MaxTreaty v1.0 – Treaty Types and Enforcement Modes

This document defines the types of treaties supported by MaxTreaty and the corresponding enforcement modes. It distinguishes between OSS, institutional, governmental, and private configurations and outlines how enforceability is implemented in each context.

# 1. Treaty Classification

MaxTreaty supports the following treaty types:

* - Public OSS Treaties: signed by maintainers or collectives, published openly
* - Institutional Treaties: created and validated within a defined organization or project
* - Governmental Treaties: issued by states, municipalities, or public authorities
* - Private Treaties: limited-scope, non-public treaties used within closed environments

# 2. Treaty Lifecycle

Each treaty has a lifecycle consisting of:  
- Creation and signing  
- Registration (optional for public discovery)  
- Enforcement (through capsule reference binding)  
- Revision (with hash-locked update trace)  
- Expiry or revocation (voluntary or auto-expired)

# 3. Enforcement Mechanisms

Enforcement is based on verifiable conditions:

* - All signed deployments must match a valid treaty.ref
* - Conflict or invalid treaties are blocked by MaxDeploy and MaxBridge
* - Audit capsules (via MaxAudit) embed the treaty fingerprint

# 4. Delegated Treaty Enforcement

Treaties may delegate enforcement to capsules in other systems. For example:  
- MaxDeploy may reject a rollout based on treaty class mismatch  
- MaxBridge may mark a capsule as orphaned due to invalid treaty path  
- MaxReg may refuse policy approval due to unrecognized treaty authority

# 5. Scope-Specific Validity

Treaties can define their own validity scope, e.g.:  
- By domain (e.g., \*.project.org)  
- By capsule type (e.g., only Bridge capsules)  
- By jurisdiction or licensing body  
These scoping rules are verifiable via the treaty.scope.json definition.