Cascade Deactivation Dependencies

by Nick Clark | Published April 25, 2026

What It Specifies

Each adaptation declares its dependencies: required adaptations, supported adaptations, conflicting adaptations. Deactivation events propagate through the dependency graph; dependent adaptations admit against the new dependency state.

Cascade-deactivation events are governance-credentialed. The triggering deactivation, the cascade evaluation, and the resulting cascade-deactivations all enter lineage; downstream audit can traverse the cascade structurally.

Why It Matters Structurally

Deactivation without cascade evaluation produces architectural inconsistency. Dependent adaptations may continue to operate against deactivated dependencies; the resulting operation is unstable.

Cascade deactivation produces structural consistency. Dependencies are explicit; deactivation propagates; the resulting state is consistent.

How It Composes With Mesh Operation

The architecture defines the dependency-declaration format, the cascade-evaluation primitives, and the cascade-recording. Implementations apply the architecture; deactivation operations proceed within the framework.

Cascade composes with other features. Cross-jurisdictional cascade, byzantine-robust cascade under disputed dependencies, and dispute mechanism for cascade disputes all build on the cascade primitive.

What This Enables

Defense adaptation operations gain structurally-supported cascade deactivation. Civilian critical-infrastructure adaptation operations gain the same.

The architecture also supports cascade evolution. As adaptation-dependency patterns mature, cascade protocols update through governance procedures.