MaxDeploy v1.0 – Standalone Capability and System Coupling

This document defines how MaxDeploy functions as a standalone deployment enforcement system and how it connects with the broader MaxOneOpen architecture. It differentiates minimal deployment scenarios from full system-integrated operations.

# 1. Minimal Standalone Operation

MaxDeploy can operate without the presence of MaxAudit, MaxReg, or MaxBridge in environments where full system integration is not possible. In this mode, it enforces basic policy control using local manifest logic and embedded signature verification.

Minimum required components:

* - manifest.deploy.yaml with valid structural logic
* - operator capsule with assigned role signature
* - local hash-checking and rollback tracing

# 2. System-Coupled Operation (MaxOneOpen Integrated)

When operating inside a MaxOneOpen instance, MaxDeploy links to MaxReg for policy enforcement, to MaxAudit for trust propagation and certification, and to MaxBridge for source integrity and fork validation.

System-coupled behavior includes:

* - Active deployment rejection on failed policy match (MaxReg)
* - Audit-trace recording in IATL with trust signal propagation (MaxAudit)
* - Cross-environment traceability of forked deployments (MaxBridge)

# 3. Use in OSS-Only Environments

MaxDeploy can be used in pure OSS repositories to enforce governance over CI/CD pipelines, artifact releases, or public package distribution. OSS maintainers can embed pre-approved policy triggers and validate operator signatures before release publishing.

# 4. Behavior in Hybrid Infrastructure

In hybrid environments (e.g., enterprise + public OSS), MaxDeploy ensures consistent deployment logic by linking external policy references into the internal manifest system. It supports capsule synchronization across federated systems and rejects orphaned or invalidated fork attempts automatically.