MaxTreaty v1.0 – Audit Trail Structure and TreatyProof Export

This document defines the structure of the audit data generated by MaxTreaty and how treaty-related proof elements can be exported for verification, certification, or sovereign inspection. All treaty operations generate a consistent, signed trail of events.

# 1. Audit Trail Composition

Each treaty-related operation produces audit materials including:

* - treaty.manifest.yaml
* - treaty.scope.json
* - capsule.signature.sig
* - optional: delegation.tree.json and treaty.fingerprint.hash

# 2. TreatyProof Structure

The exported TreatyProof bundle may include:  
- `treatyproof.meta.json` (issuer, date, keyID, scope)  
- `treatyproof.integrity.sig` (full hash chain signature)  
- `treatyproof.anchor.ref` (governance link, optional notarization)

# 3. Export Targets

TreatyProof bundles can be exported to:

* - OSS registries (e.g., GitHub, IPFS)
* - Treaty catalogs or legal certification providers
* - Internal governance registries or audit sandboxes

# 4. Cross-System Audit Integration

MaxTreaty audit exports are interoperable with MaxAudit and MaxDeploy. Each TreatyProof can be imported as a verified audit condition for rollout approvals, governance classification or capsule transparency proofs.

# 5. Version and Integrity Checking

All exported proof files must include:  
- embedded treaty version and compatibility hash  
- origin capsule or anchor reference  
- timestamp of treaty execution or registration