PARCEP PROJECT

Analysis 04

High-Level Design, Schema, and Synchronization Protocol

A distinguished-engineer view of the X.PARCEP solution: a federated architecture for Child Online Protection across device, network, and platform; a complete data schema; and the PARCEP-Sync protocol designed for IETF development.

Prepared for: X.PARCEP editors (Broadcom Europe Ltd. / Vodafone Group), ITU-T SG17 Q1/Q3
Date: May 18, 2026 **Status:** Draft v0.1 for discussion

Executive Summary

This document is the high-level design companion to the PARCEP requirements catalogue (Analysis 03). It presents the architecture of a federated Child Online Protection (COP) solution, the data schema needed to express household policies, credentials and operational artifacts, and the PARCEP-Sync synchronization protocol that runs between the Policy Administration Point (PAP), the Policy Decision and Enforcement Points (PDP and PEP), and the federated Policy Information Points (PIPs). The protocol is specified at a level of detail that is intended to be carried into the IETF for full standardization.

Three architectural decisions anchor the design. First, the trust model is federated: jurisdictional PIPs (rather than a single global authority) issue the credentials, taxonomies, and signed classifier-model packages that PDPs need. Second, the household policy is authored and stored at the guardian's PAP; no central database of children's data is created. Third, policy distribution and event reporting between the PAP and the household's PEPs run inside a per-household Messaging Layer Security (MLS, IETF RFC 9420) group, providing forward secrecy, post-compromise security, asynchronous group operations, and natural support for adding or removing PEPs as the dependant acquires new devices or as co-guardians join.

The data schema is split into five layers — identity, credentials, policy, operational, and discovery — and uses verifiable credentials (SD-JWT-VC and ISO/IEC 18013-5 mDL) compatible with eIDAS 2 / EUDI Wallet for the cross-jurisdictional pieces (guardianship, age, classifier-model attestation). The wire format is JSON-LD for human-readable, web-compatible exchanges (policy authoring, web dashboards, regulatory audit) and CBOR/COSE (RFCs 8949 / 9052) for compact, embedded-friendly exchanges (event telemetry, network-side enforcement signals, IoT-class PEPs).

The PARCEP-Sync protocol defines fourteen methods — ENROLL, WELCOME, PUBLISH, UPDATE, REVOKE, ACK, EVENT, DIGEST, APPEAL, RESOLVE, ATTEST, FETCH, NOTIFY, plus an extensibility primitive — with a small, regular request-response semantics modeled on the IETF MIMI (More Instant Messaging Interoperability) and MLS architectures. Discovery uses DNS SVCB records (RFC 9460) and a .well-known/parcep endpoint; authentication uses MLS credentials backed by guardianship and device VCs. The protocol is

transport-agnostic above the security boundary; the canonical transport is MLS over QUIC for the household group, with mTLS over HTTPS as a discovery/enrollment bootstrap.

Section 11 closes with the path to ITU-T and IETF specification, framed as a coordinated track between ITU-T SG17 (the X.PARCEP Recommendation, the data model and conformance profiles) and a new IETF Working Group (the PARCEP-Sync wire protocol, the MLS profile, the DNS discovery).

1. Architecture overview

PARCEP is a federated, four-tier system. Tier 1 is the federation of jurisdictional trust anchors (PIPs) that issue and attest the credentials, taxonomies, and signed classifier-model packages on which the rest of the system depends. Tier 2 is the Policy Administration Point (PAP), guardian-controlled, where the household policy is authored, stored, and signed. Tier 3 is the Policy Decision and Enforcement layer (PDP + PEP), distributed across device, application, network, and platform constituencies; each PEP subscribes to the household policy and enforces it locally. Tier 4 is the dependant — the rights-bearing subject of the policy, who receives the transparency notices and exercises the appeal channel mandated by the requirements catalogue.

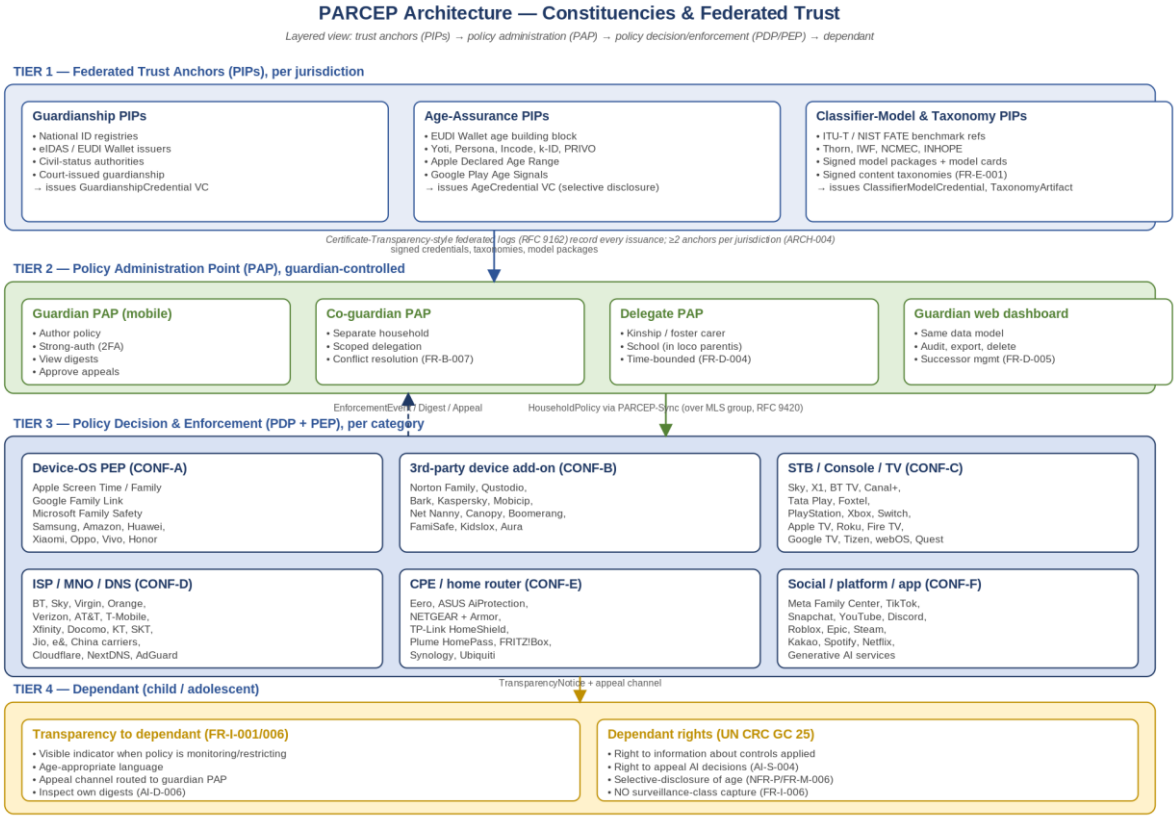

Figure 1 — PARCEP constituencies and federated trust topology (Diagram D1).

The federated design is the central architectural choice. A purely centralized system would concentrate the most sensitive class of data in existence — children's identities, ages, household policies, and enforcement events — in one place, contradicting the X.PARCEP non-goal of avoiding a centralized database of children's data. A purely decentralized peer-to-

peer system would lack the anchoring needed for content taxonomies, classifier models, and identity-grade age credentials. The federation pattern, modeled on DNS PKI, eIDAS 2, Certificate Transparency, and the IETF MLS/MIMI work, allows multiple independent jurisdictional roots, public auditability of issuances, and per-jurisdiction regulatory adaptation, while keeping the household policy under the guardian's control.

2. Functional decomposition

The functional decomposition follows the IETF RFC 3198 / ISO/IEC 29146 policy-management terminology. The PAP is the guardian-controlled authoring surface; the PDP makes per-request decisions against the policy; the PEP enforces the decisions; the PIP attests credentials and signs distributed artifacts.

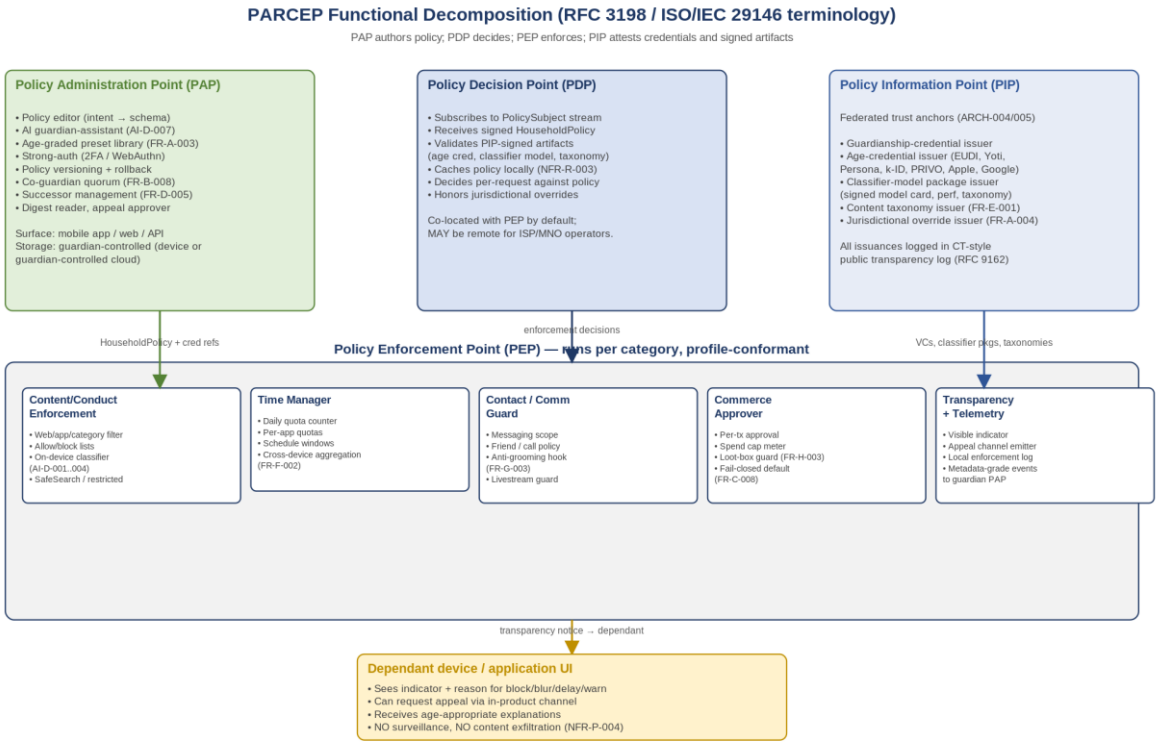

Figure 2 — PAP, PDP, PEP, and PIP — functional decomposition (Diagram D2).

2.1 Policy Administration Point (PAP)

The PAP is implemented as a guardian-controlled application (mobile, web, or a guardian-controlled cloud service). It must authenticate the guardian using strong (two-factor) authentication and must persist the canonical household policy under the guardian's authority. The PAP offers an age-graded preset library to make low-friction onboarding possible (FR-A-003) and may offer an AI guardian-assistant that renders a natural-language policy intent into a draft policy for guardian review (FR-A-005 / AI-D-007). The PAP is the source of all PUBLISH, UPDATE, REVOKE, and RESOLVE messages.

2.2 Policy Decision Point (PDP)

The PDP is, by default, co-located with the PEP. It receives the signed HouseholdPolicy from the PAP, validates the PIP-signed artifacts referenced by the policy (classifier-model

packages, taxonomies, jurisdictional overrides), caches the policy locally for offline operation, and evaluates the policy against runtime requests. For ISP / MNO operators, the PDP MAY be remote (co-located with the operator's policy engine) while the PEP remains in the access network.

2.3 Policy Enforcement Point (PEP)

The PEP is the component that blocks, allows, paces, delays, blurs, or warns. It is implemented per category (CONF-A through CONF-G in the requirements catalogue) and decomposes into five reusable sub-modules: content/conduct enforcement (including on-device AI classifiers), the time manager, the contact/communication guard, the commerce approver, and the transparency-and-telemetry emitter. The PEP is also the source of all EVENT messages and the entry point for APPEAL messages from the dependant.

2.4 Policy Information Point (PIP)

The PIPs are federated trust anchors operating per jurisdiction. They issue and attest the credentials and artifacts that PDPs need: guardianship credentials, age credentials, classifier-model packages, content taxonomies, jurisdictional override descriptors. Every issuance is recorded in a public transparency log modeled on IETF RFC 9162 (Certificate Transparency v2). At least two independent PIPs are required per jurisdictional function (ARCH-004) so that no single PIP failure or compromise blocks enforcement.

3. Federated trust model

PARCEP's trust assumptions follow Internet-PKI conventions, extended for selective-disclosure verifiable credentials. The root trust set is, per jurisdiction, a small federation of PIPs. The PIPs issue verifiable credentials (SD-JWT-VC and mDL formats) that PEPs verify offline against pinned PIP roots. Every issuance is publicly logged so that silent issuance or rogue PIP behavior is detectable by independent auditors.

3.1 Credential chain

Three credential chains converge at the PEP:

- Guardianship chain: PIP → GuardianshipCredential → guardian's PAP signature on the HouseholdPolicy.
- Age chain: PIP → AgeCredential → dependant's selective-disclosure presentation at the PEP ("holder is ≥13 in jurisdiction X").
- Classifier-model chain: PIP → ClassifierModelCredential → model package consumed by the on-device classifier; model package signature verified at load time.

3.2 Revocation

PARCEP REQUIRES every credential to carry a revocation endpoint. Two revocation mechanisms are supported: short-lived credential validity (where the credential is re-issued frequently and revocation is implicit by non-renewal); and status-list endpoints (W3C VC Status List 2021 / IETF Token Status List) for long-lived credentials. PEPs MUST check revocation at policy load time and at a policy-defined refresh interval.

3.3 Cross-jurisdictional verification

A PEP operating in jurisdiction A may receive a credential issued in jurisdiction B (e.g., a dependant traveling, a school in a different country, a guardian on a foreign assignment). PARCEP defines a cross-jurisdictional verification rule: a PEP MUST accept a credential from any PIP whose root is on a PEP-recognized trust list. The trust list is, by default, the union of the PEP's home jurisdiction trust list and the trust lists explicitly imported by the guardian via the PAP.

4. Data schema

The schema is split into five layers: identity (Guardian, CoGuardianRelation, Dependant, Device, AccountBinding); credentials (GuardianshipCredential, AgeCredential, ClassifierModelCredential, TaxonomyArtifact, JurisdictionalOverride); policy (HouseholdPolicy, PolicySubject, plus the six clause types: ContentPolicy, TimePolicy, ContactPolicy, CommercePolicy, ConductPolicy, TransparencyPolicy); operational (EnforcementEvent, TransparencyNotice, AppealRequest, GuardianDigest); and discovery (PEPRegistration, PIPDescriptor).

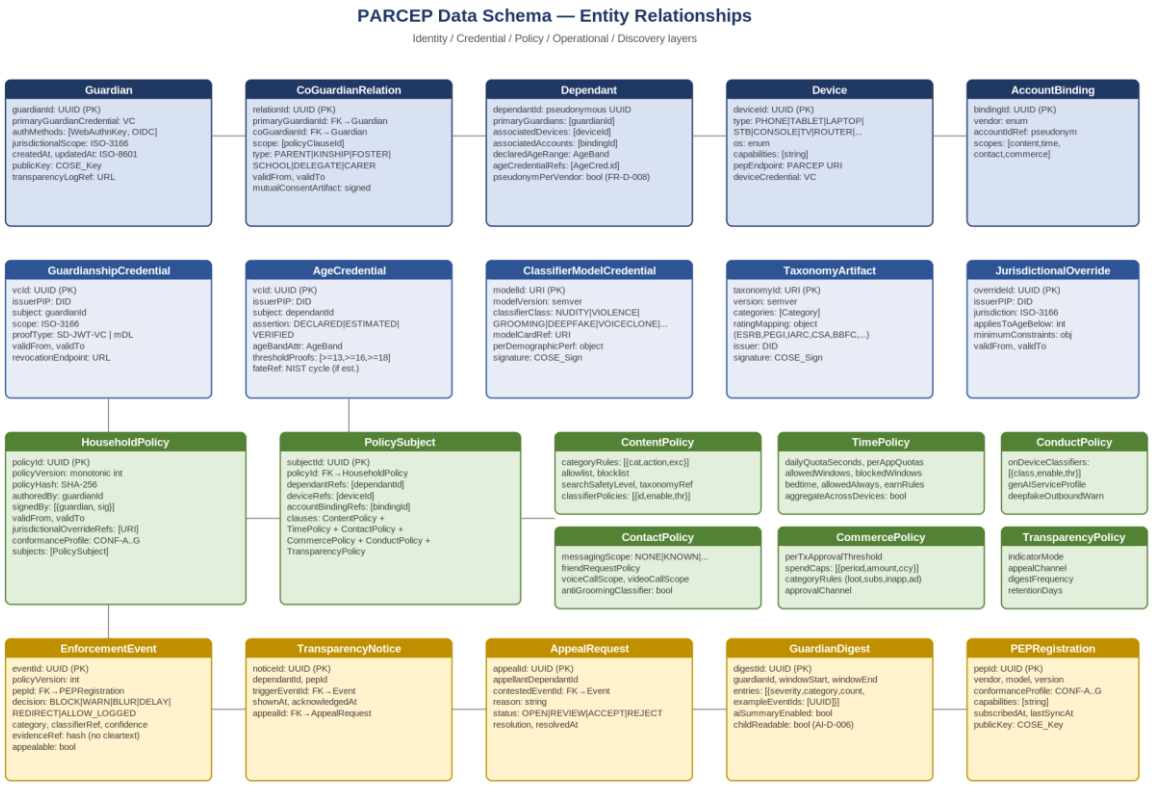

Figure 3 — Entity-relationship diagram (Diagram D3).

4.1 Identity layer

```
// Guardian — a responsible adult with policy-authoring authority
Guardian {
  guardianId           : UUID                       // canonical PK
  primaryCredential    : GuardianshipCredential     // VC from PIP
  authMethods          : [WebAuthnKey | OIDCBinding]
```

```
  jurisdictionalScope  : [ISO-3166-1-alpha-2]
  publicKey            : COSE_Key                  // for PARCEP-Sync signing
  transparencyLogRef   : URI                       // CT-style log entry
  createdAt, updatedAt : Timestamp (RFC 3339)
}

// CoGuardianRelation — binds two guardians for the same dependant set
CoGuardianRelation {
  relationId           : UUID                      // PK
  primaryGuardianId    : UUID  → Guardian.guardianId
  coGuardianId         : UUID  → Guardian.guardianId
  scope                : [PolicyClauseId]          // subset they may administer
  type                 : enum { PARENT, KINSHIP, FOSTER, SCHOOL, DELEGATE, CARER }
  validFrom, validTo   : Timestamp                 // time-bounded (FR-D-004)
  mutualConsent        : COSE_Sign1                // both signatures
}

// Dependant — the rights-bearing subject of the policy
Dependant {
  dependantId          : UUID                      // pseudonymous PK
  primaryGuardians     : [UUID  → Guardian.guardianId]
  associatedDevices    : [UUID  → Device.deviceId]
  associatedAccounts   : [UUID  → AccountBinding.bindingId]
  declaredAgeRange     : AgeBand                   // guardian-asserted; FR-M-001
  ageCredentialRefs    : [URI   → AgeCredential.vcId]
  pseudonymPerVendor   : bool                      // FR-D-008
}

// Device — a device the dependant uses
Device {
  deviceId             : UUID                      // PK
  type                 : enum { PHONE, TABLET, LAPTOP, DESKTOP, STB, CONSOLE,
                                TV, ROUTER, SMART_SPEAKER, XR_HEADSET, OTHER }
  os                   : enum
  capabilities         : [string]                  // FR-K capability tags
  pepEndpoint          : URI                       // for direct PARCEP-Sync
  deviceCredential     : DeviceCredential          // VC, optional
}

// AccountBinding — a per-vendor account the dependant uses (Apple ID, Google,
// Meta, TikTok, etc.) — pseudonymised to prevent cross-vendor correlation.
AccountBinding {
  bindingId            : UUID
  vendor               : enum
  accountIdRef         : pseudonym                 // not the raw account id
  scopes               : [enum { CONTENT, TIME, CONTACT, COMMERCE, CONDUCT }]
}
```

4.2 Credentials layer

```
// GuardianshipCredential — VC asserting that a guardian has authority over
// a specified jurisdictional scope. Issued by a PIP.
GuardianshipCredential {
  vcId                 : URI                       // PK
  issuerPIP            : DID  → PIPDescriptor.pipId
  subject              : UUID → Guardian.guardianId
  scope                : [ISO-3166-1-alpha-2]
  proofFormat          : enum { SD_JWT_VC, ISO_MDL, JSON_LD_VC }
  validFrom, validTo   : Timestamp
  revocationEndpoint   : URI
}

// AgeCredential — selective-disclosure VC asserting an age band and/or
// the boolean answers to age-threshold predicates (e.g., "holder >= 13").
AgeCredential {
```

```
  vcId                 : URI                          // PK
  issuerPIP            : DID
  subject              : UUID → Dependant.dependantId   // never disclosed
                                                          // to a verifier
  assertion            : enum { DECLARED, ESTIMATED, VERIFIED }
  ageBandAttribute     : AgeBand                      // disclosable
  thresholdProofs      : { ">=13": bool, ">=16": bool, ">=18": bool }
  benchmarkRef         : NIST_FATE_Cycle | null       // when ESTIMATED
  heaaConformance      : bool                         // when VERIFIED (Ofcom HEAA)
  validFrom, validTo   : Timestamp
  revocationEndpoint   : URI
}

// ClassifierModelCredential — VC binding a signed classifier model package
// to a class (NUDITY, GROOMING, DEEPFAKE, VOICECLONE, ...) and to a PIP.
ClassifierModelCredential {
  modelId              : URI                          // PK
  modelVersion         : SemVer
  classifierClass      : enum { NUDITY, VIOLENCE, SELF_HARM, HATE,
                                GROOMING, DEEPFAKE, VOICECLONE, MALWARE, OTHER }
  modelCardRef         : URI                          // training, perf, ethics
  perDemographicPerf   : { skinTone: {...}, ageBand: {...}, gender: {...} }
  adversarialRobustness: { tested: bool, methods: [string] }
  signature            : COSE_Sign1                   // by issuerPIP
}

// TaxonomyArtifact — signed content-category taxonomy plus the cross-
// regional rating mapping required by FR-E-003.
TaxonomyArtifact {
  taxonomyId           : URI                          // PK
  version              : SemVer
  categories           : [Category]
  ratingMapping        : { ESRB: {...}, PEGI: {...}, IARC: {...}, CSA: {...},
                           BBFC: {...}, CERO: {...}, USK: {...}, ACB: {...} }
  issuer               : DID
  signature            : COSE_Sign1
}

// JurisdictionalOverride — minimum policy constraints set by a regulator
// (e.g., CAC for under-14 in CN, ARCOM for under-18 in FR).
JurisdictionalOverride {
  overrideId           : URI                          // PK
  issuerPIP            : DID
  jurisdiction         : ISO-3166-1-alpha-2
  appliesToAgeBelow    : int
  minimumConstraints   : {
    time:    { maxDailyMinutes, curfewStart, curfewEnd, ... },
    content: { mandatoryBlockCategories: [...], ... },
    contact: { mandatoryRestrictions: [...] },
    commerce:{ noPurchase: bool, noLootBox: bool, ... }
  }
  validFrom, validTo   : Timestamp
}
```

4.3 Policy layer

```
// HouseholdPolicy — the signed root object the PAP publishes
HouseholdPolicy {
  policyId             : UUID                         // immutable
  policyVersion        : int                          // monotonic per policyId
  policyHash           : SHA-256                      // for ACK matching
  authoredBy           : UUID → Guardian.guardianId
  signedBy             : [{ guardianId, signature: COSE_Sign1 }]
  validFrom, validTo   : Timestamp
  jurisdictionalOverrideRefs: [URI → JurisdictionalOverride.overrideId]
```

```
  conformanceProfile   : enum { CONF_A, CONF_B, CONF_C, CONF_D,
                                CONF_E, CONF_F, CONF_G }
  subjects             : [PolicySubject]
  extensions           : [Extension]               // namespaced
}

// PolicySubject — binds a clause-set to a dependant + device + account set
PolicySubject {
  subjectId            : UUID
  policyId             : UUID → HouseholdPolicy.policyId
  dependantRefs        : [UUID → Dependant.dependantId]
  deviceRefs           : [UUID → Device.deviceId]
  accountBindingRefs   : [UUID → AccountBinding.bindingId]
  clauses : {
    content      : ContentPolicy,
    time         : TimePolicy,
    contact      : ContactPolicy,
    commerce     : CommercePolicy,
    conduct      : ConductPolicy,
    transparency : TransparencyPolicy
  }
}

ContentPolicy {
  taxonomyRef          : URI → TaxonomyArtifact.taxonomyId
  categoryRules        : [{ category, action: ALLOW|BLOCK|WARN, exceptions: [URI] }]
  allowlist, blocklist : [URI pattern]
  searchSafetyLevel    : enum { OFF, MODERATE, STRICT }
  classifierPolicies   : [{ modelId, enable: bool, threshold: 0..1 }]
  ratingScheme         : enum (region-specific)
}

TimePolicy {
  dailyQuotaSeconds        : int | null
  perAppQuotas             : [{ appCategory|appId, seconds }]
  allowedWindows           : [{ daysOfWeek, startTime, endTime, tz }]
  blockedWindows           : [{ ... }]
  bedtime                  : { start, end }
  allowedAlways            : [{ appId | category }]
  earnRules                : [{ trigger, bonusSeconds }]
  aggregateAcrossDevices   : bool
}

ContactPolicy {
  messagingScope           : enum { NONE, KNOWN, MUTUAL, APPROVED, ALL }
  friendRequestPolicy      : enum { BLOCK, APPROVE, ALLOW }
  voiceCallScope           : enum
  videoCallScope           : enum
  livestreamReceive        : bool
  livestreamBroadcast      : bool
  antiGroomingClassifier   : bool                    // FR-G-003
  blockedIdentities        : [opaque identifier]
}

CommercePolicy {
  perTxApprovalThreshold   : { amount, currency } | 'ALWAYS'
  spendCaps                : [{ period: enum, amount, currency }]
  categoryRules            : [{
    category: enum { IN_APP, SUBSCRIPTION, LOOT_BOX, AD_CLICK, REAL_MONEY_GAMBLING },
    action:   enum { BLOCK, APPROVE, ALLOW }
  }]
  approvalChannel          : enum { PAP_PUSH, EMAIL, BOTH }
}

ConductPolicy {
```

```
  onDeviceClassifiers      : [{ modelId, enable: bool, threshold: 0..1 }]
  generativeAIServiceProfile : {
    consentRequired        : bool,
    contentToggles         : { sexual: bool, violence: bool, selfHarm: bool,
                               weapons: bool, romanticRoleplay: bool },
    distressEscalation     : bool,
    quietHours             : [{ start, end }],
    timeBounds             : { maxDailyMinutes },
    modelAttestationRequired : bool
  }
  deepfakeOutboundWarn     : bool                    // AI-T-004
  voiceCloneInboundWarn    : bool                    // AI-T-005
}

TransparencyPolicy {
  indicatorMode            : enum { PROMINENT, DISCREET }
  appealChannel            : URI | enum
  digestFrequency          : enum { NONE, DAILY, WEEKLY }
  digestDetail             : enum { CATEGORY_ONLY, INSTANCE_REFS }
  childReadableDigest      : bool                    // AI-D-006
  retentionDays            : int
  ageGradedVocabulary      : LocaleTag
}
```

Default age-band postures (referenced by FR-A-003 and used by the preset library) are illustrated in Table 1 below. Jurisdictional overrides may raise but not lower these defaults.

AgeBand value	Range	Default policy posture (illustrative; jurisdictional overrides apply)
UNDER_6	0–5	Walled garden; allowlist only; no messaging; no purchase; no AI services; allowed-always: 1-2 educational apps.
AGE_6_9	6–9	Allowlist-dominant; no messaging without parent approval; no GenAI; daily 1h cap; supervised browsing.
AGE_10_12	10–12	Curated allow + categorical filter; messaging mutual-followers-only; GenAI off by default; daily 2h cap; bedtime.
AGE_13_15	13–15	Categorical filter; messaging known-only; GenAI quiet hours + no romantic role-play; loot-box block; bedtime; daily cap.
AGE_16_17	16–17	Categorical filter on extreme harms; messaging unrestricted; GenAI permitted with safety filters; spend cap; bedtime advisory.
OVER_18	≥18	PARCEP no longer applies as a child-protection regime.

Table 1 — Default age-band policy postures (illustrative; jurisdictional overrides apply).

4.4 Operational layer

```
// EnforcementEvent — metadata-grade record of a PEP decision
EnforcementEvent {
  eventId              : UUID                      // PK
  policyVersion        : int
  pepId                : UUID → PEPRegistration.pepId
  timestamp            : Timestamp
  decision             : enum { BLOCK, WARN, BLUR, DELAY, REDIRECT,
                                ALLOW_LOGGED }
  category             : enum (from TaxonomyArtifact)
```

```
  classifierRef        : URI → ClassifierModelCredential.modelId | null
  confidence           : 0..1 | null
  evidenceHash         : SHA-256                    // no cleartext
  appealable           : bool
  appealId             : UUID | null
}

TransparencyNotice {
  noticeId             : UUID
  dependantId          : UUID
  pepId                : UUID
  triggerEventId       : UUID → EnforcementEvent.eventId
  shownAt, acknowledgedAt : Timestamp | null
  appealId             : UUID | null
}

AppealRequest {
  appealId             : UUID                       // PK
  appellantDependantId : UUID
  contestedEventId     : UUID
  reason               : string
  status               : enum { OPEN, UNDER_REVIEW, ACCEPTED, REJECTED }
  resolution           : string | null
  resolvedAt           : Timestamp | null
}

GuardianDigest {
  digestId             : UUID                       // PK
  guardianId           : UUID
  windowStart, windowEnd : Timestamp
  entries              : [{
    severity           : enum { INFO, NOTICE, WARNING, CRITICAL },
    category           : enum,
    count              : int,
    exampleEventIds    : [UUID]
  }]
  aiSummary            : { enabled: bool, text: string | null,
                           modelRef: URI }
  childReadable        : bool                       // AI-D-006
}
```

4.5 Discovery layer

```
PEPRegistration {
  pepId                : UUID                       // PK
  vendor, model, version : string
  conformanceProfile   : enum { CONF_A..CONF_G }
  capabilities         : [string]
  subscribedAt, lastSyncAt : Timestamp
  publicKey            : COSE_Key
}

PIPDescriptor {
  pipId                : DID                        // PK
  jurisdiction         : ISO-3166-1-alpha-2
  rootKey              : COSE_Key
  certificateChain     : [X.509 | COSE]
  issuanceCapabilities : [enum { GUARDIANSHIP, AGE, CLASSIFIER, TAXONOMY,
                                 OVERRIDE }]
  endpoints            : { issue: URI, status: URI, log: URI, .well_known: URI }
  transparencyLogRef   : URI                        // RFC 9162-style
}
```

5. PARCEP-Sync protocol — overview

PARCEP-Sync is the wire protocol that runs between the PAP, the PEPs, and the PIPs. It is small (14 methods plus an extension primitive), symmetric in shape (every request gets an acknowledgement), and transport-neutral above the security boundary. The default transport is a per-household Messaging Layer Security group (IETF RFC 9420) over QUIC; an mTLS-over-HTTPS fallback is provided for bootstrap and for non-MLS-capable legacy PEPs.

5.1 Why MLS as the canonical transport

MLS provides four properties that directly serve PARCEP's design: (i) authenticated group membership with cryptographic proof of who is in the group (PAP, PEPs, optional co-guardian PAP), (ii) forward secrecy and post-compromise security so that the compromise of a single PEP does not retroactively reveal the household policy history, (iii) asynchronous group operations so a newly added PEP (a child's new device, a new app) can be admitted without coordinating with all existing members, and (iv) a clean key-rotation primitive used when a PEP is removed (guardian removes a vendor's PEP after a security incident; co-guardian leaves).

The household MLS group has the guardian PAP as administrator and the household's PEPs as members. The PAP issues policy as MLS application messages signed by the authoring guardian's credential. Each PEP receives the message decrypted by its own ratchet state.

5.2 Methods at a glance

Method	Sender → Receiver	Purpose	Idempotent?
ENROLL	PEP → PAP	PEP registers itself with the PAP: presents COSE_Key, conformanceProfile, capabilities, vendor/model/version. Returns a PEPRegistration.	Yes (re-enroll)
WELCOME	PAP → PEP	MLS Welcome message (RFC 9420) adding the PEP to the household MLS group. Carries the ratchet tree and current epoch.	No
PUBLISH	PAP → MLS group	Publishes a new HouseholdPolicy version. Signed by the authoring guardian; delivered as an MLS application message to all PEP subscribers.	Idempotent by policyVersion
UPDATE	PAP → MLS group	Publishes a policy delta (subset of clauses changed). Carries previous and new policyVersion for ordering.	Idempotent by version pair
REVOKE	PAP → MLS group	Revokes a current policy. PEPs fall back to the prior valid policy or to the policy-defined fail-safe state.	Yes
ACK	PEP → PAP	Acknowledges receipt and applied state of a PUBLISH/UPDATE/REVOKE. Carries policyVersion, status (APPLIED|REJECTED|DEFERRED), reasonCode.	Yes
EVENT	PEP → PAP	Emits an EnforcementEvent (metadata only, no cleartext). Includes decision, category, optional classifierRef, evidence hash, confidence.	Yes (by eventId)

DIGEST	PAP → Guardian UI	Periodic guardian-readable digest aggregating events into severity-categorized highlights. May include AI summary (AI-D-006).	Yes
APPEAL	PEP → PAP	Routes an AppealRequest from a dependant to the guardian. Carries contestedEventId, free-text reason, requested resolution.	Yes (by appealId)
RESOLVE	PAP → PEP	Guardian's resolution of an appeal: ACCEPT (one-time allow), REJECT, or POLICY_DELTA (patch the policy).	Yes (by appealId)
ATTEST	PEP → PIP	PEP attests its capabilities and conformance profile to a PIP for a public registry entry.	Yes
FETCH	PEP → PIP	Fetches a signed artifact: classifier model package, taxonomy artifact, jurisdictional override descriptor.	Yes (by artifact hash)
NOTIFY	MLS group → all	MLS-level notification (epoch change, member add/remove, conflict notice). Out-of-band relative to policy messages.	—

Table 2 — PARCEP-Sync methods, sender → receiver, purpose, idempotency.

6. Discovery, bootstrap, authentication

6.1 Discovery

A PEP needs to locate the PAP for a household before it can ENROLL. PARCEP defines two discovery mechanisms:

- DNS SVCB records (RFC 9460) under the household-issued domain. A guardian receives a household identifier at PAP setup (e.g., "household-3f9b@parcep.example"); PEPs query SVCB for _parcep._tcp.parcep.example and resolve the PAP's enrollment endpoint.
- Per-PEP .well-known/parcep endpoint as defined by IETF RFC 8615. A PEP that hosts a guardian-facing onboarding flow advertises the PAP enrollment URI under https://<pep-host>/.well-known/parcep so the PAP can complete pairing out-of-band.

6.2 Bootstrap

Bootstrap proceeds in three steps. The guardian, at the PAP, scans a QR code emitted by the new PEP (a device, an app, an STB, a router admin page). The QR code encodes the PEP's pubkey, conformance profile, and a fresh nonce. The PAP sends an ENROLL request to the PEP over mTLS (using a one-time bootstrap certificate). On success the PAP issues an MLS Welcome message that adds the PEP to the household MLS group; thereafter all PAP↔PEP communication flows in the MLS group.

6.3 Authentication

Two authentication primitives are used. At the MLS layer, every member is authenticated by its MLS credential (an X.509 or COSE credential bound to the member's pubkey). At the application layer, every guardian-authored message (PUBLISH, UPDATE, REVOKE, RESOLVE) is signed by the authoring guardian's GuardianshipCredential. PEPs and PIPs are

authenticated by their respective DeviceCredential and PIPDescriptor.publicKey. Strong-auth (2FA / WebAuthn) at the PAP guards the guardian's signing key.

7. Wire format

PARCEP-Sync uses a dual wire format. Policy authoring, web dashboards, and human-readable audit use JSON-LD with W3C Verifiable Credentials. Event telemetry, network-side signaling, and IoT-class PEP exchanges use CBOR with COSE signatures (RFC 8949 / RFC 9052) for compactness and embedded-device parsability.

7.1 Example: PUBLISH HouseholdPolicy (JSON-LD)

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://parcep.itu.int/ns/v1"
  ],
  "type": ["HouseholdPolicy"],
  "policyId":      "urn:uuid:c1d2e3f4-...",
  "policyVersion": 6,
  "validFrom":     "2026-05-18T00:00:00Z",
  "validTo":       "2027-05-18T00:00:00Z",
  "conformanceProfile": "CONF_A",
  "signedBy": [
    { "guardianId": "urn:uuid:guardian-...",
      "proof":      { "type": "DataIntegrityProof", "cryptosuite": "eddsa-2022",
                      "created": "...", "proofValue": "..." } }
  ],
  "jurisdictionalOverrideRefs": [
    "https://pip.fr.example/overrides/under-18-arcom-v3"
  ],
  "subjects": [
    {
      "subjectId": "urn:uuid:subject-...",
      "dependantRefs": ["urn:uuid:dependant-anon-1"],
      "deviceRefs":    ["urn:uuid:device-phone-1", "urn:uuid:device-tablet-1"],
      "clauses": {
        "content": {
          "taxonomyRef":  "https://pip.eu.example/taxonomies/v4",
          "categoryRules":[
            {"category":"adult_sexual",  "action":"BLOCK"},
            {"category":"violence_extreme","action":"BLOCK"}
          ],
          "classifierPolicies":[
            {"modelId":"https://pip.example/models/nudity-v1.2.0",
             "enable":true, "threshold":0.85}
          ]
        },
        "time": {
          "dailyQuotaSeconds": 7200,
          "aggregateAcrossDevices": true,
          "bedtime": {"start":"21:30","end":"07:00"}
        },
        "transparency": {
          "indicatorMode":"PROMINENT",
          "appealChannel":"app://parcep/appeal",
          "digestFrequency":"WEEKLY",
          "childReadableDigest": true,
          "retentionDays": 30
        }
      }
    }
```

```
  ]
}
```

7.2 Example: EnforcementEvent (CBOR / COSE)

```
// CBOR diagnostic notation; the wire form is canonical CBOR (RFC 8949),
// signed with COSE_Sign1 (RFC 9052) by the PEP's deviceCredential key.

{
  1:  h'a1b2c3...',                  // eventId (UUID, 16 bytes)
  2:  6,                              // policyVersion
  3:  h'pep1...',                     // pepId
  4:  1747512000,                     // timestamp (epoch sec)
  5:  "BLOCK",                       // decision
  6:  "adult_sexual",                // category
  7:  "https://pip.example/models/nudity-v1.2.0",  // classifierRef
  8:  0.91,                           // confidence
  9:  h'sha256(evidence)...',         // evidenceHash
  10: true                            // appealable
}
```

8. Key sequence flows

The following five sequence diagrams illustrate the canonical PARCEP-Sync exchanges. They show the wire-level interactions; the surrounding state machines are specified in the IETF Internet-Draft (Section 11).

8.1 PEP enrollment and first policy distribution

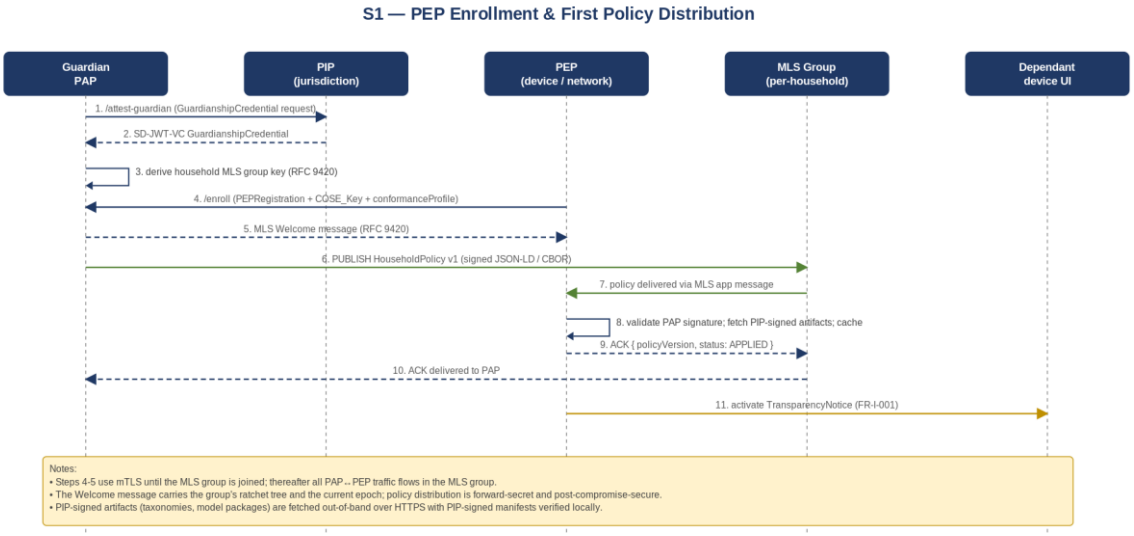

Figure 4 — PEP enrollment and first policy distribution (Diagram S1).

8.2 Enforcement decision, transparency notice, appeal

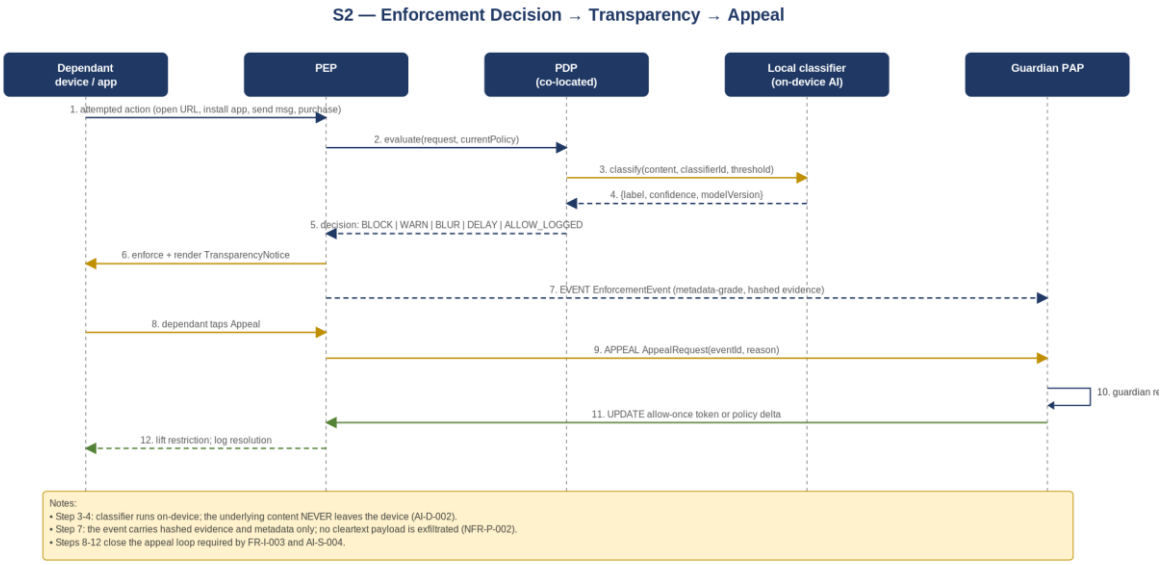

Figure 5 — Enforcement → transparency notice → appeal loop (Diagram S2).

8.3 Co-guardian concurrent update with conflict

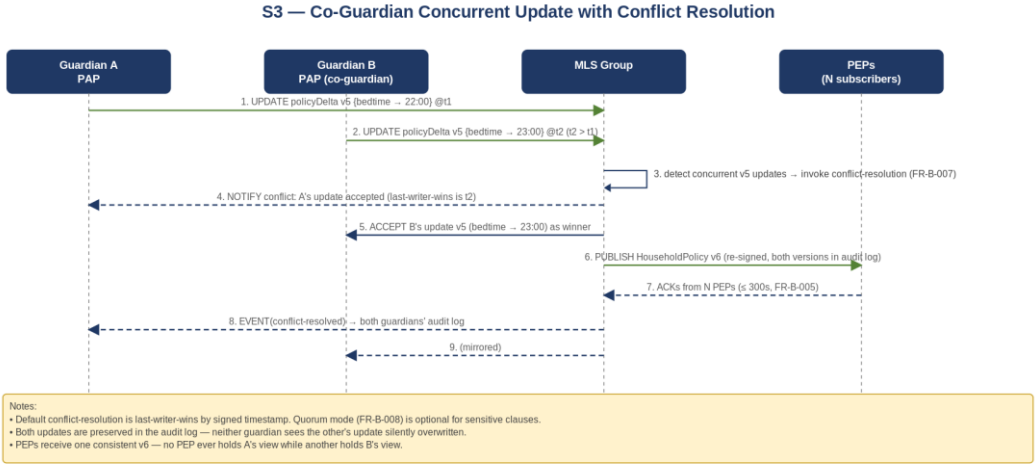

Figure 6 — Co-guardian concurrent update; conflict resolution (Diagram S3).

8.4 Age credential consumption with selective disclosure

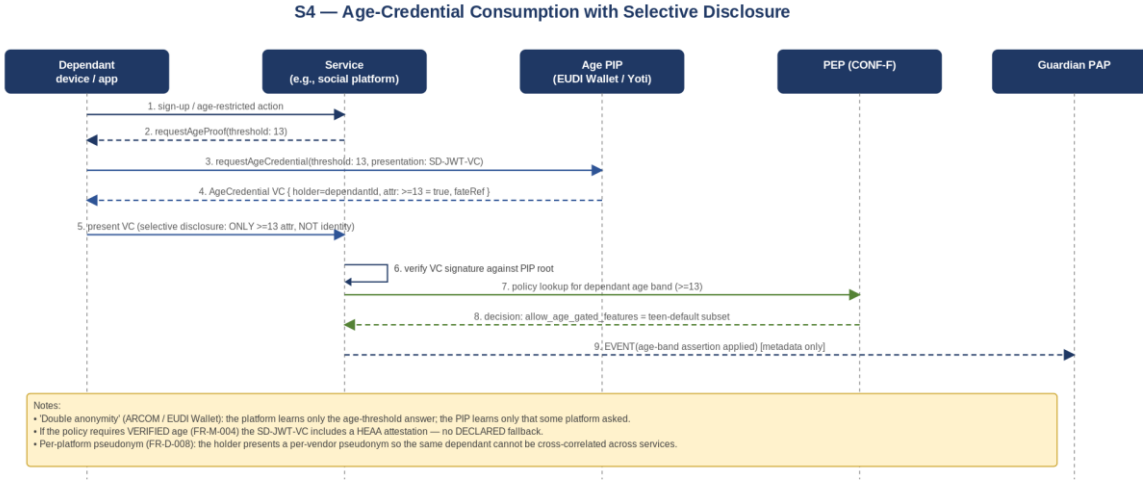

Figure 7 — Age credential consumption (double-anonymity flow) (Diagram S4).

8.5 Cross-network continuity and offline degradation

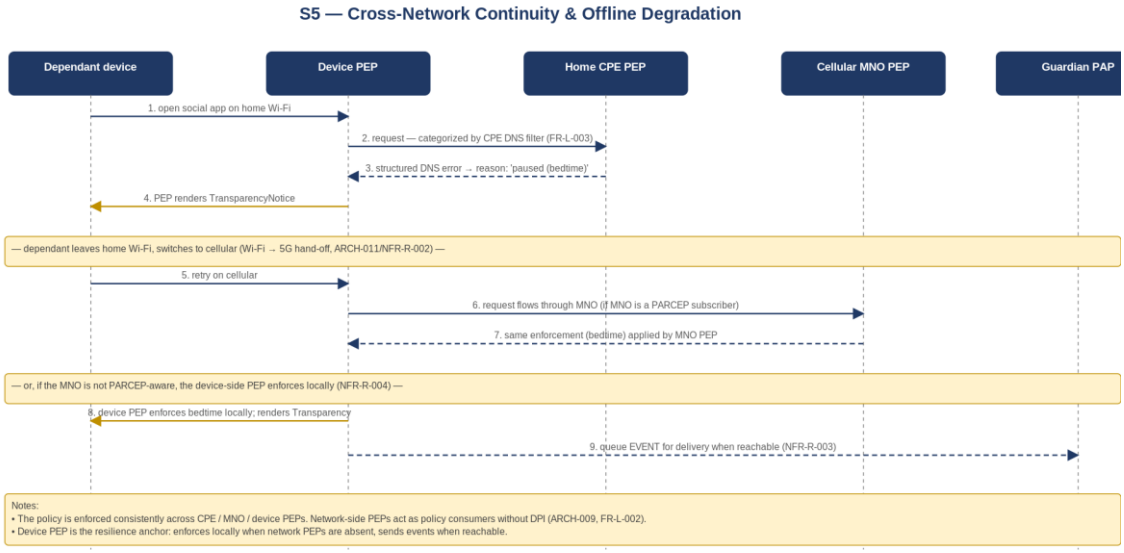

Figure 8 — Hand-off between home Wi-Fi and cellular; degraded operation (Diagram S5).

9. Error model, idempotency, and conformance testing

9.1 Error model

PARCEP-Sync uses a small structured error model. Every response carries a status (one of { OK, REJECTED, DEFERRED, ERROR }) and a structured reasonCode (one of: BAD_SIGNATURE, EXPIRED_CREDENTIAL, REVOKED_CREDENTIAL, UNSUPPORTED_PROFILE, INVALID_SCHEMA, JURISDICTIONAL_BAR, CAPABILITY_MISSING, OUT_OF_ORDER, RATE_LIMITED, TEMPORARY_UNAVAILABLE, INTERNAL). Free-text reason details MAY accompany the reasonCode but SHALL NOT be relied on by clients.

9.2 Idempotency and ordering

All policy-changing methods (PUBLISH, UPDATE, REVOKE) are idempotent under repeat with the same (policyId, policyVersion) tuple. Out-of-order delivery is allowed: if a PEP receives UPDATE v8 before PUBLISH v7, it requests v7 via FETCH and applies in order. The MLS application-message ordering provides a strong baseline; PARCEP-Sync adds policyVersion as an application-layer monotonic counter so that conflicts (Section 8.3) can be detected and resolved deterministically.

9.3 Conformance testing

PARCEP defines a reference test suite that exercises every method, every error path, every cross-PEP semantic-equivalence requirement (FR-C-002), and every conformance-profile requirement subset (CONF-A through CONF-G). Conformance reports are themselves expressed in the PARCEP schema and signed; PEPs can present their conformance attestation in ENROLL so that a guardian PAP can verify before subscribing the PEP.

10. Security and privacy analysis

10.1 Threat model — protocol layer

In addition to the threat model captured in Analysis 03 (Section 7), the protocol layer faces three concrete classes of adversary. (a) The malicious vendor PEP: a PEP that subscribes to the MLS group but attempts to exfiltrate policy, send false events, or refuse enforcement silently. Defense: vendor attestation in ENROLL; reference-test conformance; EVENT signing keyed to the PEP's deviceCredential. (b) The compromised PIP: a trust anchor that issues fraudulent credentials. Defense: minimum two independent PIPs per jurisdiction (ARCH-004); CT-style transparency logs (RFC 9162); PEP-side detection of unlogged issuances. (c) The MLS-group adversary: a former member attempting to read post-eviction traffic. Defense: MLS PCS — when a PEP is removed, the group key is rotated so future traffic is opaque to the evicted member.

10.2 Privacy properties

The protocol provides three structural privacy properties. First, data minimization: a PEP receives only the policy attributes its conformance profile requires, and only for the dependants bound to its subjects. Second, content non-disclosure: EVENT messages carry hashed evidence and metadata, never cleartext payload. Third, cross-vendor unlinkability: per-vendor pseudonym for the dependant (FR-D-008) ensures that even if two PEPs collude, they cannot trivially link the same dependant across their respective accounts without explicit guardian opt-in.

10.3 Cryptographic agility

PARCEP-Sync is crypto-agile. Default suites at v1: Ed25519 / X25519 (signing / key exchange), ChaCha20-Poly1305 (AEAD), SHA-256 (hashing). The protocol negotiates suites via MLS ciphersuites. A post-quantum suite (ML-KEM for key exchange, ML-DSA for signing) is recommended as a SHOULD requirement at v1 and SHALL at v2 (NFR-S-004).

11. Path to ITU-T and IETF standardization

The work-split this design implies is the classic split between the policy world (ITU-T) and the wire world (IETF). The recommended track is:

11.1 ITU-T SG17 (X.PARCEP Recommendation)

- Scope: the data model (Sections 4.1-4.5), the conformance profiles (CONF-A through CONF-G), the threat model, the regulatory mapping, the architecture decision (Sections 1-3).
- Vehicle: the existing X.PARCEP work item in Q1/Q3 SG17 carried by Broadcom and Vodafone.
- Deliverables across the 2025–2028 study period: a stable data model by the end of Year 1; a stable set of conformance profiles by Year 2; the full Recommendation consented in Year 3.

11.2 IETF (PARCEP-Sync wire protocol)

- Scope: the wire protocol (Sections 5-9) — methods, wire format, MLS profile, DNS SVCB discovery, .well-known endpoint, error model, conformance test scaffolding.
- Vehicle: a Birds-of-a-Feather (BoF) session at IETF, followed by a new Working Group. Candidate WG name: PARCEP-Sync. Sponsoring Area: ART (application) or SEC (security), with cross-area collaboration with MLS and MIMI.
- Initial deliverables: (i) the architecture and terminology Internet-Draft; (ii) the PARCEP-Sync wire-protocol I-D referencing MLS RFC 9420; (iii) the MLS profile I-D specifying the household-group ciphersuites, member attestation, and policy-message format; (iv) the discovery I-D for SVCB and .well-known/parcep.
- Interfaces with adjacent WGs: MLS (RFC 9420 baseline), MIMI (publish/subscribe semantics), DNSOP (structured DNS error data used for network-side enforcement signaling), OAUTH (SD-JWT-VC), and DICE (constrained-device profiles for CONF-E CPE / IoT).

11.3 Cross-SDO coordination

Two other standards bodies will need to be liaised. ISO/IEC JTC 1/SC 27 (information security) owns 18013-5 (mDL) and 23220-series (mobile identity) — required for the cross-format compatibility of AgeCredential and GuardianshipCredential. W3C VC Working Group owns Verifiable Credentials Data Model 2.0 and SD-JWT-VC — required for the JSON-LD wire format. The European Commission's eIDAS 2 / EUDI Wallet Architecture Reference Framework should be liaised so the AgeCredential is consumable by the Wallet's age-verification building block out of the box.

12. Open questions for the editorial team

- D-Q1. Should the default wire format be JSON-LD or CBOR? (Current design: dual; both at v1.)
- D-Q2. Should MLS be the only canonical transport, or should an mTLS-over-HTTPS profile be defined as a peer alternative (not just bootstrap)?
- D-Q3. Should the PAP be required to be device-resident (better privacy) or may it be a guardian-controlled cloud service (better availability)? Current design: both; guardian's choice.
- D-Q4. Should a school acting in loco parentis be modeled as a CoGuardianRelation (current design) or as a separate role with its own credential format?
- D-Q5. Should the age-band granularity be the five-band default (Section 4.3 Table 1), or should PARCEP support a finer grain (per-year) where the underlying age credential allows it?
- D-Q6. Should the PARCEP MLS group be at most one (per household) or should multiple groups coexist (e.g., one per dependant)? Current design: one per household, with PolicySubject providing per-dependant scoping inside.
- D-Q7. Should the protocol carry an explicit "jurisdictional re-evaluation" event when the dependant crosses a border? Current design: implicit via cross-jurisdictional verification at PEP.

- D-Q8. Should the reference test suite be normative (every implementation MUST pass) or informative? Current design: normative for conformance, informative for differentiation.

References

1. X.PARCEP — Draft baseline (Cxxxx Rev. 1). https://www.itu.int/ITU-T/workprog/wp_item.aspx?isn=20100
2. PARCEP Analysis 03 — Requirements Extending X.PARCEP (companion document in this project folder). https://www.itu.int/ITU-T/workprog/wp_item.aspx?isn=20100
3. IETF RFC 9420 — Messaging Layer Security (MLS). https://datatracker.ietf.org/doc/html/rfc9420
4. IETF MLS Working Group. https://datatracker.ietf.org/wg/mls/about/
5. IETF MIMI Working Group (More Instant Messaging Interoperability). https://datatracker.ietf.org/wg/mimi/about/
6. IETF RFC 9162 — Certificate Transparency v2. https://datatracker.ietf.org/doc/html/rfc9162
7. IETF RFC 3198 — Terminology for Policy-Based Management. https://datatracker.ietf.org/doc/html/rfc3198
8. ISO/IEC 29146:2024 — A framework for access management. https://www.iso.org/standard/86013.html
9. IETF RFC 8949 — Concise Binary Object Representation (CBOR). https://datatracker.ietf.org/doc/html/rfc8949
10. IETF RFC 9052 — CBOR Object Signing and Encryption (COSE). https://datatracker.ietf.org/doc/html/rfc9052
11. IETF RFC 9460 — Service Binding (SVCB) and HTTPS Resource Records. https://datatracker.ietf.org/doc/html/rfc9460
12. IETF RFC 8615 — Well-Known Uniform Resource Identifiers. https://datatracker.ietf.org/doc/html/rfc8615
13. IETF RFC 8484 — DNS over HTTPS (DoH). https://datatracker.ietf.org/doc/html/rfc8484
14. IETF RFC 9462 — Discovery of Designated Resolvers. https://datatracker.ietf.org/doc/html/rfc9462
15. IETF RFC 9463 — Discovery of Network-designated Resolvers. https://datatracker.ietf.org/doc/html/rfc9463
16. IETF DNSOP — Structured DNS Error Data. https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-structured-dns-error
17. W3C Verifiable Credentials Data Model 2.0. https://www.w3.org/TR/vc-data-model-2.0/
18. IETF SD-JWT-VC. https://datatracker.ietf.org/doc/html/draft-ietf-oauth-sd-jwt-vc
19. ISO/IEC 18013-5 — Mobile driving licence (mDL). https://www.iso.org/standard/69084.html
20. EU Digital Identity Wallet — Architecture Reference Framework. https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework
21. EU Commission — DSA Guidelines on the Protection of Minors (July 2025). https://digital-strategy.ec.europa.eu/en/library/commission-publishes-guidelines-protection-minors
22. UK Ofcom — Highly Effective Age Assurance (HEAA) guidance, April 2025. https://www.ofcom.org.uk/siteassets/resources/documents/consultations/category-1-10-weeks/statement-age-assurance-and-childrens-access/part-3-guidance-on-highly-effective-age-assurance.pdf?v=395680
23. NIST FATE Age Estimation and Verification benchmark. https://pages.nist.gov/frvt/reports/aev/fate_aev_report.pdf
24. IWF — How AI is being abused to create CSAM (2026). https://www.iwf.org.uk/about-us/why-we-exist/our-research/how-ai-is-being-abused-to-create-child-sexual-abuse-imagery/
25. WeProtect Global Alliance — Global Threat Assessment 2025. https://www.weprotect.org/wp-content/uploads/GTA-2025_EN.pdf
26. UN CRC General Comment No. 25 — Children's rights in the digital environment. https://www.ohchr.org/en/documents/general-comments-and-recommendations/general-comment-no-25-2021-childrens-rights-relation