# Module XX – License Enforcement & Fork Governance

Version: 4.1 | Classification: CTO Mandatory (Open Licensing Integrity & Structural Governance)

Scope: Fork transparency, naming protection, license integrity, capsule validation for derivatives

## 0. Purpose & License Trust Layer

This module defines mechanisms to preserve license integrity, track authorized forks, and prevent proprietary capture of MaxOneOpen. It enforces naming conventions, mandates license declarations, and ensures that any derived systems uphold the principles of openness, reproducibility, and structural compatibility.

## 1. License Declaration Capsule (LDC)

- Each deployment or fork must include a signed LDC:  
`{ license\_type, version, origin\_hash, forked: true/false, compliance\_mode, contact, declaration\_date }`  
- Must reference root governance capsule and hash of base specification  
- LDCs must be published to federation registry for observability

## 2. Naming Protection & Structural Claims

- The term 'MaxOneOpen' may only be used for forks or deployments with full capsule integrity  
- Partial forks must be declared as 'derived' or 'inspired by', not 'compliant'  
- Compatibility claims must be validated by Certification Council (Module 17 linkage)

## 3. Capsule Audit Anchor for Licensing

- LDCs are indexed via a `License Audit Capsule (LAC)`  
- Each LAC confirms hash lineage and license match  
- Incompatible forks may be listed with status: `restricted`, `isolated`, or `non-certified`

## 4. Federation Behavior Under Fork Divergence

- Federation nodes may restrict interaction with non-declared forks  
- LDC must be co-signed by receiving federation authority for operational merge  
- Conflicting forks may trigger Fork Divergence Alert Capsule (FDAC)

## 5. Enforcement & Public Disclosure

- All licensed forks must be publicly logged  
- Licensing violations may result in:  
 • Certification score nullification  
 • Governance council warning  
 • Federation access suspension  
- Public license index must be published via `/capsule/license` endpoint (see API spec)