MaxTreaty v1.0 – Treaty Capsules and Reference Integration

This document defines how treaty data is encapsulated in technical units and how these capsules can be referenced from other systems. It ensures verifiability, linkage, and auditability across sovereign deployments.

# 1. Treaty Capsule Definition

A Treaty Capsule is a signed, immutable container of governance logic. It includes:  
- treaty.manifest.yaml  
- legal.scope.json  
- capsule.signature.sig  
- optional: delegation.tree.json and anchor.ref

# 2. Referencing from Other Capsules

Treaty Capsules may be referenced via:

* - `treaty.ref` field in MaxDeploy, MaxBridge or MaxAudit
* - `anchor.ref` linkage for signed forks or governance transitions
* - `license.ref.sig` validation by MaxReg or treaty validators

# 3. Structural Integrity Requirements

Treaty Capsules must include verifiable hashes and signed fields. All references must resolve to:  
- a known signature chain  
- a registered governance anchor  
- a publicly verifiable version

# 4. Capsule Hash Linking

Each capsule that refers to a treaty must store its fingerprint hash. This hash forms the base for:  
- audit chains  
- fork protection  
- treaty version tracking

# 5. Multi-System Integration

Treaties are designed to be referenced from any MaxSystem:  
- MaxDeploy validates deployment policies  
- MaxBridge anchors policy lineage and reference inheritance  
- MaxAudit attaches treaty proofs to execution traces  
- MaxReg validates that a policy is treaty-compliant