MaxDeploy v1.0 – OSS Mode Template (Minimal Integration)

This short-form guide enables open source project maintainers to use MaxDeploy without requiring full integration with MaxOneOpen components. It defines the smallest viable deployment structure for GitHub, GitLab, or similar platforms.

# 1. Required Capsule Components

* - manifest.deploy.yaml (includes policy.ref)
* - operator.capsule.json (signed by maintainer or org)
* - optional: meta.audit.json and audit.trace.sig

# 2. Recommended Policy Settings

OSS maintainers may define their own policy profiles (e.g., contribution scope, review requirements, release rules). These can be self-hosted or referenced from an upstream trusted project.

Example:

* policy.ref: https://oss.myproject.dev/policy/v1.2.yaml

# 3. Basic Signing Workflow

1. 1. Write or modify the manifest
2. 2. Sign with your GPG key or project keychain
3. 3. Attach signature and optional audit stub
4. 4. Trigger MaxDeploy (local or CI-integrated)

# 4. Hosting Options

Capsules can be hosted:  
- Inside the Git repo (e.g., /deploy)  
- In an attached object store (e.g., IPFS, S3)  
- With transparency via GitHub Pages or artifact registries

# 5. Public Verification

Audit traces may be made public to build trust in release processes. Capsule signatures should be verifiable using open GPG or signature validation tools. If possible, publish the audit trail (meta.audit.json) with each release tag.