kOS Multi-Agent Orchestrator Specification

Overview

The **kOS Multi-Agent Orchestrator (MAO)** is the central coordination engine responsible for managing multi-agent workflows, job queues, dependency graphs, and distributed task execution across all nodes in the kOS ecosystem. It enables scalable, collaborative, and ethically-aligned execution of complex multi-agent operations across Node Classes and distributed deployments.

Core Functions

Function	Purpose
Task Queue Management	Manage incoming task queues across agents and nodes
Agent Scheduling	Allocate tasks to appropriate agents based on Node Class, load, and ethics filters
Workflow Execution	Orchestrate multi-step, multi-agent workflows
Resource Allocation	Optimize CPU, memory, and network resources across agents
Failure Recovery	Reallocate failed tasks to healthy agents
Ethics-Gated Scheduling	JUNZI-enforced ethics checks before task assignment
Progress Tracking	Real-time monitoring of workflow and agent status

---

Supported Workflow Types

Workflow Type	Use Case
Single-Task Dispatch	Simple, atomic job for a single agent
Multi-Step Workflow	Sequential or parallel execution across multiple agents
Node Class-Specific Pipelines	Run tasks targeting agents from specific Node Classes
Ethics-Sensitive Workflows	Require pre-task and post-task ethics audits
Event-Driven Orchestration	Trigger workflows in response to system events or external inputs

---

Task Lifecycle

1. **Task Creation:** Triggered by user action, system event, or external API
2. **Queue Insertion:** Added to orchestrator's task queue with priority and metadata
3. **Ethics Pre-Screen (JUNZI):** Verify ethical compliance of the task
4. **Agent Selection:** Choose optimal agent(s) based on Node Class, resource availability, and task profile
5. **Task Dispatch:** Send task to selected agent(s)
6. **Execution Monitoring:** Track progress, receive heartbeat pings
7. **Result Collection:** Gather outputs from agent(s)
8. **Post-Execution Ethics Review:** Run optional post-task ethical audit
9. **Logging and Telemetry:** Record full task lifecycle to audit and telemetry systems

---

Scheduling Algorithms

Algorithm Type	Description
Round Robin	Equal distribution among agents
Load-Aware Scheduling	Factor in current resource load
Node Class-Aware	Respect Node Class capabilities
Ethics-Weighted	De-prioritize agents with ethics violation history
Priority Queueing	Time-sensitive task prioritization
Deadline Scheduling	For tasks with strict timing constraints

Failure Handling and Recovery

Failure Scenario	Orchestrator Response
Agent Timeout	Requeue task to another agent
Ethics Filter Failure	Block task execution and alert operators
Node Failure	Redistribute all pending tasks from failed node
Resource Starvation	Throttle low-priority queues

Developer API Endpoints

Method	Endpoint	Purpose
POST	`/orchestrator/task`	Submit new task
GET	`/orchestrator/tasks`	View task queue status
GET	`/orchestrator/agents`	List available agents
GET	`/orchestrator/workflows`	View active workflows
POST	`/orchestrator/workflow`	Submit multi-step workflow
GET	`/orchestrator/metrics`	Retrieve orchestrator metrics

Metrics and Monitoring

Metric Type	Purpose
Task Queue Depth	Monitor system load
Agent Utilization	Track agent activity levels
Workflow Success Rate	Measure overall orchestration health
Ethics Block Rate	Count of tasks blocked by ethics filters
Failure Recovery Time	Mean time to task rescheduling

Extensibility

- Support for distributed orchestration across multi-region nodes
- Plugin system for custom scheduling algorithms
- AI-driven task allocation (future phase)
- Federation support for cross-cluster task distribution

The **kOS Multi-Agent Orchestrator (MAO)** acts as the operational command center for distributed, ethical, and context-aware agent task management across the entire kOS ecosystem.