# Module 08 – Add-on Forking & Domain Separation

Module ID: ADDON-FORK-008

Version: 4.1 (Revised CTO Edition)

Layer: Add-on Policy Layer

Status: RELEASE

Dependencies: Module 00, Module 06, Module 07

## 0. Purpose & Enforcement Point

This module defines the technical control logic and isolation policy for forking add-on components within MaxOneOpen v4.1. It ensures that all forks operate within declaratively defined execution boundaries and are domain-separated from origin systems unless explicitly re-linked via controlled bridges. Forked domains are ledger-anchored, trust-classified, and manifest-bound.

## 1. Fork Invocation, Binding Protocol & Domain Anchoring

Forks are created using the `ADDON\_FORK\_CREATE()` protocol:  
- `origin\_manifest\_id` (must be validated)  
- `fork\_manifest\_id` (must declare scope and trust explicitly)  
- `bridge\_flag` (YES/NO: determines re-linking rights)  
Returned: `Fork Capsule ID (FCID)` or `Violation Capsule`  
Each fork is registered in the Add-on Fork Ledger and assigned a `Domain Separation Level (DSL)` classification.

## 2. Domain Separation Levels & Fork Governance Rules

Forks are evaluated against domain separation policies:  
- `DSL-0`: Fully rebindable fork (requires admin twin signature)  
- `DSL-1`: Isolated execution, read-only bridge  
- `DSL-2`: Hard-isolated, no communication with origin allowed  
Forks inherit rules from origin only if declared in the fork manifest and confirmed by ConfigBinding (Module 05).  
Default policy: DSL-2 unless explicitly overridden.

## 3. Execution Isolation, Communication Control & Merge Logic

All forked executions are subject to sandboxing (Module 07) and domain enforcement:  
- Manifest isolation enforced at runtime  
- IO must route through verified Twin Messaging Relay (Module 09)  
- Merge back into origin only possible via `FORK\_REJOIN()` protocol, which requires:  
 → same trust tier  
 → rule conformity snapshot  
 → admin override capsule

## 4. Capsule Format & Fork Ledger Traceability

Each fork event produces the following artifacts:  
- `Fork Capsule (FC)` – `{ fork\_id, origin\_id, manifest\_ref, trust\_tier, DSL, timestamp }`  
- `Domain Separation Record (DSR)` – `{ fork\_id, dsl\_level, trust\_boundary\_hash }`  
- `Fork Violation Capsule (FVC)` – triggered on unauthorized merge or linkage attempt  
All capsules are committed via Module 13 and retrievable through Module 14 forensic queries.

## 5. Intermodular Bindings & Enforcement Points

This module interfaces with:  
- Module 06 (Manifest Engine) to validate fork definitions  
- Module 07 (Sandbox) for isolation enforcement  
- Module 09 (Twin Messaging) for communication flow control  
- Module 12 (Trust Enforcement) for tier alignment  
- Module 13 (LedgerSync) for capsule chain traceability  
- Module 16 (Admin Twin) for override handling

## CTO Validation Matrix

Module 08 (CTO Edition) guarantees the following verifiable conditions:  
- Forked add-ons are domain-separated and capsule-anchored: YES  
- Fork manifests control re-linking and trust inheritance: YES  
- Domain Separation Level (DSL) is enforced and auditable: YES  
- No fork may bypass sandbox, trust, or messaging controls: YES  
- Merge requires policy-bound capsule chain and admin override: YES