MaxDeploy v1.0 – System Architecture and Capsule Model

This document defines the internal structure, technical components, and control logic of MaxDeploy. It adheres to the TBYD process standard and represents the structural foundation for deployment operations in MaxOneOpen v4.1.

# 1. Architectural Overview

MaxDeploy operates as a deployment intent executor and validator. It processes structured capsule inputs and transforms them into verified execution events based on policy compliance, audit traceability, and trust anchor validation. Its architecture is composed of modular logic units that interact with ManifestEngine, MaxReg, MaxAudit, and MaxBridge.

# 2. Core Capsule Types

The MaxDeploy capsule system defines the following core structures:

* - manifest.deploy.yaml – declares policy-synced deployment logic and input triggers
* - meta.audit.json – contains audit trace anchor, signature hash, and ledger references
* - anchor.exec.ref – connects the deployment to a structural or treaty-bound upstream context
* - rollback.state.json – defines technical rollback options with signature control

# 3. Trigger and Control Flow

Deployments in MaxDeploy are not initiated by shell commands or automation scripts but through policy-bound execution triggers. These are verified by the ManifestEngine and require validation from MaxReg before execution proceeds. Each deployment path includes signature checkpoints, policy evaluation hooks, and rollback contracts.

# 4. System Interfaces

MaxDeploy integrates with:

* - MaxReg: For regulation enforcement and policy-based rejection of deployment intents
* - MaxAudit: For audit trace generation, immutable log registration, and trust signal emission
* - MaxBridge: For fork-check validation and cross-environment capsule reference control
* - MaxOneOpen ManifestEngine: For verification and interpretation of deployment manifests

# 5. Execution Model

Once validated, MaxDeploy generates a structured Execution Capsule, which inherits the policy, audit, and capsule state. It is then passed to the runtime environment via a cryptographically validated trigger capsule that authorizes execution only under compliant system states.