Agentathon 2026

-Developer Productivity and Tooling-

--TraceOps--

Suhas Venkata Karamalaputti | Soumya Ranjan Mishra | Shreyas Suresh

–PES University–

PROBLEM STATEMENT

There is no system today that can automatically investigate software incidents and reason about root causes and fixes.

Modern software systems fail frequently — due to configuration errors, faulty deployments, and resource exhaustion.

When incidents occur, engineers are forced to manually:

- Dig through massive volumes of logs
- Correlate behavior across services and configurations
- Guess possible root causes
- Write incident reports after the fact

This workflow is slow, error-prone, and heavily experience-dependent — leading to prolonged outages and delayed recovery.

Existing monitoring tools show alerts and dashboards — they do not perform reasoning.

EXISTING SOLUTIONS

Current tools such as monitoring dashboards, log aggregators, and alerting systems provide visibility — but not intelligence.

They can tell you that something is broken, but not why it broke or what to do next.

Metrics platforms (e.g., Prometheus, Grafana) show system health — not root causes

Log tools surface errors — but require manual investigation

On-call engineers must still correlate data across services and configurations

There is no built-in reasoning or fix planning

There is no system today that performs automated incident reasoning and fix recommendation.

OUR APPROACH

We propose **TraceOps** — a multi-agent incident investigation system that automates first-level incident triage by combining agentic reasoning with grounded system access using open-source **MCP Filesystem Server.**

Multi-Agent Workflow:

-> Ingestion Agent — Extracts failure signals from logs

-> Context Agent (MCP-grounded) — Fetches configs and repository files via MCP

-> Root-Cause Agent — Generates and ranks root-cause hypotheses

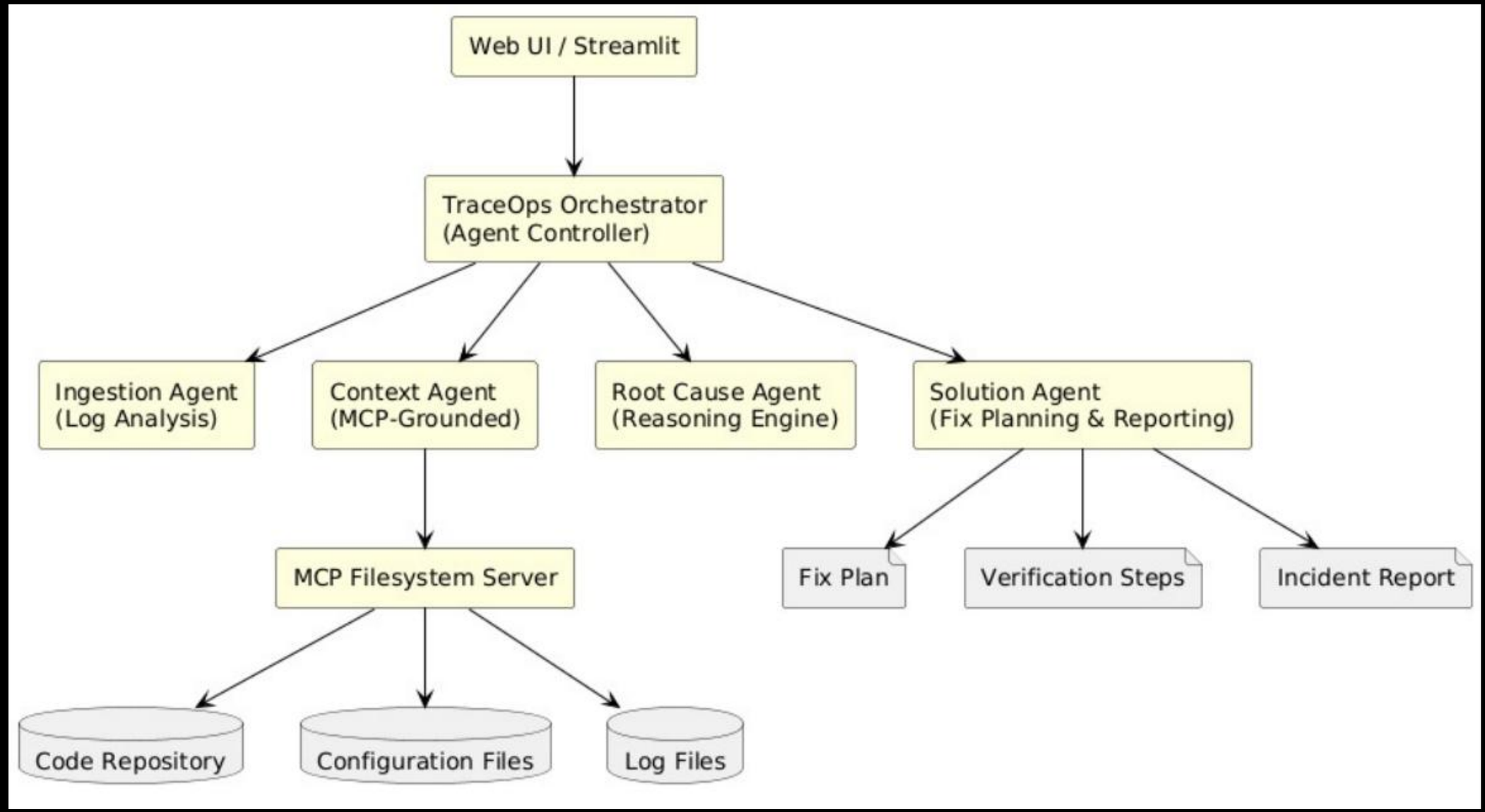
-> Solution Agent — Produces fix plans, verification steps, and incident summaries

MCP Integration:

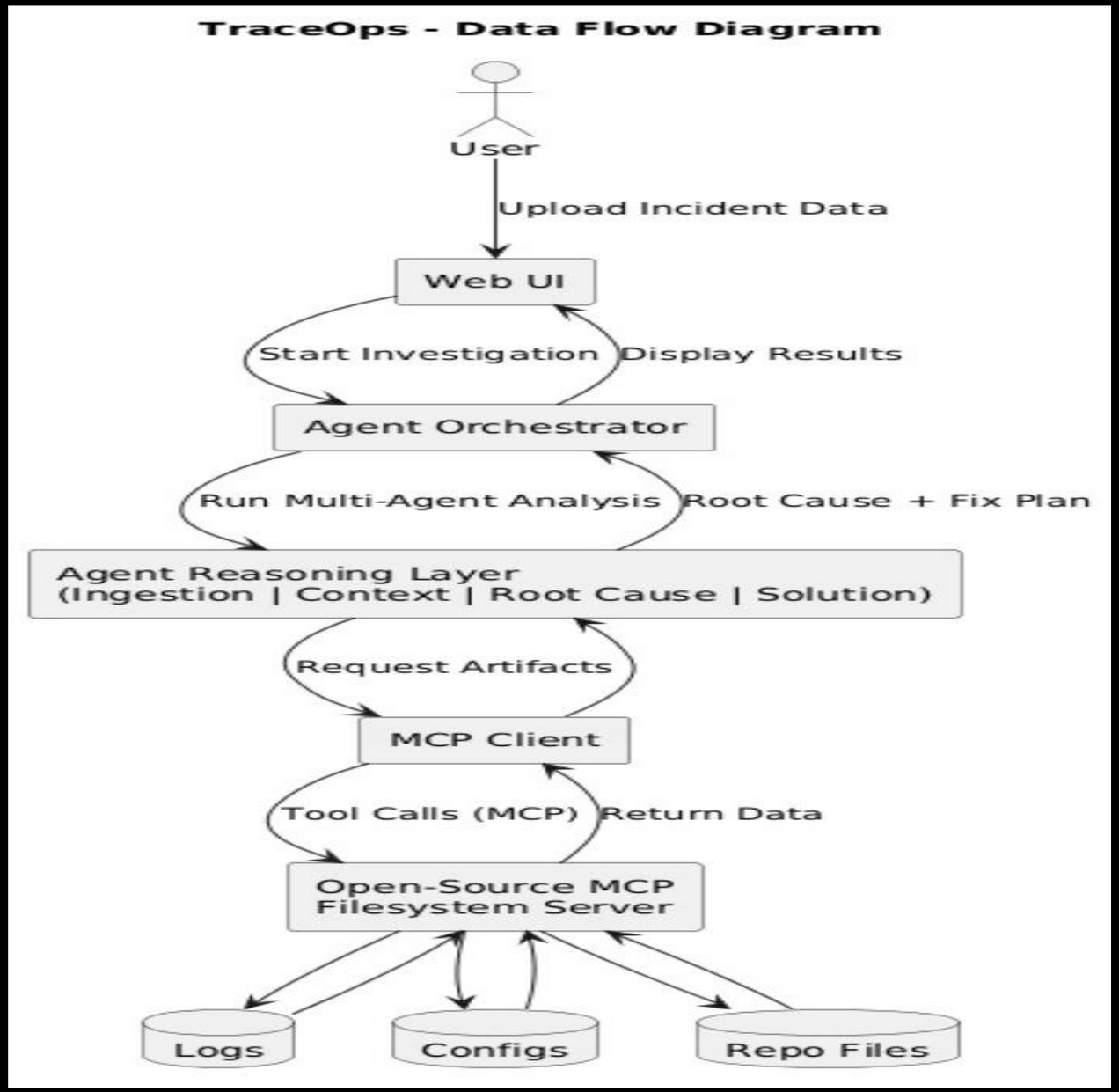
All system artifacts are accessed through an open-source MCP Filesystem Server, enabling tool-grounded reasoning and preventing hallucinated system state.

This decouples reasoning from system access and reflects a scalable, enterprise-style agent architecture.

OUR APPROACH



DATA FLOW DIAGRAM



Encode.ai

TECH STACK

PES UNIVERSITY

Application
- Python
- Streamlit (Web UI)

Agent System
- LLM-powered multi-agent pipeline
- Custom agent orchestrator
- Structured reasoning & tool calls

MCP Integration
- Open-source MCP Filesystem Server
- MCP client for grounded system access

Data Sources
- Code repositories
- Configuration files
- Incident logs

Deployment
- Local runtime (extensible to cloud)

FEASIBILITY

1. Technically achievable in hackathon scope
- Modular, pipeline-based multi-agent design
- Small, realistic system surface (logs, configs, repo)
- No dependency on live production infrastructure
- MCP Filesystem Server enables immediate grounding

2. Proven building blocks
- LLMs already perform log analysis and code understanding well
- MCP provides a standardized tool interface for system access
- Agent orchestration is deterministic and debuggable

3. Demo-ready by design
- Controlled incident scenarios
- Real codebase and log bundles
- Clear success criteria (root cause + fix plan)

4. Extensible beyond the hackathon
- Same agents can plug into GitHub, CI/CD, and log platforms via MCP
- Architecture supports deeper reasoning, monitoring, and remediation

The system is realistic, deployable, and demonstrates practical agentic reasoning.

RESEARCH AND REFERENCES

1.Model Context Protocol & Specifications

- Model Context Protocol Official Documentation(https://modelcontextprotocol.io/docs/) – Comprehensive guides on MCP architecture, transport layers, and core primitives for protocol implementation.
- MCP GitHub Repository(https://github.com/modelcontextprotocol/modelcontextprotocol) – Official specification, SDKs, and reference implementations.

2.MCP Server Implementations

- mark3labs/mcp-filesystem-server(https://github.com/mark3labs/mcp-filesystem-server) – Go-based MCP filesystem server for agent-driven file operations.
- cyanheads/filesystem-mcp-server(https://github.com/cyanheads/filesystem-mcp-server) – TypeScript-based filesystem MCP server with advanced search/replace and directory traversal.

3.Multi-Agent LLM Systems for Incident Response

- Multi-Agent LLM Orchestration for Incident Response(https://arxiv.org/html/2511.15755v2) – Demonstrates that multi-agent orchestration achieves 100% actionable recommendations vs. 1.7% for single-agent systems in incident diagnosis (348 controlled trials).
- Exploring LLM-based Agents for Root Cause Analysis(https://arxiv.org/html/2403.04123v1) – Research on LLM agents with tool access for real-world incident root cause analysis in operations contexts.
- A Survey of AIOps in the Era of Large Language Models(https://arxiv.org/html/2507.12472v1) – Comprehensive survey of LLM applications in AIOps, covering anomaly detection, RCA, and mitigation solution generation.

AI tools were used for ideation, architecture design, and draft generation.

Encode.ai

PES UNIVERSITY

Agentathon 2026

Thank You