

Sanctum SecOps LLC Certificate Policy (CP)

Status: DRAFT — v0.1 **Document OID:** 1.3.6.1.4.1.65953.23.1.1 **Version:** 0.1 **Effective date:** DRAFT — not yet effective. Pending Brian Vicente review and approval. **Document date:** 2026-07-27

DRAFT NOTICE. This document is version 0.1 and is a DRAFT. It has not been approved by Sanctum SecOps LLC policy management authority. It describes a LAB/TEST demonstration PKI hierarchy ([sanctum-root-ca-g1](#) , [sanctum-issuing-ca-g1](#) , [sanctum-pqc-ca-g1](#)) built for the Cygnus PQC console repository build. No certificate issued under this hierarchy should be relied upon for any production purpose. Section numbering follows RFC 3647 verbatim, including sections with "No stipulation," because auditors check for the presence of every numbered section.

1. Introduction

1.1 Overview

1.1.1. This Certificate Policy ("CP") specifies the policies and requirements governing the issuance, management, and use of public key certificates issued by the Sanctum SecOps Trust Foundation demonstration hierarchy that underlies the public PKI repository at [pki.sanctumsecops.com](#) and [crl.sanctumsecops.com](#).

1.1.2. This CP is written against the framework of RFC 3647, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*.

1.1.3. This CP covers a three-tier hierarchy: [sanctum-root-ca-g1](#) (root), [sanctum-issuing-ca-g1](#) (classical-algorithm issuing CA), and [sanctum-pqc-ca-g1](#) (post-quantum-flagged issuing CA). All three are LAB/TEST demonstration material as of this version; see Section 1.1 DRAFT NOTICE above.

1.1.4. This CP's own policy object identifier is `1.3.6.1.4.1.65953.23.1.1` (see Section 1.2). It is distinct from, and must not be confused with, the published production OID assignments in the Sanctum SecOps OID registry (`Sanc-Admin/sanctum-oid-registry`), including `sanctum-root-ca-g1 = 1.3.6.1.4.1.65953.1.1` , which identifies the real production root CA, not the LAB/TEST material described here.

1.2 Document name and identification

1.2.1. Document name: **Sanctum SecOps LLC Certificate Policy (CP)**.

1.2.2. Document OID: `1.3.6.1.4.1.65953.23.1.1` .

1.2.3. Version: 0.1 (DRAFT).

1.2.4. This CP cross-references, but does not redefine, the following already-published certificate policy OIDs from the production OID registry: `sanctum-policy-dv` (1.3.6.1.4.1.65953.2.1), `sanctum-policy-ov` (1.3.6.1.4.1.65953.2.2), `sanctum-policy-hv` (1.3.6.1.4.1.65953.2.3), `sanctum-policy-pqc-only` (1.3.6.1.4.1.65953.2.4), and `sanctum-policy-hybrid-composite` (1.3.6.1.4.1.65953.2.5).

1.3 PKI participants

1.3.1 Certification authorities

1.3.1.1. The demonstration hierarchy consists of `sanctum-root-ca-g1` , `sanctum-issuing-ca-g1` , and `sanctum-pqc-ca-g1` , as generated by `scripts/gen-ca.sh` .

1.3.2 Registration authorities

1.3.2.1. No stipulation. No separate registration authority function exists for this LAB/TEST demonstration hierarchy in v0.1.

1.3.3 Subscribers

1.3.3.1. For this demonstration hierarchy, the only subscriber is the sample leaf certificate `demo.pki.sanctumsecops.com` , issued for repository-plumbing demonstration purposes only.

1.3.4 Relying parties

1.3.4.1. Relying parties are any party that validates a certificate chaining to `sanctum-root-ca-g1` . Given the DRAFT/LAB status of this hierarchy, relying parties are advised not to extend any trust to it.

1.3.5 Other participants

1.3.5.1. No stipulation.

1.4 Certificate usage

1.4.1 Appropriate certificate uses

1.4.1.1. Demonstration and repository-plumbing purposes only: validating that AIA/CDP URIs resolve correctly over plain HTTP, and that the repository directory layout functions end to end.

1.4.2 Prohibited certificate uses

1.4.2.1. Certificates issued under this hierarchy must not be used to secure production traffic, must not be installed in any browser or OS trust store, and must not be relied upon for authentication, encryption, or code signing in any production system.

1.5 Policy administration

1.5.1 Organization administering the document

1.5.1.1. Sanctum SecOps LLC. Custodian: Brian Vicente, Founding Principal | Architect.

1.5.2 Contact person

1.5.2.1. Brian Vicente, brian@sso-labs.com (per the custodian contact on file in the published OID registry).

1.5.3 Person determining CPS suitability for the policy

1.5.3.1. Brian Vicente, pending formal policy management authority designation. DRAFT — not yet finalized.

1.5.4 CP approval procedures

1.5.4.1. DRAFT — no stipulation until Brian Vicente formally approves v1.0.

1.6 Definitions and acronyms

1.6.1. **CA** — Certification Authority. **CP** — Certificate Policy. **CPS** — Certification Practice Statement. **CDP** — CRL Distribution Point. **AIA** — Authority Information Access. **CRL** — Certificate Revocation List. **PQC** — Post-Quantum Cryptography. **ML-DSA** — Module-Lattice Digital Signature Algorithm (FIPS 204). **ML-KEM** — Module-Lattice Key Encapsulation Mechanism (FIPS 203). **SLH-DSA** — Stateless Hash-Based Digital Signature Algorithm (FIPS 205). **PEN** — Private Enterprise Number (IANA).

2. Publication and repository responsibilities

2.1 Repositories

2.1.1. The repository is published at `http://pki.sanctumsecops.com/` (AIA/ certificate retrieval) and `http://crl.sanctumsecops.com/` (CRL retrieval), served over plain HTTP per Section 4.9.7 and Section 6.

2.2 Publication of certification information

2.2.1. This CP, the associated CPS, all CA certificates, and all CRLs are published under `public/{docs,pem,aia,crl}/` in this repository.

2.3 Time or frequency of publication

2.3.1. DRAFT — no stipulation on a production publication cadence. For this lab hierarchy, CRLs are issued once at generation time by `scripts/gen-ca.sh` with a 7-day (issuing CA), 1-day (delta), and 365-day (root) validity window.

2.4 Access controls on repositories

2.4.1. The repository is intentionally public and world-readable. See Hard Rule 4 in the build context: all PKI repository directories are world-traversable (`o+x`) and files world-readable (`o+r`), verified by `scripts/fix-perms.sh` . No authentication gate is applied to `pki.` or `crl.` hostnames.

3. Identification and authentication

3.1 Naming

3.1.1 Types of names

3.1.1.1. X.501 Distinguished Names are used in all certificate subjects, e.g. `CN=demo.pki.sanctumsecops.com` .

3.1.2 Need for names to be meaningful

3.1.2.1. Subject CNs must be meaningful and correspond to the actual entity (hostname, CA name) being represented.

3.1.3 Anonymity or pseudonymity of subscribers

3.1.3.1. Not permitted. No stipulation beyond that for this demonstration hierarchy.

3.1.4 Rules for interpreting various name forms

3.1.4.1. No stipulation.

3.1.5 Uniqueness of names

3.1.5.1. Subject names must be unique within the scope of the issuing CA's `index.txt` database.

3.1.6 Recognition, authentication, and role of trademarks

3.1.6.1. No stipulation.

3.2 Initial identity validation

3.2.1 Method to prove possession of private key

3.2.1.1. Standard PKCS#10 CSR signature verification, per `openssl req -new`.

3.2.2 Authentication of organization identity

3.2.2.1. No stipulation for this LAB/TEST hierarchy — no real organization identity is validated; subjects are demonstration/lab material only.

3.2.3 Authentication of individual identity

3.2.3.1. No stipulation.

3.2.4 Non-verified subscriber information

3.2.4.1. All subscriber information in this demonstration hierarchy is non-verified lab material.

3.2.5 Validation of authority

3.2.5.1. No stipulation.

3.2.6 Criteria for interoperation

3.2.6.1. No stipulation.

3.3 Identification and authentication for re-key requests

3.3.1 Identification and authentication for routine re-key

3.3.1.1. No stipulation. Re-key is out of scope for this v0.1 demonstration hierarchy.

3.3.2 Identification and authentication for re-key after revocation

3.3.2.1. No stipulation.

3.4 Identification and authentication for revocation request

3.4.1. No stipulation for v0.1. A revocation runbook is planned for a future version of this CP, aligned with `sanctum-pki-cert-ops` operational guidance.

4. Certificate life-cycle operational requirements

4.1 Certificate application

4.1.1 Who can submit a certificate application

4.1.1.1. For this demonstration hierarchy, only the repository build process (`scripts/gen-ca.sh`) submits certificate applications (CSRs), for the sample leaf `demo.pki.sanctumsecops.com`.

4.1.2 Enrollment process and responsibilities

4.1.2.1. No stipulation beyond the automated CSR generation performed by `scripts/gen-ca.sh`.

4.2 Certificate application processing

4.2.1 Performing identification and authentication functions

4.2.1.1. No stipulation for this lab hierarchy; see Section 3.2.

4.2.2 Approval or rejection of certificate applications

4.2.2.1. No stipulation.

4.2.3 Time to process certificate applications

4.2.3.1. No stipulation.

4.3 Certificate issuance

4.3.1 CA actions during certificate issuance

4.3.1.1. Certificates are issued via `openssl x509 -req ... -CA ... -CAkey ...` as scripted in `scripts/gen-ca.sh`, embedding the applicable CDP and AIA extensions described in Section 4.9.7 and Section 7.

4.3.2 Notification to subscriber by the CA of issuance of certificate

4.3.2.1. No stipulation.

4.4 Certificate acceptance

4.4.1 Conduct constituting certificate acceptance

4.4.1.1. No stipulation.

4.4.2 Publication of the certificate by the CA

4.4.2.1. Issued CA certificates and the sample leaf are published in DER form under [public/aia/](#) and in PEM form under [public/pem/](#).

4.4.3 Notification of certificate issuance by the CA to other entities

4.4.3.1. No stipulation.

4.5 Key pair and certificate usage

4.5.1 Subscriber private key and certificate usage

4.5.1.1. Demonstration/lab use only. See Section 1.4.2 prohibited uses.

4.5.2 Relying party public key and certificate usage

4.5.2.1. Relying parties should not extend production trust to this hierarchy. See Section 1.3.4.

4.6 Certificate renewal

4.6.1 Circumstance for certificate renewal

4.6.1.1. No stipulation. Out of scope for v0.1.

4.6.2 Who may request renewal

4.6.2.1. No stipulation.

4.6.3 Processing certificate renewal requests

4.6.3.1. No stipulation.

4.6.4 Notification of new certificate issuance to subscriber

4.6.4.1. No stipulation.

4.6.5 Conduct constituting acceptance of a renewal certificate

4.6.5.1. No stipulation.

4.6.6 Publication of the renewal certificate by the CA

4.6.6.1. No stipulation.

4.6.7 Notification of certificate issuance by the CA to other entities

4.6.7.1. No stipulation.

4.7 Certificate re-key

4.7.1 through 4.7.7. No stipulation for v0.1. Certificate re-key is out of scope for this demonstration hierarchy.

4.8 Certificate modification

4.8.1 through 4.8.7. No stipulation for v0.1.

4.9 Certificate revocation and suspension

4.9.1 Circumstances for revocation

4.9.1.1. No stipulation for v0.1; the demonstration hierarchy does not currently revoke any certificate. `scripts/gen-ca.sh` provisions an `index.txt` entry for the sample leaf sufficient to support revocation tooling in a future version.

4.9.2 Who can request revocation

4.9.2.1. No stipulation.

4.9.3 Procedure for revocation request

4.9.3.1. No stipulation.

4.9.4 Revocation request grace period

4.9.4.1. No stipulation.

4.9.5 Time within which CA must process the revocation request

4.9.5.1. No stipulation.

4.9.6 Revocation checking requirement for relying parties

4.9.6.1. Relying parties that choose to validate this demonstration hierarchy's certificates should check the applicable CRL via the plain-HTTP CDP URIs embedded in the certificate (see Section 4.9.7), consistent with RFC 5280 guidance not to require HTTPS for CRL retrieval.

4.9.7 CRL issuance frequency

4.9.7.1. Base CRLs are issued with a 7-day validity for `sanctum-issuing-ca-g1` and `sanctum-pqc-ca-g1`, and a 365-day validity for the offline `sanctum-root-ca-g1`, matching typical offline-root operational practice (Reference Architecture point 1, `sanctum-pki-cert-ops`).

4.9.7.2. A delta-style CRL, `sanctum-issuing-ca-g1+.crl`, is published with a 1-day validity, following the widely-used "+" delta CRL naming convention. This is a documented limitation: it is not a byte-for-byte RFC 5280 formal Delta CRL (id-ce-deltaCRLIndicator, OID 2.5.29.27), because OpenSSL 3.5.5's `ca -gencrl` CLI does not support setting that extension via its config-file mechanism. See `scripts/gen-ca.sh` inline comments.

4.9.7.3. **All CDP URIs are plain `http://`, never `https://`.** This is a deliberate, non-negotiable design choice per RFC 5280 guidance: validating the TLS certificate of an HTTPS CRL-distribution host would itself require fetching a CRL to check that very host's certificate, creating a circular dependency. `crl.sanctumsecops.com` and `pki.sanctumsecops.com` must never have HSTS or an HTTP→HTTPS redirect.

4.9.8 Maximum latency for CRLs

4.9.8.1. No stipulation for v0.1.

4.9.9 On-line revocation/status checking availability

4.9.9.1. No stipulation. No OCSP responder is deployed for this demonstration hierarchy in v0.1.

4.9.10 On-line revocation checking requirements

4.9.10.1. No stipulation.

4.9.11 Other forms of revocation advertisements available

4.9.11.1. No stipulation.

4.9.12 Special requirements re: key compromise

4.9.12.1. No stipulation.

4.9.13 Circumstances for suspension

4.9.13.1. No stipulation. Certificate suspension is not implemented.

4.9.14 Who can request suspension

4.9.14.1. No stipulation.

4.9.15 Procedure for suspension request

4.9.15.1. No stipulation.

4.9.16 Limits on suspension period

4.9.16.1. No stipulation.

4.10 Certificate status services

4.10.1 Operational characteristics

4.10.1.1. Certificate status is available only via CRL at this time; see Section 4.9.9.

4.10.2 Service availability

4.10.2.1. No stipulation for v0.1 uptime commitments.

4.10.3 Optional features

4.10.3.1. No stipulation.

4.11 End of subscription

4.11.1. No stipulation.

4.12 Key escrow and recovery

4.12.1 Key escrow and recovery policy and practices

4.12.1.1. No key escrow is performed. No stipulation beyond this.

4.12.2 Session key encapsulation and recovery policy and practices

4.12.2.1. No stipulation.

5. Facility, management, and operational controls

5.1 Physical controls

5.1.1 through 5.1.8. No stipulation for v0.1. This demonstration hierarchy's key material is generated and stored only within this repository build sandbox and is not deployed to any physical facility.

5.2 Procedural controls

5.2.1 through 5.2.4. No stipulation for v0.1.

5.3 Personnel controls

5.3.1 through 5.3.8. No stipulation for v0.1.

5.4 Audit logging procedures

5.4.1 through 5.4.8. No stipulation for v0.1.

5.5 Records archival

5.5.1 through 5.5.8. No stipulation for v0.1.

5.6 Key changeover

5.6.1. No stipulation for v0.1.

5.7 Compromise and disaster recovery

5.7.1 through 5.7.4. No stipulation for v0.1.

5.8 CA or RA termination

5.8.1. No stipulation for v0.1.

6. Technical security controls

6.1 Key pair generation and installation

6.1.1 Key pair generation

6.1.1.1. Root CA: RSA 4096-bit. Issuing CAs (`sanctum-issuing-ca-g1` , `sanctum-pqc-ca-g1`): ECDSA P-384. Sample leaf: ECDSA P-256. All generated via `openssl genpkey` inside `scripts/gen-ca.sh`.

6.1.1.2. **PQC note:** `sanctum-pqc-ca-g1` is a demonstration CA flagged for future PQC signing. OpenSSL 3.5.5 as configured in this sandbox does not provide a default provider for ML-DSA CA-certificate signing operations invoked through the `openssl ca` / `openssl req` CLI used by `scripts/`

`gen-ca.sh`; the CA's own certificate is therefore signed with ECDSA P-384 today, with PQC/hybrid leaf issuance profiles defined in Section 7.1.3 for when a PQC-enabled provider is integrated. This is a documented gap, not a silent substitution.

6.1.2 Private key delivery to subscriber

6.1.2.1. No stipulation.

6.1.3 Public key delivery to certificate issuer

6.1.3.1. Standard PKCS#10 CSR.

6.1.4 CA public key delivery to relying parties

6.1.4.1. Via the public repository (`public/aia/` , `public/pem/`).

6.1.5 Key sizes

6.1.5.1. See 6.1.1.1. For the target production PQC profile (Section 7.1.3): ML-DSA-65 (FIPS 204), ML-KEM-768 (FIPS 203), SLH-DSA-128f (FIPS 205), and the X25519+ML-KEM-768 hybrid — matching the algorithm set surfaced by `/api/algorithms` in the Cygnus console.

6.1.6 Public key parameters generation and quality checking

6.1.6.1. No stipulation.

6.1.7 Key usage purposes

6.1.7.1. Per the `keyUsage` and `extendedKeyUsage` extensions defined in `scripts/gen-ca.sh`'s generated `openssl.cnf` templates (`v3_root` , `v3_intermediate_ca` , `v3_leaf`).

6.2 Private key protection and cryptographic module engineering controls

6.2.1 through 6.2.11. No stipulation for v0.1. This demonstration hierarchy's private keys are stored as plain files under `pki-repo/ca/*/private/` with mode 600, for build/demo purposes only — not FIPS 140-3 validated HSM storage. Production key custody requirements are deferred to a future CPS version aligned with `sanctum-pki-cert-ops` (offline air-gapped root, YubiKey/HSM key protection).

6.3 Other aspects of key pair management

6.3.1 Public key archival

6.3.1.1. No stipulation.

6.3.2 Certificate operational periods and key pair usage periods

6.3.2.1. Root: 20 years (7300 days). Issuing CAs: 10 years (3650 days). Sample leaf: 825 days. See `scripts/gen-ca.sh` `DAYS_ROOT`, `DAYS_ICA`, `DAYS_LEAF` constants.

6.4 Activation data

6.4.1 through 6.4.3. No stipulation for v0.1.

6.5 Computer security controls

6.5.1 through 6.5.2. No stipulation for v0.1.

6.6 Life cycle technical controls

6.6.1 through 6.6.3. No stipulation for v0.1.

6.7 Network security controls

6.7.1. No stipulation for v0.1. Note: `crl.sanctumsecops.com` and `pki.sanctumsecops.com` must be served over plain HTTP with no HSTS and no forced HTTPS redirect, per Section 4.9.7.3.

6.8 Time-stamping

6.8.1. No stipulation for v0.1. No RFC 3161 timestamp authority is deployed.

7. Certificate, CRL, and OCSP profiles

7.1 Certificate profile

7.1.1 Version number(s)

7.1.1.1. X.509 v3 for all CA and leaf certificates.

7.1.2 Certificate extensions

7.1.2.1. CA certificates carry `basicConstraints` (critical, `CA:true`), `keyUsage` (critical, `keyCertSign / cRLSign`), `subjectKeyIdentifier`, `authorityKeyIdentifier`, `crlDistributionPoints` (plain `http://`), and `authorityInfoAccess` (plain `http://`, `caIssuers`).

7.1.2.2. Leaf certificates additionally carry `extendedKeyUsage` (`serverAuth`, `clientAuth`) and `subjectAltName`.

7.1.3 Algorithm object identifiers

7.1.3.1. **Classical profile** (`sanctum-issuing-ca-g1`): RSA/ECDSA with SHA-256/SHA-384, as generated today.

7.1.3.2. **PQC profile** (target, `sanctum-pqc-ca-g1` , cross-referencing `sanctum-policy-pqc-only` = 1.3.6.1.4.1.65953.2.4 in the published registry):

- **ML-DSA-65** (FIPS 204) — primary post-quantum signature algorithm.
- **ML-KEM-768** (FIPS 203) — post-quantum key encapsulation mechanism.
- **SLH-DSA-128f** (FIPS 205) — stateless hash-based signature, fast variant, used as a conservative fallback signature algorithm.

7.1.3.3. **Hybrid profile** (target, cross-referencing `sanctum-policy-hybrid-composite` = 1.3.6.1.4.1.65953.2.5): **X25519 + ML-KEM-768** hybrid key exchange, pairing classical X25519 ECDH with ML-KEM-768 KEM shares for harvest-now-decrypt-later mitigation during the PQC transition period.

7.1.3.4. These four algorithm identifiers match the algorithm set exposed by the Cygnus console's `/api/algorithms` endpoint (`ML-KEM-768` , `ML-DSA-65` , `SLH-DSA-128f` , `X25519+ML-KEM-768`), and are registered informationally at `1.3.6.1.4.1.65953.23.4.1 – .23.4.4` in this repository's `registry.json` (see Section 1.2.4 and `public/oid/registry.json`).

7.1.4 Name forms

7.1.4.1. No stipulation.

7.1.5 Name constraints

7.1.5.1. No stipulation.

7.1.6 Certificate policy object identifier

7.1.6.1. This CP: `1.3.6.1.4.1.65953.23.1.1` . The companion CPS: `1.3.6.1.4.1.65953.23.1.2` .

7.1.7 Usage of Policy Constraints extension

7.1.7.1. No stipulation.

7.1.8 Policy qualifiers syntax and semantics

7.1.8.1. No stipulation.

7.1.9 Processing semantics for the critical Certificate Policies extension

7.1.9.1. No stipulation.

7.2 CRL profile

7.2.1 Version number(s)

7.2.1.1. X.509 v2 CRLs, as confirmed by `openssl crl -text` output showing `Version 2 (0x1)` (zero-indexed encoding) for all CRLs generated by `scripts/gen-ca.sh`.

7.2.2 CRL and CRL entry extensions

7.2.2.1. `X509v3 CRL Number` is present on all base and delta CRLs. See Section 4.9.7.2 for the documented delta-CRL-indicator limitation.

7.3 OCSP profile

7.3.1 Version number(s)

7.3.1.1. No stipulation. OCSP is not deployed in v0.1.

7.3.2 OCSP extensions

7.3.2.1. No stipulation.

8. Compliance audit and other assessments

8.1 Frequency and circumstances of assessment

8.1.1. No stipulation for v0.1.

8.2 Identity/qualifications of assessor

8.2.1. No stipulation.

8.3 Assessor's relationship to assessed entity

8.3.1. No stipulation.

8.4 Topics covered by assessment

8.4.1. No stipulation.

8.5 Actions taken as a result of deficiency

8.5.1. No stipulation.

8.6 Communication of results

8.6.1. No stipulation.

9. Other business and legal matters

9.1 Fees

9.1.1 through 9.1.5. No stipulation for v0.1. This demonstration hierarchy is issued at no cost and provides no commercial certificate issuance service.

9.2 Financial responsibility

9.2.1 through 9.2.3. No stipulation.

9.3 Confidentiality of business information

9.3.1 through 9.3.3. No stipulation.

9.4 Privacy of personal information

9.4.1 through 9.4.9. See the companion Privacy Policy at [public/policy/privacy-policy.md](#), itself DRAFT v0.1.

9.5 Intellectual property rights

9.5.1. No stipulation.

9.6 Representations and warranties

9.6.1 CA representations and warranties

9.6.1.1. No stipulation. This demonstration hierarchy makes no representations or warranties; see the DRAFT NOTICE at the top of this document.

9.6.2 RA representations and warranties

9.6.2.1. No stipulation.

9.6.3 Subscriber representations and warranties

9.6.3.1. No stipulation.

9.6.4 Relying party representations and warranties

9.6.4.1. See the companion Relying Party Agreement at [public/policy/relying-party-agreement.md](#) , DRAFT v0.1.

9.6.5 Representations and warranties of other participants

9.6.5.1. No stipulation.

9.7 Disclaimers of warranties

9.7.1. This demonstration hierarchy and this CP are provided "as is" with no warranty of any kind.
DRAFT — subject to change before v1.0.

9.8 Limitations of liability

9.8.1. No stipulation for v0.1.

9.9 Indemnities

9.9.1. No stipulation.

9.10 Term and termination

9.10.1 Term

9.10.1.1. This CP is effective only upon formal approval; as DRAFT v0.1 it has no legal effective date.

9.10.2 Termination

9.10.2.1. No stipulation.

9.10.3 Effect of termination and survival

9.10.3.1. No stipulation.

9.11 Individual notices and communications with participants

9.11.1. No stipulation.

9.12 Amendments

9.12.1 Procedure for amendment

9.12.1.1. No stipulation for v0.1. Future amendments will increment the version number in this document's header and be republished at [public/docs/CP.html](#) and [public/docs/CP.pdf](#) .

9.12.2 Notification mechanism and period

9.12.2.1. No stipulation.

9.12.3 Circumstances under which OID must be changed

9.12.3.1. No stipulation.

9.13 Dispute resolution provisions

9.13.1. No stipulation.

9.14 Governing law

9.14.1. No stipulation for v0.1. Anticipated: State of New York, United States, consistent with Sanctum SecOps LLC's registered address.

9.15 Compliance with applicable law

9.15.1. No stipulation.

9.16 Miscellaneous provisions

9.16.1 Entire agreement

9.16.1.1. No stipulation.

9.16.2 Assignment

9.16.2.1. No stipulation.

9.16.3 Severability

9.16.3.1. No stipulation.

9.16.4 Enforcement (attorneys' fees and waiver of rights)

9.16.4.1. No stipulation.

9.16.5 Force majeure

9.16.5.1. No stipulation.

9.17 Other provisions

9.17.1. No stipulation.

Generated from a single Markdown source (`docs/CP.md`) by `scripts/render-docs.py` , which produces both `public/docs/CP.html` and `public/docs/CP.pdf` so the two can never diverge. Do not hand-edit the HTML or PDF outputs.

Rendered 2026-07-27 11:56 UTC by `scripts/render-docs.py` from a single Markdown source. HTML and PDF outputs are generated from the same source and can never diverge.