards are also emerging, especially for specific industry sectors.

In response to these growing demands, research has increasingly focused on AI governance practices and the ethical exploration of AI technologies. However, much of the research so far has been conceptual and high-level, imagined as static, lacking concrete, practical guidelines that can be directly applied in day-to-day engineering work and business decision-making. In addition, AI governance is often considered only during the development phase of AI projects, rather than throughout their entire lifecycle. For example, a recent survey identified over 100 ethics frameworks, but many were identified as too abstract to be readily translated into actionable designs for AI systems (Prem, 2023). In addition, the MIT identifies 777 risks for AI technologies classified in 7 categories broadening the notion of just relying on regulations to be compliant with public and employees' fears (MIT, 2024). One example of a conceptual framework that somewhat considers practical application is the hour-glass model for AI governance (Mäntymäki, 2022) that differentiates the environmental, organizational, and AI system levels. At the AI system level, where the system level tries to make the connection from the higher level, abstract concepts and principles to the system lifecycle.

While industrial adoption of MLOps (Kreuzberger, 2023) for inhouse ML systems rather than LFMs has made advances in addressing the technical aspects of AI/ML system development and engineering, it often overlooks socio-technical concerns, such as fairness, accountability, human oversight, and business impact. Concurrently, tools and practices have been introduced to support AI system development, such as MLFlow (https://mlflow.org/), Google's model cards (https://modelcards.withgoogle.com/), and IBM's AI fact sheets (https://www.ibm.com/docs/en/software-hub/5.1.x?topic=services-ai-factsheets). While these tools effectively capture static snapshots of information during AI model development, there remains a valuable opportunity to enhance them with holistic, continuous tracking capabilities for AI systems in production—helping to better align with the requirements of the AI Act and similar regulations.

2.5 Risk Assessment and Decision Support Methods for E-commerce

There is an emerging industry trend towards agentic and multi-agent system architectures, representing the next phase in integrating large language model (LLM)-based conversational interfaces with toolchains and capabilities for generating actionable insights (Alvarez, 2024). Merchant-facing agents are designed to automate complex workflows, enabling dynamic adjustment of merchandising rules and the derivation of analytical insights from behavioral data.

A parallel development is the deployment of consumer-facing shopping assistants, where accuracy and scalability are of critical importance. These agents aim to enhance the personalized shopping experience by facilitating natural language interactions at various touchpoints within merchant storefronts (Ramachandran, 2024).

2.5.1 Key Risk Assessment and Decision Support Considerations

A number of critical factors must be addressed when assessing the viability and robustness of agentic architectures in e-commerce personalization scenarios. These include cost-efficiency, scalability, security, latency, and output accuracy, all of which are essential for ensuring reliable and safe deployment.

2.5.1.1 Cost-Efficiency and Scalability

Merchants exhibit diverse operational profiles, and their resource requirements may fluctuate significantly, particularly during influencer-driven sales campaigns or other demand spikes. As such, the ability to dynamically scale the underlying agentic infrastructure is essential to maintain operational continuity and cost-effectiveness. Elastic scaling mechanisms, including auto-provisioning of compute resources and stateless agent orchestration, are therefore critical components of resilient deployment strategies.

2.5.1.2 Latency

In consumer-facing applications, low-latency responses are critical for maintaining seamless user experiences. Any degradation in response time can negatively impact key performance indicators such as site engagement and conversion rates. Therefore, ongoing latency monitoring—particularly with respect to web performance metrics like Core Web Vitals—is essential to ensure that the integration of agentic systems does not compromise frontend responsiveness (Miernik, 2024).

2.5.1.3 Accuracy

Many decision-support tasks in merchant environments require precise analysis of structured tabular and time-series data. While LLMs offer powerful language capabilities, they often lack the numerical reasoning accuracy required for these domains. As such, hybrid approaches combining LLMs with traditional machine learning or statistical tools are essential. Architectures based on protocols such as MCP, A2A, or ACP can facilitate tool-augmented reasoning by allowing LLMs to interface with specialized analytical components.

Given that some agentic actions—such as pricing changes or inventory updates—can have a substantial business impact, the risk of hallucination or misalignment must be minimized. Consequently, a human-in-the-loop (HITL) mechanism remains necessary for approving high-stakes or irreversible actions (Ehtesham, 2025).

2.5.2 Security

In addition to risks related to system performance, security concerns need to be addressed as well. These include at least the following.

2.5.2.1 Prompt Injection and Output Manipulation

Large Language Models (LLMs) are vulnerable to prompt injection attacks, where adversarial user inputs are crafted to manipulate the model's intended behavior. These attacks pose a significant risk in consumer-facing applications, where user-generated content—such as reviews or queries—can carry hidden instructions that trigger unintended actions (Liu, 2023). The threat becomes more severe in multi-agent systems, where a single compromised prompt can propagate between LLM agents, leading to cascading failures and loss of control across the system (Lee, 2024).

2.5.2.2 Data Leakage

LLMs fine-tuned or deployed in contexts containing sensitive merchant or customer data are at risk of inadvertently disclosing proprietary information or personally identifiable data during inference. This is especially true if memory or retrieval mechanisms are not adequately scoped or sandboxed. To mitigate these risks, robust data governance frameworks (e.g. greatexpectations.io)— access controls, and auditable logging should be implemented (Zhang et al, 2024).

3. Methods for Integrated Business and Model Monitoring

In the realm of commercial Large Foundational Models (LFMs), integrated business and model monitoring is crucial for ensuring sustained performance and alignment with organizational objectives. This chapter explores methodologies for continuous monitoring with interactive observability, emphasizing the dynamic nature of LFMs and their potential for performance degradation, inconsistency, and unpredictability. By detailing components such as data collection, storage, analysis, visualization, and alerting, this section provides a comprehensive framework for maintaining model reliability and transparency, facilitating informed decision making. Additionally, it highlights the importance of aligning model outputs with business KPIs and addresses the complexities of monitoring in different settings, such as with multi-agent architectures and log anomalies. Through proactive monitoring, organizations can optimize their LFM deployments, ensuring they remain effective and aligned with strategic goals.

3.1 General components of an on-premise monitoring system

Successful implementation of machine learning models does not end with their deployment in production. In fact, this stage marks the beginning of a new critical cycle: continuous monitoring. Model monitoring has become an essential component of the machine learning lifecycle, as it ensures that models continue to deliver value over time. Unlike traditional software, which tends to maintain predictable behavior, machine learning models operate in dynamic environments, exposed to changing data and evolving real-world conditions. Therefore, continuous monitoring is indispensable for detecting and addressing performance degradation or unexpected behavior, thereby ensuring the reliability and effectiveness of the machine learning system as a whole.

The importance of model monitoring manifests across multiple dimensions. First, it allows for maintaining model performance over time. By nature, models are subject to a gradual decline in predictive power. This degradation, often referred to as "drift," can occur due to changes in the distribution of input data, evolving underlying patterns in the data, or even errors in the data ingestion process itself. Continuous monitoring provides the necessary tools to detect such drift and take corrective actions, such as retraining the model with recent data or modifying its architecture, for instance, by incorporating adaptive learning techniques, adding drift detection layers, or adjusting model complexity to better capture new data patterns (Martyr, 2025; Paka, 2023).

In addition, monitoring is fundamental for ensuring model reliability. A reliable model produces consistent and accurate results, minimizing errors and unexpected predictions. Monitoring model health (which includes tracking service availability, prediction latency, and resource utilization) makes it possible to quickly identify and resolve any issues that may compromise the system's reliability.

Another important aspect of monitoring is that it contributes to transparency. In many scenarios, it is essential to understand how a model reaches its decisions, especially in critical applications where accountability and responsibility are paramount. Explainability techniques, which can be integrated into the monitoring process, offer insights into the importance of different input features and the internal logic of the model, facilitating the identification of potential biases or functional errors.

Finally, monitoring also plays a key role in optimizing both the model and the supporting infrastructure. Data collected through monitoring can reveal inefficiencies in the model or inference pipeline, allowing for performance improvements and reductions in operational costs.

The model monitoring process involves several key stages, each with its own considerations and challenges. These stages are data collection, data storage, data analysis, visualization, alerting, and corrective action and are described in detail in the following subsections.

3.1.1 Data Collection

This foundational stage involves capturing all relevant signals associated with the model's behavior and environment. This includes model inputs and outputs, confidence scores, model version, user and system-level logs, performance metrics (e.g., latency, throughput), and contextual metadata such as timestamps and user segments.

The design of this layer must balance observability with privacy and storage efficiency. Special attention should be paid to which fields are collected, especially in regulated environments where data minimization and anonymization are mandatory. For real-time systems, it's also crucial to support high-throughput data capture without introducing latency.

Some examples of possible tools:

- MLflow: Records input parameters, outputs, and model versions.
- Apache Kafka / Fluentd / Logstash: Streaming ingestion of high-volume logs and metrics.
- Prometheus: Micro-matching ingestion of high-volume metrics.
- OpenTelemetry: For standardized telemetry across systems and services.

3.1.2 Data Storage

Once data is collected, it must be stored efficiently for future analysis, compliance audits, and model debugging. This stage involves organizing large volumes of structured and unstructured data, ensuring integrity, accessibility, and adherence to retention policies.

Storage choices should align with data velocity, query performance, and regulatory constraints. For example, time-series databases are ideal for monitoring metrics over time, while object stores are better suited for log archives and model artifacts.

Some examples of possible tools:

- Prometheus: For time-series metrics collection.
- PostgreSQL + TimescaleDB: For time-series metric queries and structured logs.
- Delta Lake / Snowflake: Versioned and analytics-ready storage, supporting large-scale batch and streaming data.

3.1.3 Data Analysis

With data stored and accessible, the next step is processing and interpreting it to extract meaningful insights. This includes computing key performance indicators (e.g., precision, recall), analyzing drift (both data drift and concept drift), detecting outliers or anomalies, and summarizing behavior through reports. (Goh, 2024; Chen, 2024)

Depending on the context, analysis may range from simple descriptive statistics to advanced unsupervised anomaly detection. This layer can also feed automated retraining pipelines or governance dashboards.

Some examples of possible tools:

- NannyML: Performance estimation without immediate ground-truth labels.
- Ragas: Suite with multiple performance scores (e.g., to detect hallucination).
- Pandas, PySpark, Scikit-learn: For custom statistical or ML-based analyses.

3.1.4 Visualization

Effective communication of insights is key to making monitoring actionable. This stage focuses on creating dashboards and visual tools that help technical and non-technical stakeholders understand the current state and trends of model performance.

Dashboards should be role-specific: operational teams monitor latency and resource usage, while data scientists track accuracy and drift indicators. Ideally, visualizations are updated in near real-time and allow drill-downs into specific model versions or data segments.

Some examples of possible tools:

- Grafana: For real-time metric dashboards using sources like Prometheus or Loki.
- Kibana: For log analysis and anomaly detection with Elasticsearch backends.
- Apache Superset / PowerBI / LUCA BDS: For rich, business-aligned views of monitoring data.

3.1.5 Alerting

Proactive alerting is essential to detect issues before they escalate into failures. This stage involves defining thresholds or anomaly triggers for key metrics (e.g., sudden drop in F1 score, latency spikes, unexpected drift) and routing notifications to the appropriate teams.

Alerting systems should support multi-channel delivery and incident prioritization, as well as mechanisms for auto-resolving alerts or suppressing noise. Some teams also integrate alert logs into issue tracking systems like Jira or ServiceNow.

Some examples of possible tools:

- Prometheus + Alertmanager: Industry-standard alerting stack for metrics with communication channels (Slack / Microsoft Teams / Email).
- Apache Airflow: To trigger automated alerting workflows.

3.1.6 Corrective Action

Monitoring only creates value if issues are addressed. This final stage closes the feedback loop by enabling corrective measures, which may include retraining the model, tuning hyperparameters, adjusting preprocessing pipelines, or resolving infrastructure issues (e.g., autoscaling problems, GPU memory overflow).

A well-integrated system may initiate semi-automated responses: for instance, a drop in accuracy could trigger data validation, and if confirmed, launch a retraining job. However, human-in-the-loop governance is essential in regulated environments to validate changes before redeployment.

Some examples of possible tools:

- Apache Airflow: To trigger automated workflows (e.g., retraining or redeploying).
- MLflow / Kubeflow: Automate retraining and deployment steps.
- Terraform + Kubernetes: To adjust infrastructure configuration if needed.
- GitOps frameworks: For managing versioned model rollbacks or upgrades.

In conclusion, model monitoring is not a one-time task. Instead, it is a continuous and essential process for the long-term success of any machine learning system in production. By embracing best practices and choosing the right tools across all stages, from data collection to corrective action, organizations can ensure their models remain performant, reliable, and transparent. This ongoing vigilance not only preserves the value generated by machine learning solutions but also strengthens trust, accountability, and operational excellence.

3.2 Monitoring LFM output and reacting to alerts

Large Foundational Models (LFM) have transformed the intelligent systems in different domains (healthcare, legal, e-commerce and more), However the power and flexibility these LFM models provide is remarkable but their outputs in production settings could be inconsistent, unpredictable or unsafe. The random nature of generation, continuous prompt evolving and fine-tuning cycles can contribute to potential performance degradation, hallucination or undesirable behavior, that often goes undetected until they start creating problems. So, there is a need for a continuous, context-aware and proactive monitoring system that can detect anomalies in model behavior and trigger timely alerts. This research will investigate a unified monitoring framework that addresses these three dimensions of risks.

- Contextual Anomaly Detection (e.g., contextual precision/recall/relevancy)
- Fine-Tuning Anomaly detection (Overfitting, Distributional shift)
- Prompt Behavioral drift and Response consistency.

3.2.1 Contextual Anomaly Detection

Currently, most LLMs rely on static test-sets or surface level metrics like BLEU, ROUGE, F1, which assume a single ‘correct’ answer and often fail to capture semantic relevance. They lack dynamic context check; they might compare the outputs with benchmarks offline but may not continuously vet new output against up-to-date context. To address this shortcoming, a more robust evaluation method is needed which can monitor metrices like Contextual Precision, Contextual Recall, and Contextual Relevancy. This method could possibly use some other LLM Model to semantically assess how well an output aligns with its retrieved context. In practice, each user query, its LFM output, and its associated context can be evaluated and scored against predefined thresholds. If the score is low; meaning output is misaligned or hallucinated, the system can raise an alert in real time. These metrics are useful in Retrieval Augmented Generation (RAG) (Lewis, 2020) which often hallucinate by making up the facts or by ignoring context. These errors have major real-world consequences, even legal and financial issues. By incorporating context-aware metrices in MLOps pipelines we can catch and correct these problems as they happen which will not only increase performance but also enrich trust, transparency and safety in high-stakes environments. Currently there are some tools like Deep Eval and Galileo, which work on the same principles but require more exploration.

3.2.2 Fine-Tuning Anomaly detection

Monitoring LFM is critical as they evolve through Fine-tuning, we need to prevent silent failures like overfitting, distributional drift or behavioral regression. Fine-tuning is often treated as a ‘black box’ and gives limited visibility to what changes during fine-tuning. Traditional approaches to monitor these fine-tune models involve static validation and A/B testing, which are time-consuming and can miss subtle regression and may fail to capture real-time silent anomalies. This lack of proactive monitoring may silently degrade the model by losing generalization, hallucinating or forgetting previously learnt capabilities (often noticed after deployment and their impact has already occurred).

A few modern strategies to overcome this issue could be; firstly, to have an anomaly detection system which instruments the training process by tracking signals like loss curves, gradient norm distributions, data distribution statistics. For example, we can continuously monitor validation perplexity and class distribution, sudden loss spikes, vanishing gradients, or a growing gap between train and validation performance. These can indicate overfitting or data issues. We can also track similarities between fine-tuning data and live input through KL divergence algorithm on embedding distributions. If a new batch of user queries has features very different from the fine-tune set, an alert fires. In short, this approach means to build such monitoring system that collects training logs and data stats, apply drift detection algorithm on data during and after each training run.

Another approach implemented in parallel to the above one could be to do checkpoint regression monitoring, which compares model behavior (by computing delta) before and after fine-tuning using different factual metrices, output entropy and latent representational shifts to flag significant behavioral regressions. This system may generate reports and alerts before deployment; it could even be in a CI/CD pipeline, which will provide a safety net for models.

3.2.3 Prompt Behavioral drift and Response consistency

Traditional LLMs monitoring is based on tracking general and most common metrices, like latency, accuracy or other static ones but fail to detect when the same prompt starts yielding inconsistent, off-tone, or factually degraded outputs. This might be due to model updates or domain drift or due to silent deployment changes. This blind spot leads to undetected semantics or behavioral drift, where the model responds differently to the same prompt over time, which erodes reliability. To address this, a possible approach could be to continuously track “prompt-response consistency” by maintaining a baseline library of canonical prompt-output pairs and measuring changes in embedding similarity, sentiment, topic alignment, and structural features (like response length on n-gram diversity).

Another possibility to monitor prompt drift could be through comparison of incoming user prompts over time. It can be done by clustering incoming prompts in embedding space and comparing them against historical distribution to catch shifts in user intent or problem domain. For example, a 30% drop in cosine similarity or a sudden spike in response entropy can trigger alerts.

Both of these approaches enable unsupervised detection of output degradation and behavioral shifts. It goes beyond static validation sets and can enable real-time analytics. These metrics can highlight when retraining or prompt adjustment is needed as monitoring prompt differences over time determines if user behavior is changing, which means the model needs to be updated.

3.3 Business KPI monitoring

The successful deployment of Large Foundational Models (LFMs) in e-commerce requires close alignment with clearly defined business Key Performance Indicators (KPIs). Platforms such as AWS SageMaker offer integrated monitoring capabilities that enable tracking of cost-efficiency metrics—including infrastructure expenditure, inference costs, and compute resource utilization. These insights support teams in optimizing resource allocation by balancing operational costs against model latency and predictive accuracy. The business KPI monitoring can be closely associated with monitoring service level objectives (SLOs) (Beyer, 2016).

Moreover, commercial monitoring tools like Amazon CloudWatch, alongside open-source solutions such as Grafana—when paired with metrics backends like Prometheus—can establish direct connections between model performance and core e-commerce outcomes. These outcomes may include conversion rates, customer engagement levels, and revenue per session. By correlating LFM performance data with relevant business KPIs, organizations can continuously assess the real-world effectiveness of their deployed models, thereby ensuring sustained alignment with overarching business objectives (Chinoy, 2024).

It is equally important to adopt monitoring standards such as OpenTelemetry, which provide a unified, vendor-neutral framework for observability across distributed systems (Liu, 2025). A notable example of an advanced implementation of OpenTelemetry for multi-agent systems is OpenLLMetry, which extends these practices specifically to the agentic domain (Traceloop, 2025).

3.4 Monitoring Multi-Agent Architectures

In multi-agent architectures based on the Model-Context Protocol (MCP) or similar frameworks (e.g. A2A, ACP), agents collaborate by exchanging structured context representations to fulfil complex user requests. While MCP facilitates streamlined communication and coordination among agents, ensuring operational robustness necessitates comprehensive monitoring mechanisms that are specifically tailored to the intricacies of agent-based workflows (Aldridge, Brooker, & Sivasubramanian, 2025).

Frameworks such as Langfuse and LangSmith offer specialized functionality for tracing and monitoring multi-agent systems. These tools provide end-to-end visibility into request flows across interconnected agents, capturing critical metrics including response latency, execution paths, and error rates. Such detailed instrumentation supports rapid issue diagnosis, performance tuning, and informed decisions regarding system scalability (Langfuse, 2024).

Crucially, effective monitoring in MCP-based agentic systems must extend beyond the performance of individual agents. It must also encompass the orchestration layer—capturing inter-agent interactions, data dependencies, and full execution traces. This systems-level observability ensures that organizations can verify the reliability and responsiveness of complex agent workflows, while continuously aligning system behavior with enterprise KPIs and operational objectives.

3.5 Log anomaly detection

Logging helps detect errors and misbehavior in LFM-powered software systems. Such systems, consisting of a large number of components, may produce logs far too voluminous for a developer to effectively monitor and interpret by manual inspection. Simple rule-based methods can help catch common errors, but they may be inadequate for dealing with novel or unexpected failures (Landauer, 2023). This has motivated the development of machine-learning methods for anomaly detection in log files. Unsupervised methods are usually preferred as labelling data by hand would be costly and would have to be repeated regularly to deal with data drift.

Many unsupervised ML techniques have been applied to log anomaly detection – some deep-learning based and others not. The former includes RNNs, transformers, and CNNs, while the latter include SVMs, decision trees, clustering, and many other techniques (Landauer, 2023). In recent years, academic research has focused on transformer-based models. Various commercial cloud providers offer log anomaly detection services; additionally, some open-source tools, offering varying levels of integration with MLOps workflows, exist.

An issue in log anomaly detection not sufficiently addressed by existing methods is the handling of large numbers of log types, which presents two challenges. Firstly, training a separate model for each log type can be resource-intensive, whereas a single-model approach may struggle with accuracy (Zang, 2024). Secondly, anomaly scores should be comparable between log types, but some log types typically have more variation than others even in the absence of actual errors.

An emerging method for log anomaly detection and localization, Ladle (Mylläri, 2025), has been designed particularly with the multi-log-type case in mind. The development of the method has taken place partially within the current project. However, further research is required to determine how the method should be configured and integrated into MLOps practices in the case of LFM-based services.

4. Methods for Continuous risk management and validation

With commercial AI-deployments, continuous risk management and validation are central for integrating generative AI tools and large language models (LLMs) into product development environments. This section outlines methods to address the concerns that limit the adoption of these technologies, emphasizing the need for structured threat modeling and decision-making frameworks. It aims to provide actionable guidelines for identifying and mitigating risks throughout the development lifecycle. Additionally, the chapter explores strategies for managing data, model, operational, and security risks. These approaches ensure that AI systems remain reliable, transparent, and compliant, fostering trust and alignment with business objectives and regulatory requirements.

4.1 Threat modelling

As the AI revolution progresses, there is an urge to update and reshape risk management methods to meet the emerging needs of software development environments that increasingly integrate genAI tools and LLMs. According to surveys, genAI tools are already well used in development tasks (*AI | 2024 Stack Overflow Developer Survey*), providing improvement in code quality and documentation, as well as faster results (*Gen AI Tests: Productivity Statistics & Analysis*, 2024). However, security concerns limit their adoption. Therefore, project managers and CIOs should be better informed of the risks and their mitigation strategies.

Under these challenges, there is a demand for a clear decision structure and responsibility mapping for AI project risks. For this purpose, one needs to formalize how decisions are made about risk acceptability, and which technical and human actors hold responsibility. One possible strategy to address this is to introduce a structured way to document, learn form, and re-apply insights from past risk events or mitigation efforts and to develop practical, workshop-based guidelines for identifying, evaluating, and managing risks throughout the technical development lifecycle.

The resulting guidelines should be designed for direct integration into the operational models used in software and AI development – particularly those that support iterative deployment, automation, and continuous learning cycles as in machine learning operations (MLOps) frameworks. In forming these guidelines, multiple sources of information, such as literature reviews and expert interviews, should be utilized. The findings would eventually be integrated into development practices.

4.1.1 Literature review

As a part of research effort into managing foundation model operations (FMOps) related risks, a literature review was conducted. This literature review will be published as a thesis by Sulevi Sihvola. This thesis forms the baseline of our current understanding. The thesis considers usage of genAI tools through the software lifecycle and what kind of risks and mitigation strategies are related to each part. The material for the thesis has been collected from peer-reviewed articles, non-reviewed articles, white papers, blogs, and books. An eclectic collection was chosen as there isn’t much research done in this novel phenomenon and the rapid developments of the genAI and LLM tools.

4.1.1.1 Research methodology of the literature review

The literature review was conducted from 42 articles, 26 blogs, 6 white papers, 21 books, and 10 standards or frameworks. Search for the material was done using Google Scholar, Scopus, Aalto-Primo, and O’Reilly. In these services, search terms were different combinations of “AI”, “genAI”, “LLM”, “Risk”, “Risks”, “Software development”, “Software Development Life Cycle”. Other sources included Slack discussions in Siili Solutions, news articles, and word of mouth.

The purpose of the literature review was to find risks that arise when using genAI and LLM tools in software development. At the same time there is a need to build a larger point of view for the research questions i.e., the whole software development life cycle, traditional risks associated with software development, rules and regulations that might affect AI usage, and existing guidelines or governance frameworks. Regarding the ELFMo project, the central contribution of the thesis is to outline best practices and risk mitigation strategies for usage of genAI software development tools.

4.1.1.2 Findings

As preliminary findings, the risk of using AI-tools in software development was identified in four thematic categories:

1. Technological Risks:
   - Prompt Injection, Indirect Prompt Injection, Insecure Output Handling, Training Data Poisoning, Model Denial-of-Service, Supply Chain, Performance, Hallucinations, Lines of Code, Code Quality
2. Human-Centered Risks:
   - Loss of Critical Thinking, Loss of Skill Learning/Acquisition, Human Trust in AI/HCI, Developers Downplaying Risks, Bias
3. Governance Risks:
   - Data Leakage, GDPR, IP/Licensing, Ethical Risk
4. Operational Risks:
   - Agents, Autonomy, Integration

Many of the risks aren’t novel risks in software development life cycle but are in novel settings and amplified by genAI and LLM tools and require new ways to mitigate them. As an example, humans are the source of the risks, especially when they are downplaying it, they don’t even understand the risks, or they don’t even know about the risks (Duke, 2022). GenAI might aggravate these problems such as loss of critical thinking and loss of skill learning (Mollick, 2024).

There are frameworks and standards, but they do not take direct action towards software development life cycle and genAI tools that might be part of it. They are more focused on AI solutions developed and deployed by organizations, and risks associated with these systems. There are of course overlaps in risks with these systems as many of them are technically based on similar LLM solutions.

4.2 The AI-Driven ERP Platform

The AI-Driven ERP Platform aims to improve Enterprise Resource Planning (ERP) by transitioning from monolithic systems to a modular design, and to support microservices approach. By embedding AI and Large Foundation Models (LFMs), the platform will automate processes, improve decision-making, and deliver an improved user experience, while ensuring regulatory compliance and ethical AI operation.

4.2.1 Continuous Risk Management Strategies

4.2.1.1 Data Risk Management

ERP data is dynamic and heterogeneous (financial records, HR data, supply chain metrics, etc.). LFMs need high-quality, representative, and unbiased data to operate reliably and ethically. Risks include low-quality inputs, privacy issues (e.g. GDPR), among others.

Strategies:

1. Automated Data Quality Monitoring: Implement checks for missing data, anomalies, and schema changes across all integrated ERP modules.
2. Privacy Protection: Apply pseudonymization and differential privacy where necessary to prevent leakage of personal and sensitive information.

Baseline for Validation:

1. Track data completeness, and conformity against ERP schemas.
2. Use past ERP process execution metrics (such as cycle times, error rates, approval rates, etc.) as reference baselines.

4.2.1.2 Model Risk Management

LFMs in ERP scenarios should produce accurate, relevant, and explainable outputs for decision support and automation. Risks involve hallucinations, bias, and degradation of performance over time.

Strategies:

1. Performance Monitoring: Use benchmark datasets (HR, Finance, Logistics) to regularly evaluate model predictions against expected outputs.
2. Explainability Tools: Integrate LIME, SHAP or similar tools for prediction interpretation.
3. Retraining and Fine-tuning Pipelines: Establish schedules or triggers (performance drops) for model retraining or fine-tuning using fresh ERP data.

Baseline for Validation:

1. Benchmark accuracy, precision, recall, and latency.
2. Validate retrained models in staging environments before deployment.

4.2.1.3 Operational Risk Management

ERP AI systems must communicate through various modules and APIs, ensuring consistent and performant operations. Risks include integration failures, downtime, version mismatches, and microservices constraints.

Strategies:

1. Continuous Integration and Deployment (CI/CD): Automate deployment workflows with rollback capabilities.
2. API Monitoring: Monitor API uptime, latency, and error rates for microservices and ERP module integration points.
3. End-to-End Testing: Automate regression tests simulating typical ERP communications.

Baseline for Validation:

1. Monitor system availability and the percentage of time the ERP platform and AI services are operational and accessible by users and connected systems (microservices, APIs, and ERP modules).
2. Track API uptime and performance metrics and failed communications.

4.2.1.4 Security and Privacy Risk Management

Handling sensitive business and personal data requires solid security controls, especially given regulatory requirements (GDPR, AI Act). Risks include unauthorized access to LFMs.

Strategies:

1. Role-based Access Control (RBAC): Enforce strict access controls for AI services and datasets.
2. Encryption and Secure Communication: Encrypt data for the protected exchange of information; implement API security best practices.

Baseline for Validation:

1. Verify encryption settings and TLS certificates periodically.
2. Document and review outcomes of security testing assessments.

By adopting these strategies, the ERP platform will ensure that the LFMs remain dependable, transparent, and compliant throughout their lifecycle, supporting continuous validation and trustworthiness aligned with business and regulatory needs of the use case.

5. Conclusions

This document emphasizes the transformative potential of Large Foundation Models, offering enhanced protection and efficiency while addressing key challenges such as privacy, data quality, and transparency. By adopting structured risk assessment and decision support frameworks, organizations can navigate the complexities of LFM deployment and utilize their benefits effectively. The document addresses a variety of techniques aimed at risk detection and management, combining edge and cloud models, relevant evaluation metrics, privacy-by-design principles, and explainability enhancements to build trust and compliance. Continuous monitoring and risk management strategies are essential to ensure the long-term success and reliability of LFMs. By addressing these challenges and opportunities, the ELFMo project aims to empower the industry to harness the potential of LFMs, while meeting stringent requirements for consumer protection and regulatory compliance.

References

AI | 2024 Stack Overflow Developer Survey (2024). https://survey.stackoverflow.co/2024/ai

Aldridge, N., Brooker, M., & Sivasubramanian, S. (2025). Open protocols for agent interoperability part 1: Inter-agent communication on MCP. *AWS Open Source Blog*. https://aws.amazon.com/blogs/opensource/open-protocols-for-agent-interoperability-part-1-inter-agent-communication-on-mcp/

Alvarez, G. (2024, October 21). Gartner Top 10 Strategic Technology Trends for 2025. Gartner. https://www.gartner.com/en/articles/top-technology-trends-2025

Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). Site reliability engineering: how Google runs production systems. O'Reilly Media, Inc.

Chen, Y., Fu, Q., Yuan, Y., Wen, Z., Fan, G., Liu, D., Zhang, D., Li, Z & Xiao, Y. (2023). Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (pp. 245-255). https://doi.org/10.1145/3583780.3614905

Chinoy, H., & Liu, A. (2024). KPIs for gen AI: Measuring your AI success. Google Cloud Blog. https://cloud.google.com/transform/gen-ai-kpis-measuring-ai-success-deep-dive

Choudhuri, R., Trinkenreich, B., Pandita, R., Kalliamvakou, E., Steinmacher, I., Gerosa, M., Sanchez, C. & Sarma, A. (2024). What Guides Our Choices? Modeling Developers' Trust and Behavioral Intentions Towards GenAI. *arXiv preprint arXiv:2409.04099*. https://arxiv.org/abs/2409.04099

Duke, S. A. (2022). Deny, dismiss and downplay: developers’ attitudes towards risk and their role in risk creation in the field of healthcare-AI, In *Ethics and Information Technology*, 24(1). https://doi.org/10.1007/s10676-022-09627-0

Ehtesham, A., Singh, A., Gupta, G. K., & Kumar, S. (2025). A survey of agent interoperability protocols: Model Context Protocol (MCP), Agent Communication Protocol (ACP), Agent-to-Agent Protocol (A2A), and Agent Network Protocol (ANP). *arXiv preprint arXiv:2505.02279*. https://arxiv.org/abs/2505.02279

Emaminejad, N., & Akhavian, R. (2022). Trustworthy AI and robotics: Implications for the AEC industry. In *Automation in Construction*, 139, 104298. https://doi.org/10.1016/j.autcon.2022.104298

European Union (2024). The AI Act. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689

Gen AI Tests: Productivity Statistics & Analysis (2024). https://campaign.siili.com/ai-powered-development-whitepaper

Gog, H. W., Mueller, J., Auner, N. & Thyagarajan, A. (2024). Benchmarking Hallucination Detection Methods in RAG https://cleanlab.ai/blog/rag-tlm-hallucination-benchmarking/

Intel (2023). Edge AI – Innovative capabilities at the edge. https://www.intel.com/content/www/us/en/learn/edge-ai.html

Kreuzberger D., Kühl N. & Hirschl S. (2023), "Machine Learning Operations (MLOps): Overview, Definition, and Architecture,". In *IEEE Access*, vol. 11, pp. 31866-31879, https://doi.org/10.1109/ACCESS.2023.3262138.

Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. In *Machine Learning With Applications*, 12, 100470. https://doi.org/10.1016/j.mlwa.2023.100470

Langfuse. (2024). Observability for LLM-based applications. https://www.langfuse.com/docs/overview

Lee, D., & Tiwari, M. (2024). Prompt infection: LLM-to-LLM prompt injection within multi-agent systems. *arXiv preprint arXiv:2410.07283*. https://arxiv.org/abs/2410.07283

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in neural information processing systems*, *33*, 9459-9474. https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

Liang, C. J., Le, T. H., Ham, Y., Mantha, B. R., Cheng, M. H., & Lin, J. J. (2024). Ethics of artificial intelligence and robotics in the architecture, engineering, and construction industry. In *Automation in Construction*, 162, 105369. https://doi.org/10.1016/j.autcon.2024.105369

Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., Liu, Y., Wang, H., Zheng, Y., & Liu, Y. (2023). Prompt injection attack against LLM-integrated applications. *arXiv preprint arXiv:2306.05499*. https://arxiv.org/abs/2306.05499

Liu, G., & Solomon, S. (2025). AI agent observability – Evolving standards and best practices. https://opentelemetry.io/blog/2025/ai-agent-observability/

MIT (2024). MIT Researchers Create an AI Risk Repository, https://ide.mit.edu/insights/mit-researchers-create-an-open-ai-risk-repository/

Market.us (2025). Edge AI for Cybersecurity Market Tech Growth at USD 643.2Bn. https://market.us/report/edge-ai-for-cybersecurity-market/

Martyr, R. (2025). Understanding Model Drift and Data Drift in LLMs https://orq.ai/blog/model-vs-data-drift

Miernik, M. (2024, October 12). The impact of AI chatbots on Core Web Vitals. https://www.reffine.com/en/blog/the-impact-of-ai-chatbots-on-core-web-vitals

Mollick, E. (2024) Co-intelligence: living and working with AI. New York: Penguin Publishing Group.

Mylläri, J., Aalto, T. & Nurminen, J.K. (2025) Ladle: a method for unsupervised anomaly detection across log types. In *Automated Software Engineering* 32, 34. https://doi-org.libproxy.helsinki.fi/10.1007/s10515-025-00504-w

Mäntymäki, M., Minkkinen, M., Birkstedt, T., & Viljanen, M. (2022). Putting AI ethics into practice: The hourglass model of organizational AI governance. *arXiv preprint arXiv:2206.00335*. https://arxiv.org/abs/2206.00335

Paka, A (2023). How to Monitor LLMOps Performance with Drift Monitoring https://www.fiddler.ai/blog/how-to-monitor-llmops-performance-with-drift

Prem, E. (2023). From ethical AI frameworks to tools: a review of approaches. In *AI Ethics* 3, 699–716. https://doi.org/10.1007/s43681-023-00258-9

Rahmati, M. (2025). Towards Explainable and Lightweight AI for Real-Time Cyber Threat Hunting in Edge Networks. *arXiv preprint arXiv:2504.16118*. https://arxiv.org/abs/2504.16118

Raj, H., Gupta, V., Rosati, D., & Majumdar, S. (2023). Semantic consistency for assuring reliability of large language models. *arXiv preprint arXiv:2308.09138*. https://arxiv.org/abs/2308.09138

Ramachandran, A. (2024). A survey of agentic AI, multi-agent systems, and multimodal frameworks: Architectures, applications, and future directions. ResearchGate. https://www.researchgate.net/publication/387577302_A_Survey_of_Agentic_AI_Multi-Agent_Systems_and_Multimodal_Frameworks_Architectures_Applications_and_Future_Directions

Traceloop (2025). OpenLLMetry [Software]. GitHub. https://github.com/traceloop/openllmetry

Weidinger, L., Raji, I. D., Wallach, H., Mitchell, M., Wang, A., Salaudeen, O., Bommasani, R., Ganguli, D., Koyejo, S. & Isaac, W. (2025). Toward an evaluation science for generative AI systems. *arXiv preprint arXiv:2503.05336*. https://arxiv.org/abs/2503.05336

Weisz, J. D., Muller, M., He, J., & Houde, S. (2023). Toward general design principles for generative AI applications. *arXiv preprint arXiv:2301.05578*. https://arxiv.org/abs/2301.05578

White house (2023). FACT SHEET: President Biden Issues Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence. https://bidenwhitehouse.archives.gov/briefing-room/statements-releases/2023/10/30/fact-sheet-president-biden-issues-executive-order-on-safe-secure-and-trustworthy-artificial-intelligence/

Wu, D., Gu, J. C., Yin, F., Peng, N., & Chang, K. W. (2024). Synchronous faithfulness monitoring for trustworthy retrieval-augmented generation. *arXiv preprint arXiv:2406.13692*. https://arxiv.org/abs/2406.13692

Zamora, F. (2025). Status Update: Is AI Breaking Down or Creating Barriers in AEC?. In *AI Architecture*, https://architizer.com/blog/inspiration/industry/status-update-is-ai-breaking-down-or-creating-barriers-in-aec/

Zang, R., Guo, H., Yang, J., Liu, J., Li, Z., Zheng, T., Shi, X., Zheng, L., & Zhang, B. (2024). MLAD: A Unified Model for Multi-system Log Anomaly Detection. *arXiv preprint arXiv:2401.07655*. https://arxiv.org/abs/2401.07655

Zhang, S., Ye, L., Yi, X., Tang, J., Shui, B., Xing, H., Liu, P., & Li, H. (2024). “Ghost of the past”: Identifying and resolving privacy leakage from LLM’s memory through proactive user interaction. *arXiv preprint arXiv:2410.14931*. https://arxiv.org/abs/2410.14931

Zhou, C., Liu, X., Yu, C., Tao, Y., & Shao, Y. (2024). Trust in AI-augmented design: Applying structural equation modeling to AI-augmented design acceptance. In *Heliyon*, 10(1). https://doi.org/10.1016/j.heliyon.2023.e23305