# Module 09 – Secure Twin Messaging & Scoped IO Relay

Module ID: COMM-TWIN-009

Version: 4.1 (Revised CTO Edition)

Layer: Communication Control Layer

Status: RELEASE

Dependencies: Module 00, Module 03, Module 07

## 0. Purpose & Enforcement Point

This module defines the secure, scoped and identity-bound messaging framework used by actors and add-ons in MaxOneOpen v4.1. All messages must pass through a role-verified Twin Relay, obey trust boundary conditions, and remain traceable via audit capsules. No unscoped or unverified IO is permitted inside or outside the system.

## 1. Messaging Structure, Invocation Format & Twin Relay Logic

Messaging occurs via the `TWIN\_SEND()` function:  
- `from\_identity\_id`, `to\_identity\_id`  
- `operation\_context`, `payload`, `TTL`  
Messages are routed by the Twin Relay Controller (TRC), which:  
- verifies sender/receiver role and trust alignment  
- binds the message to a Twin Message Capsule (TMC)  
- applies TTL, rule constraints, and scope boundary control  
- denies or delays messages outside allowed relay scope

## 2. Message Capsule Schema & Enforcement Fields

Each valid message is encapsulated in a TMC with the following format:  
`{ capsule\_id, from\_id, to\_id, operation\_type, scope\_class, TTL, hash\_anchor, timestamp }`  
Messages failing rule or trust validation generate a `Relay Rejection Capsule (RRC)`.  
Message capsules are stored for retrieval and trace validation in Module 14.

## 3. Scoped IO Enforcement, Trust Filtering & Role Matching

The relay system enforces communication limits via:  
- manifest-declared messaging scopes (Module 06)  
- trust-tier compatibility checks (Module 12)  
- role identity verification (Module 03)  
- sandbox interaction boundaries (Module 07)  
Any trust downgrade, unknown role, or boundary breach is blocked, logged, and flagged as a security event.

## 4. Relay Integrity Capsule, Violation Logs & Audit Trace

Artifacts generated:  
- `Twin Message Capsule (TMC)` – for every successful message event  
- `Relay Rejection Capsule (RRC)` – for blocked or failed messages  
- `Scope Validation Snapshot (SVS)` – context of message control decision  
All logs are bound via Module 13 and visible through Module 14 forensic tools.

## 5. Intermodular Bindings & Trust Chain Hooks

This module integrates with:  
- Module 03 (Identity & Role) to check message sender/recipient  
- Module 06 (Manifest) to validate scope declaration  
- Module 07 (Sandbox) to limit add-on IO  
- Module 12 (Trust Enforcement) to enforce tier alignment  
- Module 13 (LedgerSync) to store capsule logs  
- Module 14 (Audit Capsule) to trace message lineage

## CTO Validation Matrix

Module 09 (CTO Edition) guarantees the following verifiable conditions:  
- All IO occurs through identity-bound, capsule-logged relay: YES  
- Messages respect manifest scope, TTL, and trust boundaries: YES  
- Violations are denied, logged, and trigger forensic trace: YES  
- Message logs are cryptographically linked to actors and roles: YES  
- No external or internal channel may bypass relay enforcement: YES