Strategic Direction for DPKI Implementation

Complementing Conventional PKI with DPKI

Young B. Park

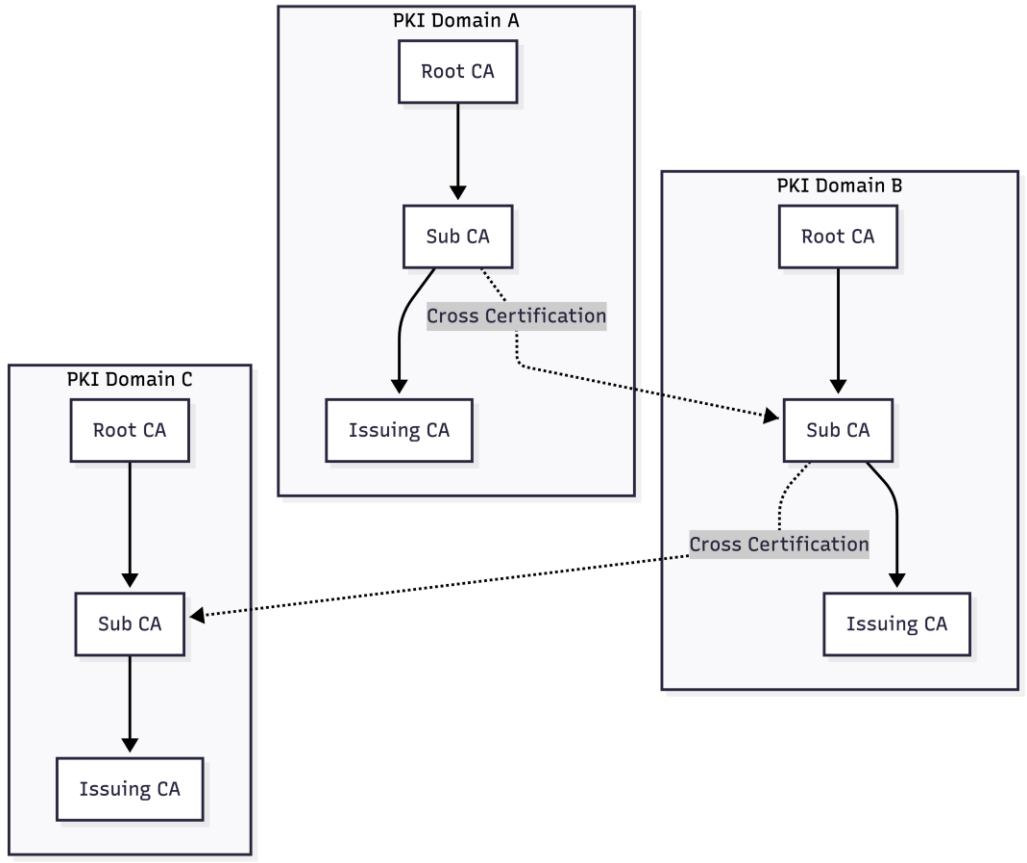
Dankook University, Korea

01 Issues in Conventional PKI

DPKI

Issues

- Long certificate chains
- Complex path validation
- Trust dilution
- Increased validation latency
- Operational dependency on CRL/OCSP
- Difficulty in worldwide PKI scalability



02 We still need to know: “Who issued and under what policy”

DPKI

◆ Established identity proofing and certificate issuance

- CA/RA processes already support identity validation with clearly assigned legal, organizational, and audit responsibilities.
- Certificates are issued under defined CP(Certificate Policy)/CPS(Certification Practice Statement) policies

◆ Mature X.509-based ecosystem

- Existing applications and relying parties are designed around PKI validation
- Replacing this ecosystem immediately is unrealistic

02 We still need to know: “Who issued and under what policy”

DPKI

Policy and compliance framework

- Conventional PKI provides governance through certificate policies, certification practice statements, audits, and operational controls
- These are necessary for accountability and regulatory acceptance

CA remains the authoritative source

- DPKI can distribute and verify certificates status information
- But the original authority for certificate usance, revocation, and lifecycle decisions still comes from CA/RA operations

03 Strategic Direction

DPKI

Preserve the existing PKI structure

- To ensure stable and secure interoperability across systems, the current PKI framework should remain intact with minimal structural modifications, Integrate decentralized PKI components only as extensions where necessary.

Prioritize CA security and operational integrity

- DPKI design should preserve the security and reliability of existing CAs with minimal operational changes. The introduction of a Decentralized CA (DeCA) enables a new CA model for connectivity and interoperability in decentralized network environments.

Retain core DPKI concepts

- The fundamental concepts of the existing DPKI model should be preserved to maintain security guarantees and compatibility with the current PKI ecosystem. CA-handled data should be categorized into creation, deletion, modification, and reference functions, with reference functionality distributed across the decentralized network and synchronized with DeCA-managed data to ensure consistency, integrity, and reliability.

04 Design Principle: Extend PKI, Do Not Replace it

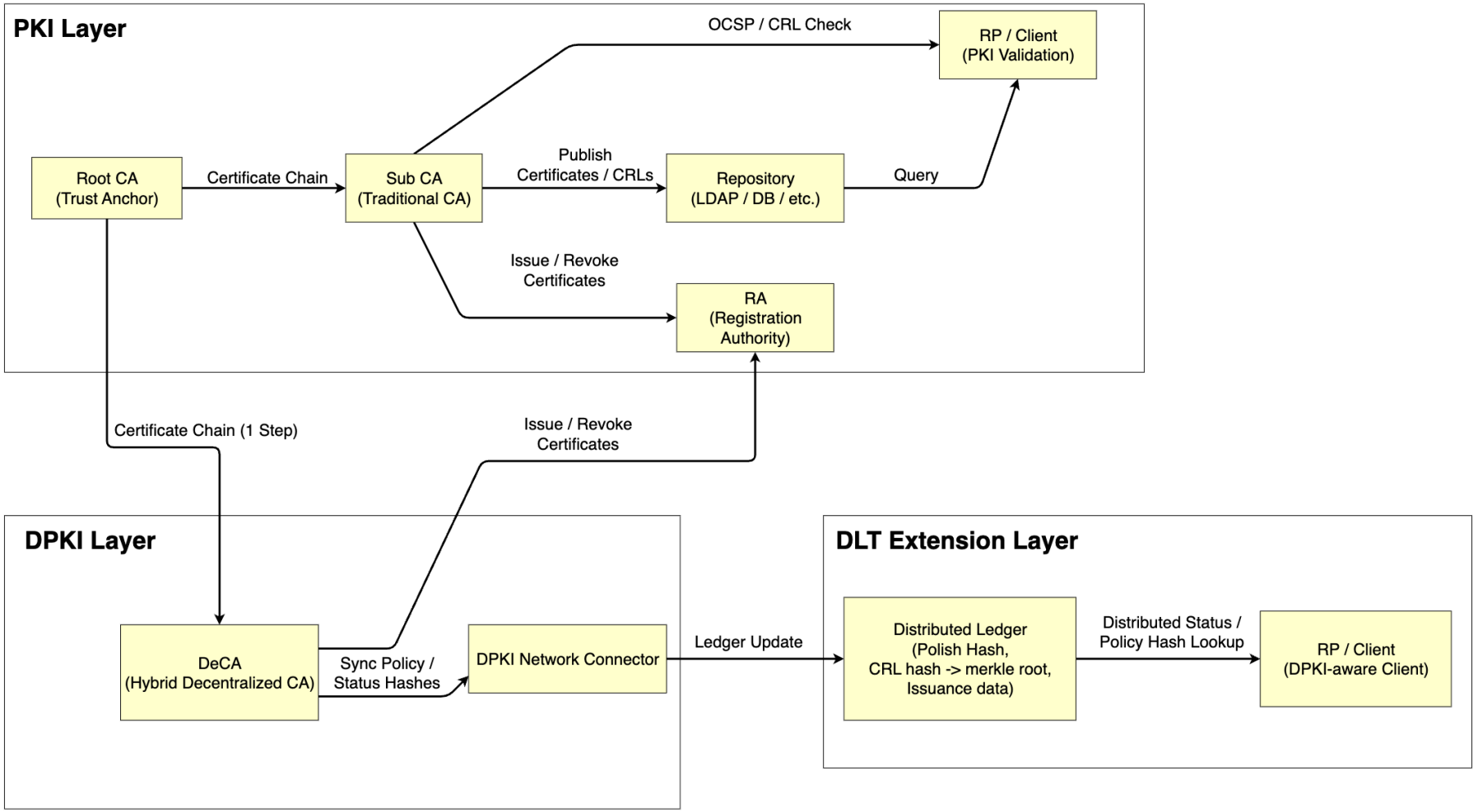
DPKI

- **The proposed implementation keeps the existing PKI structure intact while introducing DPKI and DLT components as extension layers.**
  - Existing **CA/RA** remains responsible for certificate issuance, revocation, and identity validation
  - **DeCA** bridges conventional PKI and decentralized trust infrastructure
  - **DPKI Network connector** synchronizes policy and certificate status hashes
  - **DLT layer** provides distributed lookup for certificate status and policy information
  - Relying parties can validate certificates through either conventional PKI services or DPKI-aware lookup

05 PKI-DPKI Interworking Architecture

DPKI

Overview



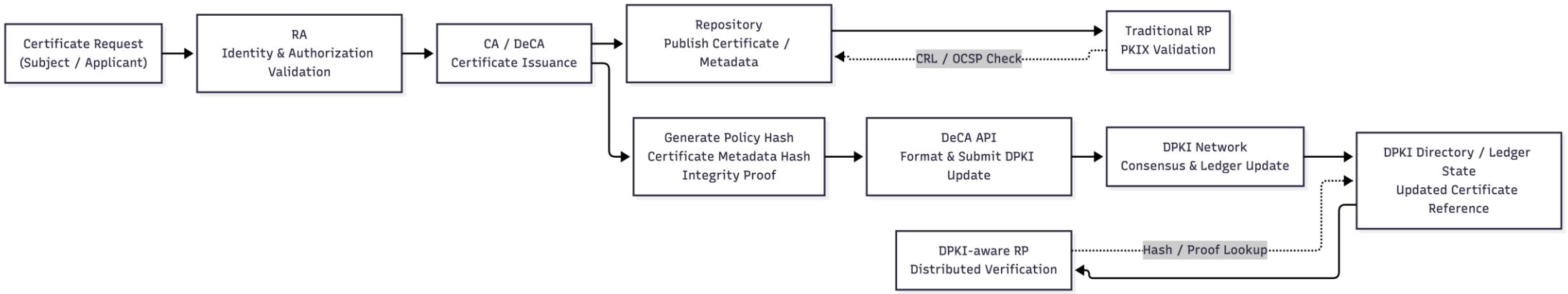
06 Components and Roles

DPKI

Component	Layer	Role
Root CA	Traditional PKI Layer	Serves as the trust anchor; issues certificates to subordinate CAs; maintains top-level CP/CPS; preserves the foundation of the X.509 trust hierarchy.
Sub CA	Traditional PKI Layer	Issues end-entity certificates; generates CRLs/OCSP responses; operates strictly under traditional PKI mechanisms; does not interact with the DPKI network.
RA (Registration Authority)	Traditional PKI Layer	Validates subject identity and authorization; approves certificate requests; assists in renewal and revocation identity checks; maintains registration audit logs.
Repository (LDAP/DB/etc.)	Traditional PKI Layer	Stores and distributes certificates, CRLs, and metadata; supports legacy relying parties; ensures backward compatibility with classic PKI clients.
RP (Traditional Relying Party)	Traditional PKI Layer	Performs certificate path validation using traditional PKIX rules; retrieves CRLs/OCSP data from the repository; does not use the DPKI ledger
DeCA (Decentralized CA)	DPKI Layer	Performs all functions of a traditional Sub CA while additionally generating and syncing Policy Hashes, CRL Hashes, and integrity proofs with the DPKI network.
DeCA API (DPKI Interface)	DPKI Layer	Acts as the interface between DeCA and the DPKI network; formats and publishes Policy Hashes, CRL Hashes, and proofs to the ledger; ensures secure state synchronization.
DPKI Network Layer	DLT Extension Layer	Maintains the decentralized ledger containing immutable policy, revocation, and proof data; supports validator, auditor, sync, and repository nodes; provides tamper-resistant transparency.
DPKI-aware RP	DLT Extension Layer	Performs traditional certificate validation plus decentralized verification using ledger-stored hashes and proofs; supports offline validation and improved trust transparency.

07 Certificate Lifecycle

DPKI



- **RA validates** the subject's identity and authorization before certificate issuance.
- **CA/DeCA issues** the certificate under the existing PKI lifecycle process.
- **Repository publishes** certificates and metadata for traditional relying parties.
- **DeCA generates** policy hashes, certificate metadata hashes, and integrity proofs.
- **DeCA API submits** DPKI updates to the decentralized network.
- **DPKI network records** validated references through consensus and ledger update.
- **DPKI-aware RP verifies** certificate references using ledger-stored hashes and proofs.

08 Role of Blockchain(DLT) in DPKI

DPKI

- **Blockchain does not replace PKI, but complements it**
- **Acts as a trust extension layer for PKI state commitment**
- **Blockchain(DLT) provides:**
  - State publication(policy hash & revocation)
  - Ordering of updates
  - Integrity and immutability
  - Public verifiability

09 Core Responsibilities of the Blockchain(DLT)

DPKI

State Commitment

- Publishes hashes of:
  - Policy
  - Revocation set(e.g., Hash tree root)

Ordering

- Ensures consistent sequence of updates
- Defines the latest valid state

09 Core Responsibilities of the Blockchain(DLT)

DPKI

Integrity

- Detects any tampering via hash mismatch

Verification Achor

- Serves as the reference point for proof validation

Auditability

- Enables full lifecycle traceability

10 Data stored On-Chain

DPKI

Minimal Commitment Data Only

◆ Policy Hash

- H(policy_document)

◆ Revocation Root

- HashTreeRoot(revoked_serials)

◆ Metadata:

- Issuer ID
- Epoch / Version
- Update Timestamps

◆ Proof Anchors

- Hash tree roots / Hash references

11 Data Kept Off-Chain

DPKI

Do NOT store:

- Full X.509 certificates
- Personal Identity Information
- Full CRL / revocation lists
- Policy documents (full text)
- Proof data (Hash tree paths)

Reasons

- Scalability
- Privacy protection
- Performance efficiency
- Reduced storage overhead

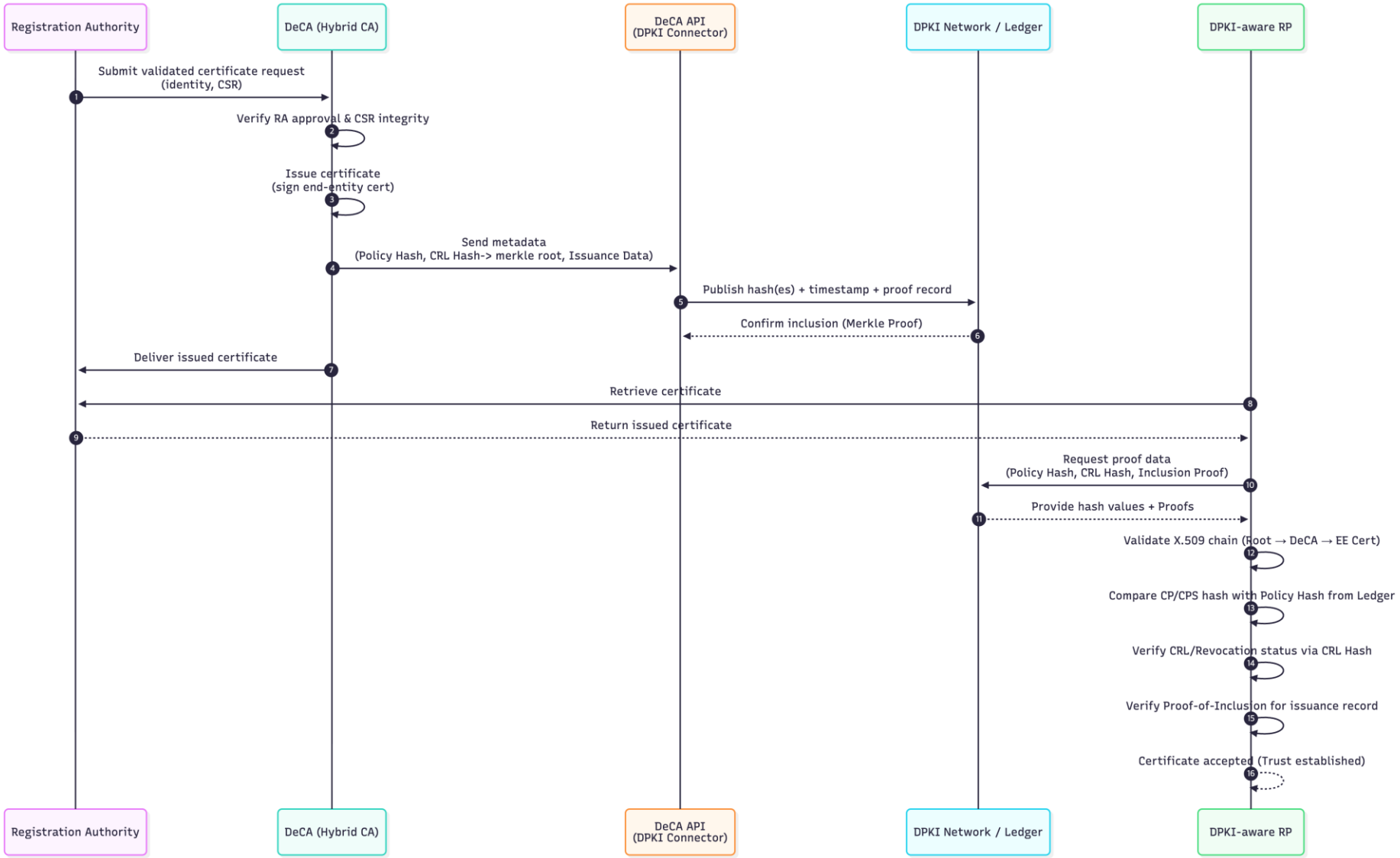
12 Smart Contract vs Direct Ledger Storage

DPKI

Aspect	Smart Contract Storage	Direct Ledger
Execution Model	On-chain logic execution	No execution (data-only)
Complexity	High	Low
Gas Cost	High	Low (or gas-free)
Flexibility	Dynamic logic possible	Fixed data structure
Security Surface	Larger (code vulnerabilities)	Smaller
Transparency	Requires contract interpretation	Directly readable
Performance	Slower	Faster
Suitability for DPKI	Overkill	Highly suitable
Network example	Ethereum, Hyperledger Besu	Bitcoin, Hyperledger Besu

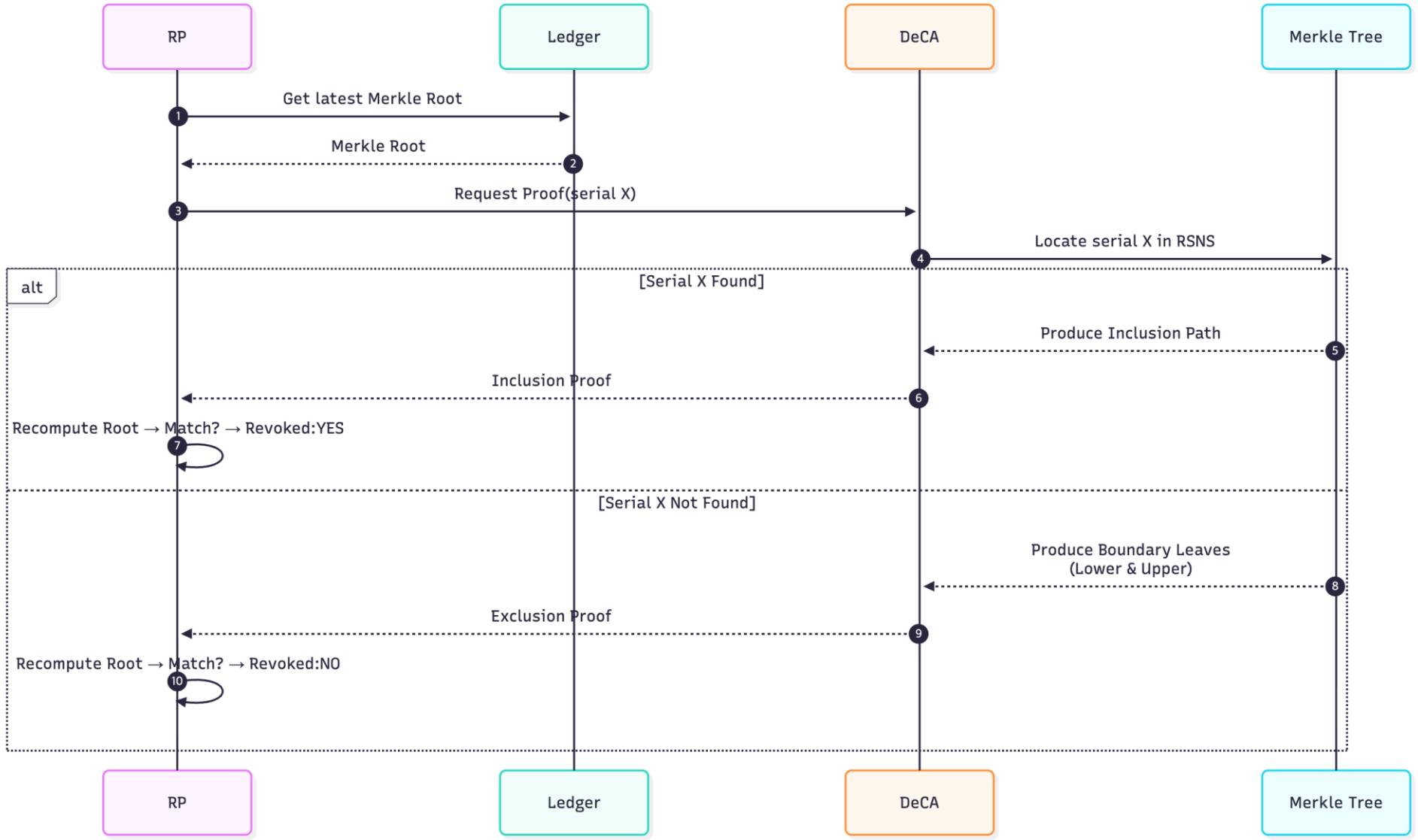
13 Certificate Issuance & Verification vid DeCA with DPKI Network

DPKI



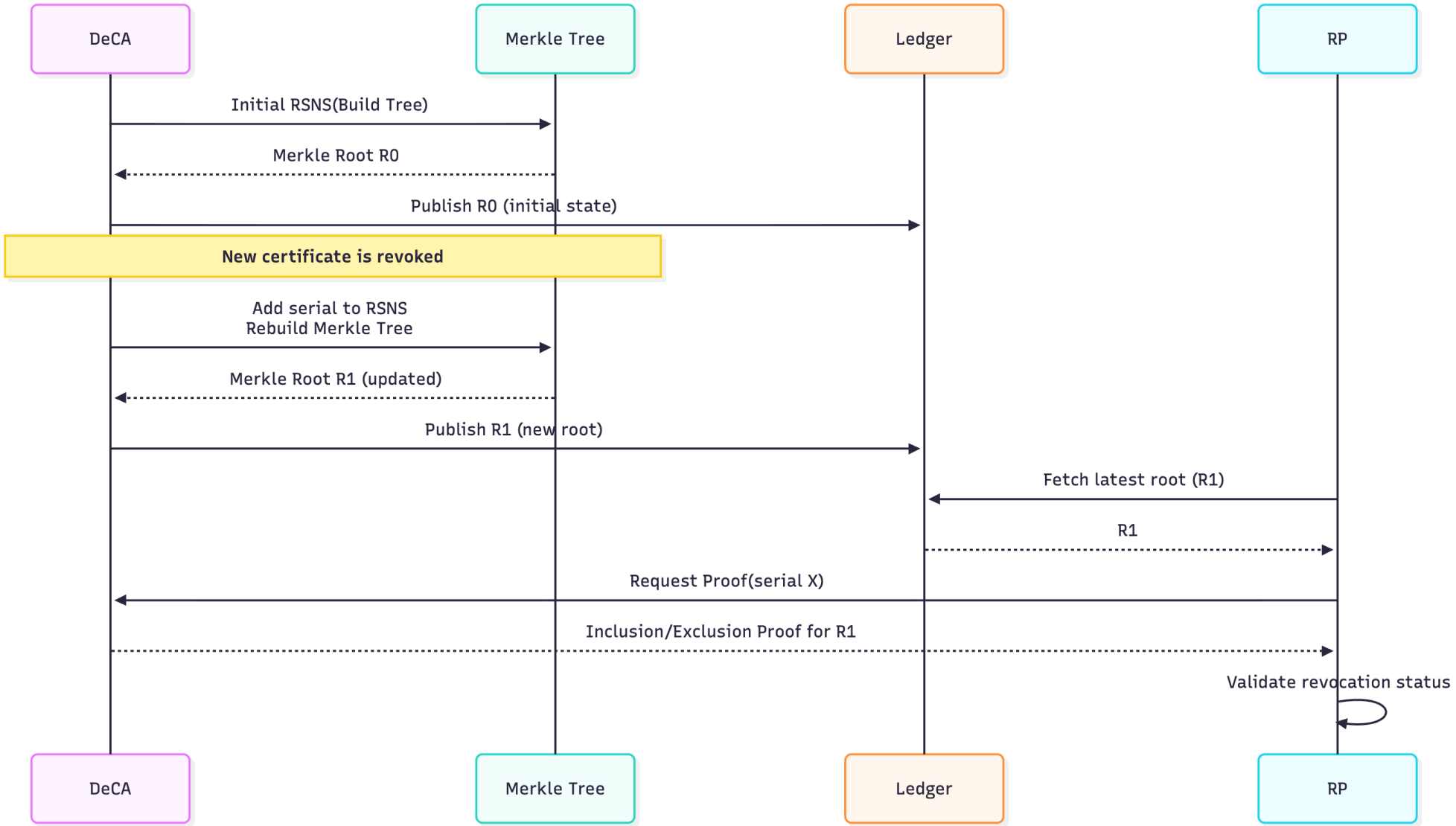
14 Inclusion & Exclusion Proof Sequence

DPKI



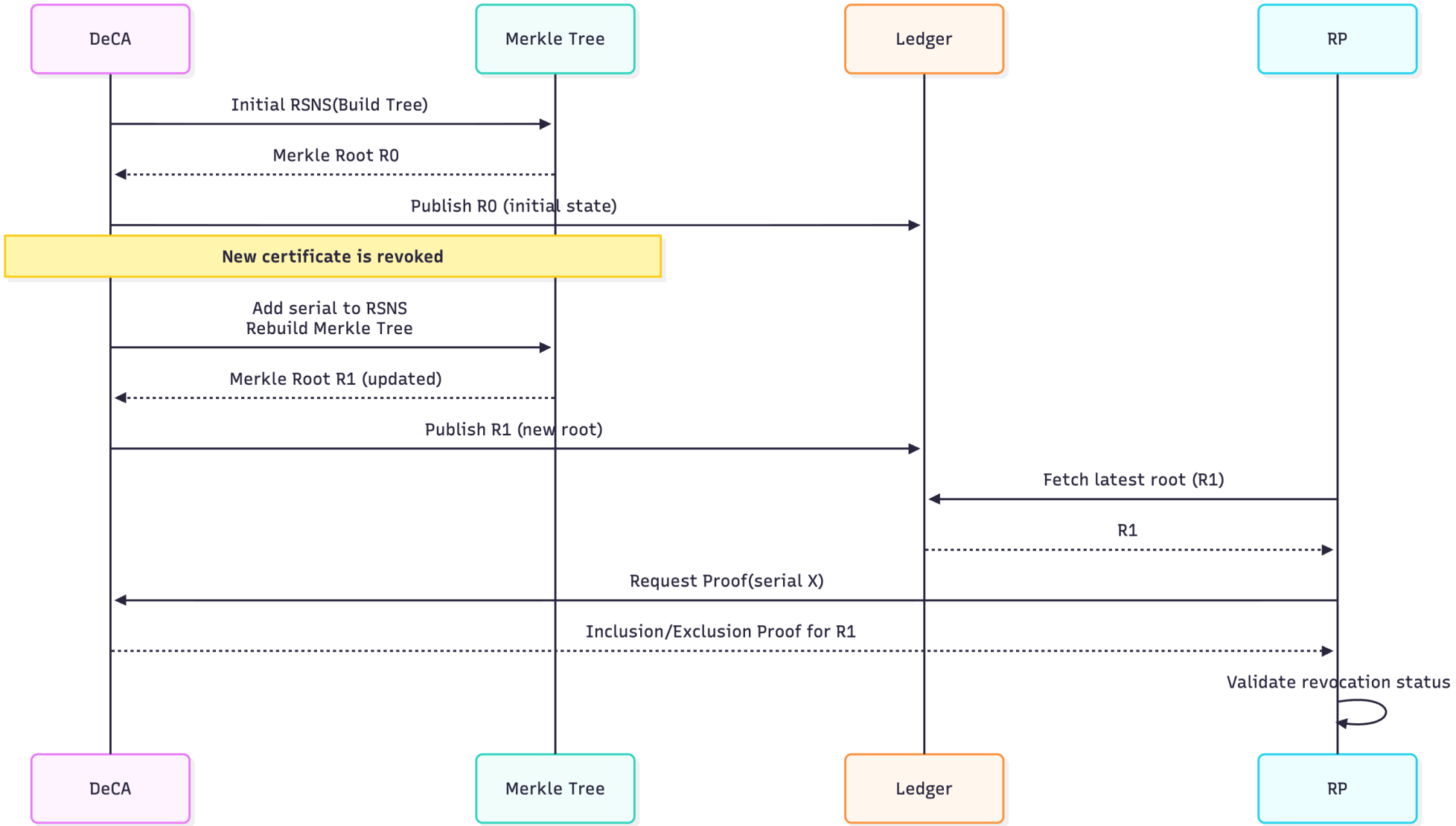
15 Update CRL Hash Tree

DPKI



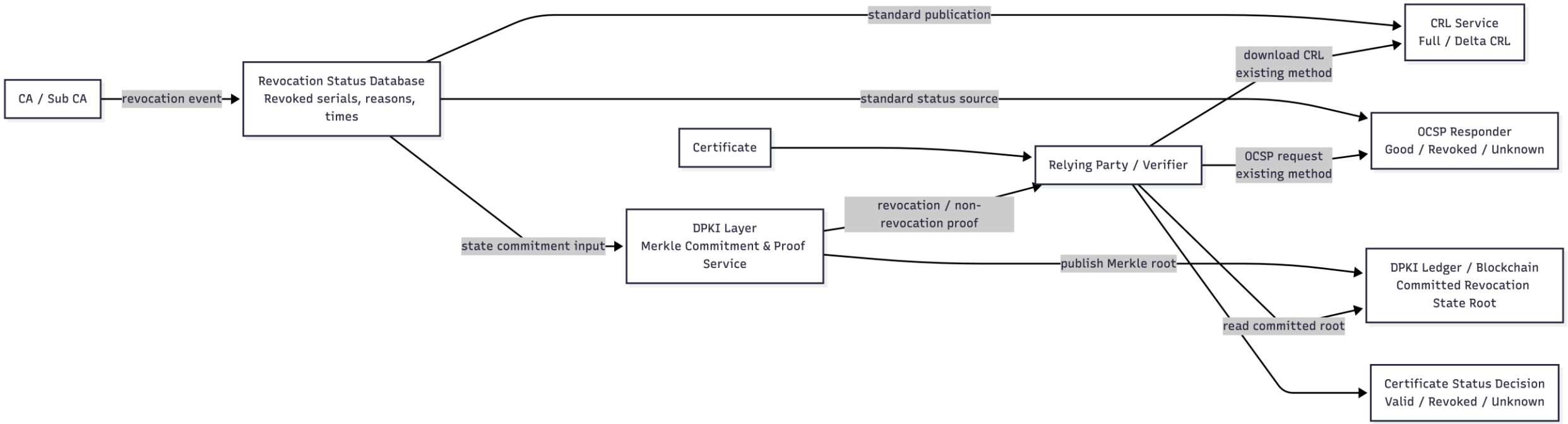
16 Update CRL Hash Tree

DPKI



16 CRL/OCSP and DPKI-Based Revocation Verification Flow

DPKI



16 Revocation Verification Methods

DPKI

Aspect	CRL	OCSP	DPKI Merkle Commitment
Verification method	Download a full or delta revocation list and search for the certificate serial number	Query an OCSP responder for the status of a specific certificate serial number	Verify a compact proof against a Hash tree root committed to the DPKI ledger
Network dependency	Requires access to a CRL distribution point	Requires real-time access to an OCSP responder	Requires access to the committed ledger root and a revocation or non-revocation proof
Data size	Increases as the number of revoked certificates grows	Small per-certificate response	Small proof size, typically logarithmic to the committed revocation set
Freshness	Depends on the CRL publication interval	Relatively real-time	Depends on the ledger update interval
Privacy	The CA does not necessarily learn which certificate is being checked	The responder can observe which certificate is being queried	Can reduce query exposure depending on how proofs are distributed
Centralized bottleneck	CRL repository or distribution server	OCSP responder	Ledger consensus layer or proof provider
Integrity basis	CA-signed CRL	CA- or OCSP-signed response	Ledger-committed revocation state root
Main advantage	Simple and widely supported	Efficient per-certificate status checking	Transparent, auditable, and suitable for distributed verification
Main limitation	Large CRLs and stale status information	Availability and privacy concerns	Requires ledger integration and proof management
Suitable use case	Batch or offline-style revocation checking	Online, near-real-time certificate status checking	Verifiable revocation transparency and decentralized PKI validation

Thank you

End Of Document