# Module 01 – System Sovereignty & Execution Control

Module ID: CORE-CTRL-001

Version: 4.1 (Revised CTO Edition)

Layer: Core Architecture Layer

Status: RELEASE

Dependencies: Module 00

## 0. Purpose & Enforcement Point

This module defines the technical foundation for execution control, operational sovereignty, and runtime authority enforcement in MaxOneOpen v4.1. It guarantees that no execution path may begin, persist, or complete unless it is manifest-declared, capsule-bound, and governance-verified. This module is the origin anchor for all identity, policy, and runtime capsule flows.

## 1. Control Logic, Invocation Interface & Scope

Execution is only permitted via the Sovereign Execution Gate (SEG), which:  
- validates the incoming Manifest Capsule (MC);  
- authenticates the triggering identity (TIT);  
- injects all applicable rule capsules (RCs) via ConfigBinding;  
- allocates an Execution Session Capsule (ESC).  
The invocation format is:  
`EXEC\_CALL(manifest\_id, identity\_token, operation\_type)`  
Returns: `ESC\_ID` or `Violation Capsule`.  
Input schema, binding results, and capsule generation are audit-sealed and recorded.

## 2. Runtime Policy Injection & Scope Control

Upon invocation, the module injects policy capsules from Module 04 (MaxReg) using the ConfigBinding engine. Session validation includes:  
- Scope Map Resolution (role, action, domain);  
- Trust-Level Filtering (via Module 12);  
- Policy-Conflict Resolution Matrix.  
If resolution fails or returns `BLOCK`, the operation is terminated pre-execution and the violation is logged.

## 3. Trigger Logic, Isolation & Exit Enforcement

Each execution session has:  
- strict memory and IO boundaries;  
- runtime expiry window (max TTL);  
- violation and rollback triggers on capsule mismatch.  
Exit is only permitted via:  
- `OPERATION\_SUCCESS` → Capsule Commit to Ledger (Module 13)  
- `OPERATION\_FAIL` → Rollback Capsule and Audit Flag  
- `TRUST BREACH` → Emergency Suspension Trigger

## 4. Audit Artifacts, Forensic Fields & Capsule Schema

Every session produces:  
- Execution Session Capsule (ESC): includes scope, caller, TTL, hash, outcome  
- Policy Injection Record (PIR): list of injected rules, hashes, enforcement result  
- Session Trace Summary (STS): timestamp, trigger origin, runtime envelope  
ESC Capsule Schema includes:  
`{ esc\_id, manifest\_id, role\_id, operation, hash\_anchor, outcome, capsule\_timestamp }`  
All artifacts are stored via LedgerSync (Module 13) with audit-read access by certified roles.

## 5. Intermodular Control & Trigger Graph Anchors

This module triggers or validates:  
- Module 03 (Identity Check via TIT)  
- Module 04 (Policy Capsule Retrieval)  
- Module 05 (ConfigBinding Injection)  
- Module 13 (Ledger Commit)  
- Module 14 (Forensic Capsule Snapshot)  
Failure at any trigger point results in a capsule-bound violation record.

## CTO Validation Matrix

Module 01 (CTO Edition) guarantees the following verifiable conditions:  
- All executions are bound to manifests and declared identities: YES  
- Runtime rule enforcement is active and logged: YES  
- Exit points are structured and rollback-safe: YES  
- Audit capsules include role, origin, and trust score: YES  
- No undeclared execution path can bypass enforcement: YES