

MCP × A2A Platform for the Enterprise

~Security by Spec~

Masaki TSUKADA
Mitsubishi Electric Corporation

Introduction

Masaki TSUKADA

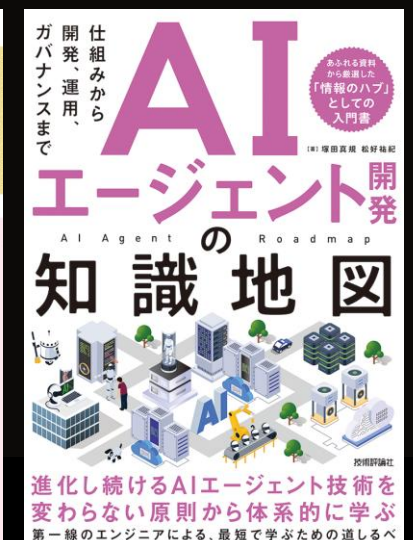
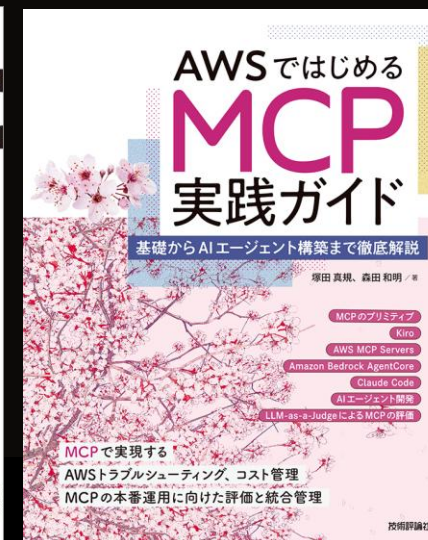
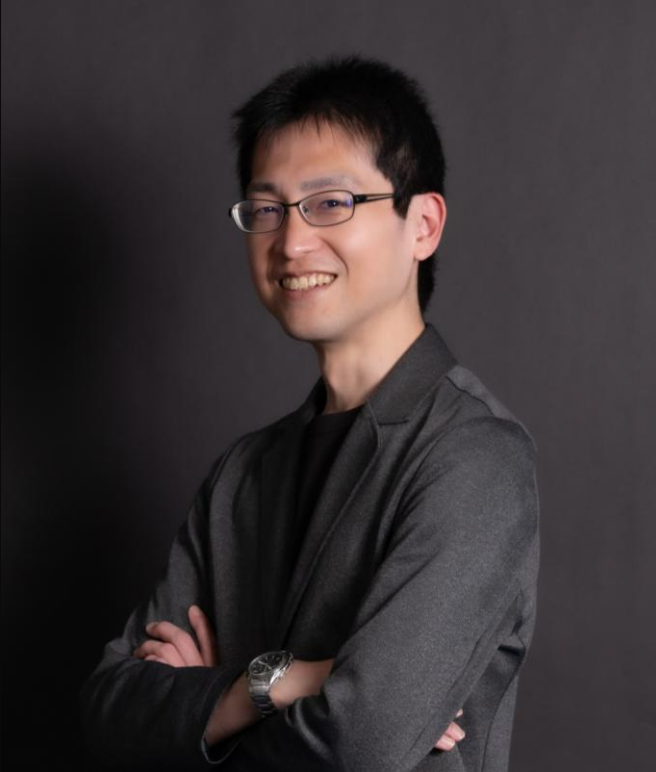
Mitsubishi Electric Corporation

Digital Innovation Group

AI Transformation Innovation Center

Role:

Technical Lead for the Enterprise AI Agent Platform



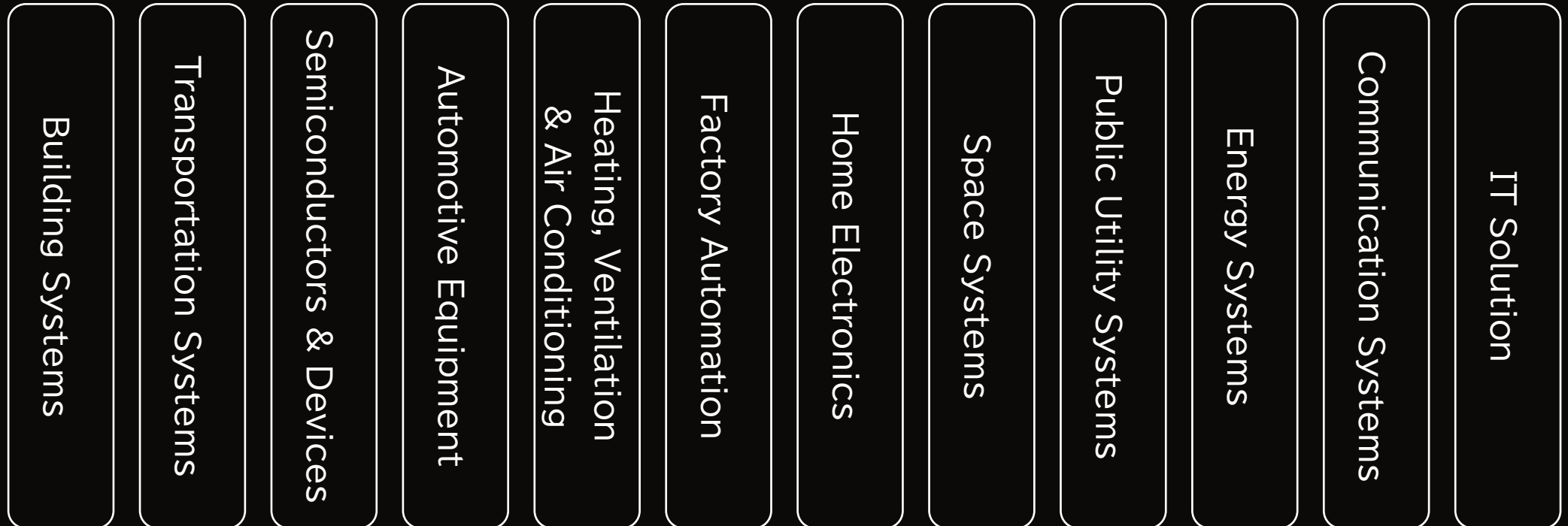
Introducing Mitsubishi Electric Corporation

Founded : Jan. 15, 1921

Consolidated Employees : 150,386

Revenue : ¥5,894.7 billion

12 Business Domains



Introducing Mitsubishi Electric Corporation

Founded : Jan. 15, 1921

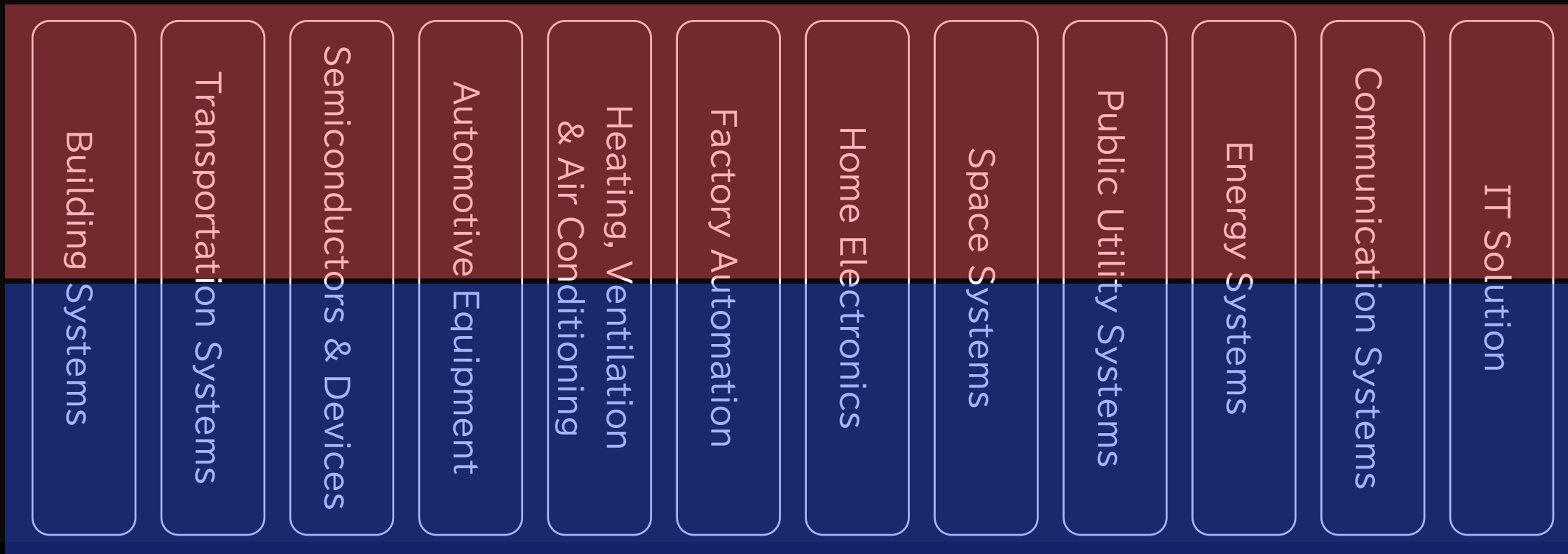
Consolidated Employees : 150,386

Revenue : ¥5,894.7 billion

12 Business Domains

For
Internal
operations

For
Business



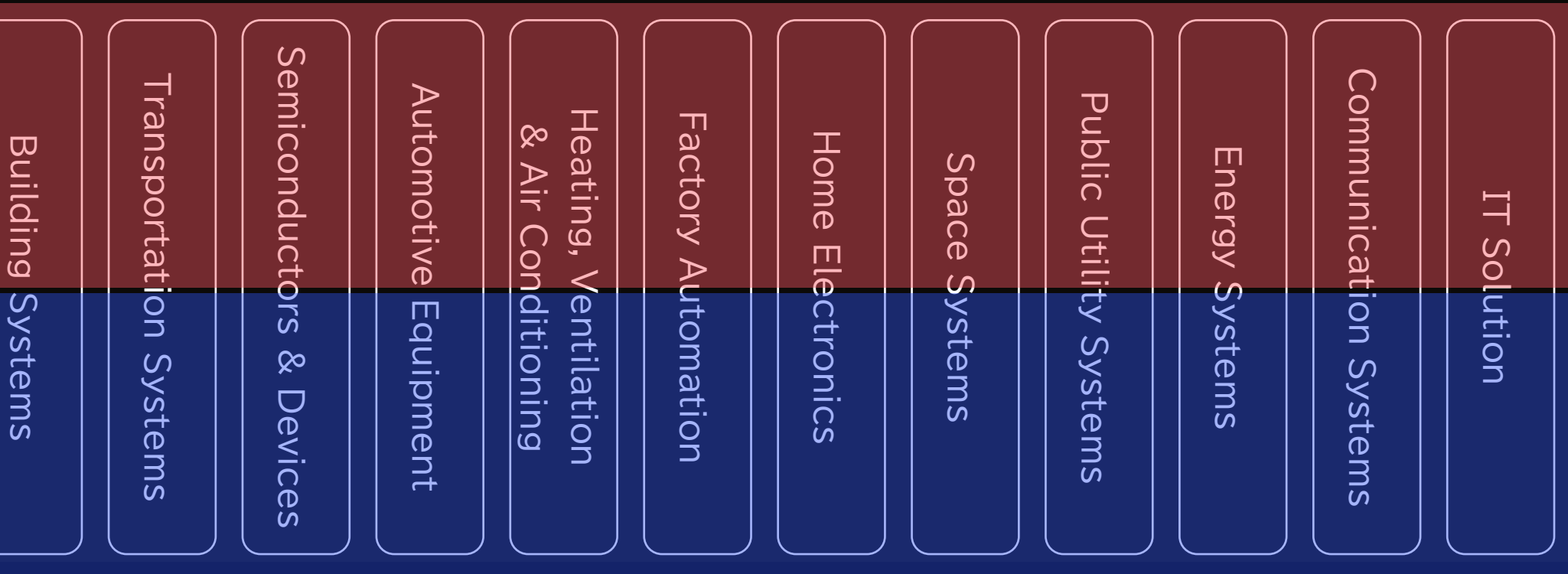
Introducing Mitsubishi Electric Corporation

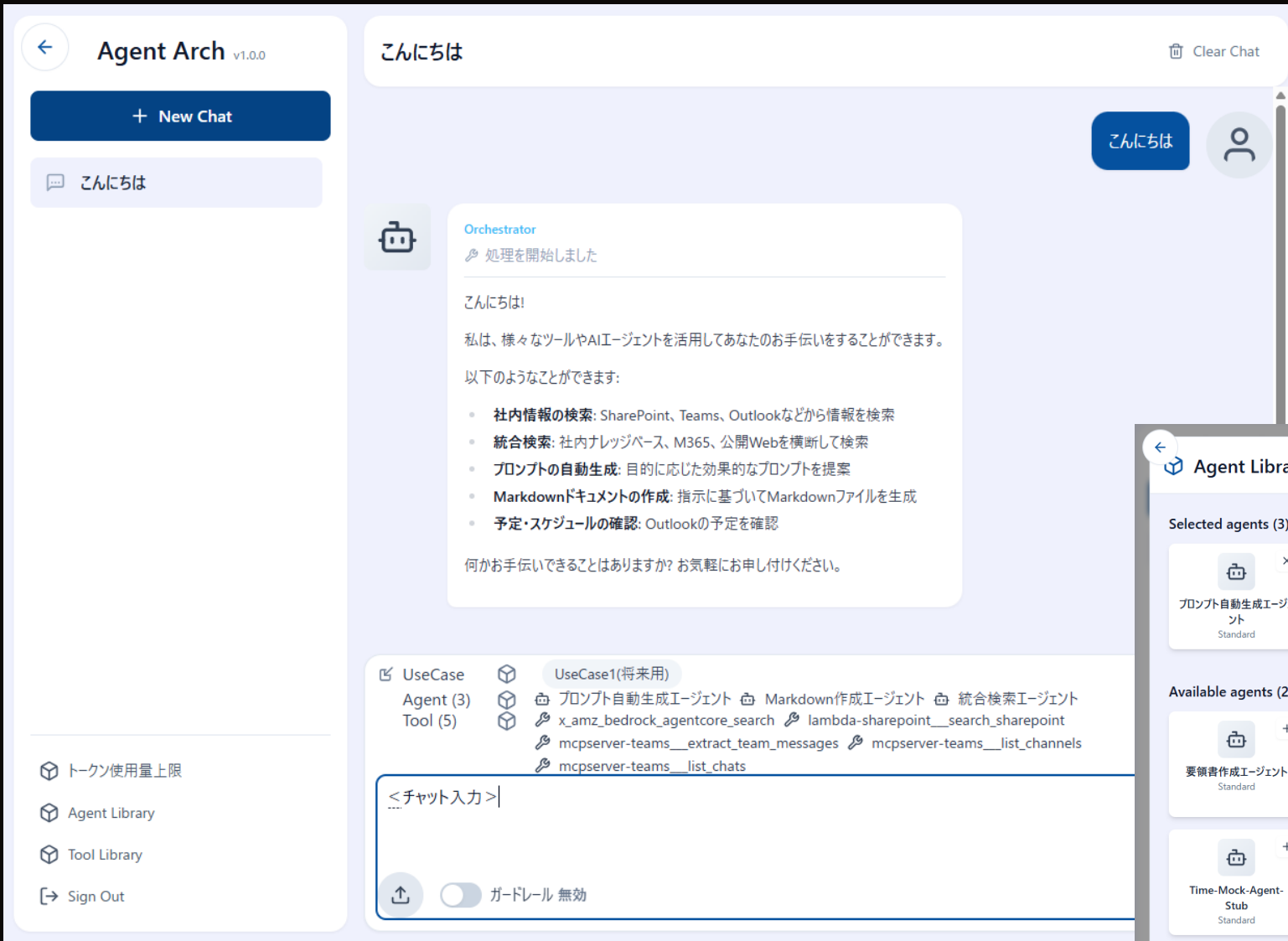
In internal operations,
the needs are common across business domains.

One Enterprise-wide Multi-Agent Platform

For
Internal
operations

For
Business

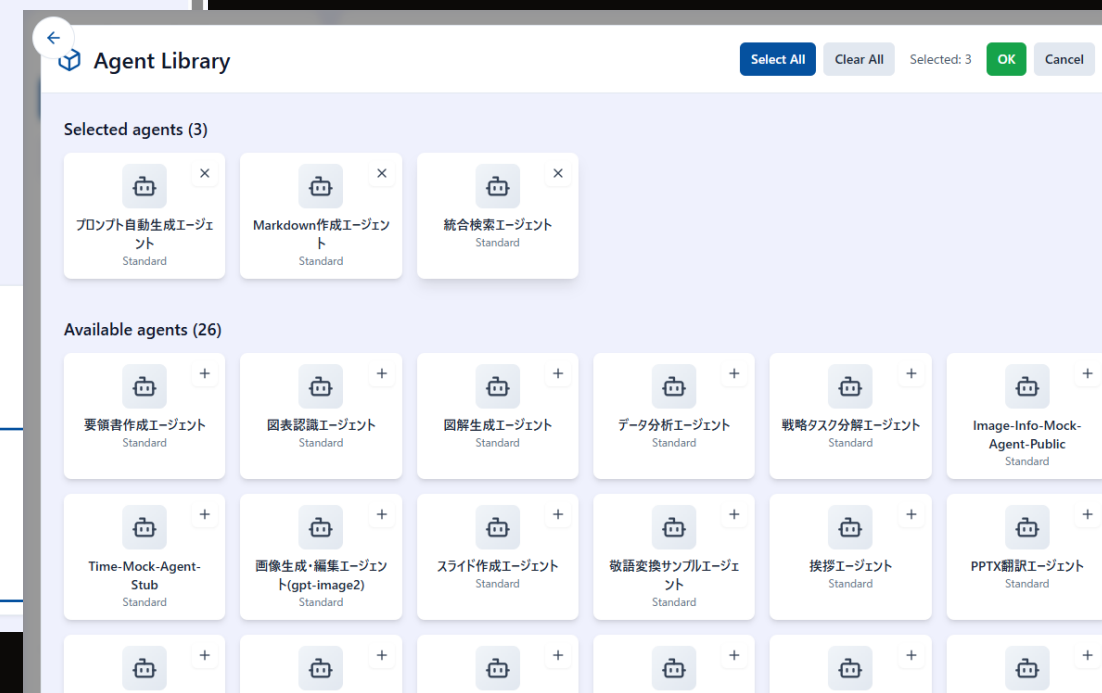




Chat Screen

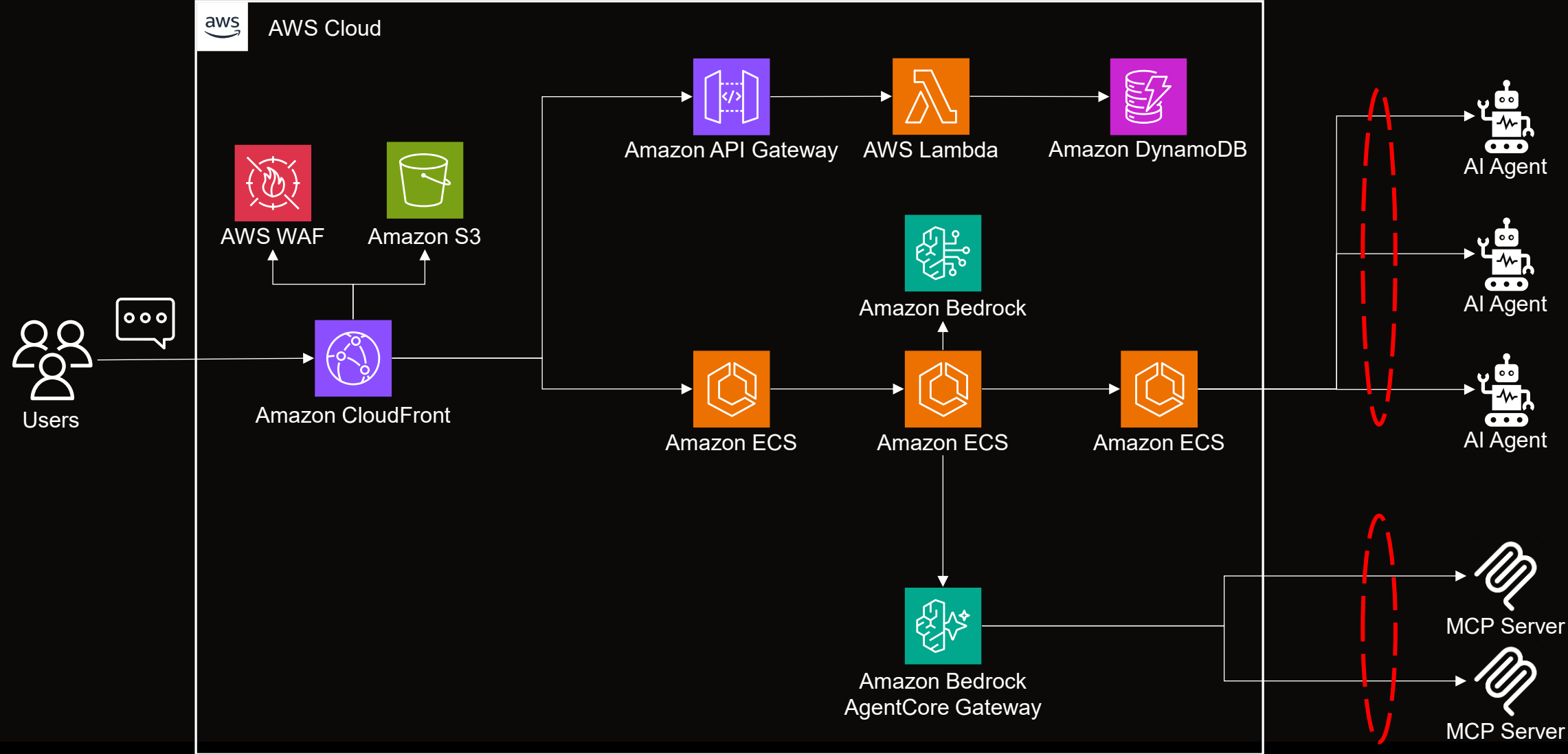


Token Usage Dialog

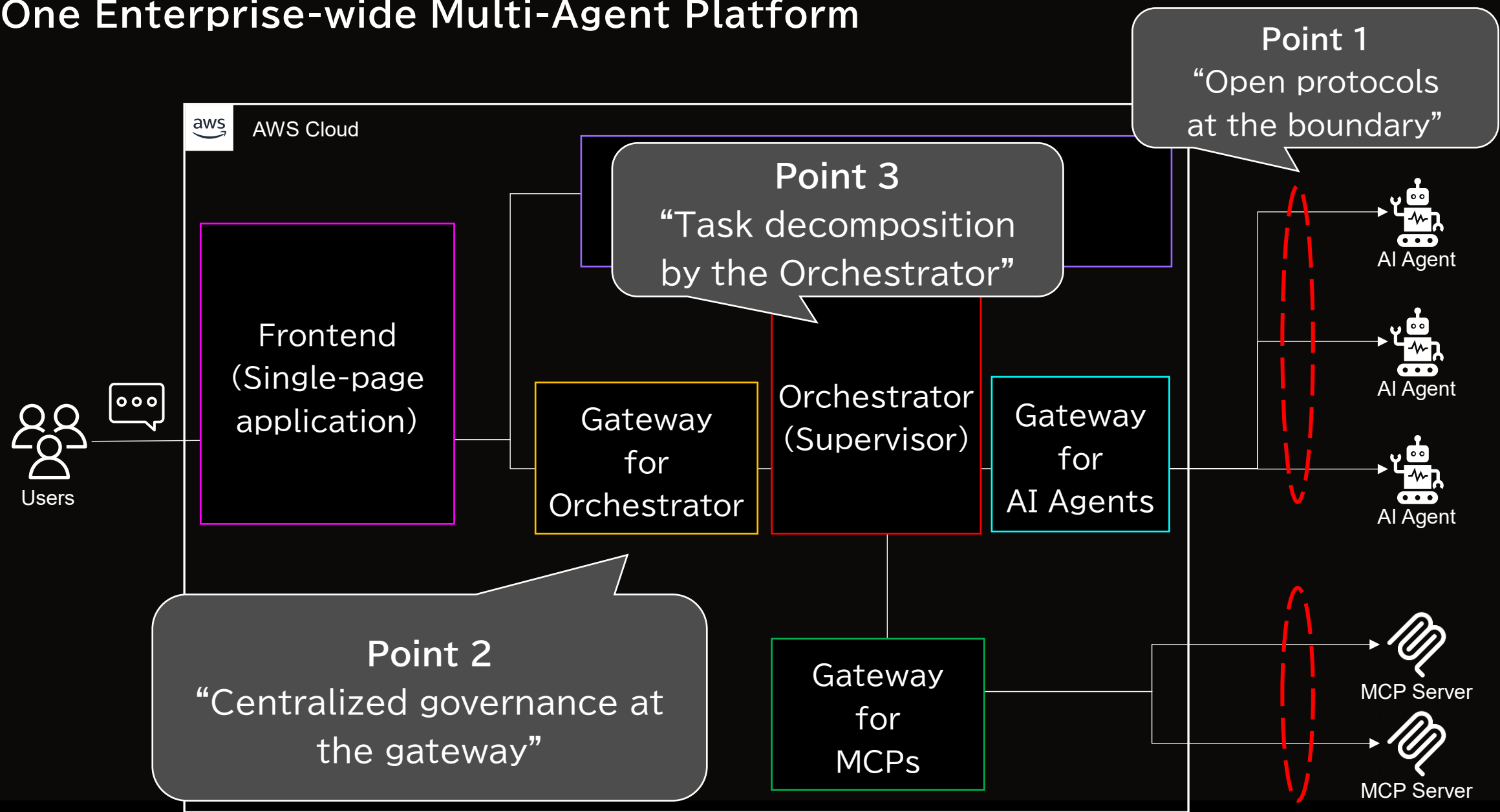


AI Agent Selection Dialog

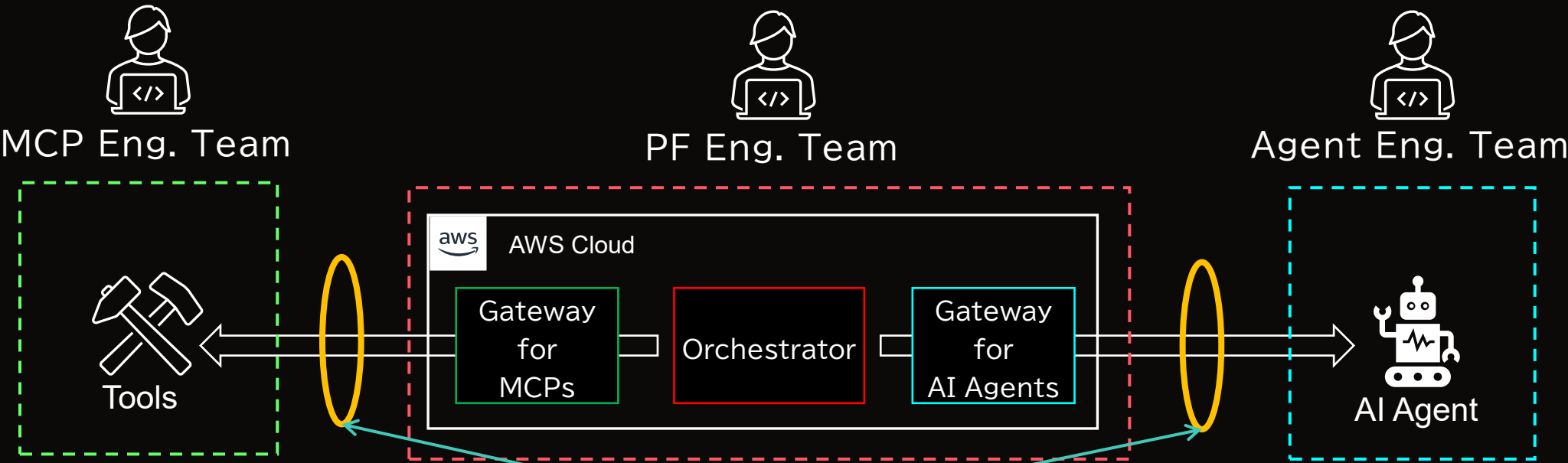
One Enterprise-wide Multi-Agent Platform



One Enterprise-wide Multi-Agent Platform



Point 1 “Open protocols at the boundary”



Teams must align on how modules connect

Standardize cross-team interfaces

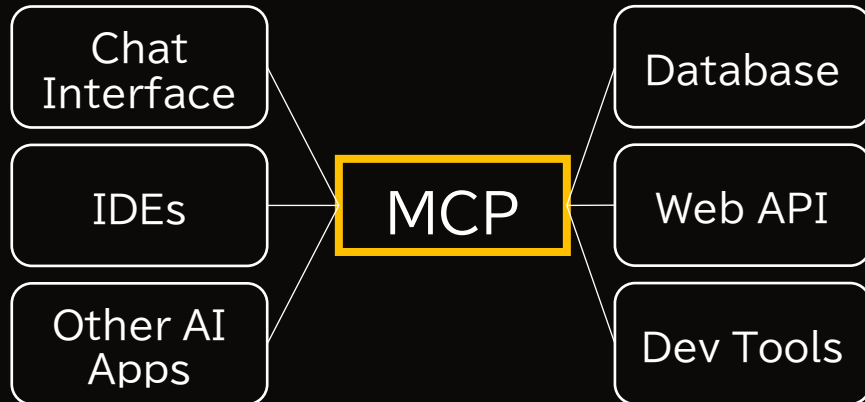


Point 1 “Open protocols at the boundary”

MCP (Model Context Protocol)

Like a USB-C port for AI applications

- open standard protocol
- connecting AI applications to external systems

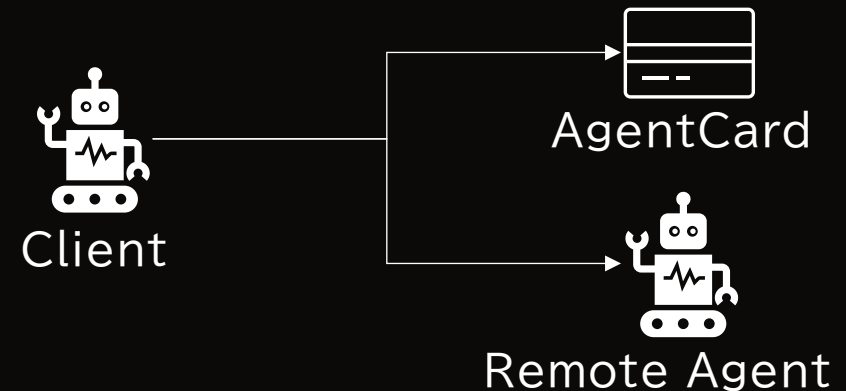


Build once,
reuse across AI applications

A2A (Agent2Agent)

A Common Language
for Agent Interoperability

- open standard protocol
- communication and collaboration between AI agents

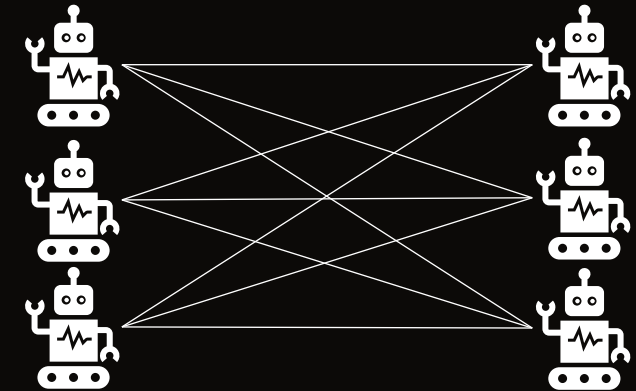


Build independently,
connect across agents

Point 2 “Centralized governance at the gateway”

Governance required at enterprise scale

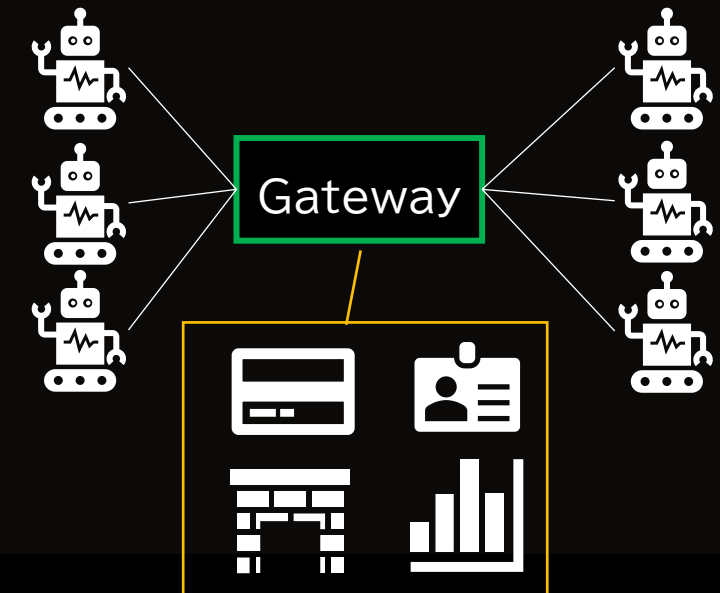
- a. Separate authentication across services
- b. Complex permission management across agents
- c. Limited visibility into per-user token usage
- d. Inconsistent guardrails across applications



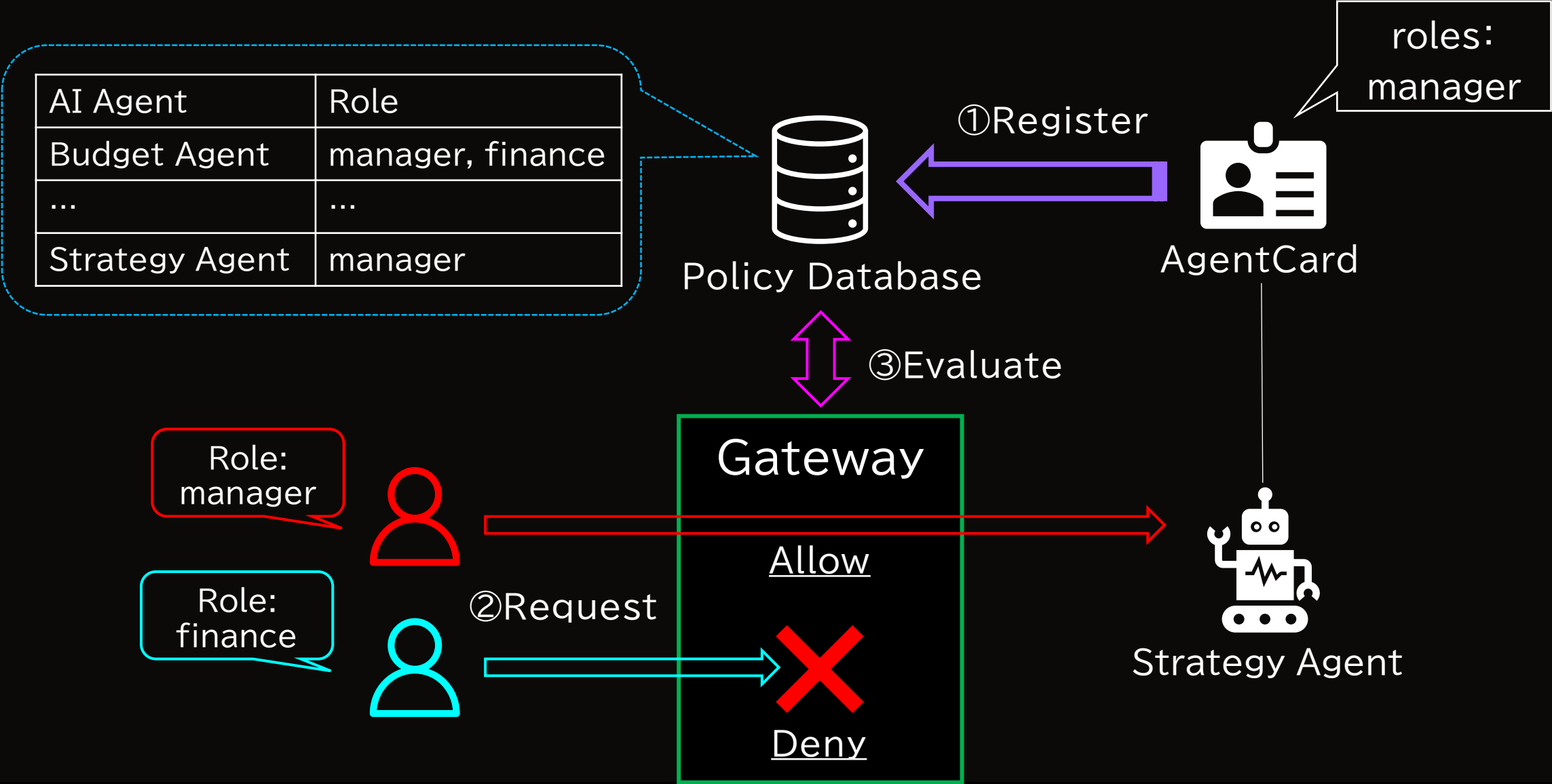
Our Approach



- a. Integration with the enterprise authentication platform
- b. AgentCard-Based Authorization
- c. Per-user token usage tracking and control
- d. Centralized guardrail enforcement



AgentCard-Based Authorization



Point 3 “Task decomposition by the Orchestrator”

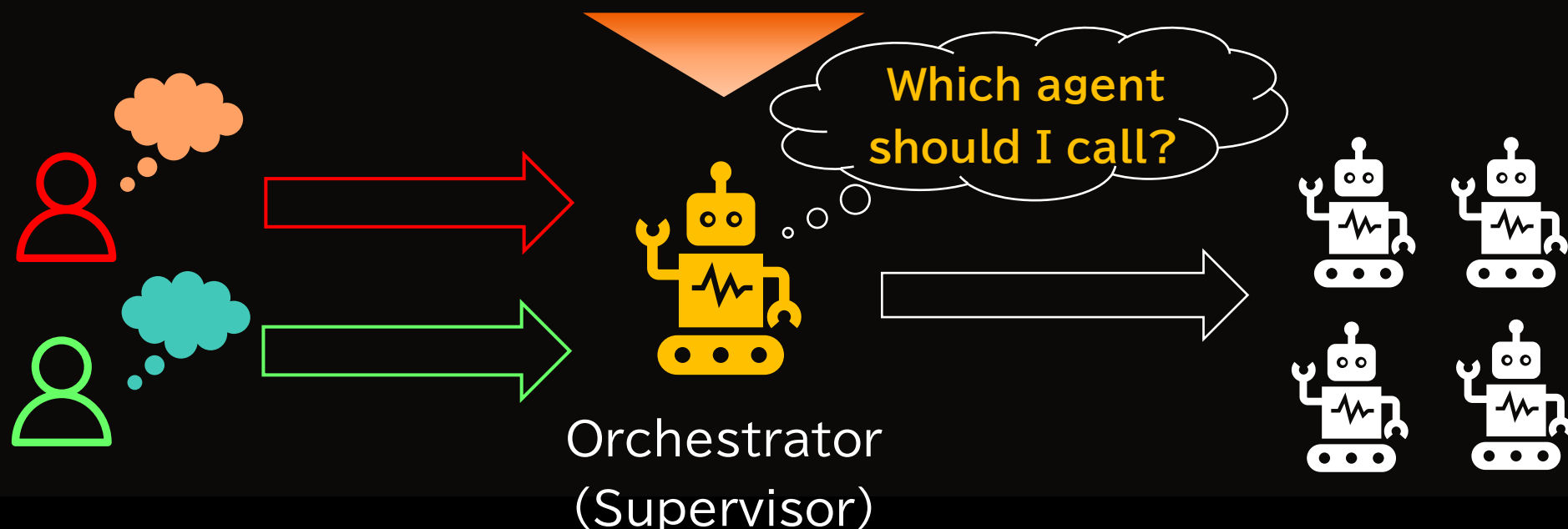
Users across the company have diverse needs



Problem

- The required agents vary by request
- Complex requests require multi-turn coordination across agents

Our Approach





What we built

Enterprise-wide Multi-Agent Platform

1. Open protocols at the boundary
2. Gateways for Governance
3. Orchestrator for diverse requests



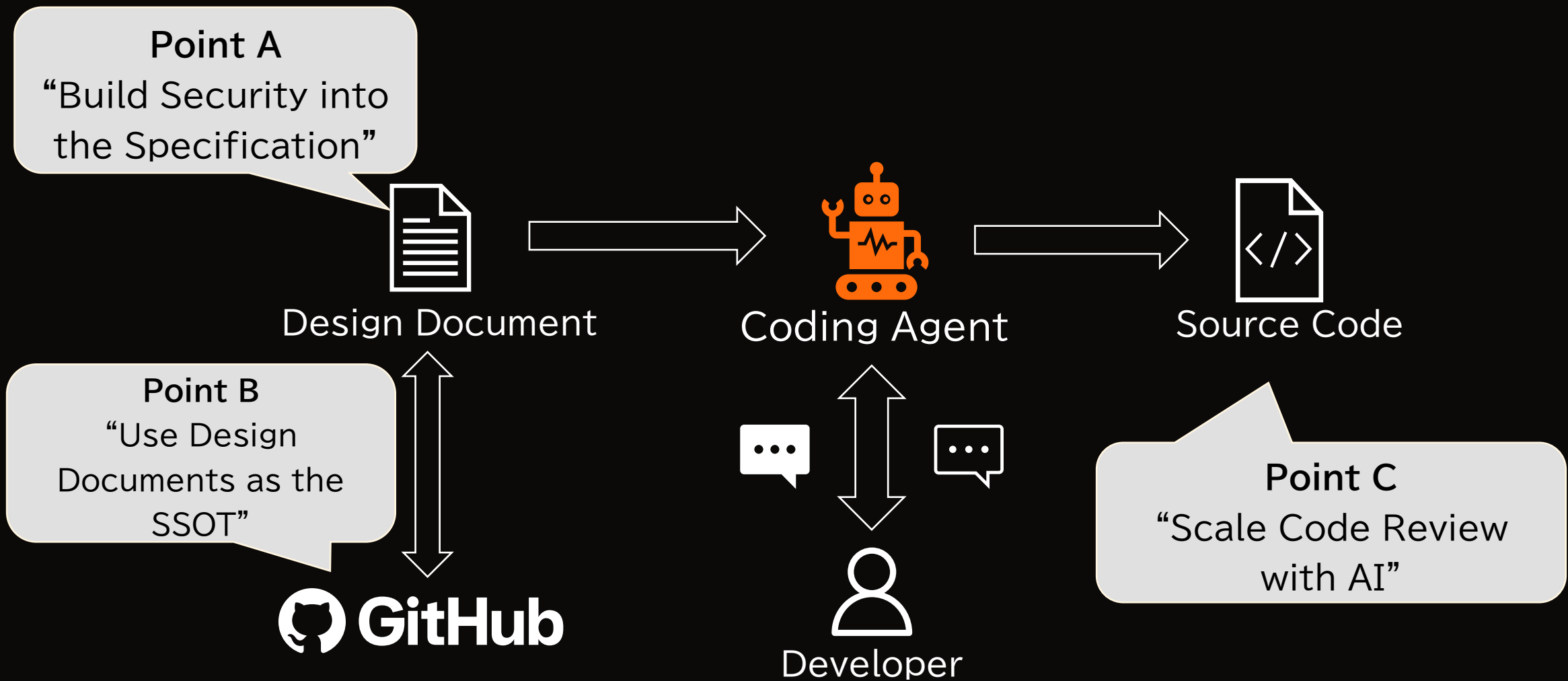
How we built it

Developing With AI

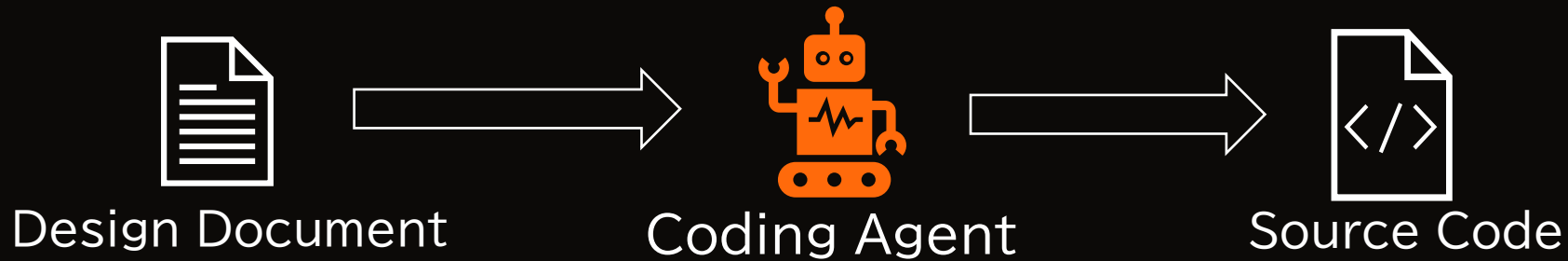
- A) Build Security into the Specification
- B) Use Design Documents as the SSOT
- C) Scale Code Review with AI

How we built it

Development Led by Coding Agents



Point A “Build Security into the Specification”



Clearly Defined
Requirements



Implemented as intended

Undefined
Requirements



May be omitted
or
implemented incorrectly



A well-defined specification is essential.
Security is no exception.

Point A “Build Security into the Specification”

What to Secure & How to Implement it

- OWASP Top 10 for LLM Applications
- OWASP Top 10 for Agentic Applications
- Internal Security Policies, etc



LLM



Design Document



Following the guidelines?
Any gaps or omissions?

Check for weaknesses from an attacker's perspective

- MITRE ATT&CK Enterprise
- MITRE ATLAS

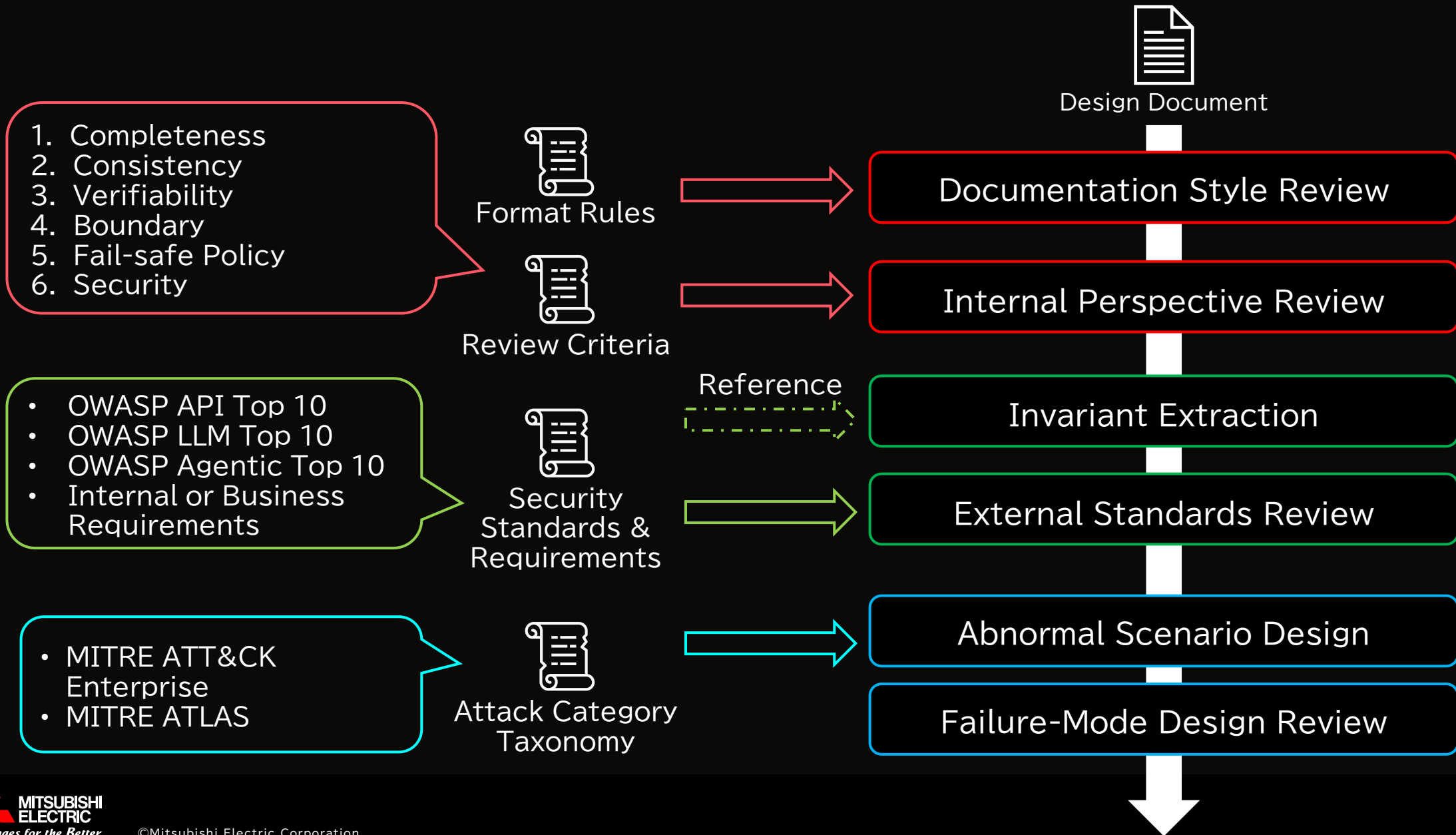


LLM

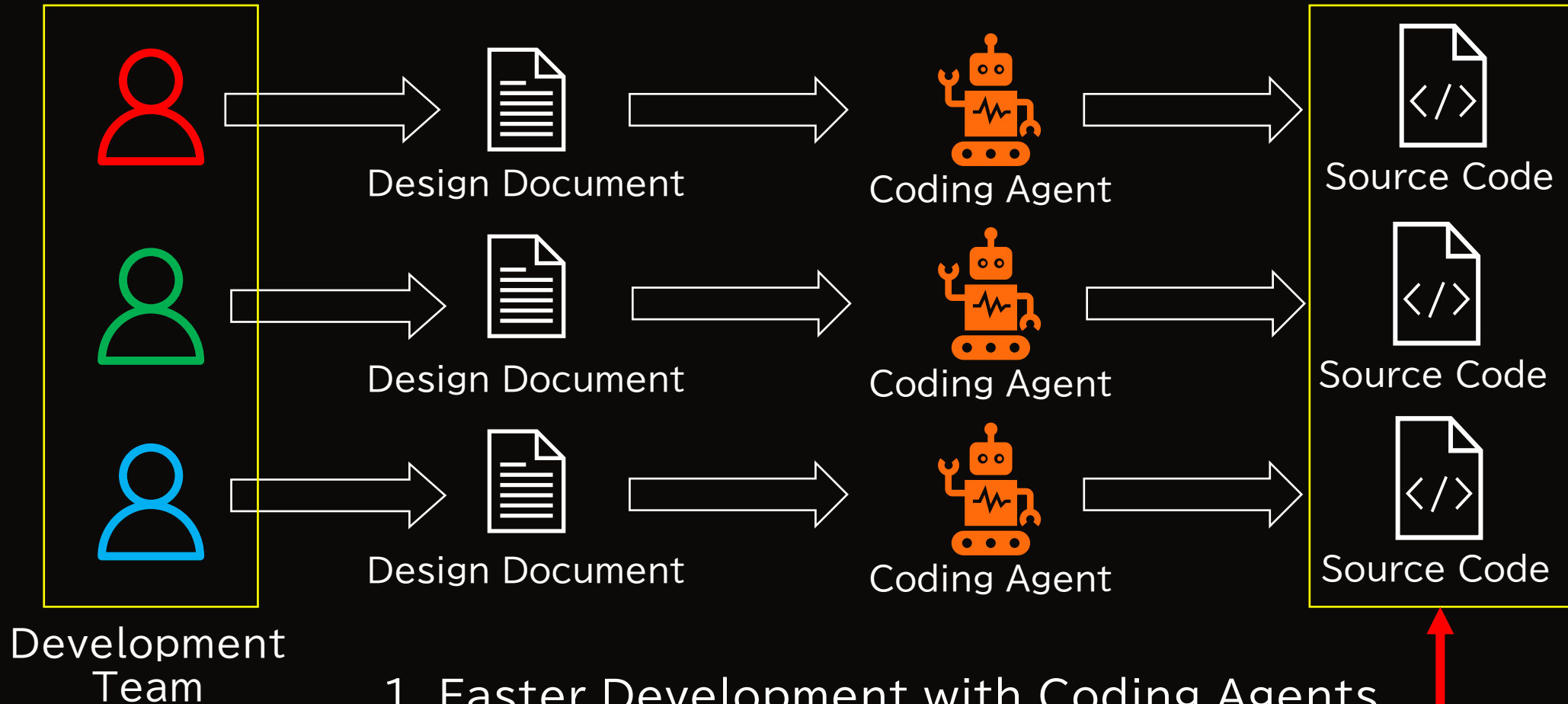


Any attack paths or vulnerabilities?
How can they be mitigated?

Point A “Build Security into the Specification”

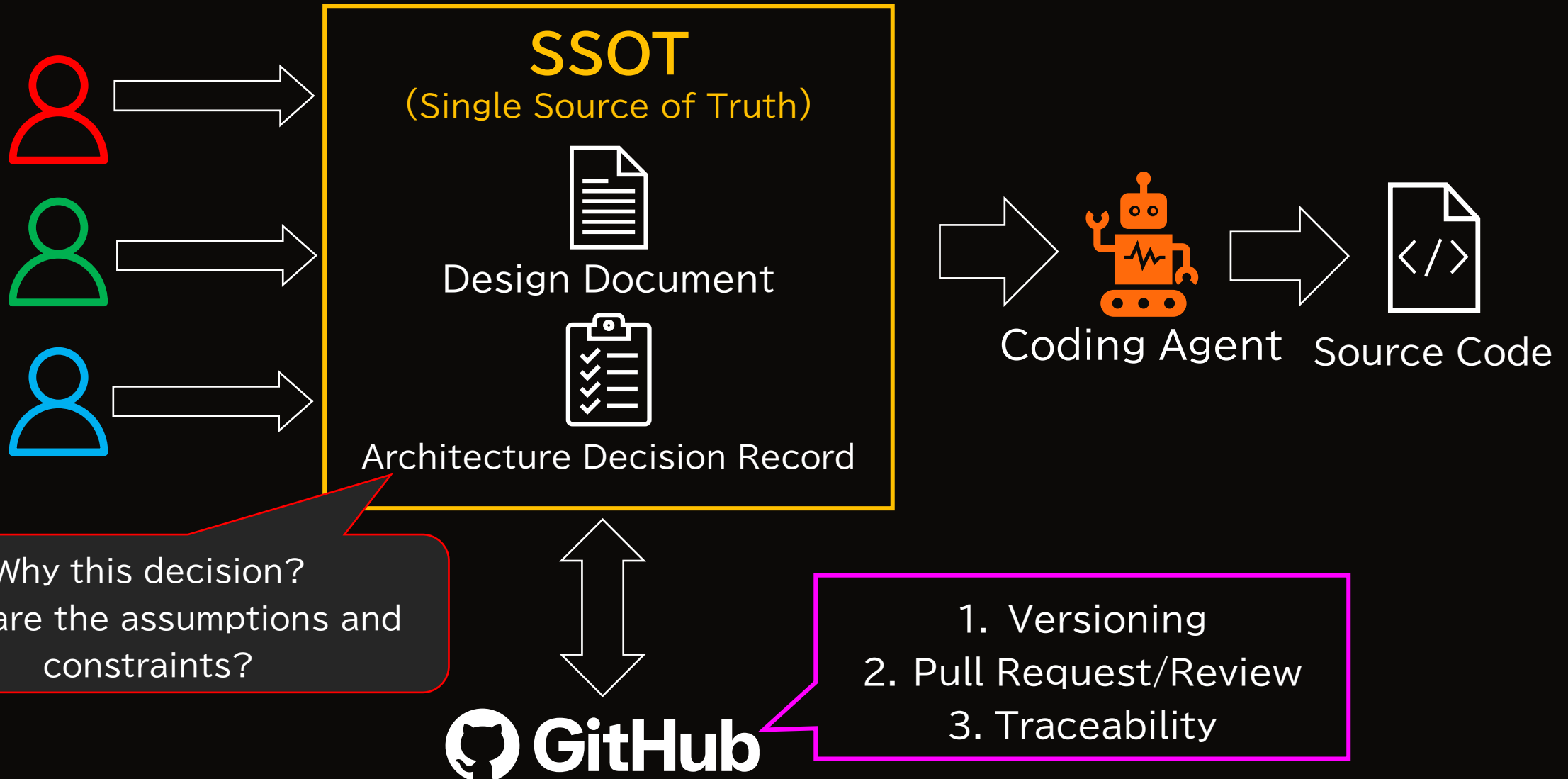


Point B “Use Design Documents as the SSOT”

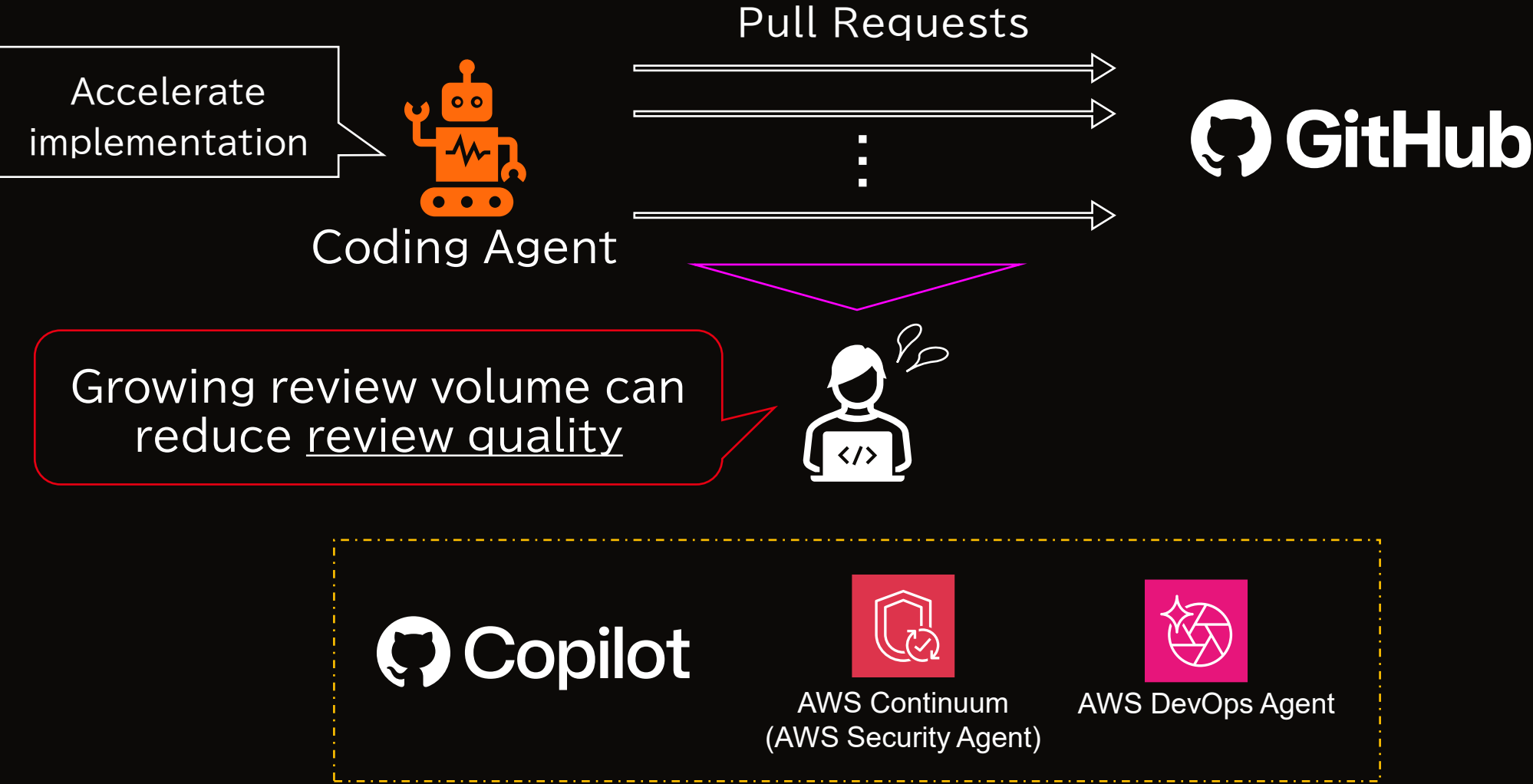


1. Faster Development with Coding Agents
2. More Parallel Development
3. Harder to Keep Designs in Sync
4. **Implementation Inconsistencies**

Point B “Use Design Documents as the SSOT”



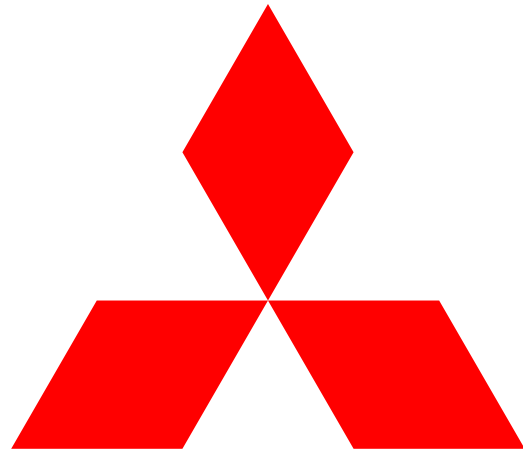
Point C “Scale Code Review with AI”



Ensure Quality and Efficiency with AI-Assisted Reviews

Conclusion

- Build an enterprise-wide platform
 - Use MCP and A2A for interoperability
 - Use gateways for governance
 - Use an orchestrator for diverse requests
- Establish AI-driven development
 - Build security into the specification
 - Use design documents as the SSOT
 - Scale code review with AI



**MITSUBISHI
ELECTRIC**

Changes for the Better