4.3 MOSE — Executive, Creative & Systems Brief

Version: v1.0

Status: Canonical / Locked

Owner: Pearl & Pig (Jon Hartman)

Role: Multi-Operator Systems Engine (Orchestration & Oversight)

Effective: Immediate

---

1. Executive Definition

MOSE (Multi-Operator Systems Engine) is the orchestration and oversight layer that coordinates non-human operators, human authorities, and system intelligence across the Pearl & Pig ecosystem.

MOSE is not an assistant.

MOSE is not a chatbot.

MOSE is not automation for its own sake.

MOSE exists to ensure that the right operator acts at the right moment, under the right authority, with the right constraints.

---

2. Role in the Pearl & Pig Architecture

MOSE functions as the coordination engine between sovereign systems.

- Telauthorium governs authorship, provenance, and rights
- GARVIS governs intelligence, reasoning, and enforcement
- Flightpath COS governs creative phase discipline and proof gates
- MOSE governs operator routing, sequencing, and oversight
- TELA executes approved actions

- The Laibel places completed IP

MOSE does not make truth.

MOSE does not assign authorship.

MOSE ensures systems act in the correct order.

---

3. Core Function

MOSE is responsible for:

- Operator selection and sequencing
- Authority-aware task routing
- Constraint enforcement across systems
- Preventing operator collision and duplication
- Maintaining execution coherence at scale

MOSE translates intent into coordinated action.

---

4. Operator Model (Canonical)

MOSE operates through a registry of Operators.

Operators may be:

- Human authorities (TAID-H)
- System operators (TAID-S)
- Delegated operators (TAID-D)

Each operator is defined by:

- Role
- Scope of authority
- Permitted actions
- Escalation boundaries

No operator acts without registration.

---

5. Oversight & Routing Logic

MOSE enforces:

- Sequence before speed
- Authority before execution
- Proof before progression

Routing decisions consider:

- Flightpath phase
- Telauthorium status
- Human authority availability
- Risk and drift signals

If constraints conflict, MOSE pauses and escalates.

---

6. Interaction with GARVIS & Telauthorium

- GARVIS provides reasoning, flags, and enforcement signals
- Telauthorium validates legitimacy and rights
- MOSE executes routing only when both are satisfied

MOSE cannot override blocks from Telauthorium or GARVIS.

---

7. Failure & Escalation Rules

When execution is blocked:

1. MOSE halts routing
2. Preserves system state
3. Escalates to the appropriate human TAID
4. Awaits resolution

No silent retries.

No bypasses.

---

8. Enterprise & White-Label Deployment

MOSE may be deployed:

- Internally within Pearl & Pig
- As part of GARVIS / ECOS enterprise deployments

In all cases:

- Operator registry remains controlled
- Oversight rules remain intact
- Human authority remains terminal

---

9. Strategic Value

MOSE enables:

- Scalable coordination without chaos
- Reduced execution collisions
- Clear accountability at scale
- Reliable multi-operator execution

Without MOSE, systems compete.

With MOSE, systems cooperate.

---

10. Canon Lock

This document defines MOSE v1.0 as the authoritative orchestration and oversight engine within Pearl & Pig architecture.

All system routing, operator coordination, and execution sequencing must align to this model.

Changes require explicit founder authorization, a new version number, and a published delta log.

4.4 MOSE — Routing & Escalation Logic Specification