

## **MaxAudit – Structural Fork Editor & Registry Mutation Interface**

Version: 1.0

Issued by: SAC, Take Back Your Data (TBYD)

License: MaxOneOpen License v2.2 – Structurally Enforced

### **1. Objective**

This module defines the interface logic, usage policy, and audit compatibility constraints of the Fork Editor. It enables authorized operators to propose, structure, and register structural forks of MaxOne systems while maintaining audit integrity.

### **2. Fork Editor Functionality**

- Visual interface to edit MaxInstance structures
- Define or remove modules, roles, trust paths
- Annotate rationale and operational constraints
- Export as signed Fork Profile with unique Fork ID

### **3. Mutation Rules**

- Forks must remain compliant with MaxAudit interface specification
- Core audit logic must remain unmodified
- Modified modules must be explicitly identified and hashed
- Profiles must declare ZKP compatibility status

### **4. Registry Integration**

- New forks submitted to Fork Registry via offline-signed package
- Registry assigns versioned Fork ID
- Change log is appended to Fork's public lineage
- Fork can then be validated via Verifier or Dongle (if activated)

### **5. Governance Options**

- Forks may be public or private
- Private forks are locally usable but not certifiable
- Public forks become auditable across jurisdictions
- Registry may apply expiration or review periods

### **6. UI Requirements**

- Schema editor with drag-drop modular logic
- Audit preview: pre-check against MaxAudit core engine
- Export function with signed profile manifest