# Module 04 – MaxReg: Governance Rule Capsules & Compliance Binding

Module ID: CORE-GOV-004

Version: 4.1 (Revised CTO Edition)

Layer: Core Architecture Layer

Status: RELEASE

Dependencies: Module 00, Module 01, Module 03

## 0. Purpose & Enforcement Point

This module defines the formal structure, classification, and enforcement logic for system-wide rules in MaxOneOpen v4.1. It ensures that all operational decisions, role privileges, AI behavior, fork permissions, and audit conditions are governed by immutable rule capsules. Rules are manifest-mappable, cryptographically bound, and enforced at runtime through integration with ConfigBinding (Module 05).

## 1. Rule Structure, Classification & Identifier Schema

Every rule in MaxOneOpen is defined as a Governance Rule Capsule (GRC) with the following schema:  
`{ rule\_id, classification\_type, applies\_to, enforcement\_level, expiration, rule\_logic\_hash }`  
Rule classifications:  
- SYSTEM: structural constraints (fork limits, execution anchors)  
- ROLE: operator restrictions (scope, override conditions)  
- TRUST: trust-tier conditions (min/max)  
- AI: inference boundaries (model trust, containment)  
- AUDIT: logging conditions (capsule types, fields)  
Each rule is versioned, immutable, and hash-anchored.

## 2. Rule Anchoring, Registration & Lifecycle Logic

Rules are declared via the `RULE\_REGISTER()` function and stored in the MaxReg Capsule Ledger. Rules can be:  
- static (unchangeable for lifetime);  
- lifecycle-bound (expiring on event or time);  
- fork-specific (scoped to fork lineage ID).  
Rules are linked to manifests by `manifest\_rule\_map` and validated at execution via Module 06 and Module 01 triggers.

## 3. Runtime Evaluation, Conflict Resolution & Override Logic

Rules are evaluated at runtime via ConfigBinding (Module 05) in the following sequence:  
1. Load all `manifest\_rule\_map` entries  
2. Evaluate per rule type in priority order: SYSTEM → ROLE → TRUST → AI → AUDIT  
3. Apply result: `ALLOW`, `BLOCK`, or `ESCALATE`  
4. Log result in `Rule Evaluation Capsule (REC)`  
Override only permitted with explicit Escalation Capsule and admin twin confirmation (Module 16)

## 4. Capsule Format & Enforcement Artifacts

Rule Capsules and evaluation traces include:  
- `Governance Rule Capsule (GRC)` – the original rule object  
- `Rule Evaluation Capsule (REC)` – result of runtime enforcement  
- `Rule Conflict Snapshot (RCS)` – if contradictory rules encountered  
Each capsule includes:  
`{ capsule\_id, rule\_id, context, outcome, hash\_anchor, timestamp }`

## 5. Intermodular Bindings & Enforcement Path

This module is directly linked to:  
- Module 01 (Execution Control) for runtime invocation  
- Module 03 (Role Map) for role-bound rule validation  
- Module 05 (ConfigBinding) for injection and resolution  
- Module 12 (Trust Enforcement) for trust-scope rule handling  
- Module 13 (LedgerSync) for rule commit and audit anchoring

## CTO Validation Matrix

Module 04 (CTO Edition) guarantees the following verifiable conditions:  
- Every system rule is capsule-bound, immutable, and auditable: YES  
- Rule enforcement occurs at runtime with full trace: YES  
- Conflicts are detected and resolved with logged override: YES  
- Role- and trust-based rule constraints are enforced: YES  
- No rule executes or applies outside manifest linkage: YES