# MaxSuite – AppPolicy Lifecycle and Expiry

Version: 1.0

Module ID: CTO-T1

Document Type: Lifecycle and Validity Control

---

## 1. Purpose

This document defines the full lifecycle of application-specific policy capsules used by MaxSuite. It includes activation, expiration, override logic, signature management, and formal retirement handling.

## 2. Lifecycle Phases

• Draft:

- Created by a registered authority or MaxReg node.  
- Not yet active; allows internal validation or simulation.  
- May include test binaries and uncommitted policy references.

• Active:

- Policy capsule is signed and registered.  
- Actively governs app execution based on role, scope, and signature validation.  
- Included in runtime capsule manifest as `policy\_id`.

• Expired:

- Time-limited policies become inactive after `valid\_until` timestamp.  
- Any capsule referencing an expired policy must be blocked or rerouted to fallback logic.

• Overridden:

- A temporary override capsule is linked with a new validity window and reason code.  
- Must include `override\_id`, `signature`, `reason`, `valid\_until` fields.

• Retired:

- Policy is formally revoked and archived.  
- Must be listed in MaxReg retirement registry.  
- Execution attempts must be frozen or redirected.

## 3. Validation Requirements

- Capsule must include `policy\_version`, `valid\_from`, `valid\_until`, and `signature\_hash`.  
- All transitions (Active → Expired, Override → Retired) must be audit-logged.  
- Policies must never persist outside of version-controlled MaxReg context.