MaxTreaty v1.0 – Conflict Detection and Arbitration Mechanism

This document describes how MaxTreaty identifies, flags, and resolves conflicts between overlapping, revoked, or manipulated treaty references. It also defines arbitration logic for capsules with unclear or competing treaty lineage.

# 1. Types of Conflicts

MaxTreaty supports detection of the following conflict types:

* - Duplicate treaty IDs with diverging content
* - Expired or revoked treaties still referenced in live capsules
* - Orphaned references to non-verified treaty anchors
* - Capsule claiming multiple contradictory treaty classes or policies

# 2. Conflict Detection Mechanism

MaxTreaty validates incoming capsules or treaty references against the active treaty registry. Conflicts are flagged when:  
- A treaty.ref hash mismatch is detected  
- A signature chain has been broken or revoked  
- Multiple treaties claim to govern the same scope without cross-validation

# 3. Arbitration Logic

Arbitration proceeds as follows:

* - Preference is given to the treaty with the stronger anchor.ref chain
* - Valid class assignments (e.g., Class A over Class B) override unanchored claims
* - Historical capsules may defer to a treaty.version fallback anchor
* - System-level override may be triggered by MaxBridge or governance operator

# 4. Conflict Audit Trail

Conflicts and their resolution are stored in a `treaty.conflict.log.json` audit capsule. This includes:  
- capsule ID(s) involved  
- treaty IDs and version numbers  
- arbitrating mechanism applied  
- result and system action (accept, reject, quarantine)

# 5. External Notification and Disclosure

All conflict events above a defined severity threshold are publicly exportable. MaxTreaty provides notification hooks to:  
- MaxBridge (capsule lineage adjustment)  
- MaxAudit (risk-flagged audit trail)  
- MaxReg (declassification or license adjustment)