

## Research Overview

- This work proposes a hierarchical multi-agent architecture for vision-based interaction in Mixed Reality (MR).
- The target system connects a Unity MR environment to an LLM-powered backend, utilizing a supervisor agent to route visual queries to specialized sub-agents.
- Deploying a full multi-agent team introduces latency challenges; therefore, the foundation must be optimized for real-time interaction.
- This prototype serves as a proof-of-concept, isolating and evaluating the core Vision Agent to determine the real-time viability of an LLM-grounded backend before broader multimodal integration.

## Background

### What is Multi-Agent?

- A multi-agent system divides a complex task across specialized agents instead of relying on a single monolithic model.
- This structure supports modularity by separating responsibilities such as coordination, reasoning, and response generation.

### Why Multi-Agent for MR?

- Recent work on proactive AR agents highlights the value of context-aware assistance that reduces interaction burden during everyday tasks.
- A multi-agent design provides a scalable way to organize these functions, allowing the system to easily expand as richer MR inputs and APIs are introduced.

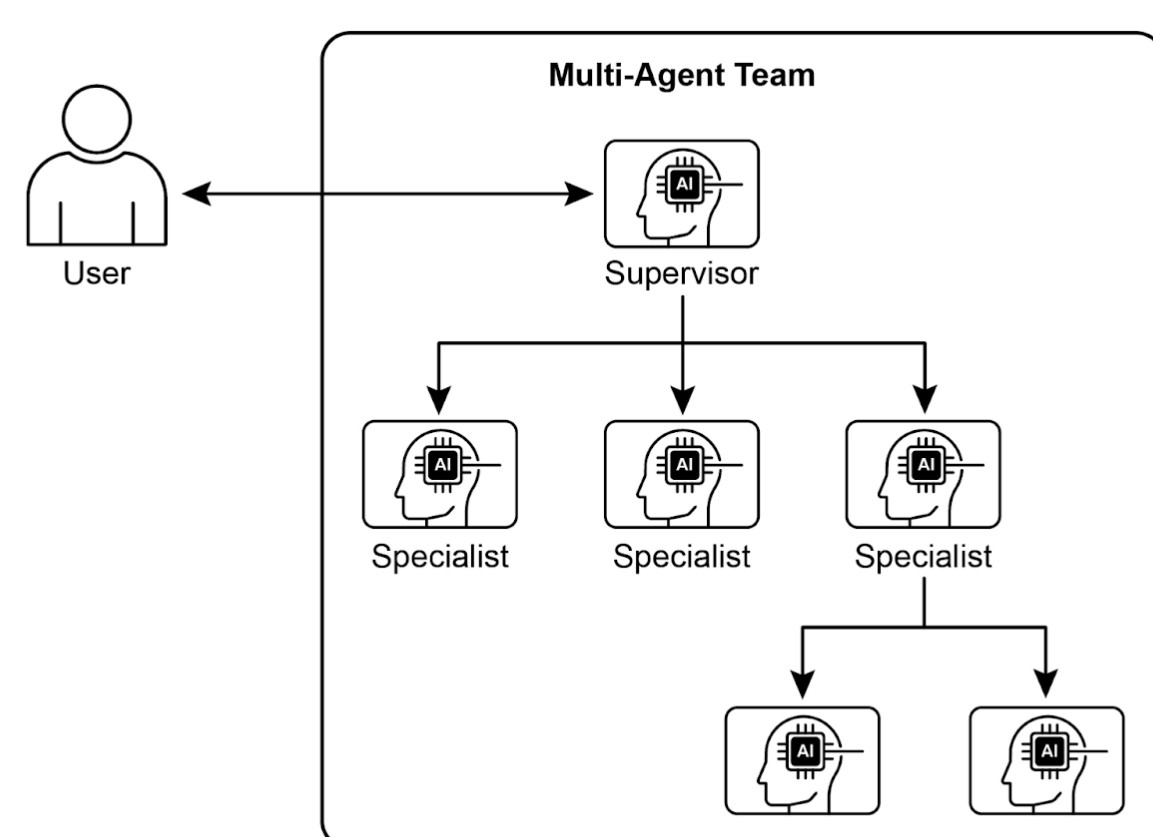


Figure 1. Illustrative example of Hierarchical multi-agent architecture for MR scene understanding

## System Architecture & Setup

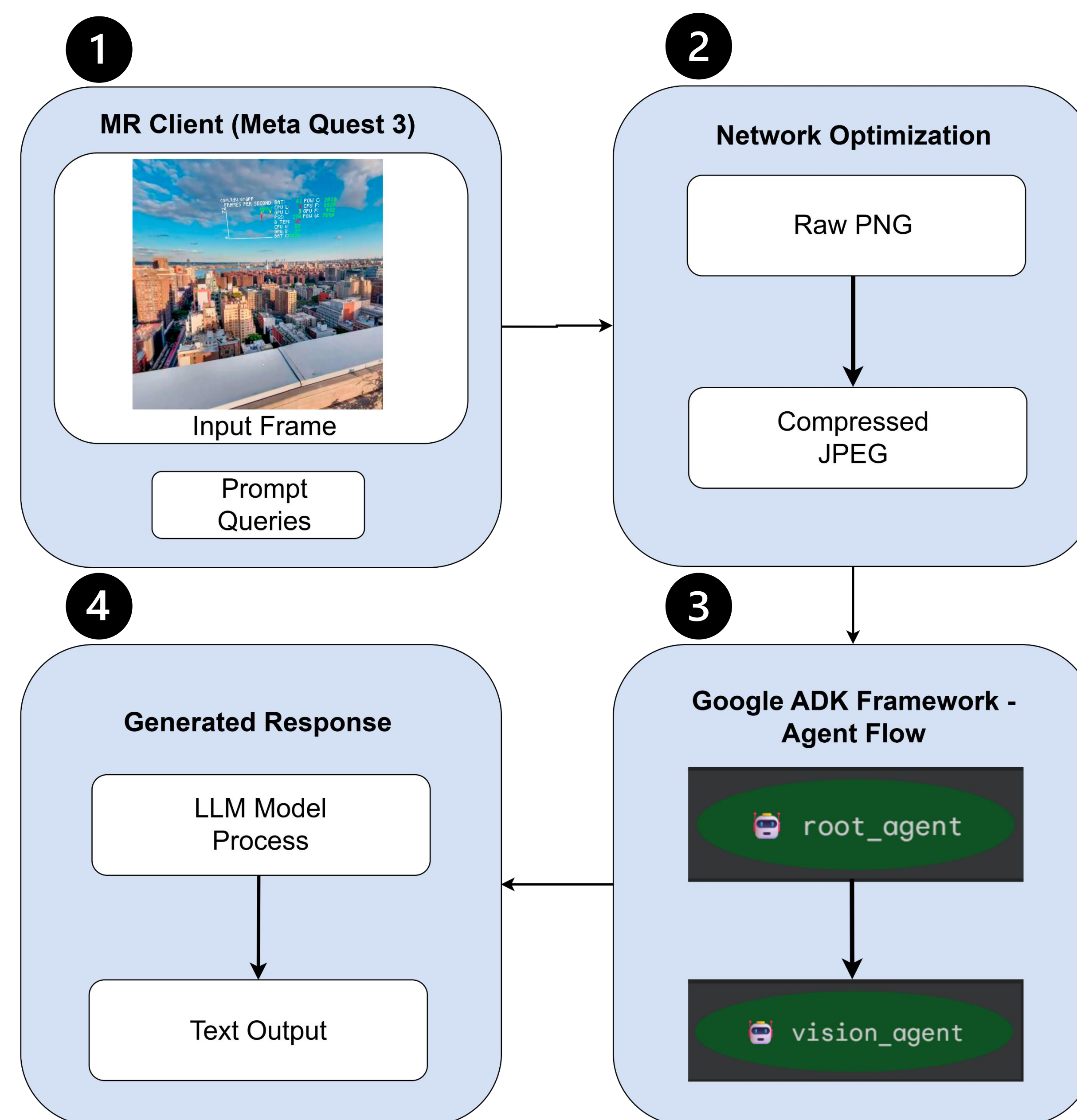
- The Additive Latency Problem:** Initial system testing revealed that sequential routing (User -> Supervisor -> Vision Agent) introduced unacceptable additive latency for Mixed Reality, effectively doubling response times.

Equation 1. Latency model used for OpenTelemetry-based tracing

$$T_{total} = T_{network} + \Delta t_{supervisor} + \Delta t_{vision} + T_{LLM}$$

- Phase 1 Isolation Strategy:** To build a viable hierarchical system, the heaviest computational node—the Vision Agent—must first be optimized. This experimental setup temporarily bypasses the supervisor to evaluate the visual-processing path in isolation.
- Benchmarking Pipeline:** Local headset frames are loaded from disk, dynamically compressed from PNG to JPEG to minimize network payload, and processed directly by the Vision Agent via a Google ADK runner. The core vision model evaluated in this benchmarking pipeline is Gemini 2.5 Flash.

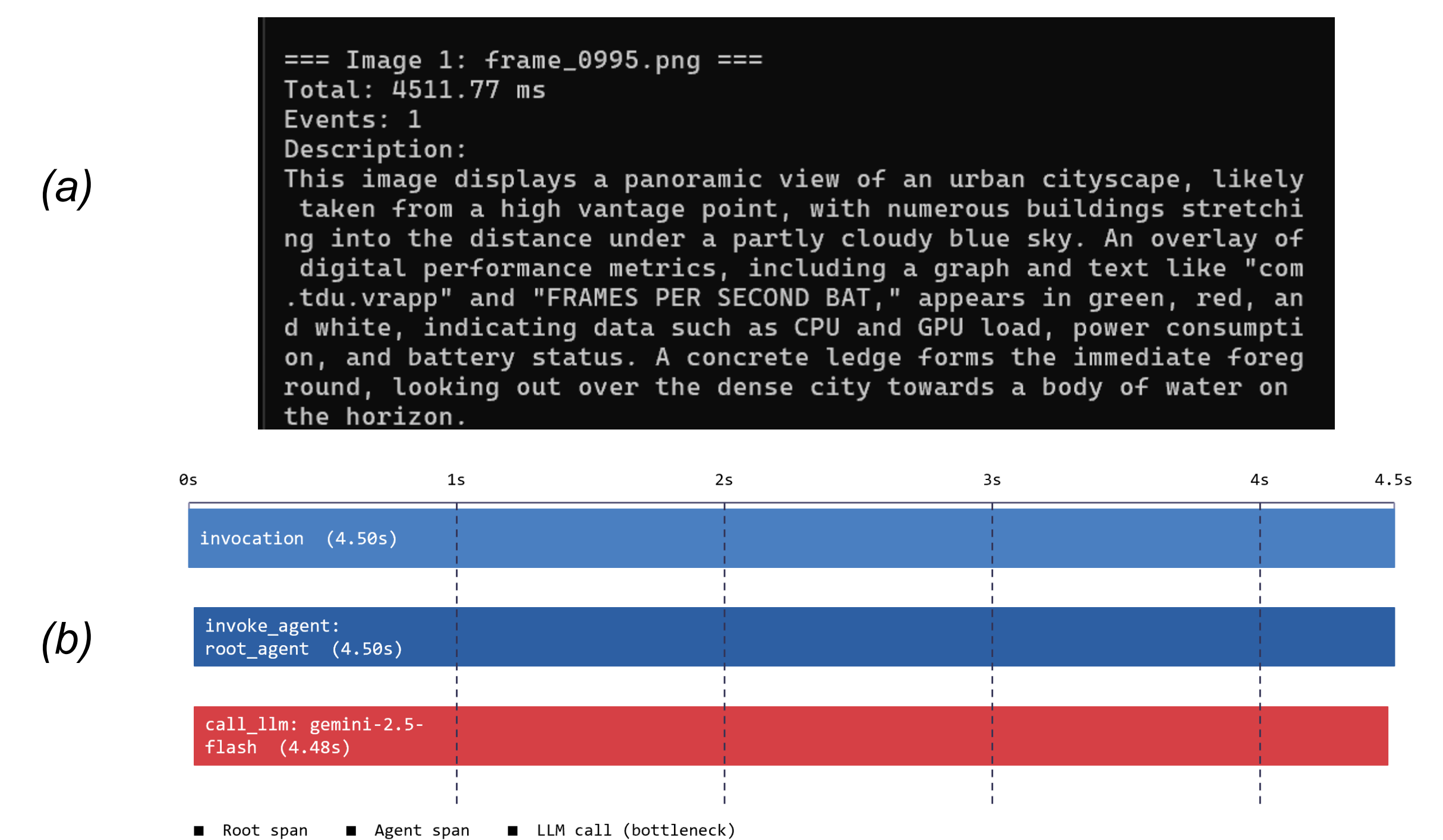
Figure 2. Visual-Processing Benchmarking Pipeline



## Preliminary Results & Optimization

- The Reasoning Bottleneck:** Benchmarking the core vision model (Gemini 2.5 Flash) revealed highly unpredictable total response times. Across multiple isolated test runs on the exact same frame, latency fluctuated significantly, ranging from roughly 4.4 to 7.2 seconds.
- Latency Variance:** OpenTelemetry traces confirmed this variance is directly tied to the model's internal chain-of-thought reasoning phase. For an identical input image, the model's thoughts\_token\_count swung wildly from 183 to 570 tokens per run, proving that current reasoning models introduce unpredictable cognitive overhead that breaks real-time MR interaction limits.

Figure 3. (a) System Benchmarking: Representative execution log illustrating total response cycle for one frame b) Distributed Trace Visualization



## Future Work

- Telemetry Granularity:** Server-side tracing is needed to separately measure network latency and pure LLM inference time.
- Model Optimization:** The critical next step is migrating the Vision Agent to a non-reasoning, lightweight model to eliminate cognitive overhead and force latency under a stable 2-second threshold.
- Live Unity Integration:** Transitioning the pipeline from evaluating local disk frames to processing live camera streams directly from the Meta Quest 3 headset.

## References

- A. Gullí, *Agentic Design Patterns: A Hands-On Guide to Building Intelligent Systems*. Cham, Switzerland: Springer Cham, 2025, ch. 7, "Multi-Agent Collaboration," pp. 97–113, doi: 10.1007/978-3-032-01402-3.
- G. Lee *et al.*, "Sensible Agent: A Framework for Unobtrusive Interaction with Proactive AR Agents," in *Proc. 38th Annu. ACM Symp. User Interface Software and Technology (UIST '25)*, 2025, doi: 10.1145/3746059.3747748.
- Google Cloud, "Overview of Agent Development Kit," *Vertex AI Agent Builder Documentation*, Apr. 2, 2026.