# Module 06 – Manifest Engine & Threat-Resilient Declaration Layer (CTO Reinforced Edition)

Module ID: MANIFEST-DECL-006

Version: 4.1 (CTO Reinforced Edition)

Layer: Component Declaration, Trust Filtering, Threat-Aware Execution Control

Status: RELEASE

Dependencies: Module 00, 05, 07, 12, 13, Threat Model

## 0. Purpose & CTO Enforcement Scope

This module defines the system-wide manifest declaration and runtime binding logic for all executable components in MaxOneOpen v4.1. It includes threat-aware validation, replay-protected registration, trust tier enforcement, and sandbox isolation. It replaces all previous manifest structures and includes binding to the threat model for CTO-grade score validation.

## 1. Unified Manifest Capsule Schema

All components must declare a `Manifest Capsule (MC)`:  
`{ component\_id, manifest\_type, scope\_class, trust\_tier, rule\_links[], signature, origin\_hash, threat\_profile[], execution\_flag }`  
Manifest Types:  
- CORE\_MODULE  
- OSS\_ADDON  
- AI\_EXTENSION  
- POLICY\_AGENT  
- SYSTEM\_FORK  
Every manifest is verified against:  
- Trust tier mapping rules (Module 12)  
- Threat vectors (see Section 5)  
- Signature lineage from declared source  
- Capsule chain presence (Module 13)

## 2. Execution Control & Enforcement Hooks

Execution is only permitted after full manifest validation and trust clearance:  
- All sandbox-bound components (e.g. OSS\_ADDON) must set `execution\_flag = sandbox\_only`  
- Policy-bound execution requires live Rule Capsule references  
- Execution failures or trust mismatch emit `Manifest Violation Capsule (MVC)`  
- Repeated failures lead to suspension and quarantine  
Audit trail for all executions is enforced via Module 13 (Ledger) and Module 14 (Audit Engine).

## 3. Threat-Aware Validation & Replay Protection

Each manifest undergoes threat-scanning using:  
- Replay Hash Conflict Detection (RHC)  
- Signature Chain Validation (SCV)  
- Tier Escalation Prevention (TEP)  
- Add-on Escalation Filter (AEF)  
Any risk trigger generates:  
- `Replay Block Capsule (RBC)`  
- `Trust Violation Capsule (TVC)`  
- `Manifest Quarantine Capsule (MQC)`  
Threat profiles are matched against the global Threat Model (CTO Layer).

## 4. Capsule Chain Binding & Audit Mapping

All manifests must anchor to the Ledger (Module 13) and support forensic replay via Module 14.  
Each manifest commit generates:  
- `Manifest Commit Capsule (MCC)` with scope, hash, zone  
- Optional `Audit Scope Capsule (ASC)` for public view  
Uncommitted manifests are invalid and non-executable.

## 5. Threat Vector Classification & Mitigation Binding

Linked from Global Threat Model:  
- Signature Spoofing → Signature Chain Verification (Module 13)  
- Replay Injection → Replay Hash Guard Capsule  
- Add-on Escalation → Tier Isolation, Sandbox Gate  
- Rule Injection Bypass → Rule Integrity Check Capsule  
- Trust Zone Drift → ZONE\_ENFORCE Capsule with override rejection  
Each mapped vector must produce an auditable mitigation capsule on violation.

## 6. Visual Execution Path & Manifest Enforcement Flow

Flow overview (abstract structure):  
Component Manifest → Threat Scan → Trust Map Check → Capsule Bind → Ledger Commit → Audit Trace Enabled  
Violations → Quarantine / Replay Alert / Certification Block

## 7. CTO Score Linkage & Certification Prerequisite

This module is mandatory for full certification (Module 17). Without threat-aware manifest enforcement and capsule linkage, any deployment is disqualified from CTO scoring. Manifest capsules must pass replay, signature, and scope validation before execution or scoring eligibility.