Weekly Sprint Plan: MAS Architecture & MCP Foundation

This presentation outlines Sprint 1 objectives focused on implementing a working MCP Server skeleton and finalizing the Agent Orchestration Architecture to begin coding the main flow.

Sprint 1 Goal: Architecture Lock-Down & Model Context Protocol (MCP) Foundation

Goal (This Week)

Implement a working skeleton of the **MCP Server (v1)** and finalize the **Agent Orchestration Architecture** to begin coding the main flow.

Target Outcome

Working `FastAPI` server hosting the initial `MCP` schema with `POST/PUT` endpoints. `LangGraph` flow diagram locked down.

Key Strategic Decisions

Orchestration

LangGraph selected for deterministic, stateful flow control (critical for FEA reliability).

Deployment

Azure Container Apps chosen for free-tier development and seamless enterprise transition.

Core Novelty

`MCP` is the focus for research publication.

Next Step: Begin implementing **Planner** and **Executor** agent logic next week.

Technical Justification: Reliability & Enterprise Alignment

This section justifies why **LangGraph** was chosen over other options and how this choice aligns with the Azure enterprise contract.

Orchestration Choice: LangGraph

Why not Conversational? Frameworks like `AutoGen` / `CrewAI` are conversational; `LangGraph` provides **explicit, stateful, graph-based control**. This is essential for managing multi-step, high-stakes FEA jobs where order and reliability are non-negotiable.

Core Innovation: Model Context Protocol (MCP)

The `MCP` acts as the system's shared memory (the `LangGraph` state). It is hosted via `FastAPI` to ensure data **Standardization** and **Protocol Reliability** for all agent communications, preventing data loss between Abaqus steps.

Deployment Strategy: Develop Locally, Deploy on Azure

Framework

`LangGraph` (Python) is **Cloud-Agnostic** for local development.

Hosting

Deploy to **Azure Container Apps (ACA) Free Tier** (Scales-to-Zero) for cost-effective testing.

LLM/Reasoning

Agents call the **Azure OpenAI Service** (the key asset of the Curtin contract) for complex reasoning tasks.

Overall Benefit: The architecture is **Reliable** (LangGraph control) and **Scalable** (ACA deployment).

Research Strategy: Publishing the System & Methodology

This section outlines the publication goal, focusing on the novelty of the system design and target venues.

Publication Target

Aiming for a **Systems/Design Paper** (Application/Methodology Track), *not* a product demo or pure algorithm paper.

Core Novelty / Contribution

1. **MAS Architecture:** A robust `LangGraph` architecture designed for precise control over external, resource-intensive scientific APIs (Abaqus).
2. **The MCP:** Defining a generalizable data standard/protocol for managing the state and context of FEA simulations.

Framing the Paper

Focus on the **System Architecture** and **Design Methodology** (Concurrency, Fault Tolerance) rather than just the final result. Quantify the benefit as **Speedup Factor** and **Reliability** compared to manual/scripted workflows.

Venues

Engineering (Primary Focus)

- *Automation in Construction*
- *ASCE Journal of Computing in Civil Engineering*

AI (Secondary Focus)

AAAI/IJCAI (Applications Track) if the MCP design is highly generalizable.