MedLattice

A Quantum-Resistant, Compartmentalised Electronic Patient Record

---

Technical Specification, Version 1.0

Permissioned EVM consortium chain · ML-KEM-1024 · ML-DSA-87 · SLH-DSA

Element-granular disclosure · FHIR R5 · Three reference smart contracts

Document	MedLattice Technical Specification
Version	1.0
Status	**DRAFT — SUBMITTED FOR APPROVAL**
Date	25 August 2026
Prepared by	Dr Neal Aggarwal
Classification	Confidential — architecture and design
Regulatory scope	HIPAA/HITECH · UK GDPR & DPA 2018 · EU GDPR & EHDS Kenya DPA Cap. 411C & Digital Health Act 2023
Accompanying artefacts	3 Solidity contracts (compiled, 61 tests passing) Python reference client (executed end to end)

Approval note. This specification is submitted for review and sign-off before implementation proceeds. Section 12 enumerates the fourteen decisions that require explicit approval; every one of them has a recommended option with the reasoning stated. Nothing in this document should be read as a claim that the system is finished — it is a design that has been made concrete enough to be criticised, with the security-critical parts implemented and tested so that the claims can be checked rather than taken on trust.

Contents

1 Executive summary

MedLattice is an electronic patient record in which **confidentiality is a property of the key material rather than a promise made by a server**, and in which **the audit trail is a property of a distributed ledger rather than a table that a database administrator can edit**. It is designed for a consortium of hospitals, laboratories, payers and public health authorities operating a permissioned Ethereum Virtual Machine (EVM) chain, and it is designed on the assumption that a cryptographically relevant quantum computer will exist within the confidentiality lifetime of the records it protects.

Three claims define the system, and each is substantiated by working code accompanying this document.

First, no protected health information ever touches the ledger. The chain carries identities, post-quantum key commitments, consent directives, capability grants, content-addressed pointers, integrity commitments and the disclosure log. The clinical content lives in an off-chain encrypted vault. This separation is what makes the design compatible with a right to erasure at all: the ledger is immutable, the content plane is not, and erasure is performed by destroying keys (§7.4).

Second, granularity is enforced cryptographically at three levels. The record is decomposed into *compartments* — a compartment being one $(\text{patient}, \text{data class}, \text{sensitivity tier}, \text{epoch})$ slice, of which there are up to twenty data classes and four tiers. Within a compartment, each canonical FHIR element is a separately encrypted *cell*, keyed from a Hierarchical Disclosure Key Tree (HDKT). Releasing a set of interior tree nodes releases exactly the leaves beneath them and nothing else, in $O(m \log(n/m))$ key material. A registration clerk handed the *entire* compartment ciphertext by a fully compromised gateway still cannot decrypt the diagnosis, because the clerk was never given a key from which that cell's key can be derived. The accompanying reference client demonstrates precisely this (§6.2, Appendix C).

Third, the cryptography is post-quantum today, not on a roadmap. Key encapsulation is ML-KEM-1024 (FIPS 203, NIST Category 5) hybridised with X448 under the X-Wing combiner; clinical attestation is ML-DSA-87 (FIPS 204) hybridised with Ed448; long-term notarisation of history uses the hash-only SLH-DSA family (FIPS 205) so that a future break of lattice assumptions costs future confidentiality but cannot retroactively rewrite the record. Content encryption is AES-256-GCM with an explicit key-commitment tag. Every suite carries an on-chain identifier, and migration to a successor suite requires re-wrapping data-encryption keys — not re-encrypting the corpus.

The motivation for the third claim is arithmetic rather than fashion. A paediatric genomic record generated today has a confidentiality requirement measured in decades; Kenya's Digital Health Act 2023 alone mandates twenty years of retention. Under Mosca's inequality, if the required secrecy lifetime plus the migration time exceeds the time until a cryptographically relevant quantum computer, the migration is already late. For health records that inequality is not marginal — it is violated by a wide margin (§5.1). Any ciphertext produced today under classical-only key establishment is a ciphertext that an adversary can harvest now and decrypt later.

1.1 What is delivered with this specification

Artefact	Status
`IdentityRegistry.sol` — DIDs, PQ key registry, expiring role credentials, suite agility	Compiles under solc 0.8.28; 6,296 bytes deployed
`ConsentCapabilityManager.sol` — the authorisation kernel, consent, capabilities, break-glass	Compiles; 10,260 bytes deployed
`RecordAnchorRegistry.sol` — hash-linked anchors, disclosure log, legal hold, crypto-shredding, notarisation	Compiles; 6,248 bytes deployed
Test suite — 61 assertions across all three contracts, executed on a full EVM	**61 passed, 0 failed** (Appendix B)
Python reference client — hybrid PQ envelope encryption, HDKT, role-scoped query resolution	Executed end to end (Appendix C)

Artefact	Status
Gas profile on the hot path	64,816 gas per query authorisation; 73,257 per audit write

1.2 The shape of the argument

The remainder of this document is organised so that each claim is separable and can be rejected independently. §3 gives the architecture. §4 defines what a record *is*. §5 specifies the cryptography and — importantly — states what is *not* claimed. §6 specifies granularity and the staff tiering. §7 documents the three contracts. §8 is the threat model with residual risks stated plainly. §9 maps the design onto four regulatory regimes and analyses the two places where they genuinely conflict. §12 lists what still needs a signature.

2 Scope, objectives and non-goals

2.1 Objectives

1. **Longitudinal patient record** spanning encounters, organisations and modalities, queryable at element granularity.
2. **Cryptographic minimum-necessary.** The technical safeguard that HIPAA §164.502(b) and Caldicott Principle 4 describe as an administrative duty is here implemented as a key-distribution property.
3. **Non-repudiable audit.** Every disclosure and every denial is a ledger event attributable to a capability, a purpose of use and an element set.
4. **Patient agency.** The data subject holds a controller key, can restrict any data class, can sever a care relationship, can grant a compartment to an outside clinician, and is notified of every emergency override of her restrictions.
5. **Post-quantum confidentiality with a defined migration path**, including for data already at rest.
6. **Regulatory defensibility** across four regimes simultaneously, with the conflicts between them identified rather than papered over.

2.2 Non-goals

- **MedLattice is not a database replacement.** Hospitals keep their existing systems; MedLattice is the consent, key-management, integrity and audit fabric that sits between them. A trust that cannot run a node still participates through a gateway.
- **The ledger is not a store of clinical data.** Any proposal to put a FHIR resource on chain, encrypted or otherwise, is out of scope and is rejected in §3.3.
- **This is not a token, and there is no cryptocurrency.** Gas on a permissioned QBFT chain is a scheduling and denial-of-service control, priced at zero.
- **No claim of information-theoretic security.** Everything here is computational. §5.12 states the limits explicitly.

2.3 Design principles

The seven principles the design is answerable to

P1 — The ledger holds no secrets. Every value written on chain is a commitment, a pseudonym, a pointer or a policy. Assume the ledger is public.
P2 — Authority is a budget, not a status. A capability has an expiry, a query count, a purpose and an element set. There is no standing access.
P3 — Role is necessary, never sufficient. Treatment access additionally requires a live care relationship derived from an actual encounter.
P4 — The patient's restriction outranks the role matrix, and the only overrides are emergency, statutory and public-health — each of which notifies her.
P5 — Every denial is logged. Probing behaviour must be as visible as successful access.
P6 — Assumption diversity for history. Confidentiality rests on lattices; integrity of the historical record addition-

ally rests on hashes alone.
P7 — Agility over strength alone. Every suite is identified on chain and retirable, and migration re-wraps keys rather than re-encrypting content.

3 System architecture

3.1 Layer model

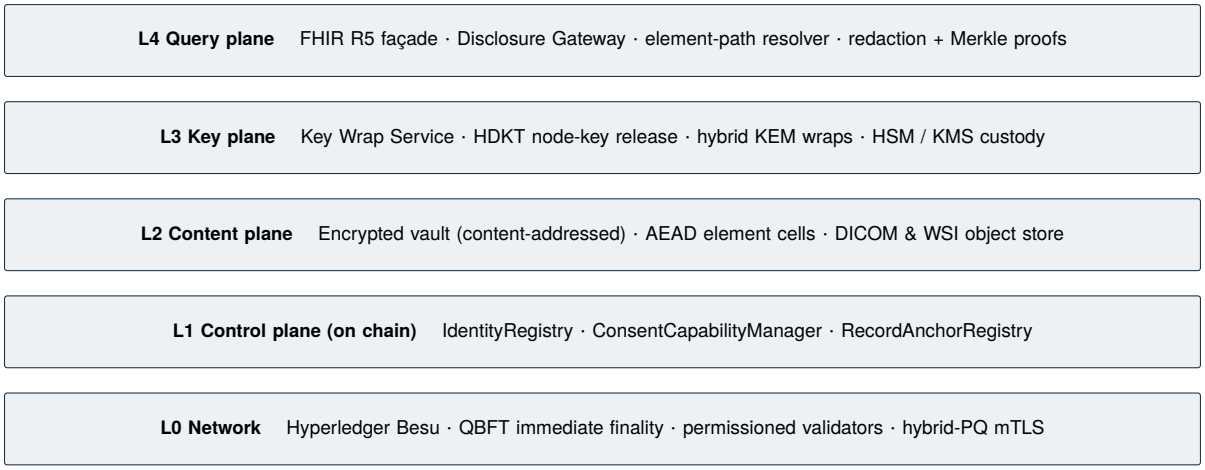

Figure 1. The five planes. Data flows downward only as ciphertext; authority flows upward only as capabilities. No layer above L1 can grant itself authority, and no layer below L3 ever holds a key.

3.2 L0 — the network

Client and consensus. Hyperledger Besu with QBFT (Istanbul BFT v2). The choice is driven by three clinical requirements that proof-of-work and proof-of-stake chains do not satisfy:

- **Immediate finality.** QBFT finalises at block commit; there are no reorganisations. A clinician must never see an authorisation that is later un-made. Probabilistic finality is unacceptable in a system where a block reorganisation would retroactively invalidate a disclosure that has already resulted in a clinical decision.
- **Known validator set.** Byzantine fault tolerance requires $n \geq 3f+1$. With a founding consortium of seven validator organisations the network tolerates two simultaneous Byzantine or failed validators. Validator membership is itself governed by an on-chain permissioning contract, so admission and expulsion are auditable.
- **Data residency control.** Kenya's Digital Health Act constrains where health data may be held (§9.4). A known validator set makes it possible to state, and prove, which jurisdictions hold which replicas.

Parameters. Block period 4 s; block gas limit 30,000,000; gas price zero with per-account rate limiting at the gateway. On the measured gas profile (§7.5) a single block accommodates about 400 query authorisations, i.e. about 100 per second sustained, before any Layer-2 or batching optimisation.

Node-to-node transport. TLS 1.3 with the hybrid `X25519MLKEM768` key-exchange group and mutual authentication using composite certificates (ML-DSA-87 + ECDSA P-384). The hybrid group is chosen for transport rather than the Category-5 profile used at rest because it is the group with real deployment maturity across TLS stacks; the asymmetry is deliberate and is justified in §5.8.

3.3 L1 — what is, and is not, on chain

On-chain / off-chain boundary

On chain: DIDs and identity kinds; commitments to post-quantum public keys and their epochs; accreditation status; expiring role credentials; consent directives and their legal-basis codes; care-relationship attestations; capability grants and their consumption counters; per-compartment anchor chains (element Merkle root, ciphertext digest, storage locator, author signature digest); the disclosure log; denial log; legal holds; erasure tombstones; epoch notarisations.
Never on chain: names, identifiers, dates of birth, diagnoses, medications, notes, images, genomic data, any FHIR resource in any form; any private key; any data-encryption key or key wrap; any free-text field that a user could populate; any plaintext justification for break-glass (only its commitment).

The prohibition on encrypted PHI on chain deserves a word, because it is the most common design error in this field. Ciphertext written to an immutable, replicated ledger is ciphertext that (i) cannot be erased, defeating UK/EU GDPR Article 17 outright; (ii) is retained by every validator, including validators in jurisdictions to which transfer may be unlawful; and (iii) will be decryptable by whoever eventually breaks the algorithm, which for a fifty-year record is a serious proposition. MedLattice therefore puts ciphertext in a vault whose replication set is a policy decision, and puts only 32-byte commitments on the ledger.

3.4 L2 — the content plane

The vault is a content-addressed encrypted object store, deployed per organisation with policy-governed replication. Each stored object is one **element cell**: the AEAD-sealed value of a single canonical FHIR element path. Objects are addressed by `SHAKE256(cell)`, so the vault is self-verifying and deduplicating, and a corrupted or substituted object is detected before decryption is attempted.

Large binary payloads — DICOM series, whole-slide pathology images — are handled as a special case: the pixel data is sealed as a single cell under the `IMAGING_PIXEL` or `PATHOLOGY` compartment, while the metadata (modality, body site, acquisition time, report impression) is decomposed into ordinary cells in `IMAGING_META`. This is what allows a radiographer to see the worklist without the report and a coder to see the report without the pixels.

3.5 L3 — the key plane

The **Key Wrap Service** (KWS) is the only component that touches key material, and it holds no long-term reader keys and no plaintext. On presentation of a capability it (i) calls `ConsentCapabilityManager.consume` on chain, which re-evaluates the live predicates and decrements the query budget; (ii) computes the minimal HDKT cover of the permitted element set; and (iii) seals those node keys to the reader's hybrid encapsulation key. Compartment root keys are held in a FIPS 140-3 Level 3 HSM or equivalent KMS, and are the object destroyed by crypto-shredding.

Two disclosure modes are supported and are a deployment choice, not a design fork:

- **Client-decrypt (default, zero-trust).** The gateway forwards sealed grants and ciphertext; decryption happens in the clinician's client. The gateway never sees plaintext. This is the mode exercised in the reference client.
- **Gateway-decrypt (thin-client fallback).** For devices that cannot hold a key — shared ward terminals, some medical devices — the gateway decrypts within an enclave. This trades a real reduction in assurance for deployability and must be approved per device class, not per organisation (Decision D-07).

3.6 L4 — the query plane

The Disclosure Gateway exposes a FHIR R5 RESTful façade plus a restricted FHIRPath query language, and implements the eight-stage resolution pipeline of §6.8. It is an *enforcement* point, not a *decision* point: every decision is made by the on-chain kernel, so a compromised gateway can deny service but cannot manufacture authority, and cannot decrypt what the key tree did not release to it.

4 The record: data model

4.1 Alignment with FHIR R5

MedLattice does not invent a clinical information model. Content is HL7 FHIR R5 resources, profiled against the International Patient Summary where an equivalent profile exists, and terminology is SNOMED CT, LOINC, ICD-11 and RxNorm/ATC. What MedLattice adds is an orthogonal *disclosure* model layered over FHIR: the compartment, the sensitivity tier and the element cell.

The reason for a separate disclosure model is that FHIR's own security labelling (`Resource.meta.security`, HL7 v3 Confidentiality codes) is advisory metadata. A server is expected to honour it. MedLattice makes the equivalent distinctions *key-bearing*, so that honouring them is not optional.

4.2 Compartments

A compartment is the unit of key management:

$$\text{cid} = \text{keccak256}\big(\text{pseudonym}_{\text{patient}} \,\|\, \text{class} \,\|\, \text{tier} \,\|\, \text{epoch}\big)$$

The twenty data classes are fixed by governance (they are bit positions in a `uint32` mask, so a role's entitlement is a single machine word):

#	Class	Principal FHIR resources	Default tier
0	`DEMOGRAPHICS`	Patient, RelatedPerson	N
1	`ADMIN_COVERAGE`	Coverage, Account, Encounter (administrative)	N
2	`VITALS`	Observation (vital-signs category)	N
3	`ALLERGY`	AllergyIntolerance	N
4	`MEDICATIONS`	MedicationRequest / Statement / Administration	N
5	`PROBLEMS`	Condition	N
6	`PROCEDURES`	Procedure	N
7	`LABS_ROUTINE`	Observation (laboratory), DiagnosticReport	N
8	`LABS_SPECIAL`	Microbiology, virology, serology incl. HIV	R
9	`IMAGING_META`	ImagingStudy metadata, report impression	N
10	`IMAGING_PIXEL`	DICOM instance data	N
11	`PATHOLOGY`	DiagnosticReport (histopathology), WSI	N
12	`CLINICAL_NOTES`	DocumentReference, Composition	N
13	`GENOMIC`	MolecularSequence, GenomicStudy, PGx	R
14	`MENTAL_HEALTH`	Psychiatric Condition, Observation, notes	R
15	`SEXUAL_REPRO`	Sexual and reproductive health, terminations	R
16	`SUBSTANCE_USE`	Substance use disorder (42 CFR Part 2 scope)	R
17	`SOCIAL_DETERM`	Housing, income, safeguarding, immigration	R
18	`ADVANCE_DIRECTIVE`	Consent, advance decisions, DNACPR	N
19	`RESEARCH_DERIVED`	Pseudonymised or aggregated derivatives	U

Compartments are not merely categories: because each has an independent root key, the set of compartments a reader can decrypt is exactly the set of root keys that have been wrapped to that reader. Adding a twenty-first class is a governance action; it does not require re-keying anything that already exists.

4.3 The sensitivity lattice

Four tiers, ordered, aligned to HL7 v3 Confidentiality:

Tier	HL7	Meaning	Typical content
T0	`U`	Unrestricted	De-identified derivatives, public health aggregates
T1	`N`	Normal	Ordinary clinical care
T2	`R`	Restricted	Behavioural health, sexual health, genomics, safeguarding
T3	`V`	Very restricted	Patient is a staff member; protected identity; active safeguarding investigation

Tier is a property of the compartment, not of the reader. A reader carries a *clearance*, and the predicate is tier $\leq$ clearance. Note that the same clinical fact can exist in two compartments at two tiers: a lithium prescription appears in `MEDICATIONS`/N as a drug and dose, and the reason for it appears in `MENTAL_HEALTH`/R. This is deliberate — an emergency physician must be able to see that the patient takes lithium without thereby acquiring the psychiatric formulation.

4.4 Element cells and canonicalisation

Within a compartment, each FHIR resource is canonicalised to a set of leaf element paths in FHIRPath-like notation (`name.given[0]`, `code.coding[0].display`, `bp.systolic`). Canonicalisation is deterministic: keys are sorted lexicographically, arrays are indexed, numbers are serialised in a single canonical form. Each leaf value is sealed independently:

$$c_p = \mathrm{AEAD}(k_p;\ v_p;\ \mathrm{AAD}_p), \qquad \mathrm{AAD}_p = \text{“MedLattice|cell”} \,\|\, \mathrm{cid} \,\|\, p \,\|\, \mathrm{suite} \,\|\, \mathrm{epoch}$$

Binding the compartment identifier, the element path, the suite and the epoch into the associated data means a cell cannot be transplanted between records, between elements, or between epochs — a cut-and-paste attack on the vault produces an authentication failure, not a plausible false record.

4.5 The element Merkle tree and the anchor

The cells of a compartment version are Merkle-committed:

$$\ell_i = \mathrm{SHAKE256}(\text{“cellleaf”} \| p_i \| c_{p_i}), \qquad R = \mathrm{MerkleRoot}(\ell_0, \ldots, \ell_{n-1})$$

R is the `elementRoot` written on chain. Two consequences follow, and they are the reason for the construction:

1. **Selective disclosure retains provenance.** A reader given four of thirteen elements receives, with each, an $O(\log n)$ Merkle path proving that cell is exactly what the author committed to — without learning anything about the other nine.
2. **Redaction is detectable but not forgeable.** A gateway can withhold; it cannot substitute.

The author signs $\text{“anchor”} \| \mathrm{cid} \| R \| \mathrm{digest}_{ct} \| n$ under ML-DSA-87 + Ed448. The signature is 4,741 bytes and lives in the vault; only its SHAKE-256 digest is anchored.

4.6 Versioning and amendment

Clinical records are amended, never overwritten. Each compartment carries a **hash-linked anchor chain**: anchor v contains the hash of anchor $v-1$. Version numbers are strictly sequential and enforced on chain

— `writeAnchor` reverts on a skipped or replayed version. `verifyChain` lets any reader confirm, from the head alone, that the sequence of amendments they have been shown is contiguous and complete. A suppressed intermediate amendment — the classic retrospective-alteration scenario in medico-legal disputes — breaks the chain and is detectable.

4.7 Epochs and unlinkability

The `epoch` component of a compartment identifier is a rotating, salted counter. Because the on-chain identifier is keccak256 of a pseudonym and an epoch, an observer with full ledger access — including a validator in another jurisdiction — cannot link a patient's compartments across epochs without the epoch salt. Epoch rotation is also the mechanism for retroactive element revocation: revoking a reader's access to an element they were once granted requires a fresh key tree, i.e. a new epoch, since one cannot un-give a derived key.

5 Cryptographic specification

5.1 The threat that sets the parameters

Mosca's inequality states that migration is already overdue when

$$X + Y > Z$$

where X is the required secrecy lifetime of the data, Y is the time required to migrate the estate, and Z is the time until a cryptographically relevant quantum computer (CRQC).

For health records the numbers are not close.

Quantity	Value for MedLattice	Basis
X — secrecy lifetime	50–90 years	A genomic or paediatric record retains identifiability for the patient's life and implicates relatives beyond it. Kenya's Digital Health Act 2023 mandates 20 years of retention as a floor.
Y — migration time	5–8 years	Estate-wide re-keying, device fleets, medical devices with decade-long lifecycles, vendor dependency.
Z — time to CRQC	2030–2040 (contested)	NIST anticipates ECDSA deprecation by 2030 and disallowance by 2035 (IR 8547); NSA CNSA 2.0 sets PQC transition milestones through 2033–2035.

$X + Y \approx 55$–98 years against $Z \approx 4$–14 years. The inequality is violated by an order of magnitude. The operational consequence is specific and unavoidable: **every ciphertext produced today under classical-only key establishment is a ciphertext an adversary can harvest now and decrypt later.** There is no retrofit for a ciphertext that has already been exfiltrated. This is why MedLattice uses post-quantum key establishment for data at rest from the first record written, and why the design treats transport hybridisation as necessary but not sufficient.

5.2 Suite QMED-KEM-1 — hybrid key encapsulation

$$\text{QMED-KEM-1} = \text{ML-KEM-1024 (FIPS 203)} \ + \ \text{X448 (RFC 7748)}$$

Encapsulation to a recipient bundle (ek, pk_X):

1. $(ss_M, ct_M) \leftarrow$ ML-KEM-1024.Encaps(ek)
2. $e \leftarrow$ fresh X448 scalar; $ct_X \leftarrow e \cdot G$; $ss_X \leftarrow e \cdot \text{pk}_X$
3. $K \leftarrow \text{SHA3-256}(\texttt{0x01} \| ss_M \| ss_X \| ct_X \| \text{pk}_X \| \texttt{"QMED-KEM-1"})$

Step 3 is the **X-Wing combiner** (Connolly, Schwabe and Westerbaan), generalised from the standardised ML-KEM-768 + X25519 instantiation to a Category-5 profile. Two points matter and are frequently got wrong in ad-hoc hybrids:

- **Concatenating the two shared secrets is not enough.** Binding ct_X and pk_X into the hash is what gives the construction its multi-user security and key-binding properties.
- **The construction is IND-CCA2 secure if *either* component is.** That is the entire point of hybridisation: ML-KEM is young, and X448 is quantum-broken. Neither alone is an acceptable bet for a fifty-year secret.

Sizes: public bundle 1,624 bytes (1,568 + 56); ciphertext 1,624 bytes; shared secret 32 bytes.

Why Category 5 at rest but Category 3 in transit

A TLS session key protects data for the lifetime of a connection — seconds. A compartment root key protects data for the lifetime of a patient — decades. The Category-3 hybrid `X25519MLKEM768` is used for transport because it is the group with broad, tested deployment across TLS stacks and hardware; the Category-5 profile ML-KEM-1024 + X448 is used for anything at rest, because the security margin is what is being bought and the size penalty is paid once per wrap rather than once per packet.

5.3 Suite QMED-AEAD-1 — content encryption with key commitment

$$\text{blob} = \underbrace{\text{nonce}}_{12} \,\|\, \underbrace{\tau}_{32} \,\|\, \text{AES-256-GCM}(k;\ v;\ \text{AAD} \| \tau)$$

$$\tau = \text{SHA3-256}(\texttt{"CTX"} \| k \| \text{nonce} \| \text{AAD})$$

AES-256-GCM is not key-committing. An adversary who holds two keys can construct a single ciphertext that decrypts successfully — with no authentication failure — under both. In a general-purpose protocol this is a curiosity; in a record system where the same blob is wrapped to dozens of readers with different entitlements, it is a *partitioning oracle* and a route to serving one reader a different clinical fact from another while both see a valid authentication tag. The explicit CTX-style commitment τ binds the key to the ciphertext and closes it. The reference implementation verifies τ in constant time before invoking GCM at all.

Nonces are 96-bit random, with a per-key counter ceiling of 2^{32} enforced at the KWS; in practice each cell key is used exactly once.

5.4 Suite QMED-SIG-1 — clinical attestation

$$\sigma = \text{ML-DSA-87}(sk_{\text{PQ}}; m) \,\|\, \text{Ed448}(sk_{\text{EC}}; m), \qquad m = \text{“MedLattice|sig|”} \| \text{message}$$

Concatenated rather than nested hybridisation: a forger must break both schemes. Verification requires both to pass. Sizes: verification bundle 2,649 bytes; signature 4,741 bytes (4,627 + 114).

This is the signature that answers “who wrote this note, and has it changed since?” It is a *clinical* signature with a medico-legal lifetime, which is precisely the case in which a hybrid is worth its size: a note signed today may be contested in 2060.

5.5 Suite QMED-NOT-1 — hash-only notarisation and assumption diversity

Everything above rests, on the post-quantum side, on the Module-LWE and Module-SIS assumptions. Lattice cryptanalysis is an active field. It is prudent to ask what survives if module lattices are broken in 2041.

The answer must be: *the integrity of the historical record*. Confidentiality cannot be recovered once lost, but a system in which an adversary can also **rewrite** history after such a break is qualitatively worse — it would make every historical record medico-legally worthless.

MedLattice therefore notarises each epoch's Merkle root under **SLH-DSA-SHA2-256s** (**FIPS 205**), whose security rests on hash functions alone, and lodges the attestation with an external timestamp authority as well as on chain. A break of lattices costs future confidentiality; it does not permit undetected revision of what was recorded.

The accompanying reference client implements a Merkle-over-Winternitz construction of the same shape at demonstration scale, and shows that a back-dated root fails verification (Appendix C, §7).

5.6 Hash functions and the Grover margin

SHAKE-256 and SHA3-256 throughout the post-quantum layer; Keccak-256 where the EVM requires it. Grover's algorithm gives a quadratic speed-up against preimage search, so a 256-bit digest retains a 128-bit post-quantum security level — which NIST accepts, and which is further protected by the fact that Grover parallelises poorly (Zalka): k quantum processors give only a $\sqrt{k}$ improvement, so the attack does not decompose into a practical distributed computation the way classical brute force does.

Archival anchors — the epoch roots intended to be meaningful past 2075 — use 384-bit SHAKE-256 output. The cost is negligible and the margin doubles.

5.7 Symmetric strength

AES-256 under Grover has a 128-bit post-quantum security level. NIST Category 5 is defined by reference to exactly this. No change is required, and proposals to move to larger symmetric keys are cargo cult.

5.8 Transport

TLS 1.3, hybrid group `X25519MLKEM768`, mutual authentication with composite certificates (ML-DSA-87 + ECDSA P-384) per the LAMPS composite-signature drafts. Client certificates are issued by the consortium CA, whose root is SLH-DSA — again for assumption diversity at the trust anchor.

Note the ordering of the defence: transport hybridisation protects a session; it does *not* protect a ciphertext sitting in a vault. The at-rest suite is what defeats harvest-now-decrypt-later, and it is the more important of the two.

5.9 Randomness

Data-encryption keys and HDKT roots are drawn from a NIST SP 800-90B compliant entropy source, with an optional quantum RNG (a photonic QRNG appliance) as an additional independent source, XOR-combined into the DRBG seed. Combining is by XOR precisely so that a compromised or non-compliant QRNG cannot make the result worse than the conventional source alone. This is a defence-in-depth measure and not a security claim: a well-seeded SP 800-90A DRBG is sufficient.

5.10 Algorithm agility and the re-wrap property

Every identity, every wrap and every anchor carries a four-byte suite identifier registered on chain. `IdentityRegistry.retireSuite` deactivates a suite; the test suite demonstrates that retiring a suite immediately deactivates every identity still bound to it (Appendix B). This is the kill-switch that makes a rapid response to a cryptanalytic result possible.

The economically decisive property is this: **migration re-wraps data-encryption keys; it does not re-encrypt content.** A petabyte-scale imaging archive migrating from QMED-KEM-1 to a successor suite requires re-encapsulating $O(\text{compartments} \times \text{readers})$ 32-byte keys — hours of work — rather than re-encrypting the archive, which would be a multi-month undertaking with a corresponding integrity risk. This is the single most important structural decision in the cryptographic design.

5.11 The residual classical dependency

Stated plainly: the EVM transaction signature is not post-quantum

Every state-changing call to the three contracts is authorised by a secp256k1 ECDSA signature, because that is what the EVM natively verifies. secp256k1 falls to Shor's algorithm. This is a real and unresolved gap, and it should not be glossed.

Four mitigations apply, in descending order of strength:

1. **The network is permissioned.** An adversary who has forged a transaction signature must additionally have obtained a validator-authorised node connection and mTLS client credentials, which are themselves hybrid-PQ. Forgery alone buys nothing.
2. **No confidentiality depends on it.** A forged transaction can *record* a false consent or a false capability; it cannot decrypt anything, because the HDKT node keys are released by the KWS under the hybrid KEM, and the KWS independently verifies the requester's ML-DSA-87 signature over the query envelope.
3. **Application-layer PQ signatures already exist.** Every capability request carries an ML-DSA-87 signature that the KWS verifies off-chain. The on-chain record is corroborating evidence, not the sole authority.
4. **A defined migration.** EIP-8051 specifies precompiles `VERIFY_MLDSA` (0x12) and `VERIFY_MLDSA_ETH` (0x13) at 4,500 gas per verification. As Besu is a client the consortium controls, the consortium ships these precompiles in its own build now, and the contracts are written against that interface, so that when the precompile stabilises on mainnet the migration is a configuration change. Related work — EIP-7932 (transaction algorithm agility), EIP-8141 (account abstraction giving per-account signature agility, expected late 2026) and Ethereum's leanXMSS consensus signatures targeted at roughly 2029 — determines the timetable for the public-chain path.

5.12 What is not claimed

Limits of the security argument

1. Nothing here is information-theoretically secure. All claims are computational and rest on the hardness of Module-LWE/SIS, the collision and preimage resistance of Keccak, and the security of AES.
2. "Quantum-proof" means *quantum-resistant under current cryptanalysis*. No one can prove a scheme secure against an unknown future algorithm. The honest formulation is: MedLattice uses the NIST-standardised Category-5 parameter sets, hybridises them with classical primitives so a break of either alone is survivable, diversifies the assumption for historical integrity, and can retire any suite from the ledger within one block.
3. The reference implementations used here (`kyber-py`, `dilithium-py`) are spec-faithful but not side-channel hardened. Production deployment requires a FIPS 140-3 validated module. This is a procurement item, not a design change.
4. FIPS 206 (FN-DSA / Falcon) is selected but unpublished as of this writing, and its floating-point Gaussian sampler carries implementation risk. It is deliberately not used. HQC, selected March 2025 as a backup KEM on a different (code-based) assumption, is a candidate for a future `QMED-KEM-2` once published — it would add genuine assumption diversity to key establishment, which is currently lattice-monocultural.
5. MedLattice does not defend against a clinician who is entitled to see a record and then photographs the screen. No cryptosystem does. It defends by making the access attributable, which is what turns a technical control into an evidential one.

5.13 Parameter summary

Function	Primitive	Standard	Level	Public	Ciphertext / Sig
Key encapsulation	ML-KEM-1024 + X448, X-Wing combiner	FIPS 203 / RFC 7748	Cat 5	1,624 B	1,624 B
Content encryption	AES-256-GCM + CTX key commitment	SP 800-38D	256-bit	—	+44 B
Clinical signature	ML-DSA-87 + Ed448	FIPS 204 / RFC 8032	Cat 5	2,649 B	4,741 B

Function	Primitive	Standard	Level	Public	Ciphertext / Sig
Notarisation	SLH-DSA-SHA2-256s	FIPS 205	Cat 5	64 B	29,792 B
Hash / KDF	SHAKE-256, SHA3-256	FIPS 202	≥128-bit PQ	—	—
Archival digest	SHAKE-256 (384-bit output)	FIPS 202	≥192-bit PQ	—	—
Ledger digest	Keccak-256	EVM-native	≥128-bit PQ	—	—
Transport	TLS 1.3, `X25519MLKEM768`, composite mTLS	RFC 8446 + PQ drafts	Cat 3	—	—

6 Granular access control

6.1 Three levels of granularity

Level	Unit	Enforced by	What defeat requires
1	Compartment	Possession of the compartment root key	Breaking ML-KEM-1024 + X448, or stealing an HSM key
2	Resource / time window	On-chain capability (`compartmentId`, `notAfter`, `maxQueries`)	Forging a ledger state transition on a permissioned BFT chain
3	Element (cell)	HDKT node-key cover	Inverting SHAKE-256

The essential difference from conventional RBAC is at level 3. In a conventional EPR, "the clerk cannot see the diagnosis" means *the application does not render it*. A SQL injection, a misconfigured API, a rogue DBA or a compromised middleware tier converts that into "the clerk can see the diagnosis". In MedLattice the clerk has never held a key from which the diagnosis cell's key is derivable. The reference client demonstrates the distinction directly: handed every ciphertext in the compartment, the clerk's grant decrypts 4 of 13 cells and fails the key-commitment check on the other 9 (Appendix C, §5).

6.2 The Hierarchical Disclosure Key Tree

Let the compartment's canonical element paths be $p_0 < p_1 < \cdots < p_{n-1}$ and let $d = \lceil \log_2 n \rceil$. Define a complete binary tree of depth d with root key k_ε — the compartment data-encryption key — and

$$k_{u\|0} = \mathrm{KDF}(k_u, \texttt{"L"}), \qquad k_{u\|1} = \mathrm{KDF}(k_u, \texttt{"R"}), \qquad k_i^{\mathrm{cell}} = \mathrm{KDF}(k_{\mathrm{leaf}(i)}, \texttt{"cell"})$$

with $\mathrm{KDF}(k, \ell) = \mathrm{SHAKE256}(\texttt{"MedLattice/v1|"} \| \ell \| k)$. This is a GGM tree: the derivation is one-way, so a node key yields every descendant and nothing else.

Grant. For a permitted leaf set S, the KWS releases the *minimal cover* — the smallest set of tree nodes whose subtrees partition S exactly. For $|S| = m$ out of n leaves the cover has size $O(m \log(n/m))$, and in the common case where the permitted elements are contiguous under the canonical ordering it is $O(\log n)$. Measured on the demonstration record: a consultant's full access to a 13-element demographics compartment costs 3 node keys (96 bytes); a nurse's 3-element view costs 3; a clerk's 6-element view costs 5.

Properties.

- *Forward granularity.* A reader can derive exactly its granted leaves. No configuration error can widen the grant, because widening requires a key the reader does not have.
- *Gateway independence.* A fully compromised gateway can leak what it was given. It cannot leak what it was not given.

- *Revocation asymmetry.* Future grants can be narrowed at zero cost. Retroactive revocation of an already-derived key is impossible, and requires a new epoch — which is why epochs exist and why epoch rotation is an operational routine rather than an exception (§10).

6.3 The four predicates

Access is the conjunction of four independent conditions, evaluated on chain by `ConsentCapabilityManager.evaluate`, a `view` function returning a decision and a stable machine-readable reason code.

The authorisation kernel

(1) ROLE. The requester holds a live, unexpired, organisation-scoped role credential whose class mask covers the data class and whose clearance meets the tier.
(2) RELATIONSHIP. An unexpired care relationship, derived from an actual FHIR Encounter and attested by a gateway, links requester to patient. Required for `TREAT`, `ETREAT` and `PATRQT`; waived for secondary uses that operate on pseudonymised projections.
(3) CONSENT. The patient has not restricted this data class, or has explicitly named this grantee.
(4) PURPOSE. The declared purpose of use is one the role may assert for this class — so a billing officer can never assert treatment purpose, whatever else is true.

Predicate (2) is the one most often omitted in blockchain-EPR designs and it is the one that matters most in practice. Role alone is not a licence: an oncology consultant with no relationship to a patient is not entitled to that patient's record, and the great majority of real-world inappropriate-access incidents in health systems are exactly this case — a legitimately credentialled member of staff looking up a neighbour, a celebrity or an ex-partner. The test suite asserts that no seniority defeats it (Appendix B).

6.4 Staff tiering — the entitlement matrix

The matrix below is the governance-set default. It is data, not code: it is installed by `setRoleProfile` and every change is a ledger event.

Role	Data classes	Clr	Purposes	BG	Element scope (illustrative)
R0 Patient	all own	V	PATRQT	—	all
R1 Registration clerk	DEMOGRAPHICS, ADMIN_COVERAGE	N	HOPERAT, HPAYMT	—	name, DOB, gender, identifier, contact
R2 Healthcare assistant	DEMOGRAPHICS, VITALS	N	TREAT	—	name, DOB; all vitals
R3 Registered nurse	+ ALLERGY, MEDICATIONS, PROBLEMS, ADV.DIR	N	TREAT, HOPERAT	—	problem `code.text` + status only, not notes
R4 Junior doctor	+ LABS_ROUTINE, CLINICAL_NOTES, IMAGING_META	N	TREAT, HOPERAT	—	full within class
R5 Attending / consultant	+ LABS_SPECIAL, IMAGING_PIXEL, PATHOLOGY, GENOMIC, SOCIAL_DETERM	R	TREAT, HOPERAT, HRESCH	—	full within class
R6 Pharmacist	DEMOGRAPHICS, MEDICATIONS, ALLERGY, PROBLEMS, LABS_ROUTINE	N	TREAT	—	renal/hepatic labs only
R7 Laboratory scientist	DEMOGRAPHICS, LABS_ROUTINE, LABS_SPECIAL	R	TREAT, HOPERAT	—	specimen + result, no notes
R8 Radiologist	DEMOGRAPHICS, IMAGING_*, PROBLEMS, PROCEDURES	N	TREAT	—	full within class
R9 Mental health clinician	DEMOGRAPHICS, MEDICATIONS, PROBLEMS, NOTES, MENTAL_HEALTH, SUBSTANCE_USE	R	TREAT, HOPERAT	—	full within class
R10 Emergency clinician	broad clinical + MENTAL_HEALTH, SUBSTANCE_USE, ADV.DIR	R	TREAT, ETREAT	**yes**	full within class

Role	Data classes	Clr	Purposes	BG	Element scope (illustrative)
R11 HIM coder	DEMOGRAPHICS, PROBLEMS, PROCEDURES, NOTES	N	HOPERAT	—	coded elements only, notes redacted to impression
R12 Billing	DEMOGRAPHICS, AD-MIN_COVERAGE, PROCEDURES	N	HPAYMT	—	surname, DOB, member number, procedure code + date
R13 Researcher	RESEARCH_DERIVED	U	HRESCH	—	full within class
R14 Auditor / DPO / Caldicott Guardian	*none*	U	HOPERAT	—	**content-blind** — metadata and logs only
R15 Node / system operator	*none*	U	*none*	—	**content-blind** — cryptographically excluded

Three features of this matrix are worth drawing out.

R5 does not dominate R9. A general-medicine consultant with restricted clearance still cannot read the `MENTAL_HEALTH` or `SUBSTANCE_USE` compartments, because seniority is not the same axis as specialty. This is what 42 CFR Part 2 requires in the United States and what Caldicott Principle 4 requires in the United Kingdom, and it is a distinction that ordinary hierarchical RBAC cannot express. The test suite asserts it explicitly.

R14 and R15 are content-blind by construction. The Caldicott Guardian and the DPO need to see *who accessed what, when, and why* — never the clinical content itself. The node operator, who has physical access to the servers, is excluded not by policy but by never appearing as a recipient in any key wrap. This is the property that makes outsourced hosting defensible: the hosting provider is R15.

Element scope is a computed intersection, not a promise. The gateway intersects the role's permitted element patterns with the compartment's actual paths and with the client's requested paths, and mints a capability bound to the hash of that intersection. `consume` reverts if the presented element set differs by one character.

6.5 Purpose of use

Purposes follow HL7 v3 ActReason: `TREAT`, `ETREAT`, `HPAYMT`, `HOPERAT`, `HRESCH`, `PATRQT`, `HLEGAL`, `PUBHLTH`. Purpose is asserted by the requester, bound into the capability, and written to the disclosure log. Purpose is not merely descriptive: it selects which predicates apply (relationship is required for treatment purposes and waived for pseudonymised secondary use) and which overrides are available.

Asserting a false purpose is therefore a discrete, logged, attributable act — which is what makes it a disciplinary and regulatory matter rather than an invisible one.

6.6 Consent

A directive is recorded per (patient, data class), signed by the patient's controller key, carrying a legal-basis code and a commitment to the signed FHIR `Consent` resource.

Policy	Effect
`UNSET`	Role matrix governs
`ROLE_DEFAULT`	Role matrix governs (explicitly affirmed)
`EXPLICIT_ONLY`	Role matrix insufficient; a named `grantByPatient` grant is required
`DENY`	Patient restriction: defeats the role matrix for all purposes except `ETREAT`, `HLEGAL`, `PUBHLTH`, each of which routes to the override path and notifies the patient

Directives may carry an expiry. A lapsed directive falls back to the role default rather than failing open or closed by accident — the test suite pins this behaviour, because "fails closed after the consent expires" and

"fails open after the consent expires" are both defensible-sounding and both wrong for different reasons.

Patients may also sever a care relationship unilaterally (`revokeRelationship`), which implements the UK GDPR Article 21 objection right and the NHS national data opt-out at the level of an individual clinician.

6.7 Break-glass

Emergency access relaxes the relationship and consent predicates. It never relaxes the role or clearance predicates.

Control	Implementation
Eligibility	Role profile must carry `mayBreakGlass`; R10 only in the default matrix
Justification	Non-empty commitment to a free-text justification, held off chain; empty reverts
Duration	Hard ceiling `breakGlassTTL`, default 4 h, governance-settable but capped at 24 h in code
Scope	Still bounded by the role's class mask and clearance — an ED registrar cannot break glass into `GENOMIC`
Patient notice	`PatientNotified` event emitted in the same transaction
Review	`breakGlassCount` per requester, monotonically increasing; feeds mandatory retrospective review
Patient veto	The patient may revoke the capability, which takes effect on the next query

The design intent is that break-glass be *easy to invoke and impossible to hide*. A clinician facing an unresponsive patient must not be obstructed; a clinician who invokes it eleven times in a month must be visible to the Caldicott Guardian without anyone having to go looking.

6.8 The resolution pipeline

Eight stages, in order

1. Syntactic validation. Query envelope parsed; ML-DSA-87 signature over the envelope verified against the requester's registered key epoch; nonce checked for replay; `notAfter` checked.
2. Compartment resolution. Requested resource and time window are mapped to a compartment identifier set. Non-existent compartments return `NO_SUCH_COMPARTMENT` — deliberately indistinguishable from an empty one, so that existence is not an oracle.
3. On-chain authorisation. `evaluate()` is called for each compartment; a denial returns its reason code and is written to the denial log.
4. Element intersection. The role's permitted element patterns are intersected with the compartment's actual canonical paths and with the client's requested paths. This is the "minimum necessary" computation.
5. Capability minting. `issueCapability` binds requester, compartment, purpose, window, query budget and `keccak256` of the intersected element set.
6. Metered key release. The KWS calls `consume`, which re-checks every live predicate — role revocation, identity revocation, relationship revocation, consent change — and decrements the budget. It then seals the minimal HDKT cover to the reader's hybrid encapsulation key.
7. Decryption and proof. The client decapsulates, derives cell keys, opens each cell (verifying the CTX commitment first), and verifies each cell's Merkle path against the anchored `elementRoot`.
8. Audit. `logDisclosure` writes the disclosure event: capability, requester, compartment, element-set hash, purpose, break-glass flag, anchor version.

Stages 3, 5, 6 and 8 are on chain. Stages 1, 2, 4 and 7 are off chain. The consequence is that a gateway can fail, be compromised or be malicious, and the worst it can do is deny service or leak what it was legitimately given — it cannot manufacture authority, cannot suppress the audit record of a disclosure it performed, and cannot decrypt beyond its grant.

6.9 Query envelope

```
{
  "requester":   "did:medl:prac:consultant-aggarwal",
  "keyEpoch":    2,
  "subject":     "0x8f3a…",              // patient pseudonym, epoch-salted
  "select":      ["code.text", "clinicalStatus", "onsetDateTime"],
  "where": {
    "class":     "PROBLEMS",
    "tier":      "N",
    "epoch":     42,
    "dateRange": ["2024-01-01", "2026-08-25"],
    "encounter": "Encounter/9f21…"
  },
  "purposeOfUse":     "TREAT",
  "redactionProfile": "R5_ATTENDING_DEFAULT",
  "mode":             "client-decrypt",
  "nonce":            "…",
  "notAfter":         "2026-08-25T09:05:00+03:00",
  "sig":              "<ML-DSA-87 || Ed448 over the canonical envelope>"
}
```

6.10 Worked result

The reference client runs one query — the demographics compartment of a real-shaped record — from four staff positions. The compartment has thirteen canonical elements.

Position	Elements returned	HDKT node keys released	Withheld
Registration clerk	6 of 13	5 (160 B)	address, full given names, contact system, `resourceType`
Registered nurse	3 of 13	3 (96 B)	identifier, contact, address, gender
Billing officer	3 of 13	2 (64 B)	everything but surname, DOB and member number
Consultant	13 of 13	3 (96 B)	—

And from the same four positions against the `PROBLEMS` and `MENTAL_HEALTH` compartments:

Position	`PROBLEMS`/N	`MENTAL_HEALTH`/R	`GENOMIC`/R
Registration clerk	`CLASS_NOT_ENTITLED`	`CLASS_NOT_ENTITLED`	`CLASS_NOT_ENTITLED`
Registered nurse	2 of 8 elements	`CLASS_NOT_ENTITLED`	`CLASS_NOT_ENTITLED`
Junior doctor	8 of 8 elements	`CLASS_NOT_ENTITLED`	`CLASS_NOT_ENTITLED`
Consultant	8 of 8 elements	`CLASS_NOT_ENTITLED`	6 of 6 elements
Psychiatrist	8 of 8 elements	9 of 9 elements	`CLASS_NOT_ENTITLED`
Node operator	`ROLE_CONTENT_BLIND`	`ROLE_CONTENT_BLIND`	`ROLE_CONTENT_BLIND`

Denial codes are ordered: the class predicate is evaluated before the tier predicate, so a role that lacks the class entirely returns `CLASS_NOT_ENTITLED` rather than `CLEARANCE_TOO_LOW`. The latter appears where the class *is* permitted but the tier is not — a junior doctor reading `CLINICAL_NOTES` at tier R, which the contract test suite exercises directly.

After the patient records a restriction on `MENTAL_HEALTH`, the psychiatrist's own query returns `PATIENT_RESTRICTION`; the ED registrar's break-glass invocation succeeds, notifies the patient and increments the review counter. The full transcript is Appendix C.

7 The three smart contracts

All three compile under solc 0.8.28 with `viaIR` and the optimizer at 400 runs, target EVM version Cancun. Deployed sizes are well inside the EIP-170 limit of 24,576 bytes.

Contract	Deployed size	Role
`IdentityRegistry`	6,296 B	Who exists, what keys they hold, what roles they carry
`ConsentCapabilityManager`	10,260 B	Whether a given disclosure is permitted, and the capability that authorises it
`RecordAnchorRegistry`	6,248 B	What was written, what was disclosed, what was erased

7.1 Contract 1 — `IdentityRegistry`

Binds each actor to a DID and to post-quantum key commitments, and issues expiring organisation-scoped role credentials.

Function	Caller	Effect
`registerIdentity`	governance (orgs) / accredited org (all others)	Enrol a DID with `sigSuite`, `kemSuite`, key commitments and validity window
`rotateKeys`	controller or governance	Advance `keyEpoch`; historical epoch commitments are retained
`publishKeyBlob`	controller	Force raw key bytes on chain for unconditional availability
`revokeIdentity`	controller, employing org, or governance	Immediate deactivation
`issueRole`	accredited org	Expiring, org-scoped role with a licence commitment
`revokeRole`	issuing org	Immediate
`registerSuite` / `retireSuite`	governance	Algorithm agility; retirement is the cryptanalytic kill-switch
`setAccreditation`	governance	Admit or expel an organisation
`isActive`, `hasRole`, `keyCommitmentAt`	any	Views consumed by the other two contracts and the gateway

Invariants asserted by the test suite

Organisations are enrolled only by governance. Practitioners are enrolled only by an accredited organisation. An organisation cannot issue a role to itself. **Every role expires** — a past or zero expiry reverts, and there is no perpetual clinical role. Key rotation never invalidates a historical epoch commitment, so a note signed in 2026 still verifies in 2056. Retiring a suite immediately deactivates every identity bound to it. PQ public keys are stored as 32-byte commitments plus a locator, because raw on-chain storage of a 2,592-byte ML-DSA-87 key would cost roughly 1.6 M gas per identity.

7.2 Contract 2 — `ConsentCapabilityManager`

The authorisation kernel. `evaluate()` is the whole access-control policy expressed as a pure function, and it returns a stable reason code rather than a bare boolean — which is what makes denials auditable and support tractable.

Function	Caller	Effect
`setRoleProfile`	governance	Install a role's class mask, clearance, purpose mask, break-glass and content-blind flags
`bindPatientController`	enrolling org, then incumbent	Give the patient her controller key
`recordConsent`	patient controller	Per-class directive with legal basis and document commitment
`establishRelationship`	gateway	Attest an Encounter-derived care relationship with an expiry
`revokeRelationship`	patient, gateway or governance	Sever it
`evaluate`	any (view)	The four predicates; returns `(bool, reasonCode)`
`issueCapability`	gateway	Mint a metered capability after `evaluate` passes

Function	Caller	Effect
`grantByPatient`	patient controller	Patient-directed grant to a named grantee, no relationship required
`invokeBreakGlass`	gateway	Emergency capability; TTL-capped, justified, notified, counted
`consume`	gateway	Re-check live predicates, decrement budget, return the compartment
`revokeCapability`	patient, gateway or governance	Immediate

Invariants asserted by the test suite

A capability is a budget: it has an expiry, a query count and an element-set commitment, and `consume` reverts on expiry, exhaustion or a widened element set. Revoking a role, an identity, a relationship or a consent directive takes effect on the *very next query* — there are no cached grants. Break-glass cannot exceed the role's class mask or clearance, cannot be invoked without a justification commitment, and cannot be extended past 24 hours even by governance. A patient restriction defeats an otherwise entitled clinician and routes emergency and statutory purposes to an explicit override path rather than a silent denial. A lapsed consent directive falls back to the role default. Only an authorised gateway may mint capabilities or attest relationships.

7.3 Contract 3 — `RecordAnchorRegistry`

Integrity, audit and erasure. Three responsibilities that are deliberately kept in one contract because they share the compartment head state.

Function	Caller	Effect
`writeAnchor`	vault writer	Append a version to the hash-linked chain; reverts on skipped/replayed version, revoked author or retired suite
`verifyChain`	any (view)	Confirm a caller-supplied anchor chain is contiguous and terminates at the on-chain head
`logDisclosure`	gateway	Record a completed plaintext release against the capability that authorised it
`logDenial`	gateway	Record a refused attempt
`placeLegalHold` / `liftLegalHold`	governance	Statutory or litigation hold blocking erasure
`cryptoShred`	gateway	Tombstone a compartment whose keys have been destroyed; reverts under an active hold
`notariseEpoch`	vault writer	Anchor an epoch Merkle root with an SLH-DSA attestation and external timestamp reference; strictly monotonic

Invariants asserted by the test suite

Versions are strictly sequential; a skipped or replayed version reverts. An anchor authored by a revoked clinician is refused. A tampered or truncated anchor chain fails `verifyChain`. The disclosure log cannot be written for a compartment the capability does not cover, and only a gateway may write it — a clinician can neither forge nor suppress an entry. Erasure is blocked while any legal hold is active, is idempotent, and permanently closes the compartment to further anchors. Epoch notarisations are monotonic, so no attestation can be back-dated.

7.4 Erasure, legal hold and the tombstone

`cryptoShred` does not delete data. It records that the key custodian has destroyed the compartment root key and every wrap of it, after which the ciphertext in the vault is permanently undecryptable and can be garbage-collected at leisure. The ledger retains only the tombstone: *that* an erasure occurred, when, on whose request and under which legal basis.

This is what makes an immutable ledger compatible with a right to erasure, and it is why the on-chain/off-chain boundary of §3.3 is a hard rule rather than a preference. The unresolved question is not technical but

legal — whether destruction of the key constitutes erasure of the personal data under UK/EU GDPR Article 17. The prevailing regulatory view (EDPB guidance on blockchain, and the reasoning of most national authorities) is that irreversibly rendering data unintelligible is sufficient, but this has not been definitively litigated. Decision D-11 asks for that residual legal risk to be accepted explicitly rather than by omission.

7.5 Gas profile

Measured on a full EVM at Cancun rules with the optimizer enabled.

Operation	Gas	Share of a 30 M block
`establishRelationship`	117,174	0.39 %
`issueCapability`	227,480	0.76 %
`consume` (per query)	64,816	0.22 %
`writeAnchor`	275,987	0.92 %
`logDisclosure`	73,257	0.24 %
`invokeBreakGlass`	232,490	0.78 %

The hot path is `consume` + `logDisclosure` = 138,073 gas, so a single 30 M block accommodates about 217 fully audited disclosures; at a 4-second block period that is about 54 disclosures per second sustained, before batching. A regional deployment serving 2 million patients at 8 record accesses per patient-year requires about 0.5 disclosures per second at mean load, so the design has roughly two orders of magnitude of headroom. Anchor writes are the expensive operation and are naturally batched.

7.6 Test coverage

61 assertions, 61 passing, 0 failing, executed against a full EVM implementation rather than a mock. Coverage by area:

Area	Assertions
Deployment, code size, suite registration	2
Identity enrolment, authority, duplicates, suite binding	4
Role issuance, scoping, expiry, self-issuance	4
Key rotation, historical epochs, revocation, suite kill-switch	3
Staff entitlement matrix across 10 roles	8
Consent: restriction, override routing, lapse, authority	5
Capabilities: minting, element binding, budget, live revocation, expiry	6
Patient-directed grants	2
Break-glass: TTL, scope, justification, ceiling, patient veto	6
Anchoring: sequencing, authority, author validity, chain verification	5
Disclosure log: binding, authority, denials	3
Erasure: legal hold, tombstone, idempotence, closure	4
Notarisation: attestation, monotonicity	2
Cross-contract invariants (no PHI-capable field; governance exclusivity)	2
Gas envelope	1

The invariant test worth singling out is the last but one: it walks every function of all three ABIs and asserts that no parameter or return value is a free-text `string` or a variable-length `bytes` field, with the single audited

exception of `publishKeyBlob`. This is a structural guarantee that no future contributor can add a field into which PHI could be written, and it is checked mechanically on every build rather than by review.

7.7 Governance and upgrade

The contracts are not proxy-upgradeable, deliberately: an upgradeable authorisation kernel is an authorisation kernel that a compromised governance key can rewrite retroactively. Instead:

- **Policy is data.** Role profiles, accreditations, suites, gateways and writers are all mutable state under governance control, so ordinary policy evolution needs no redeployment.
- **Code changes are migrations.** A new kernel is deployed alongside, the registry is re-pointed, and the old contract remains readable forever. Historical decisions stay auditable under the rules that were actually in force at the time — which is the property a medico-legal audit actually needs.
- **Governance is a threshold multisig** of consortium members with a timelock on anything affecting role profiles or suites, so that a policy change is visible before it takes effect. Recommended: 5-of-9, 48-hour timelock (Decision D-03).

8 Threat model and security analysis

8.1 Adversaries and mitigations

#	Adversary	Capability assumed	Primary mitigation	Residual risk
A1	Curious insider clinician	Valid credentials, valid role	Relationship predicate; compartment keys; element cover; every access logged and attributable	Access within a genuine care relationship is legitimate by construction; detection is retrospective
A2	Malicious insider clinician	As A1, plus intent	As A1, plus break-glass counters and denial logging feeding anomaly detection	Screen photography, verbal disclosure — outside any cryptosystem's scope
A3	Compromised gateway	Full control of an enforcement point	Decisions are on chain; keys are released per capability; client-decrypt mode means no plaintext transits it	In gateway-decrypt mode, plaintext for sessions in flight is exposed (hence D-07)
A4	Compromised vault / hosting provider	All ciphertext, no keys	Cell-level AEAD; no key material at rest with the data; R15 content-blind	Traffic analysis of access patterns; object-size side channels
A5	Malicious validator	Full ledger state; can propose blocks	No PHI on chain; QBFT tolerates f of $3f + 1$; epoch-salted pseudonyms defeat linkage	Metadata: which pseudonyms are active, disclosure frequency, compartment counts
A6	Validator collusion ($> f$)	Can rewrite recent history	SLH-DSA epoch notarisation lodged with an external TSA; optional anchoring to a public L1	Detection, not prevention; a colluding supermajority can halt the chain
A7	Harvest-now-decrypt-later	Passive capture of all ciphertext; future CRQC	ML-KEM-1024 + X448 at rest from record one; PQ transport	If ML-KEM *and* X448 both fall, past ciphertext is lost. Assumption diversity here is the open item — see D-09 (HQC)
A8	Key-custody compromise	HSM breach at one organisation	Per-organisation key domains; compartment keys never leave the HSM in the clear; threshold custody for tier-V	An organisation's own patients' compartments are exposed

#	Adversary	Capability assumed	Primary mitigation	Residual risk
A9	Patient key loss	Legitimate patient loses her controller key	M-of-N social/institutional recovery with PQ re-wrap; recovery is logged and delayed	Recovery quorum collusion could impersonate the patient; hence the delay and the notice
A10	Regulator / lawful access	Legal compulsion	`HLEGAL` purpose, logged; legal hold blocks erasure	By design the system complies visibly; it is not a warrant-proof store
A11	Quantum adversary vs. the ledger	Shor against secp256k1 transaction signatures	Permissioned network + PQ mTLS + application-layer ML-DSA; EIP-8051 precompile migration	Real gap until precompiles ship — see §5.11

8.2 Metadata leakage — the honest account

Even with no PHI on chain, a ledger observer learns a non-trivial amount:

- **Existence and cardinality.** How many compartments a pseudonym has, and in which classes. That a pseudonym has a `MENTAL_HEALTH` compartment at all is itself sensitive.
- **Access frequency and timing.** Frequent disclosures on a compartment suggest acuity; a burst of break-glass events suggests an emergency admission.
- **Organisational graph.** Which organisations attest relationships to which pseudonyms.

Mitigations applied: epoch-salted compartment identifiers so that longitudinal linkage requires the salt; uniform-shape events so that a `MENTAL_HEALTH` disclosure is indistinguishable on chain from a `VITALS` disclosure except by the compartment identifier, which is opaque; and pre-creation of empty compartments across all classes at enrolment, so that the *existence* of a class-14 compartment carries no information.

Not mitigated: timing and volume. A validator running traffic analysis over years can infer clinical intensity. Full mitigation would require constant-rate cover traffic, at a cost the consortium should decide on knowingly (Decision D-08) rather than have imposed.

8.3 Failure modes considered and rejected

Design that was considered	Why rejected
Encrypted PHI stored on chain	Defeats Article 17 absolutely; replicates to every jurisdiction; guarantees eventual decryption of a 50-year secret
Attribute-based encryption (CP-ABE) for granularity	Elegant, but all practical constructions are pairing-based and therefore quantum-broken; PQ ABE is not deployment-ready
Proxy re-encryption for delegated access	Same pairing problem; also concentrates a re-encryption key that is nearly as valuable as the data
One key per patient	Destroys granularity: any reader entitled to anything is entitled to everything
One key per element with no tree	$O(n)$ key material per grant; unusable for a 40-element resource, catastrophic for imaging
Public L1 with PHI pointers only	Metadata leakage to an unbounded audience; gas cost volatility; probabilistic finality unacceptable clinically
Off-chain access control with on-chain logging only	The log then records what a trusted server chose to record; the trust assumption is unchanged

9 Regulatory mapping

The system is specified against four regimes simultaneously. Where they agree, one control satisfies all four. Where they conflict — and they do, in two specific places — the conflict is identified and a resolution proposed for approval.

9.1 HIPAA / HITECH (United States)

Requirement	Citation	MedLattice control
Access control	§164.312(a)(1)	Four-predicate kernel; compartment and HDKT key separation
Unique user identification	§164.312(a)(2)(i)	DID + ML-DSA-87 key per practitioner; no shared accounts possible
Emergency access procedure	§164.312(a)(2)(ii)	`invokeBreakGlass`, TTL-capped, justified, notified, counted
Automatic logoff / session bound	§164.312(a)(2)(iii)	Capability `notAfter` and `maxQueries`
Encryption at rest	§164.312(a)(2)(iv)	AES-256-GCM per cell, PQ-wrapped keys
Audit controls	§164.312(b)	On-chain disclosure and denial log; tamper-evident by construction
Integrity	§164.312(c)(1)	Element Merkle roots, hash-linked anchor chain, ML-DSA author signatures
Transmission security	§164.312(e)(1)	TLS 1.3 hybrid PQ, composite mTLS
Minimum necessary	§164.502(b)	Element-set intersection enforced by key cover, not by policy
Accounting of disclosures	§164.528	Directly queryable from the disclosure log
BAA / vendor exposure	§164.308(b)	Hosting provider is R15, content-blind; arguably not a conduit issue at all
Substance use disorder	42 CFR Part 2	`SUBSTANCE_USE` compartment; R5 does not dominate R9

9.2 UK GDPR, DPA 2018, Caldicott and NHS assurance

Requirement	MedLattice control
Art. 9(2)(h) lawful basis for health data	`legalBasis` code recorded with every consent directive
Art. 15 subject access	`PATRQT` purpose; patient controller key; full compartment read
Art. 17 erasure	`cryptoShred` + tombstone (see §9.5)
Art. 21 objection	`revokeRelationship` and `CONSENT_DENY`
Art. 32 security of processing	The whole cryptographic specification
Art. 33/34 breach notification	Disclosure log gives the exact affected element sets within minutes rather than weeks
DPA 2018 Sch. 3 health data	Compartmentalisation and tiering
Caldicott P1 justify purpose	`purposeOfUse` mandatory and logged
Caldicott P4 minimum necessary	Element cover
Caldicott P6 comply with the law	Legal-basis codes
Caldicott P7 duty to share	Break-glass; `PUBHLTH`; the design does not obstruct legitimate sharing
National data opt-out	Class-level `CONSENT_DENY` on `RESEARCH_DERIVED`
DSPT (CAF-aligned)	Identity, access control, logging and resilience evidence is machine-generated

9.3 EU GDPR and the European Health Data Space

Regulation (EU) 2025/327 entered into force on 26 March 2025. Its staged application matters for planning:

Date	What applies
March 2027	Commission implementing acts with operational rules

Date	What applies
March 2029	Primary use of first-priority categories; secondary use for most categories; EHR-system interoperability **and logging** requirements
March 2031	Second-priority categories (imaging, laboratory results, discharge reports); secondary use for the remainder, including genomic data

Two points of alignment are worth noting. First, EHDS requires EHR systems to have a **logging component**; MedLattice's on-chain disclosure log is that component, and is stronger than the requirement because it is not under the operator's control. Second, EHDS's secondary-use regime maps cleanly onto the `RESEARCH_DERIVED` compartment and the `HRESCH` purpose: a health-data access body becomes an accredited organisation, its researchers hold R13, and every secondary-use disclosure is on the ledger.

The March 2031 date for genomic data is the binding constraint for the `GENOMIC` compartment, and is the reason genomics is tier R from the outset rather than being promoted later.

9.4 Kenya: Data Protection Act and Digital Health Act

Requirement	Source	MedLattice control
ODPC registration as controller/processor	DPA Cap. 411C	Organisational; MedLattice supplies the processing register from ledger events
Health data as sensitive personal data	DPA Cap. 411C	Tier R minimum for the sensitive classes
Notify the Digital Health Agency within 7 days of controller status	DHA 2023	Organisational
20-year retention	DHA 2023	`placeLegalHold` with authority code `KE_DIGITAL_HEALTH_ACT_2023_RETENTION_20Y`
Breach: 48 h to the Agency, 72 h to the ODPC	DHA 2023 / DPA	Disclosure log yields the affected element sets immediately; the clock is met by evidence, not by estimate
Cross-border transfer constraints	DHA 2023 + ODPC guidance	In-country validator and vault shard; see below

Kenya's Digital Health Act appears to restrict cross-border transfer of health data, though the drafting leaves open whether the restriction binds all controllers or principally the Agency itself, and the ODPC's cross-border guidance remains in consultation. The architecture handles the ambiguity structurally rather than betting on an interpretation: **vault shards are jurisdiction-pinned**, so Kenyan patients' ciphertext is replicated only to in-country vault nodes, while the ledger — which holds no PHI — replicates to all validators. If the restrictive reading prevails, no change is required. This is a concrete instance of why the on-chain/off-chain boundary is drawn where it is.

9.5 The two genuine conflicts

Conflict 1 — erasure versus statutory retention

UK/EU GDPR Article 17 gives a right to erasure. Kenya's Digital Health Act 2023 mandates 20 years of retention. For a patient with records in both jurisdictions these are directly opposed.
Resolution. `cryptoShred` is gated on `underLegalHold`. A Kenyan retention hold is placed at record creation with an explicit authority code and a 20-year expiry. An Article 17 request against a held compartment is refused, the refusal is logged with the authority that caused it, and the patient is given that authority in the response — which is what Article 17(3)(b) contemplates for processing required by law. When the hold lapses, the shred proceeds. The test suite exercises exactly this sequence.

Conflict 2 — ledger immutability versus erasure

An immutable ledger cannot forget. The GDPR requires that personal data can be erased.
Resolution. The ledger holds no personal data — only commitments, pseudonyms and policy. Erasure operates

on the content plane by key destruction. The residual question is whether the on-chain *pseudonym* is itself personal data under Article 4(1); the defensible position is that an epoch-salted keccak commitment, with the salt destroyed alongside the keys, is anonymised rather than pseudonymised, and Recital 26 then removes it from scope. **This position has not been tested by a regulator or a court**, and Decision D-11 records it as an explicit acceptance, the fallback being that the shred procedure also destroys the epoch salt — which is why the shred procedure is specified to do so.

10 Operations

10.1 Key lifecycle

Key	Custody	Rotation	On compromise
Compartment root (HDKT root)	Organisation HSM, FIPS 140-3 L3	On epoch rotation (annual, or on demand)	New epoch; re-anchor; old ciphertext remains but is no longer extended
Practitioner KEM + signing	Smartcard / secure element, or platform TEE	Annual, or on role change	`revokeIdentity`; historical signatures still verify via `keyCommitmentAt`
Patient controller	Patient device, with M-of-N recovery	Patient-initiated	Recovery quorum, delayed and notified
Gateway service	HSM	Quarterly	`setGateway(false)`; all its capabilities die at next `consume`
Consortium CA root	SLH-DSA, offline, threshold-shared	10 years	Full re-enrolment; the reason it is hash-based and offline
Epoch salt	Organisation HSM	Per epoch	Destroyed on crypto-shred, deliberately

10.2 Patient key recovery

Loss of the patient controller key must not mean loss of the record — but recovery is also the most attractive attack on patient agency. The design: M-of-N threshold with N drawn from institutional guardians (the enrolling organisation, a second clinical organisation, and a patient-nominated proxy) plus optional patient-held shares.

Recovery is (i) delayed by a governance-set window, default 72 hours; (ii) notified to every contact on file at initiation; (iii) logged on chain as a first-class event; and (iv) followed by mandatory re-wrap of the patient's compartment keys to the new controller key, so a recovered key does not retroactively read anything the old key already read.

10.3 Incident response and the breach clocks

The disclosure log turns breach notification from an estimation exercise into a query. Within minutes of an incident the operator can produce the exact set of (patient pseudonym, compartment, element set, requester, purpose, timestamp) tuples affected. This matters concretely against the tightest clock in scope — Kenya's 48 hours to the Digital Health Agency — and against Article 33's 72 hours.

Standing runbooks: gateway compromise (`setGateway(false)`, all capabilities die at next `consume`); practitioner credential compromise (`revokeIdentity`, immediate); cryptanalytic advance against a suite (`retireSuite`, which deactivates every bound identity within one block, followed by re-wrap); validator compromise (governance expulsion, QBFT continues if the remaining set satisfies $n \geq 3f + 1$).

10.4 Performance envelope

Metric	Value	Basis
Query authorisation latency	4 s worst case, sub-second typical	One QBFT block for `consume`; reads are instant from a local node
Sustained audited disclosures	about 54 / s	138,073 gas hot path against 30 M gas / 4 s
Regional load, 2 M patients	about 0.5 / s mean	8 accesses per patient-year
Headroom	about 100×	—
Key material per grant	32–160 B	HDKT cover, measured
Per-cell ciphertext overhead	60 B	12 nonce + 32 commitment + 16 GCM tag
ML-KEM wrap per recipient	1,624 B	Ciphertext size

The 4-second worst case deserves comment: it applies to the *first* query of a session, which mints a capability. Subsequent queries within the capability's budget consume it, and `consume` can be batched or pre-authorised for a clinic list, so a clinician opening a record in a consultation does not wait for a block.

11 Implementation roadmap

Phase	Duration	Content	Exit criterion
0 — Approval	—	This document; decisions D-01…D-14	Signed
1 — Foundations	3 months	Besu QBFT testnet with EIP-8051 precompiles; contracts audited; HSM integration; FIPS 140-3 PQC module procured	External audit clean; validated module in place
2 — Single-site pilot	4 months	One department, read-only shadow of an existing EPR; R1/R3/R4/R5 roles; no clinical dependency	10,000 shadow disclosures reconciled against the incumbent audit log
3 — Clinical go-live	6 months	One site, write path, break-glass, patient portal, Caldicott review workflow	Zero unexplained disclosures over 90 days; break-glass review process operating
4 — Consortium	9 months	Second and third organisations; cross-organisational relationships; patient-directed grants	Cross-site record retrieval within SLA
5 — Secondary use	6 months	`RESEARCH_DERIVED` pipeline; EHDS-aligned access body integration	First approved secondary-use study executed end to end
Continuous	—	Suite agility drills: rehearse `retireSuite` + full re-wrap twice yearly	Re-wrap of the full estate demonstrated within 72 hours

The suite agility drill in the last row is not decoration. The re-wrap property is the system's principal defence against a cryptanalytic surprise, and an untested recovery capability is not a capability.

12 Decisions requiring approval

#	Decision	Recommended	Rationale
D-01	Chain topology	Permissioned Besu QBFT, 7 validators	Immediate finality; known residency; $f = 2$ tolerance
D-02	Anchor to a public L1?	Yes, daily Merkle root only	Defeats validator supermajority collusion (A6) at negligible cost
D-03	Governance	5-of-9 multisig, 48 h timelock on role profiles and suites	Policy change visible before effect
D-04	At-rest KEM	ML-KEM-1024 + X448 (Cat 5)	50-year secrecy lifetime justifies the margin
D-05	Transport KEM	`X25519MLKEM768` (Cat 3)	Deployment maturity; session-lifetime secrets
D-06	Clinical signature	ML-DSA-87 + Ed448 hybrid	Medico-legal lifetime; forger must break both
D-07	Gateway-decrypt mode	Permit per device class only, with named approval	Real assurance reduction; must not be an organisational default
D-08	Cover traffic against timing analysis	Defer to Phase 4, decide with measured data	Cost is real; benefit is unquantified today
D-09	Second KEM assumption (HQC)	Adopt as `QMED-KEM-2` when FIPS-published	Key establishment is currently lattice-monocultural
D-10	Break-glass TTL	4 hours	Longer than a resuscitation, shorter than a shift
D-11	Accept the Article 17 / pseudonym legal position	Yes, with salt destruction on shred as the fallback	Untested by regulator or court; should be an explicit acceptance
D-12	Kenya vault residency	Jurisdiction-pinned shards from day one	Restrictive reading of the DHA costs nothing if pre-empted
D-13	Default tier for `SEXUAL_REPRO`, `SUBSTANCE_USE`, `SOCIAL_DETERM`	R	Consistent with 42 CFR Part 2 and Caldicott
D-14	Patient recovery quorum	2-of-3 institutional + 72 h delay + notification	Balances loss against impersonation

13 References

1. NIST FIPS 203, *Module-Lattice-Based Key-Encapsulation Mechanism Standard*, 13 August 2024.
2. NIST FIPS 204, *Module-Lattice-Based Digital Signature Standard*, 13 August 2024.
3. NIST FIPS 205, *Stateless Hash-Based Digital Signature Standard*, 13 August 2024.
4. NIST, *PQC Standardization Process* — FIPS 206 (FN-DSA) in development; HQC selected 11 March 2025. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization
5. NIST IR 8547, *Transition to Post-Quantum Cryptography Standards* — ECDSA/RSA deprecation 2030, disallowance 2035.
6. NSA, *Commercial National Security Algorithm Suite 2.0*.
7. Connolly, Schwabe, Westerbaan, *X-Wing: The Hybrid KEM You've Been Looking For*, IACR CiC 1(1), 2024; IETF draft-connolly-cfrg-xwing-kem.
8. Albertini, Duong, Gueron, Kölbl, Luykx, Schmieg, *How to Abuse and Fix Authenticated Encryption Without Key Commitment*, USENIX Security 2022.
9. Zalka, *Grover's quantum searching algorithm is optimal*, Phys. Rev. A 60, 2746 (1999).
10. Mosca, *Cybersecurity in an era with quantum computers: will we be ready?*, IEEE S&P 16(5), 2018.
11. EIP-8051, *Precompile for ML-DSA signature verification* (Draft). https://eips.ethereum.org/EIPS/eip-8051
12. Ethereum Foundation, *Post-quantum cryptography on Ethereum* — leanXMSS, leanVM, EIP-8141 account abstraction. https://ethereum.org/roadmap/future-proofing/quantum-resistance/
13. Regulation (EU) 2025/327 establishing the European Health Data Space; in force 26 March 2025, staged application 2027/2029/2031.
14. 45 CFR §164.312, §164.502(b), §164.528 (HIPAA Security and Privacy Rules); 42 CFR Part 2.

15. UK GDPR and Data Protection Act 2018 Schedule 3; National Data Guardian, *Caldicott Principles* (8 principles, 2020 revision).
16. Kenya Data Protection Act 2019 (Cap. 411C); Digital Health Act No. 15 of 2023.
17. HL7 FHIR R5; HL7 v3 Confidentiality and ActReason (PurposeOfUse) value sets.
18. Goldreich, Goldwasser, Micali, *How to construct random functions*, JACM 33(4), 1986 (the GGM tree underlying the HDKT).

A Selected source listings

The complete Solidity and Python sources accompany this document. Two extracts are reproduced here because they are the two places where the security argument is actually made: the on-chain authorisation kernel, and the key tree that makes element granularity cryptographic.

A.1 The authorisation kernel (`ConsentCapabilityManager.evaluate`)

```
    /// @notice Pure, side-effect-free evaluation of the four predicates.
    /// @return ok     whether access is permitted
    /// @return reason 0 = allowed; otherwise a stable machine-readable denial code
    function evaluate(
        bytes32 requester,
        bytes32 roleId,
        bytes32 orgId,
        bytes32 patient,
        uint8   dataClass,
        uint8   tier,
        uint8   purpose
    ) public view returns (bool ok, bytes32 reason) {
        if (!registry.isActive(requester))                 return (false, "IDENTITY_INACTIVE");
        if (!registry.hasRole(requester, roleId, orgId))   return (false, "ROLE_NOT_HELD");

        RoleProfile storage p = roleProfile[roleId];
        if (!p.defined)                                    return (false, "ROLE_UNDEFINED");
        if (p.contentBlind)                                return (false, "ROLE_CONTENT_BLIND");
        if (!MLT.inMask(p.classMask, dataClass))           return (false, "CLASS_NOT_ENTITLED");
        if (tier > p.clearance)                            return (false, "CLEARANCE_TOO_LOW");
        if (purpose == 0 || purpose > 15)                  return (false, "PURPOSE_INVALID");
        if ((p.purposeMask & (uint16(1) << purpose)) == 0) return (false, "PURPOSE_NOT_PERMITTED");

        // Relationship predicate. Waived only for non-treatment secondary uses
        // that operate on pseudonymised projections, and for the patient herself.
        bool needsRelationship =
            purpose == MLT.POU_TREAT || purpose == MLT.POU_ETREAT || purpose == MLT.POU_PATRQT;
        if (needsRelationship && requester != patient && !hasRelationship(patient, requester)) {
            return (false, "NO_CARE_RELATIONSHIP");
        }

        // Consent predicate.
        ConsentDirective storage c = _consent[patient][dataClass];
        uint8 policy = c.policy;
        if (policy != MLT.CONSENT_UNSET && c.expiresAt != 0 && c.expiresAt <= block.timestamp) {
            policy = MLT.CONSENT_UNSET; // lapsed directive falls back to role default
        }
        if (policy == MLT.CONSENT_DENY && requester != patient) {
            // Statutory overrides are the only escape, and they are never silent.
            if (purpose != MLT.POU_ETREAT && purpose != MLT.POU_HLEGAL && purpose != MLT.POU_PUBHLTH) {
                return (false, "PATIENT_RESTRICTION");
            }
            return (false, "REQUIRES_OVERRIDE_PATH");
        }
        if (policy == MLT.CONSENT_EXPLICIT_ONLY && requester != patient) {
            return (false, "REQUIRES_EXPLICIT_GRANT");
        }
        return (true, bytes32(0));
    }
```

A.2 The Hierarchical Disclosure Key Tree (reference client)

```
class HDKT:
    """
    GGM binary key tree over the canonical, lexicographically sorted element paths
    of a compartment.

        k_root                      = compartment data-encryption key
        k_(node,0) = KDF(k_node, "L") , k_(node,1) = KDF(k_node, "R")
        k_cell_i   = KDF(k_leaf_i, "cell")

    A grant is the minimal *cover* of the permitted leaf set: O(m log(n/m)) node
    keys rather than m leaf keys. A reader can derive every leaf beneath a node it
    was given, and — because the KDF is one-way — nothing else. Revoking an
    element from a future grant costs nothing; revoking it retroactively requires
    a compartment re-key, which is why the epoch dimension exists.
```

```
    """

    def __init__(self, paths: Sequence[str], root_key: Optional[bytes] = None):
        self.paths = sorted(paths)
        self.n = len(self.paths)
        self.depth = max(1, (self.n - 1).bit_length())
        self.size = 1 << self.depth
        self.root_key = root_key or secrets.token_bytes(32)
        self.index = {p: i for i, p in enumerate(self.paths)}

    def node_key(self, depth: int, index: int) -> bytes:
        """Key of the node at (depth, index), derived from the root downwards."""
        k = self.root_key
        for level in range(depth):
            bit = (index >> (depth - 1 - level)) & 1
            k = kdf(k, b"L" if bit == 0 else b"R")
        return k

    def cell_key(self, path: str) -> bytes:
        return kdf(self.node_key(self.depth, self.index[path]), b"cell")

    @staticmethod
    def _cover(depth: int, size: int, leaves: Iterable[int]) -> List[Tuple[int, int]]:
        """Minimal set of (depth, index) nodes whose subtrees are exactly `leaves`."""
        want = set(leaves)
        cover: List[Tuple[int, int]] = []

        def walk(d: int, i: int) -> bool:  # returns True if fully covered
            span = 1 << (depth - d)
            lo, hi = i * span, i * span + span
            present = [x for x in want if lo <= x < hi]
            if not present:
                return False
            if len(present) == span:
                cover.append((d, i))
                return True
            left, right = walk(d + 1, 2 * i), walk(d + 1, 2 * i + 1)
            return left and right

        walk(0, 0)
        return cover

    def grant(self, paths: Sequence[str]) -> List[Tuple[int, int, bytes]]:
        """The covering node keys for `paths` — this is what the wrap service releases."""
        leaves = [self.index[p] for p in paths]
        return [(d, i, self.node_key(d, i)) for d, i in self._cover(self.depth, self.size, leaves)]

    @staticmethod
    def derive_from_grant(grant: Sequence[Tuple[int, int, bytes]], depth: int,
                          leaf_index: int) -> Optional[bytes]:
        """Reader side: derive a cell key from whichever granted node covers it."""
        for d, i, k in grant:
            span = 1 << (depth - d)
            if i * span <= leaf_index < (i + 1) * span:
                key = k
                for level in range(d, depth):
                    bit = (leaf_index >> (depth - 1 - level)) & 1
                    key = kdf(key, b"L" if bit == 0 else b"R")
                return kdf(key, b"cell")
        return None
```

A.3 The hybrid KEM combiner

```
def _xwing_combiner(ss_mlkem: bytes, ss_x448: bytes, ct_x448: bytes, pk_x448: bytes) -> bytes:
    """
    X-Wing combiner shape (Connolly–Schwabe–Westerbaan), generalised from the
    standardised ML-KEM-768+X25519 instantiation to the Category-5 profile.

    Binding the ephemeral X448 ciphertext and the static X448 public key into the
    hash is what gives the construction its multi-user and binding properties;
    naive concatenation of the two shared secrets does not.
    """
    return sha3(b"\x01" + ss_mlkem + ss_x448 + ct_x448 + pk_x448 + b"QMED-KEM-1")


def kem_encapsulate(public_bundle: bytes) -> Tuple[bytes, bytes]:
    """Returns (shared_secret_32, kem_ciphertext). ct = 1568 + 56 = 1624 bytes."""
    mlkem_ek, x448_pk = public_bundle[:1568], public_bundle[1568:1624]
    ss_m, ct_m = ML_KEM_1024.encaps(mlkem_ek)
    eph = X448PrivateKey.generate()
```

```
    ct_x = eph.public_key().public_bytes(
        serialization.Encoding.Raw, serialization.PublicFormat.Raw
    )
    ss_x = eph.exchange(X448PublicKey.from_public_bytes(x448_pk))
    return _xwing_combiner(ss_m, ss_x, ct_x, x448_pk), ct_m + ct_x


def kem_decapsulate(kp: HybridKemKeypair, kem_ct: bytes) -> bytes:
    ct_m, ct_x = kem_ct[:1568], kem_ct[1568:1624]
    ss_m = ML_KEM_1024.decaps(kp.mlkem_dk, ct_m)
    ss_x = kp.x448_sk.exchange(X448PublicKey.from_public_bytes(ct_x))
    return _xwing_combiner(ss_m, ss_x, ct_x, kp.x448_pk)
```

B Contract test suite output

Executed against a full EVM implementation at Cancun rules. **61 assertions, 61 passing.**

```
Deployment
  PASS three contracts deploy under the EIP-170 code-size limit
  PASS genesis crypto suites are active (ML-KEM-1024/X448, ML-DSA-87, SLH-DSA)

Contract 1 — IdentityRegistry: enrolment, PQ keys, roles
  PASS organisations are enrolled by governance only
  PASS a non-accredited caller cannot enrol a practitioner
  PASS duplicate DID registration is rejected
  PASS an identity cannot be enrolled under a retired crypto suite
  PASS role credentials resolve and are organisation-scoped
  PASS an organisation may not issue a role to itself
  PASS roles must expire — a past expiry is rejected
  PASS role expiry is enforced by the clock, not by a job
  PASS PQ key rotation preserves historical epoch commitments
  PASS revoking an identity immediately deactivates it
  PASS retiring a suite deactivates every identity still bound to it (agility kill-switch)

Contract 2 — ConsentCapabilityManager: the staff entitlement matrix
  PASS no care relationship ⇒ no treatment access, however senior the clinician
  PASS relationships may only be asserted by an authorised gateway
  PASS registration clerk sees demographics + coverage — and nothing clinical
  PASS nurse sees vitals and meds but not clinical notes or special labs
  PASS junior doctor reads notes at NORMAL but is blocked at RESTRICTED tier
  PASS consultant clears RESTRICTED genomic data; junior doctor cannot see genomics at all
  PASS mental-health and substance-use compartments are visible only to the psychiatrist, not the consultant
  PASS billing may assert payment purpose but never treatment purpose
  PASS researcher reaches only the de-identified derived compartment, without a relationship
  PASS auditor and node operator are content-blind: no plaintext under any purpose

Contract 2 — patient consent overrides the role matrix
  PASS a patient restriction defeats an otherwise-entitled clinician
  PASS a restriction never blocks the patient reading her own record
  PASS a restriction routes emergency and statutory purposes to the override path, not to silent denial
  PASS only the patient controller may record consent
  PASS a lapsed consent directive falls back to the role default rather than failing open or closed by accident

Contract 2 — capabilities are metered budgets, not standing permissions
  PASS gateway mints a capability for an entitled clinician
  PASS a capability cannot be minted by a non-gateway caller
  PASS the element-set commitment is bound: a widened request is refused at consumption
  PASS the query budget is exhausted after maxQueries
  PASS revoking the care relationship kills an outstanding capability on the very next query
  PASS revoking the practitioner identity kills outstanding capabilities immediately
  PASS capabilities expire on the wall clock

Contract 2 — patient-directed sharing and break-glass
  PASS a patient can grant one compartment to an outside specialist with no care relationship
  PASS a patient cannot grant another patient's compartment
  PASS break-glass yields a time-boxed capability, notifies the patient, and increments the review counter
  PASS break-glass overrides a patient restriction — but only for a role licensed to do so
  PASS break-glass still cannot exceed the role's class mask or clearance
```

```
  PASS break-glass requires a non-empty justification commitment
  PASS the break-glass ceiling cannot be raised past 24 hours
  PASS a patient may revoke a break-glass capability she considers unjustified

Contract 3 — RecordAnchorRegistry: anchoring, audit, erasure
  PASS the vault writer anchors compartment version 1
  PASS versions are strictly sequential — a skipped or replayed version is rejected
  PASS only a registered vault writer may anchor
  PASS an anchor authored by a revoked clinician is refused
  PASS amendments form a hash-linked chain that a reader can verify end to end

Contract 3 — disclosure log
  PASS every plaintext release is logged against the capability that authorised it
  PASS the log cannot be written for a compartment the capability does not cover
  PASS only a gateway may write to the audit log — a clinician cannot forge or suppress entries
  PASS denied attempts are logged too, so probing behaviour is visible to the Caldicott Guardian

Contract 3 — erasure versus statutory retention
  PASS a legal hold blocks crypto-shredding (Kenya DHA 2023: 20-year retention)
  PASS lifting the hold permits erasure, and the ledger keeps only a tombstone
  PASS a shredded compartment accepts no further anchors
  PASS erasure is idempotent and cannot be replayed

Contract 3 — hash-only long-term notarisation
  PASS epoch roots are notarised under SLH-DSA-SHA2-256s with an external timestamp
  PASS epoch notarisations are monotonic — no back-dated attestation

Cross-contract invariants
  PASS no function on any contract accepts or returns plaintext PHI
  PASS governance transfer is exclusive

Gas profile (permissioned QBFT chain, 30M block gas limit)
  PASS the hot path (consume + logDisclosure) stays well inside a single block

    establishRelationship         117174 gas   ~0.391% of a block
    issueCapability               227480 gas   ~0.758% of a block
    consume (per query)            64816 gas   ~0.216% of a block
    writeAnchor                   275987 gas   ~0.920% of a block
    logDisclosure                  73257 gas   ~0.244% of a block
    invokeBreakGlass              232490 gas   ~0.775% of a block

================================================================
  61 passed, 0 failed
================================================================
```

C Reference client transcript

The complete run of `demo.py`: five compartments built from realistic FHIR resources, forty-five element cells, then the same queries issued from seven staff positions, followed by the cryptographic demonstrations that the withheld elements are unreachable rather than merely withheld.

```
══════════════════════════════════════════════════════════════════════
  MedLattice reference client — quantum-resistant granular EPR
══════════════════════════════════════════════════════════════════════

  1. Enrolling identities with hybrid post-quantum key material
──────────────────────────────────────────────────────────────────────
  enrolled 10 identities
  ML-KEM-1024 encapsulation key :  1568 bytes
  X448 public key               :    56 bytes
  ML-DSA-87 verification key    :  2592 bytes
  Ed448 public key              :    57 bytes
  on-chain commitment (keccak)  : 38983e896462f365c9ba316cc520ec01…  (32 bytes)

  2. Role entitlement matrix — classes, clearance, purposes, element scope
──────────────────────────────────────────────────────────────────────
  10 role profiles installed (these mirror ConsentCapabilityManager.setRoleProfile on chain)

  3. Constructing the record — five compartments, cell-level encryption
──────────────────────────────────────────────────────────────────────
  DEMOGRAPHICS   tier N  13 cells  elementRoot e397cf0911d4de79…  anchor v1
  VITALS         tier N   9 cells  elementRoot f2d1ef22404e243d…  anchor v1
  PROBLEMS       tier N   8 cells  elementRoot 73f2a25cafcff874…  anchor v1
  MENTAL_HEALTH  tier R   9 cells  elementRoot fbe1b5edbc91681d…  anchor v1
  GENOMIC        tier R   6 cells  elementRoot c05afbdbc4c708a8…  anchor v1

  45 element cells, each under its own HDKT-derived AES-256-GCM key.
  Author attestation: ML-DSA-87 + Ed448, 4741 bytes.
  Attestation verifies: True

  4. The same query from seven staff positions
──────────────────────────────────────────────────────────────────────

  ── Query A: the DEMOGRAPHICS compartment ──

  Registration clerk         DEMOGRAPHICS/N  ✓ 6 of 13 elements (5 HDKT node keys released)
       birthDate                          "1979-04-11"
       gender                             "female"
       identifier.value                   "NHIF-8830-4417"
       name.family                        "Achieng"
       name.given[0]                      "Mercy"
       telecom[0].value                   "+254-7xx-xxx-118"

  Registered nurse           DEMOGRAPHICS/N  ✓ 3 of 13 elements (3 HDKT node keys released)
       birthDate                          "1979-04-11"
       name.family                        "Achieng"
       name.given[0]                      "Mercy"

  Consultant                 DEMOGRAPHICS/N  ✓ 13 of 13 elements (3 HDKT node keys released)
       address.city                       "Nairobi"
```

```
      address.country                   "KE"
      address.district                  "Kibra"
      birthDate                         "1979-04-11"
      gender                            "female"
      identifier.system                 "urn:oid:2.16.404.1"
      identifier.value                  "NHIF-8830-4417"
      name.family                       "Achieng"
      … 5 more

Billing officer            DEMOGRAPHICS/N  ✓ 3 of 13 elements (2 HDKT node keys released)
      birthDate                         "1979-04-11"
      identifier.value                  "NHIF-8830-4417"
      name.family                       "Achieng"

── Query B: the clinical compartments ──

Registration clerk         PROBLEMS/N  ✗ DENIED — CLASS_NOT_ENTITLED

Registered nurse           PROBLEMS/N  ✓ 2 of 8 elements (2 HDKT node keys released)
      clinicalStatus                    "active"
      code.text                         "Stage 2 hypertension"

Junior doctor              PROBLEMS/N  ✓ 8 of 8 elements (1 HDKT node keys released)
      clinicalStatus                    "active"
      code.coding[0].code               "38341003"
      code.coding[0].display            "Hypertensive disorder"
      code.coding[0].system             "http://snomed.info/sct"
      code.text                         "Stage 2 hypertension"
      note                              "Poorly controlled; adherence limi
      onsetDateTime                     "2021-02-09"
      resourceType                      "Condition"

── Query C: restricted-tier compartments ──

Junior doctor              MENTAL_HEALTH/R  ✗ DENIED — CLASS_NOT_ENTITLED

Consultant (general med)   MENTAL_HEALTH/R  ✗ DENIED — CLASS_NOT_ENTITLED

Psychiatrist               MENTAL_HEALTH/R  ✓ 9 of 9 elements (2 HDKT node keys released)
      clinicalStatus                    "active"
      code.coding[0].code               "370143000"
      code.coding[0].display            "Major depressive disorder"
      code.coding[0].system             "http://snomed.info/sct"
      code.text                         "Recurrent depressive disorder"
      note                              "Passive suicidal ideation documen
      phq9                              19
      resourceType                      "Condition"
      … 1 more

Consultant                 GENOMIC/R  ✓ 6 of 6 elements (2 HDKT node keys released)
      APOE                              "e3/e4"
      CYP2C19                           "*2/*2 poor metaboliser"
      SLCO1B1                           "c.521TC intermediate"
      assay                             "PGx panel v4"
      consentScope                      "clinical-pgx-only"
      resourceType                      "MolecularSequence"
```

```
Psychiatrist               GENOMIC/R  ✗ DENIED — CLASS_NOT_ENTITLED

Node operator              DEMOGRAPHICS/N  ✗ DENIED — ROLE_CONTENT_BLIND

── Query D: patient restriction and emergency override ──
[patient records a CONSENT_DENY restriction on MENTAL_HEALTH]

Psychiatrist (post-restriction) MENTAL_HEALTH/R  ✗ DENIED — PATIENT_RESTRICTION

ED registrar (break-glass) MENTAL_HEALTH/R  ✓ 9 of 9 elements (2 HDKT node keys released)
     clinicalStatus                    "active"
     code.coding[0].code               "370143000"
     code.coding[0].display            "Major depressive disorder"
     code.coding[0].system             "http://snomed.info/sct"
     code.text                         "Recurrent depressive disorder"
     note                              "Passive suicidal ideation documen
     phq9                              19
     resourceType                      "Condition"
     … 1 more

  5. What a clerk holds is not merely withheld — it is unreachable
──────────────────────────────────────────────────────────────────────────────
  Clerk's granted HDKT nodes           : [(4, 3), (4, 4), (4, 7), (4, 8)]
  Key material handed over             : 128 bytes for 4 of 13 elements
  Can the clerk derive 'address.district'?   NO — no granted node covers that leaf

  Now assume the gateway is fully compromised and hands the clerk every
  ciphertext in the compartment. Without the covering node keys:
 9 of 9 non-granted cells remain undecryptable. Confidentiality is enforced by the key tree, not the server.

  6. Selective disclosure with integrity — redaction without loss of proof
──────────────────────────────────────────────────────────────────────────────
  Anchored elementRoot          : e397cf0911d4de79d311cacb81b545cfc935202459bab9bb…
  Merkle proof for 'birthDate'  : 4 sibling hashes (128 bytes)
  Proof verifies against anchor : True
  A forged cell verifies        : False
  ⇒ a reader given 4 of 13 elements can still prove those
    4 are exactly what the author signed, without ever seeing the other
    9. Redaction here does not destroy provenance.

  7. Hash-only notarisation of the epoch root
──────────────────────────────────────────────────────────────────────────────
  Epoch Merkle root             : ccdd8107fbd4b0508a89974e57fc5f9b112424e3db973fb5…
  Notary public root            : 1414c8db26aad76ef92de6bf655bc95ad9e3afbd03eb39a5…
  Attestation verifies          : True
  Back-dated root verifies      : False
  ⇒ security rests on SHAKE-256 alone. A break of module lattices would
    cost confidentiality going forward; it would not let anyone rewrite
    the historical record undetected.

  8. Immutable disclosure log
──────────────────────────────────────────────────────────────────────────────
  requester     compartment   v   pou  BG  elems
  ebad39492a5e  ad249ce0c946  1   4    -   6
  bc1fefc4cba3  ad249ce0c946  1   1    -   3
```

```
8458614a6691  ad249ce0c946  1   1    -   13
80f01e783411  ad249ce0c946  1   3    -   3
bc1fefc4cba3  b1e38a908283  1   1    -   2
af7d1415e08f  b1e38a908283  1   1    -   8
1be59e070cf5  3d8958c563b8  1   1    -   9
8458614a6691  18fb911beab4  1   1    -   6
fb711bb6d487  3d8958c563b8  1   2    yes 9

9 disclosures recorded. Every plaintext release in this
session is attributable to a capability, a purpose and an element set.

9. Cryptographic parameter summary
```

function	primitive	standard	level
Key encapsulation	ML-KEM-1024 + X448 (X-Wing combiner)	FIPS 203 / RFC 7748	Cat 5
Content encryption	AES-256-GCM + CTX key commitment	SP 800-38D	256-bit
Clinical signature	ML-DSA-87 + Ed448	FIPS 204 / RFC 8032	Cat 5
Notarisation	SLH-DSA-SHA2-256s (hash-only)	FIPS 205	Cat 5
Hash / KDF	SHAKE-256, SHA3-256	FIPS 202	≥128-bit PQ
Ledger digest	Keccak-256	EVM-native	≥128-bit PQ

```
Demonstration complete
```

D Glossary

Term	Meaning
Anchor	On-chain record of a compartment version: element Merkle root, ciphertext digest, storage locator, author signature digest, suite identifiers. Hash-linked to its predecessor.
Break-glass	Time-boxed emergency access that relaxes the relationship and consent predicates but never the role or clearance predicates.
Capability	A minted, metered authorisation bound to a requester, compartment, purpose, time window, query budget and element-set hash.
Cell	The AEAD-sealed value of one canonical FHIR element path. The atomic unit of disclosure.
Compartment	One (patient, data class, sensitivity tier, epoch) slice of the record; the unit of key management.
Content-blind	A role that may never receive plaintext under any purpose. R14 (auditor/DPO) and R15 (node operator).
Crypto-shredding	Erasure by destruction of the compartment root key and every wrap of it, leaving the ciphertext permanently undecryptable.
CRQC	Cryptographically relevant quantum computer.
CTX tag	An explicit key-commitment tag appended to an AEAD ciphertext, closing the partitioning-oracle class of attacks.
Epoch	A salted, rotating counter in the compartment identifier. Provides unlinkability across time and is the mechanism for retroactive element revocation.
HDKT	Hierarchical Disclosure Key Tree. A GGM binary key tree over canonical element paths; releasing interior nodes releases exactly the leaves beneath them.
Harvest-now-decrypt-later	Capturing ciphertext today for decryption once a CRQC exists. The threat that sets the at-rest parameters.
Legal hold	An on-chain marker blocking crypto-shredding, used for statutory retention and litigation.
ML-DSA-87	FIPS 204 lattice-based signature at NIST Category 5.
ML-KEM-1024	FIPS 203 lattice-based key-encapsulation mechanism at NIST Category 5.
Mosca's inequality	$X + Y > Z$: migration is overdue when secrecy lifetime plus migration time exceeds time to a CRQC.
Purpose of use	HL7 v3 ActReason code asserted with every query, bound into the capability and logged.
QBFT	Istanbul Byzantine Fault Tolerance v2; the consensus algorithm, giving immediate finality with $n \geq 3f + 1$.
SLH-DSA	FIPS 205 stateless hash-based signature. Used for notarisation because its security rests on hashes alone.

Term	Meaning
Tier	The sensitivity level of a compartment (U/N/R/V), compared against a role's clearance.
Tombstone	The on-chain record that an erasure occurred, retained after the content itself is unrecoverable.
X-Wing combiner	The hash-based construction that binds an ML-KEM shared secret to an elliptic-curve shared secret, its ciphertext and its public key, giving a hybrid KEM secure if either component is.

End of specification. Fourteen decisions await approval in §12.