Rethinking Secure Patient Discovery: Matching Federation with Fine-Grained Consent

Johannes Lohmöller*, Klaus Wehrle*, and Jan Pennekamp*†
*Communication and Distributed Systems, RWTH Aachen University
†Center for Computational Lifesciences, RWTH Aachen University
Aachen, Germany
{lohmoeller, wehrle, pennekamp}@comsys.rwth-aachen.de

***Abstract*—Electronic Health Records (EHRs) drive clinical care and medical research, yet hospitals, clinics, and research institutions routinely pool their records into central registries operated by a single party. Such centralization enlarges the attack surface, concentrates trust in the operator, and offers limited protection against misbehavior. Decentralized alternatives mitigate these risks but lack dynamic, fine-grained consent, and their multi-party protocols resist adaptation to this need. We present FeCaD, a deployable architecture that couples decentralized patient discovery through homomorphic encryption with dynamic, per-record consent enforcement. FeCaD allows researchers to submit encrypted predicates over distributed cohorts and to obtain encrypted indicator vectors of matching records without exposing sensitive attributes during processing. Distinctively, a per-query cryptographic consent token authorizes query classes against attribute subsets, replacing enrollment-time policies with cryptographic enforcement at evaluation time. We evaluate our open-source prototype across three representative cohorts, a clinical workload (MIMIC-IV), a domain-specific nuclear-medicine cohort, and a hospital-discharge benchmark (HCUP NIS), and find that consent-bound equality queries over 10 000 records complete on a single provider in 36 s. A hundred-provider federation returns within 14 min, placing biobank-scale discovery in the order of minutes.**

***Index Terms*—Patient Discovery, Electronic Health Records, Consent Management, Homomorphic Encryption, Cohort Discovery**

1. Introduction

Electronic Health Records (EHRs) consolidate most patient information, including diagnoses, lab results, treatment histories, and genomic data [1], [2]. Under proper authorization, such records enable researchers and practitioners to identify disease patterns [3], [4], recruit for clinical trials [5], [6], and develop personalized treatments [2], [7]. These use cases increasingly hinge on rare-disease cohorts whose incidence is too low to appear in an institutional dataset [4], [8]; consequently, discovery must happen cross-institutional.

In practice, however, patient data remains spread across hospitals, clinics, and research institutions, governed by data-protection regulations such as GDPR and HIPAA [9]. HIPAA breach reporting recorded over 61 Mio. affected individuals in 2025 [10], placing the privacy risk of cross-institutional access on a measurable footing and motivating stronger technical measures. Medical data thus tends to remain behind institutional firewalls and within narrow regulatory perimeters [11], [12], in conflict with patients' expectations of data sovereignty [13], i.e., their control over their own health information. De-identification has similarly proven insufficient: reviews of re-identification attacks consistently report high success rates against released records [12], [14], highlighting the need for cryptographic measures.

Prior work on cross-institutional patient discovery either centralizes data [15] or relies on complex multiparty computation [16], [17]. Centralized hubs concentrate privacy risk and form single points of failure, whereas existing decentralized solutions incur substantial overhead, lack expressive query mechanisms, or fail to support range or distance queries over attributes without exposing the underlying records [15]. Federated analytics pipelines, on the other hand, largely assume a fixed cohort instead of discovering it [1], [9].

Beyond the operator's perspective, patients' needs are equally important. Patients are more willing to share data when they retain granular control over its use [18], [19], so fine-grained consent facilitates participation. Yet current discovery approaches treat consent as a coarse, enrollment-time policy [20] decoupled from the queries that actually access a patient's records, reducing patient participation to a one-time decision [21]. This stance contradicts empirical findings that willingness to consent depends on *who* requests *which* attributes for *what* purpose [8], [18], [19] and curtails the deployability of otherwise sound discovery protocols.

To address this mismatch, we present FeCaD (**Fin**e-grained **C**onsent-**a**ware **D**iscovery), an architecture for discovering relevant patients across decentralized providers while preserving data confidentiality and patient sovereignty. FeCaD couples homomorphic-encryption-based discovery with a cryptographic consent mechanism through which patients dynamically authorize access to their records. Users (i.e., researchers and practitioners) thereby identify cohorts from encrypted match results, while patients retain dynamic and fine-grained control over which queries evaluate against their data. Specifically, we make the following main contributions:

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

►*Decentralized Discovery Architecture.* We transfer centralized FHE-backed discovery engines into a federated setting by integrating state-of-the-art encrypted-database techniques [22], [23] into a multi-provider architecture that eliminates the need for a single trusted third party of prior hubs. The resulting design supports complex predicates (Boolean, range, and distance comparisons) over encrypted records across institutions, advancing beyond aggregate-count MPC pipelines [16], [24], [25] and centralized FHE-based discovery [22], [23].

►*Cryptographic Consent Enforcement.* We design and integrate a fine-grained consent layer that combines a cryptographic per-query capability with a decentralized discovery protocol. Patients pre-authorize query classes against attribute subsets and dynamically consent to out-of-policy requests at query time, easing the long-standing trade-off between broad consent (low overhead, decreasing participation) and dynamic consent (high trust, high overhead). The layer composes established primitives yet enforces consent cryptographically at evaluation time, per record and per query: the evaluator blinds all rows, so that only a fresh patient signature allows the privacy proxy, which holds no decryption key, to cancel that blinding. No prior system in Table 1 allows a patient's signature to determine whether their record contributes to an encrypted federated query.

►*Implementation and Evaluation.* We implement a FeCaD prototype with five interchangeable cryptographic backends and evaluate it on three representative datasets (MIMIC-IV, a nuclear-medicine cohort, and HCUP NIS). Consent-bound queries over 10 000 records on a single provider complete in 36 s on our Lattigo BGV backend; scaling the federation to a hundred providers raises query completion time by a factor of 19×, placing biobank-scale discovery at 14 min and substantiating feasibility for clinical and research deployment.

Scope. We confine this paper to consented patient discovery, i.e., the cryptographic identification of patients matching a user's predicate, along with the consent material authorizing such identification. Orthogonal aspects, including downstream analytical workloads, integration with clinical information systems, and the legal harmonization of data reuse, we defer to future work.

Open Science. We release the FeCaD prototype on GitHub [26] and Zenodo [27] and bundle synthetic record generators so that reproducing the reported numbers does not require MIMIC-IV or HCUP NIS credentials.

Paper Organization. Section 2 introduces the setting, formalizes the threat model, and derives the design goals. Section 3 surveys related work, Section 4 presents FeCaD, Section 5 discusses its security, Section 6 evaluates our proof-of-concept implementation, and Section 7 examines FeCaD's impact and limitations before Section 8 concludes.

2. Scenario and Threat Model

We first describe the relevant entities (Section 2.1) and the threat model (Section 2.2), i.e., the adversaries against which any secure patient-discovery approach must protect, before deriving corresponding design goals (Section 2.3).

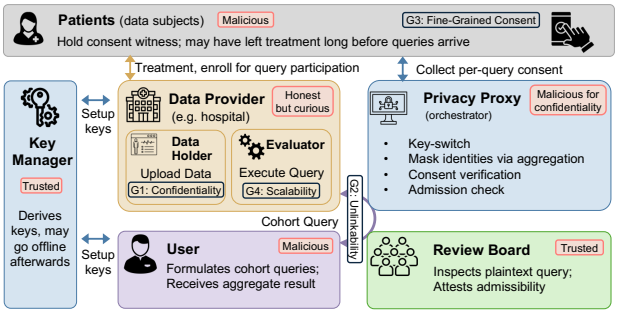


Figure 1. System model of the six entities, their roles, and their interactions; the corresponding adversaries are highlighted in red.

2.1. Involved Entities

Patient discovery returns, for a given inclusion predicate over patient attributes (e.g., demographics, diagnoses, or biomarkers), the number of records satisfying it across a federation of providers and, optionally, the row positions of the matching records, so that downstream consent requests for cohort recruitment can be targeted at the contributing providers [15], [28]. The setting comprises four entities, as shown in Fig. 1, plus an optional fifth (a review board) that screens the admission of ethically sensitive queries. *Patients* (data subjects) hold the consent material for their records and asynchronously grant or deny matching capability for incoming queries. *Data providers* (typically operated by hospitals) host the patient records and evaluate queries locally against them. *Users* (e.g., researchers or practitioners acting as data users) formulate cohort queries and submit them through a *privacy proxy* that authenticates the requester, mediates the cryptographic translation between the requester and the providers, and forwards the query without disclosing the requester's identity. To honor the information-security needs of this setting, records must remain protected at rest, in transit, and during evaluation (i.e., in use), so that plaintext is never reconstructed beyond the aggregate results delivered to the user. At the same time, each patient retains exclusive control over the consent witness that determines whether their records contribute to a given evaluation. The optional *review board* handles query admission for high-risk scenarios, e.g., through a committee decision, by inspecting the plaintext query content, attesting to its admissibility, and forwarding the query to the proxy.

2.2. Threat Model

We classify each entity by its trust level and by the information it may attempt to recover. Data providers are *honest-but-curious* [24]: they adhere to the protocol but can attempt to infer the contents of queries, the requester's identity, or the records that match a query [15]. Within the provider role, we conceptually distinguish a *data holder* that custodies its records from an *evaluator* that only executes the predicate and may be outsourced to a public cloud; whether the two coincide is a design choice discussed in Section 4 and Section B.3. The privacy proxy is *untrusted* for confidentiality,

yet must correctly orchestrate query forwarding and consent validation. Users are adversarial: they may craft queries that target individuals and collude with up to one data provider.

We argue that treating providers as honest-but-curious and users as adversarial is realistic: providers operate under HIPAA, GDPR, and corresponding auditing that penalizes protocol deviation, whereas users, once authorized, face weaker obligations on the queries they issue [12]. Patients are trusted to provide their own consent witness, but may not learn anything beyond the query they are asked to consent to. We further assume that the cryptographic primitives instantiated by any concrete design remain unbroken and that no party loses control of the secret material assigned to it. The optional review board is trusted regarding query content: it observes the plaintext query during admission and attests to its admissibility under the agreed policy. Any further trusted bootstrap role, e.g., a one-time key distributor, is assumed to act honestly during setup and to be unavailable to subsequent query flows.

2.3. Design Goals and Problem Statement

Given the setting and threat model, reasonable approaches should provide four properties: **(P1)** providers do not learn which of their records match a query, nor the query content itself; **(P2)** users do not learn which provider contributed a matching record; **(P3)** providers do not learn the identity of the issuing user; and **(P4)** no party other than the data subject can forge a consent witness, which binds to a single query descriptor (no cross-query replay). These properties rule out the trivial linkage attacks in which a single party connects a query to a provider or a record to a user.

Building on these properties, we derive four design goals that any patient-discovery system targeting our setting should meet. **G1 (Confidentiality):** records and intermediate evaluation state remain confidential against any party other than their owner. **G2 (Unlinkability):** no entity beyond the board and proxy links a query to its requester, and only the contributing provider traces an accessible result to its source. **G3 (Fine-grained consent):** patients authorize query classes against attribute subsets, enforced beyond organizational means. **G4 (Practical scalability):** discovery latency remains within hours for record volumes representative of real biobanks [29] (up to 10^7 records). Jointly, G1 and G2 motivate decentralized storage over centralized hubs, together with a cryptographic trust base; we survey this design space in Section 3.

3. Preliminaries and Related Work

FeCaD builds on three prior lines of work. Section 3.1 reviews centralized discovery hubs, decentralized health-data networks, and notification services for medical data. Section 3.2 surveys searchable and homomorphic encryption schemes underlying recent encrypted-database engines, and Section 3.3 introduces the broad and dynamic consent regimes.

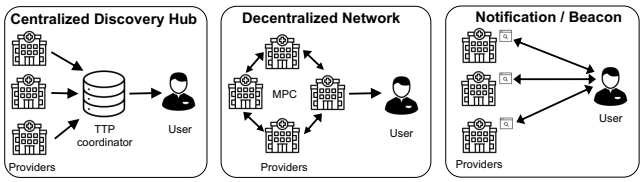


Figure 2. Abstract comparison of the concepts surveyed in this subsection.

3.1. Approaches Primed for Querying Medical Data

We focus on systems that target cross-institutional cohort identification using (semi-)structured medical attributes: organizationally governed discovery hubs, cryptographic decentralized analytics, and notification-style services. We order them from highest deployment maturity to comparatively narrow but cryptographically protected designs, reflecting developments from G1/G2 compliance through organizational means to cryptographic primitives.

Centralized Discovery Hubs. Clinical research networks such as PCORnet and TriNetX route cohort queries through a coordinating hub that returns aggregated counts to the requester [25], [30]. The architecture rests primarily on organizational governance: the hub is trusted to blind the query before fan-out, and approved users obtain identical privileges. PCORnet-style networks, therefore, lack any mechanism for cryptographic per-patient access policies and, because every query traverses a single coordinator, remain structurally vulnerable to a compromised or curious hub. Industrial federations such as TriNetX [31] and the Observational Health Data Sciences and Informatics (OHDSI) network [7] extend this pattern across hundreds of providers and serve as a baseline against which cross-institutional discovery designs compete. These hubs concentrate privacy risk and constitute single points of failure, while existing decentralized alternatives add significant overhead or lack expressive query mechanisms; to the best of our knowledge, no secure and practical mechanism supports range or distance queries over such medical attributes without exposing records [15].

Decentralized Health Data Networks. Throughout this paper, we reserve *federated* for systems in which participants cooperate under a central coordinator (typically retaining a single trust anchor), and *decentralized* for systems whose providers cooperate without one. Federated analytics enables data analysis across providers without exchanging patient-level records and emerged in response to legal barriers to centralization [16], [35]. Cryptographic instantiations such as UnLynx [24], MedCo [33], and FAMHE [16] combine secure multiparty computation with verifiable shuffles to hide both inputs and access patterns. Their security rests on an *anytrust* assumption [24], i.e., at least one party from a fixed set of computing parties must remain honest, thereby anchoring trust in the organizational coordination among providers. While the design principles for such networks emphasize data locality and harmonization over centralization [36], [37], the cryptographic enforcement of these principles remains non-standardized [38]. Personal Health Train architectures ship analytical pipelines to provider data [1], [9], yet, as with

TABLE 1. COMPARISON OF FECAD WITH EXISTING APPROACHES THAT, HOWEVER, FAIL TO CONSIDER ANY FORM OF DYNAMIC CONSENT.

System	Arch.	Trust Model	Query Capability	Consent	Evaluation Scale	G1	G2	G3	G4
Discovery hubs and clinical-research federations (organizational governance)					G1: Confidentiality G2: Unlinkability G3: Fine-grained consent G4: Scalability				
PCORnet [30]	Fed.	TTP hub	aggregate counts (CDM)	per-site IRB	11 CDRNs, goal $\geq$1M pat. each (2014)	○	○	○	●
TriNetX [31]	Fed.	vendor TTP	Boolean, aggregate statistics	per-site IRB	200+ HCOs, ∼250M pat. (as of 2025)	○	○	○	●
OHDSI [7]	Fed.	coordinating center	SQL on OMOP CDM (local)	per-site IRB	∼800M pat. across hundreds of DBs	○	○	○	●
Beacon [32]	Fed.	per-site data controller	Boolean membership	none	per-beacon (variant presence)	◐	○	○	●
Distributed cryptographic analytics (anytrust, MPC/MHE)									
UnLynx [24]	Dec.	anytrust (n srv.)	encr. Boolean, aggregate	none*	400k rec., 20 DBs (24 min)	●	◐	○	◐
MedCo [33]	Dec.	anytrust + DP	cohort sel. + DP counts	none*	8k pat. (∼1M obs.), 3 SPUs	●	◐	○	◐
FAMHE [16]	Dec.	anytrust MHE	survival, GWAS aggregates	none*	1662 pat., 96 DPs	●	◐	○	◐
Centralized FHE-backed databases and discovery									
HE3DB [22]	Cent.	TTP (DB admin)	SQL (arith.+logic FHE)	none	10k rows TPC-H	●	○	○	◐
ArcEDB [34]	Cent.	TTP (DB admin)	SQL (amortized modular HE)	none	10–32k rows	●	○	○	◐
Engorgio [23]	Cent.	TTP (DB admin)	SQL (quantized FHE)	none	2–16k rows TPC-H, SIFT1M vectors	●	○	○	●
PatDiscover [15]	Cent.	HbC DB server	Bool, enum, range, distance	none	500k pat. (single host)	●	◐	○	●
FeCaD (Ours)	**Dec.**	**trusted authority**	**Bool, enum, range, dist.**	**fine-grained dynamic**	10^4–10^7 **pat., 1–100 DPs, 3 datasets**	●	●	●	●

Symbols: ● supported, ◐ partial or restricted, ○ absent. The *Consent* column refers to built-in technical enforcement.
"none" systems may still operate under external IRB/ethics governance. * admission handled by collective authority.

the cryptographic networks above, the cohort is presupposed. Neither family provides the discovery primitive that FeCaD concentrates on.

Notification Mechanisms. Primarily targeting genomic discovery, Beacon-style services federate web endpoints that answer presence queries over predicates with coarse binary responses and rely largely on organizational governance [39]. Repeated probing has nevertheless been shown to enable membership inference and allele re-identification even with such coarse outputs [32], [40]. Implemented protocols provide no cryptographic per-patient consent binding in the sense of G3. Beacon-style notifications, therefore, cover a complementary slice of the design space and do not extend to the attribute-rich, consent-bound discovery primitive that FeCaD focuses on.

3.2. Cryptographic Database Techniques

We now turn from concrete deployment approaches to the cryptographic primitives that enable secure query evaluation and revisit relevant systems built on them. Searchable encryption (SE) and public-key encryption with keyword search (PEKS) let a server locate ciphertexts matching a token without learning the underlying records [28], [41], [42]. While efficient for exact-match lookups, these primitives do not support range, distance, or aggregate queries, which limits their applicability to cohort discovery on numeric attributes [15]. Homomorphic encryption (HE), by contrast, supports arbitrary arithmetic over ciphertexts [43], [44], [45] at the cost of higher per-operation latency, and the available schemes differ widely in overhead, maintainable precision, and supported operations [46].

Recent generic fully encrypted database engines indicate that HE-backed query evaluation becomes feasible even at larger scales. HE3DB [22] compiles SQL operators into TFHE circuits, ArcEDB [34] implements modular operators on top of CKKS, and Engorgio [23] applies quantization to reduce ciphertext expansion. All three target a centralized setting and address neither distributed providers nor per-record consent. To the best of our knowledge, PatDiscover [15] is the only prior system that applies such database techniques to patient discovery, albeit in a centralized setting.

Beyond ciphertext indistinguishability, all of the above engines remain subject to access- and search-pattern leakage [47], [48], [49], [50], a side channel that any encrypted database architecture only partially constrains. We revisit this channel and its implications in Sections 5.2 and 7.

3.3. Giving Consent

Informed consent constitutes a central principle of biomedical research, introduced in the Belmont Report [51] and later refined for population-scale studies [52]. Empirical studies report that patients' willingness to share clinical data depends less on the data itself than on *the recipient*, *the purpose*, and *the revisability of the decision* [8], [18], [19], [53].

Two consent models dominate today, occupying opposite ends of a trade-off. Under *broad consent*, a patient authorizes a broad class of future research uses at enrollment [20], [54], minimizing friction yet potentially reducing participation when patients learn retrospectively of secondary uses they did not anticipate. Under *dynamic consent*, by contrast, each request triggers a patient interaction [21], [55], minimizing the consented scope at the cost of high coordination overhead and a risk of non-response that can stall time-sensitive research. FeCaD resolves this trade-off: our cryptographic consent token decouples policy expression from per-query interaction, so that a patient pre-authorizes a class of queries (recovering broad-consent throughput) while retaining dynamic-consent power over out-of-policy requests.

Organizational Efforts. National initiatives such as the German Medical Informatics Initiative (MII) and the US PCORnet, together with biobank consortia, have established the legal and organizational scaffold for cross-institutional data reuse [9], [29], [30]. These efforts rely on standardized broad-consent forms to render data *discoverable* yet provide little technical control once access has been granted [54], and broad consent in particular cannot enforce query-time constraints on the recipient or the purpose.

Research Gap. Table 1 contrasts the families surveyed above against G1–G4: (1) organizational hubs scale well but provide no cryptographic confidentiality or per-patient consent, (2) cryptographic federations approach G1/G2 but inherit anytrust assumptions and presuppose the cohort, and (3) FHE-backed databases satisfy confidentiality yet remain centralized and without consent management. No prior system satisfies G1–G4 jointly, which is the gap FeCaD targets.

4. FeCaD System Design

FeCaD integrates and decentralizes recent FHE-based encrypted databases such as HE3DB [22] and Engorgio [23], complementing them with an orchestration layer that mediates key handling, query admission, cryptographic consent, and result aggregation. For each query, the user receives an encrypted indicator vector whose set positions the privacy proxy, but not the user, can resolve to the contributing providers. The user learns only that such positions exist, which supports targeted follow-up consent requests for cohort recruitment without disclosing aggregate-level provider participation.

Within each provider, we conceptually distinguish the data-holder role (custodian of sk_h and the encrypted records) from the evaluator role (executor of homomorphic predicates). Both may be co-located or separated to simplify operational needs or obtain stronger guarantees against a malicious provider, as discussed in Sections B.3 and 7.2.

4.1. Architectural Overview

The architecture comprises four functional entities (users, data providers, a privacy proxy with a key manager, and patients) organized into three layers (data, orchestration, and consent) that interact according to the discovery protocol shown in Fig. 3. Queries progress independently: distinct queries, whether issued by the same user or different users, do not share per-query state at the providers or the proxy beyond the key-switching keys (KSKs) and blinding material established at setup, so multiple queries proceed concurrently without cryptographic coupling.

User. The *user* (e.g., a researcher or practitioner) encrypts the query predicate under its FHE key sk_u, signs the submission with a BLS12-381 key, and consumes the aggregated encrypted indicator vector returned by the proxy. Users are external clients without access to provider, patient, or proxy secret material and need not remain online between submission and final decryption.

Data Layer (data providers). Records, e.g., structured EHRs or derivatives, reside at the institutions (usually hospitals) that hold them and never leave their domain in plaintext. Each *data provider* encrypts its records under a provider-local FHE key sk_h and evaluates incoming predicates homomorphically against this ciphertext store, thereby satisfying G1 (Confidentiality) at rest and in use and confining the trust base to the entity that already holds the data. Per-provider sampling of sk_h additionally prevents

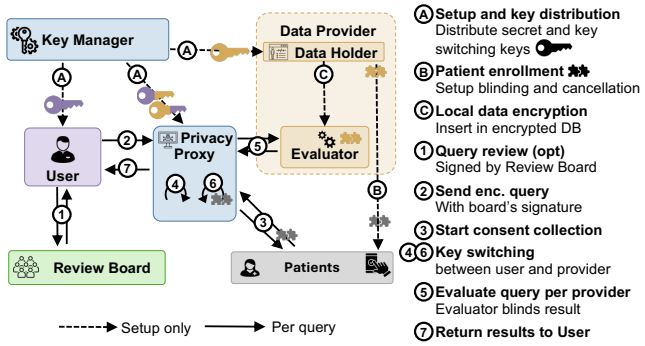


Figure 3. Protocol flow sequence of the FeCaD discovery. At query time, four entities (user, privacy proxy, evaluator, patient) cooperate via the seven query-time steps ①–⑦ detailed in Section 4.2. Section B states the protocol.

colluding providers from sharing a decryption capability across institutions.

Orchestration layer (privacy proxy, key manager). The *key manager* runs only at setup: it derives the per-provider key-switching key pair $(\mathrm{ksk}_{u\to h}, \mathrm{ksk}_{h\to u})$ from the user and provider secret keys made temporarily available to it, distributes each sk to its owning entity, and can operate air-gapped under a trustworthy operator, or be deleted once the set of users and providers is fixed. At runtime, a conceptually centralized *privacy proxy* mediates between users and data providers without holding any FHE secret key, storing the two key-switching keys per provider, translating the user's encrypted literals into the provider's evaluation domain, and translating aggregated results back afterward (cf. Section 4.2). We aggregate partial results at the proxy before returning them to the user, satisfying G2 (Unlinkability), while leaving query evaluation itself to the provider so that the proxy never observes plaintext predicates or records. The same indirection also separates the user's identity from the provider's view. If scalability becomes a concern, the proxy may run as multiple parallel instances, either horizontally or vertically scaled, in which case users additionally learn the cardinality of providers associated with a given privacy proxy.

Consent Layer (patients, privacy proxy). *Patients* hold the blinding scalar and the BLS12-381 signing key established at enrollment and may, upon query, release a freshly signed consent envelope that authorizes the corresponding cancellation (Section 4.3). Without such an envelope, the proxy cannot strip the provider-side blinding, so the user learns nothing about non-consenting records, which realizes G3 (Fine-grained consent) as a cryptographic capability.

Figure 3 situates the four entities along the interaction that constitutes the layers above; we specify the protocol in Section B and provide some intuition in the following.

4.2. FeCaD's Secure Patient Discovery Protocol

The protocol enables a user to submit an encrypted predicate, e.g., $\mathrm{Age} > 50 \wedge \mathrm{Diagnosis} = \mathrm{Diabetes}$, and to receive an encrypted indicator vector whose set positions encode the matching records without revealing the predicate, the record contents, or the per-provider decomposition of

matches to any other party [15], [28]. The privacy proxy may retain the position-to-provider mapping required for downstream cohort follow-up (which is beyond the scope of this paper). We instantiate four query classes that cover the predicate patterns common in cohort discovery (cf. Section 2): Boolean predicates, exact and approximate enumerations, continuous ranges, and distance metrics on numerical attributes. Following the step numbering of Fig. 3, an optional review board ① admits the query, the user ② submits the predicate, encrypted under its public key pk_u; patients ③ release per-query consent envelopes by polling; the privacy proxy ④ applies $\mathrm{ksk}_{u\to h}$ to align each literal with the addressed provider's evaluation key; evaluators ⑤ evaluate the predicate locally against their blinded ciphertext store; the proxy ⑥ aggregates the encrypted partial results, applies the per-query consent cancellation and $\mathrm{ksk}_{h\to u}$ to return the indicator vector under sk_u; and the user ⑦ decrypts.

Key Switching. FeCaD relies on the key-switching primitive provided by each underlying scheme to decouple the user and data-provider domains without exposing either secret key to the proxy: the proxy holds the two switching keys $\mathrm{ksk}_{u\to h} = \mathrm{KSKGen}(sk_u, sk_h)$ and $\mathrm{ksk}_{h\to u} = \mathrm{KSKGen}(sk_h, sk_u)$ derived at setup (cf. Section B for a formal definition under RLWE), but must not hold sk_u or sk_h directly. Backends instantiate this abstract proxy interface with their scheme-native construction: RLWE-style KSKGen [56] for BGV (Lattigo) and CKKS (Engorgio), BFV proxy re-encryption (PatDiscover), and TLWE identity key switching for TFHE (HE3DB) (cf. Section 6.1). When multiple providers participate, sk_h is sampled independently per provider and the proxy stores one switching-key pair per provider, preventing colluding providers from decrypting one another's switched literals. We omit user and provider indices for readability.

4.3. Cryptographic Consent Enforcement

FeCaD realizes dynamic consent as a cryptographic capability that enables the unblinding of per-record indicators on a fresh patient signature, allowing queries to be evaluated (but not their results to be accessed) when only a subset of patients consents. At enrollment, the patient samples a single blinding scalar r_p that remains fixed for the lifetime of the consent relation. The provider stores $\mathrm{Enc}(+r_p)$ over the patient's rows and applies it on every record after evaluation, while the proxy holds the matching cancellation material $\mathrm{Enc}(-r_p)$, but applies it only to records of consenting patients (cf. Section B.3 for the underlying arithmetic). The scalar r_p never appears in plaintext outside the patient, so the per-query token only authorizes the cancellation, but does not carry r_p. Slots whose patient withholds consent are replaced by a public zero before key switching, i.e., they decrypt to exactly the value of a non-match and thus are indistinguishable.

Polling-based Token Release. FeCaD adopts a polling model in which the proxy publishes a public query descriptor $qd = H(Q, \nu, D)$ on a consent endpoint, comprising a cryptographic hash of the query, a non-expert-accessible description of its contents to be shown to users, and a per-query nonce. If the review board is involved, the description D undergoes expert review and is signed accordingly. Each patient (or a delegated policy agent) periodically retrieves pending descriptors, evaluates a local policy predicate π_p or queries the patient interactively, and, on a match, releases a BLS12-381 signature $\sigma_p \leftarrow \mathrm{BLS.Sign}(sk_p^{\mathrm{BLS}}, H(qd \,\|\, ct_{C,p}))$ that binds the consent to the particular cancellation ciphertext, which the proxy admits within a bounded consent window. The proxy combines the per-patient signatures into a single aggregate σ_{qd}, so the consent transcript is constant in size regardless of how many patients consent. As each envelope binds a distinct $ct_{C,p}$, verification itself is one pairing-product check over the n consenting public keys. Non-matching queries yield no response, so polling proceeds in parallel with homomorphic evaluation at the providers. The descriptor commits to the query without revealing the predicate template, the encrypted literals, or the requester identity (cf. Section 5.2).

Per-query Envelope. Each patient jointly signs the descriptor and the cancellation ciphertext, $\sigma_p \leftarrow \mathrm{BLS.Sign}(sk_p^{\mathrm{BLS}}, H(qd \,\|\, ct_{C,p}))$, which pins the cancellation to both qd and $ct_{C,p}$. The proxy, therefore, cannot recombine the cancellation ciphertext of one query with the signature of another (no cross-query replay). By design, $ct_{C,p}$ is constructed by the patient at query time under the provider's public key pk_h, so the patient never holds any FHE secret-key material, and a smartphone-class client with a thin encrypt-only library suffices. Our current implementation reuses an enrollment-time cancellation ciphertext to keep the patient client minimal. The envelope binding (signature over $H(qd \,\|\, ct_{C,p})$) prevents replay, while shifting $ct_{C,p}$ generation to query time (closing the stockpiled-material exposure of Section B.3) remains future work. Logistically, per-query construction is feasible whenever the per-patient ciphertext is small (kilobytes to a few megabytes), which holds for BGV, BFV, and TFHE. CKKS parameter sets, by contrast, inflate the payload beyond the capacities we consider common for constrained patient devices (cf. Section 6.1).

Takeaway. FeCaD separates evaluation from consent coordination: the provider blinds every per-record indicator with material it cannot itself reverse, and the proxy applies cancellation only against freshly authorized records.

5. Security Discussion

We establish that FeCaD satisfies the four confidentiality properties of Section 2. Section 5.1 first derives P1 to P3 from the semi-honest security of the underlying FHE engines and the key-switching architecture, Section 5.2 characterizes the residual query leakage as an explicit access-pattern function, and Section 5.3 analyzes the implications of collusion under our threat model.

The four properties P1 to P4 build on different assumptions. P1 reduces to the IND-CPA security of the underlying engine against the proxy and any network observer. Against a curious provider that holds sk_h and could therefore decrypt the switched literals, it resides on the honest-but-curious restraint of Section 5.2. The consent layer weakens neither, since it only adds ciphertexts under pk_h, but does not decrypt

anything. P3 follows from the bidirectional key switching of Section 5.1, which leaves no user-identifying material in the ciphertext that a provider sees. In contrast, P2 follows from the proxy-side aggregation of Section 4.2, whose aggregate view is an equality of distributions once the per-provider results are summed. P4 reduces to the EUF-CMA security of BLS12-381 [57], since a forged or replayed witness is a signature on a descriptor the patient never signed. Likewise, consent hiding is an equality of distributions, because Filter_C replaces non-consenting slots with a public zero.

5.1. Data Confidentiality

P1 reduces directly to the semi-honest security proofs of the underlying FHE engines: HE3DB [22], Engorgio [23], the BFV-based PatDiscover [15], and the BGV-based Lattigo backend used in our prototype, all of which instantiate IND-CPA-secure schemes whose ciphertext indistinguishability reduces to the hardness of (Ring-)LWE. The server's view during evaluation consists exclusively of valid FHE ciphertexts, with no intermediate decryption, matching our honest-but-curious provider assumption. Thus, a curious provider cannot distinguish records of distinct patients with non-negligible advantage, even under arbitrary predicates issued by a malicious user. The per-patient blinding r_p never appears in plaintext, and both $\mathrm{Enc}(+r_p)$ at the provider and $\mathrm{Enc}(-r_p)$ at the proxy remain RLWE ciphertexts under the same hardness assumption. Hence, the consent layer adds no new distinguisher to a curious provider's view.

Regarding P2, providers return encrypted partial results under their own secret keys. After key switching at the provider (which already identifies providers via network metadata and key identity), these partial results carry no cryptographic metadata identifying the contributing institution, so the user's decryption of the aggregate reveals the combined value but not the per-provider decomposition. This guarantee would degrade if the user received unaggregated per-provider results, because the presence or absence of a non-trivial partial result discloses whether a given provider held any matching records, making proxy-side aggregation prior to result delivery a prerequisite (cf. Section 4.2). Following [15], the downstream cohort-recruitment step (translating the indicator vector into actionable patient contact) intentionally relaxes P2 at the proxy and is out of scope here.

For P3, the bidirectional key switching of Section 4.2 ($\mathrm{ksk}_{u\to h}$ on the literals, $\mathrm{ksk}_{h\to u}$ on the aggregate) is the relevant primitive. This transformation avoids embeding user-identifying material in the output and does not require the proxy to hold a decryption key [15]. Since the resulting ciphertext's distribution depends only on the provider's evaluation key and the Ring-LWE noise, providers cannot distinguish queries originating from different users, and a compromised proxy gains no plaintext beyond what the encrypted query already communicates. We discuss the residual exposure under a compromised provider in Section 5.3.

5.2. Query Privacy

Queries split into a plaintext *predicate template* (column references, comparison operators, and logical structure) and FHE-encrypted *literals* encoding the comparison values. Literals are encrypted under a user-aligned key, such that the proxy applies $\mathrm{ksk}_{u\to h}$ to transform them into ciphertexts under the provider's evaluation key without ever exposing their plaintext to the proxy or the provider. Against the proxy and any network observer, the resulting ciphertext inherits the IND-CPA guarantee of the underlying FHE scheme [22], [23], so neither party recovers any literal value with non-negligible advantage. The argument against a curious provider is more subtle: once $\mathrm{ksk}_{u\to h}$ has switched the literal ciphertexts into the provider's evaluation domain, they decrypt under sk_h, which the provider holds. The honest-but-curious assumption forbids the provider from invoking its decryption oracle on the forwarded literals. To extend literal secrecy to a malicious provider, the entity holding sk_h must be separated from the entity that evaluates the homomorphic predicate: a *hospital* role owns sk_h, encrypts the records, and ships the ciphertexts together with an evaluation key to the *provider*, which holds no decryption key and therefore cannot extract literal values regardless of behavior. Hiding the predicate template from a curious proxy or provider lies outside FeCaD's scope.

Access patterns are likewise leaked to the provider, i.e., the set of record indices touched during evaluation, which reconstruction [47] and volume-leakage attacks [48], [49] exploit to recover fractions of plaintext given sufficient query volume. FeCaD limits this leakage because backend evaluation is exhaustive, with each record being evaluated against each predicate regardless of a match, i.e., the access pattern degenerates to the full database index set, revealing only the database cardinality $|\mathrm{DB}|$ [22], [23], which providers already know as sole holders of the plaintext records. Exhaustive evaluation also prevents related attacks, e.g., search-pattern leakage [50], which would be exploitable when repeated queries on the same dataset reveal records accessed repeatedly. Timing side channels that correlate evaluation duration with predicate complexity or dataset selectivity remain outside scope. The consent protocol also communicates the query descriptor $qd = H(Q, \nu, D)$ of Section 4.3 to patients. Since H is a cryptographic hash and ν a per-query nonce, qd commits to the query without revealing the predicate template, the encrypted literals, or the requester identity. We discuss these limitations in Section 7.2.

5.3. Entity Collusion

Up to five entities may participate in FeCaD (user, proxy, providers, patients, and an optional review board); we discuss their coalitions and the implications for data confidentiality, omitting any patient colluding against their own consent (which invalidates the decision). A *user with one provider* correlates matches with that provider's plaintext records, yet other providers' contributions remain encrypted under the user's key and aggregated without provider tags, preserving P2 against the rest of the network. The *review board with one*

provider reproduces the same deanonymization by leaking the plaintext predicate observed during admission, so P1 and P3 no longer hold against that provider, and no further. The board's view of any single query is bounded by the predicate and its ciphertext. A *proxy colluding with one provider* hands the switched literal ciphertexts to the sk_h-holder and recovers the plaintext literals on any honest-but-curious deviation, although P4 still prevents aggregation of non-consenting records, and the hospital/evaluator role split (cf. Sections B.3 and 5.1) prohibits the literal leakage. *User with proxy* collusion permits arbitrary query submission, but P4 continues to bind record contribution to a fresh patient signature, reducing leakage to the access pattern of Section 5.2. The *review board with the user* can admit arbitrary patient-tracing queries; introducing a k-of-n threshold reviewer (cf. Section 7.2) would restore quorum-style resistance in this case. *Multiple colluding providers* gain no more than the union of their individual views, since the per-provider sk_h (cf. Section 4.2) restricts a provider to its own switched literals. Evaluators split from the data holder may collude freely without learning any confidential information. *Review board with proxy* yields no plaintext information beyond the board's admission view, bounded by its post-review erasure of (Q, r) (cf. Section B). Outside the threat model, a *malicious provider* that holds both hospital and evaluator roles (the prototype default) can pre-cancel records of non-consenting patients by homomorphically negating its stored blinding ciphertext.

Takeaway. No admitted coalition breaks P1 to P4 beyond records already accessible to its colluding members; the remaining coalitions lie explicitly outside our realistic threat model and motivate the hardening directions of Section 7.2.

6. Evaluation

We instantiate FeCaD as a Go prototype with five interchangeable cryptographic backends and measure query completion time, microbenchmarks, and consent-binding cost across three medical cohorts. Section 6.1 introduces the implementation, Section 6.2 describes the experimental setup, Section 6.3 reports query completion time and per-phase microbenchmarks, and Section 6.4 examines scaling along records per data provider, federation size, and dataset diversity.

6.1. Implementation

We implement FeCaD as a Go coordination layer with six REST services, one per protocol entity, backed by a pluggable cryptographic backend interface that abstracts key generation, key switching, predicate compilation, and homomorphic evaluation. The user holds the sole researcher key sk_R, each data provider holds a locally sampled sk_P, the privacy proxy holds no secret-key material, and each patient holds a per-patient blinding scalar r_p together with a BLS12-381 signing key. This key layout requires the privacy proxy as a confidentiality boundary for P2 and P3: the proxy observes encrypted literals, encrypted partial results, and aggregate BLS signatures, but never plaintext records, queries, or per-patient secret material (cf. Sections 5.1 and 5.2). The consent layer uses a BLS12-381 aggregate signature, such that consenting patients sign the per-query envelope $H(qd \parallel ct_{C,p})$ once, the privacy proxy folds the per-patient signatures into a single σ_{multi}, and a single pairing-product verification gates the per-query unblinding for the entire cohort (cf. Section 4.3).

Backends. We instantiate the backend interface with four FHE schemes covering the design space of slot-packed RLWE arithmetic, gate-level TLWE, and approximate CKKS, plus a plaintext baseline that relies on the same coordination layer without homomorphic operations. The Lattigo BGV backend is our native-Go implementation tailored to FeCaD's predicate workload: BGV with $\mathrm{LogN} = 15$ and ring dimension $N = 32768$ (N slots per ciphertext), plaintext modulus $t = 65537$, and a modulus chain of $\mathrm{LogQ} = 50 + 24 \times 30$ together with $\mathrm{LogP} = 2 \times 55$ bits, yielding $\mathrm{LogQP} = 880$ bits at 24 multiplicative levels, equivalent to 128 bit classical security per the HE standard [58]. We evaluate comparisons as Lagrange indicator polynomials over the target set, choosing the smaller of the target set and its complement as the basis and returning $1 - P(ct)$ when the complement was used, thereby bounding the polynomial degree by the size of the smaller set. For equality at $t = 65537$, we apply the Fermat-exponentiation fast path $\mathbb{1}[x = a] = 1 - (x - a)^{t-1}$. The remaining three FHE backends wrap prior work: HE3DB [22] via its TFHE C++ core, Engorgio [23] via an OpenFHE worker at ring dimension 2^{18}, and a reimplementation of the PatDiscover construction [15] that retains PatDiscover's BFV slot packing and proxy re-encryption primitive but substitutes the same Lagrange indicator polynomial used on Lattigo BGV. We omit ArcEDB because Engorgio supersedes its operator coverage [23]. Range bounds traverse the privacy proxy in the clear, since the Paterson–Stockmeyer comparison expansion requires the bound at evaluation time (cf. Section 5.2).

Giving Consent. To isolate cryptographic latency from human latency, the prototype simulates immediate consent: each addressed patient signs the per-query envelope as soon as the privacy proxy advertises it, so the measured completion latency captures the cryptographic pipeline but omits the human response time, which will likely dominate in practice. The polling endpoint of Section 4.3 fits an insurance-issued mobile application that holds the per-patient signing key and the blinding scalar r_p, provisioned at enrollment, and retrieves pending query descriptors on a configurable schedule or via mobile push notifications.

6.2. Experimental Setup

All measurements run on a single node with two Intel Xeon Platinum 8160 processors, 192 GB RAM, and local SSD storage, with the six FeCaD entities co-located as separate services on distinct loopback endpoints. We simulate database sizes via sampling from the medical datasets listed in Table 2, and each data point reports the median of 30 independent runs, with the interquartile range as error bars.

Query Suite. We exercise the four query classes of Section 4.2 through a panel of representative queries derived from the medical literature and the PatDiscover query

TABLE 2. DATASETS USED IN SECTION 6. RECORD COUNTS REFER TO THE SOURCE CORPUS THAT WE SAMPLE.

Dataset	Records	Attr. (num/bool)	Predicate scope
MIMIC-IV [59]	365k pat.	8 (3/5)	ICU cohort; lab values, age, and ICD-10
Nuclear-med. [15]	500k pat. (synth)	19 (14/5)	Diagnosis, tumor info, treatment history, age
HCUP NIS [60]	$\sim 7.1 \times 10^6$ disch.	7 (1/6)	ICD-10 diagnoses, demographics

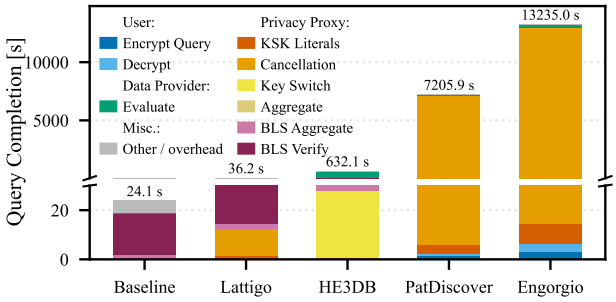


Figure 4. Query completion time at the user for the four FHE backends and the plaintext baseline on a Boolean equality query (q_1) over 10 000 MIMIC-IV records at one data provider, decomposed by per-query phases of Fig. 3.

set: a single-attribute equality predicate (e.g., *Diagnosis* = *Diabetes*), an attribute-range predicate (e.g., Age $\in [50, 80]$), a conjunctive Boolean predicate combining an equality and a range, and a distance predicate over a continuous attribute (e.g., a tumor position within $\pm\Delta$ of a reference point). Queries are represented as SQL strings, as shown in Section D.

6.3. Performance

FeCaD's query completion time decomposes into a one-time cryptographic setup, an amortizable data-ingestion phase at each data provider, and a recurring per-query critical path on which the cryptographic backend dominates. We focus the main text on the recurring per-query workload, since user-perceived cost is query completion time, and defer the one-time setup cascade to Section C.

Backend Comparison. Figure 4 compares the five backends on a Boolean equality query over 10 000 MIMIC-IV records at a single data provider, decomposed into the per-phase contributions of Fig. 3. The Lattigo BGV backend completes the reference query in 36 s, within a factor of 1.5× of the plaintext baseline at 24 s. HE3DB [22] follows at 632 s, the PatDiscover-style BFV backend [15] at 7206 s, and the Engorgio CKKS backend trails at 13 235 s, yielding a spread of 366× across the four FHE backends. When most enrolled patient consent, the per-patient cancellation ciphertext governs the completion time: HE3DB carries the smallest, whereas the BFV (PatDiscover) and CKKS (Engorgio) backends inflate it until consent enforcement is the dominant cost.

Microbenchmarks. Provider-side homomorphic evaluation accounts for the bulk of query completion time only for HE3DB (92 %). It is 6 % on Lattigo, 1 % on PatDiscover, and 2 % on Engorgio, where the per-query consent cancellation

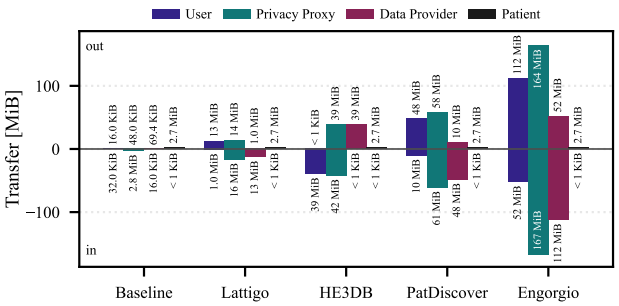


Figure 5. Per-entity outbound and inbound network transfer per query of the five backends on q_1 over 10 000 MIMIC-IV records at one data provider. The patient bar sums the consent envelopes of all 10 000 patients.

dominates instead. For Lattigo, the remaining time is mostly spent on the proxy-side. Figure 5 reports the corresponding per-query network transfer, dominated by encrypted literals and the per-record result, while the consent envelopes of all 10 000 patients account for less than 3 MiB. One-time costs are larger: per-provider switching-key material ranges from 176 MiB on Lattigo to 444 MiB on Engorgio (PatDiscover at 186 MiB), and the encrypted record store per provider ranges from 100 MiB to 896 MiB in the same ordering.

Consent Binding. By design of the exhaustive evaluation, homomorphic evaluation at the data provider is invariant to the consenting share of patients (cf. Section B). However, the consent fraction still influences the proxy-side cancellation and the verification of the aggregate consent envelope, both linear in the number of consenting patients. For a single patient granting consent at query time, the on-device cryptographic work reduces to one BLS12-381 signature of 48 B over the per-query envelope $H(qd \,\|\, ct_{C,p})$, and the empirical network payload to the privacy proxy is 140 B (≈280 B with HTTP framing), well within the capabilities of a smartphone-class client. Per query, the patient ships only 140 B to the proxy, which may optionally cache $ct_{C,p}$ at enrollment and reuse it across subsequent queries to spare bandwidth (cf. Section 4.3). The consent panel of Fig. 6 visualizes this relation: completion time grows from 13 s at 1 % participation to 43 s at full participation, where cancellation and envelope verification together account for 64 % of the time. Per-query consent is therefore the dominant cost of G3 at full participation.

Takeaway. FeCaD on Lattigo BGV completes the reference query in 36 s on 10 000 records at one provider, supports per-patient consent at 0.53 ms per decision, and runs 17.5× faster than the next-fastest FHE backend.

6.4. Scalability

We examine FeCaD's scalability along the per-provider record count, the federation size, and dataset diversity, with records and federation size contributing independently to query completion time. We rely on the Lattigo backend (lowest latency in Section 6.3) and report the remaining backends only if they govern asymptotic behavior.

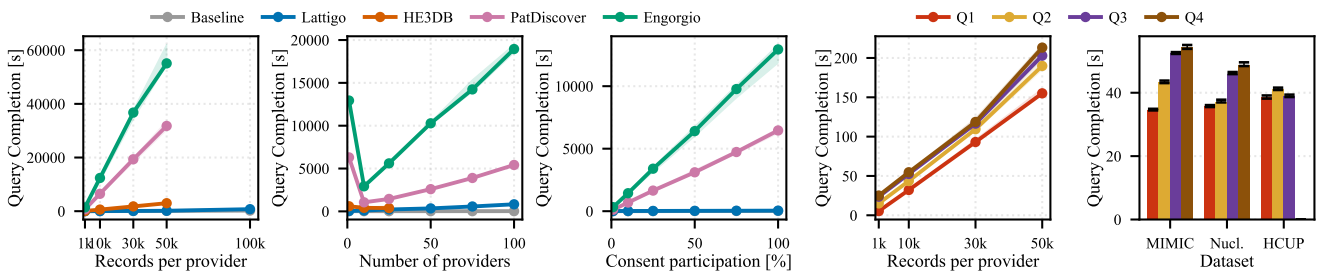


Figure 6. Scaling along records per provider (left), network size (center-left), consent participation (center), records per provider isolated to Lattigo across the four query classes (center-right), and different datasets (right) (cf. Section D).

Records per Provider. Varying the per-provider record count from 1000 to 100 000 on Lattigo yields completion times between 5.3 s and 751 s, growing near-linearly in the record count (log-log exponent 0.94). Growth concentrates in the consent layer at the privacy proxy, where per-patient cancellation and aggregate-envelope verification are linear in the enrolled cohort (cf. Section 6.3): at 100 000 records, cancellation accounts for 529 s and envelope verification for 171 s of the total. Provider-side evaluation remains at 8.6 s at the same scale, since the $N = 32\,768$-slot BGV ciphertext amortizes the indicator polynomial across a slot block. Cancellation grows disproportionately between 50 000 and 100 000 records (52 s to 529 s), where the cohort first spans four slot blocks instead of two. Engorgio and PatDiscover scale near-linearly for the same reason, from 1396 s and 708 s at 1000 records to 55 093 s and 31 771 s at the upper end of 50k records. HE3DB scales near-linearly from 64 s to 2975 s (slope 59.4 ms/record, $R^2 = 1.00$), as gate-level TFHE comparison evaluates each row individually while its TLWE cancellation ciphertexts stay small.

Number of Providers. Growing a federation from 1 to 100 data providers over 10 000 records partitioned equally across providers on Lattigo raises completion latency from 44 s to 825 s. Growth concentrates in the proxy's wait time for per-provider responses, which rises from 12 s at one provider to 544 s at one hundred, whereas proxy-side aggregation remains at ca. 0.36 s regardless of federation size. Since our experimental setup co-locates all simulated providers on one node, they share CPU at the homomorphic-evaluation stage and the observed near-linear growth may only reflect single-node CPU contention. The plaintext baseline grows sub-linearly from 22 s to 25 s (log-log exponent 0.020, $R^2 = 0.91$), confirming that federation-coordination overhead alone is sub-linear and isolating the cryptographic component as the dominant growth driver. Aggregate outbound traffic at the privacy proxy reaches 100 MiB on Lattigo and 1000 MiB on PatDiscover at 100 providers.

Deployment Variants. The measurements above assumed a single merged data-holder/evaluator entity, co-located providers on one node, and no communication latency or bottlenecks. We now revisit each of these three simplifications. Splitting the data holder from the evaluator leaves query completion time unchanged at 36 s (36 s merged): the one-time store transfer of step © transfers 290 MiB in 0.26 s, and evaluation alone slows down by 13 % (2.1 s to 2.4 s). Distributing the hundred providers over separate machines, six per host, reduces completion time from 825 s to 325 s (5.4 min), a factor of 2.5×, with 96 s at 25 and 155 s at 50 providers; the federation curve of Fig. 6 is therefore subject to contention among co-located providers. Emulating wide-area links between the entities with `netem` raises completion time at one provider by 1.02× at 20 ms RTT and 1 Gbit/s, and by 1.12× at 50 ms and 100 Mbit/s, so the wide-area penalty is small compared to the cryptographic work.

Dataset Diversity. Repeating the reference query on Lattigo across the three cohorts of Table 2 yields 43 s on MIMIC-IV [59], 41 s on the HCUP NIS cohort [60], and 37 s on the nuclear-medicine cohort [15]. The spread tracks the domain of the attribute the query compares against, which sets the degree of the Lagrange indicator polynomial: the 8-bit `creatinine_q` range of the MIMIC-IV reference query costs more than the Boolean and 2-bit equality predicates of the HCUP and nuclear-medicine queries (cf. Section D), even though the HCUP cohort is by far the largest of the three. FeCaD's query times on the reference predicate remain feasible when the Lattigo or HE3DB backend is chosen.

Takeaway. The Lattigo backend grows near-linearly along the per-provider record axis (log-log exponent 0.94), where per-patient consent enforcement dominates, and sub-linearly in federation size under single-node CPU contention. Combining the two near-independent axes places a hundred-provider deployment within 14 min, supporting cohort discovery toward the continental-biobank regime of [29] under cryptographic per-patient consent.

7. Discussion

We now characterize some discrepancies between this baseline and potential clinical deployment, as well as the regulatory and interoperability environments.

7.1. Compliance with Design Goals

FeCaD meets all four goals of Section 2.3. G1 holds under standard Ring-LWE assumptions: records, query literals, and intermediate state remain ciphertexts at providers and proxy (Sections 5.1 and 5.2), with the only plaintext exposure occurring at the optional review board, by design. G2 follows

from bidirectional key switching and proxy-side aggregation, with Section 5.3 characterizing potential collusion. G3 builds on the per-query envelope, which binds a BLS-signed witness to the cancellation ciphertext with unblinding conditional on a fresh patient signature. The prototype's policy representation is intentionally coarser than the class- and attribute-level granularity that G3 targets (cf. Section 6.1). G4 is supported by Section 6: consent-bound queries over 10 000 records on a single Lattigo provider complete in 36 s, end-to-end latency grows sub-linearly with federation size under co-located contention, and a hundred-provider deployment completes within 14 min, supporting the biobank regime targeted in [29].

7.2. Limitations

Cost, Depth, and the CKKS Footprint. The per-operation latency and multiplicative-depth bounds of fully homomorphic encryption constitute a price of end-to-end confidentiality. However, we argue that cohort discovery is sufficiently offline-tolerant to absorb these costs [15]. GPU acceleration [61] and quantization [23] can further reduce them. We also found CKKS parameter sets that yield per-patient blinding material at terabyte scales for biobank cohort sizes, which we report as a negative result for the CKKS family.

Structural Hardening. The prototype supports both a merged deployment, in which one host holds sk_h and evaluates, and a split one, in which the hospital encrypts and discards the blinding scalars while the evaluator holds only an evaluation key and the ciphertext store. The merged deployment satisfies the honest-but-curious model but lets a malicious provider decrypt the key-switched query literals via sk_h (cf. Section 5.3). Splitting provides a remedy, at the cost of one store transfer at enrollment, but may not reflect deployment realities. The optional review board can be hardened analogously: a k-of-n threshold reviewer whose admission witness is a BLS aggregate signature would raise the collusion threshold.

Residual Leakage and Revocation Semantics. The plaintext predicate template (column references, operators, connectives) traverses the proxy and the provider in the clear, and exhaustive backend evaluation collapses the access pattern to the full database index set [22], [23]. Hence, the residual channel reduces to the template itself and remains exposed to reconstruction-style attacks [47], [48], [49], [50]. A fixed-shape oblivious compilation at admission, combined with Path ORAM [62] on the evaluation traffic, would suppress these channels [23]. The consent token invalidates future capabilities if revoked but cannot invalidate in-flight evaluations or result ciphertexts already at the user.

Complementary PETs. We position synthetic patient data [63], [64] and trusted execution environments [13] as complementary efforts. Synthetic cohorts can augment data for learning objectives, but cannot guide recruitment to identifiable individuals. A hybrid TEE/HE design that confines only consent-witness verification to an enclave reduces the hardware attack surface to a small kernel while keeping the records under FHE.

7.3. Regulatory Compliance & Clinical Adoption

A definitive legal assessment lies outside our expertise, yet the technical foundation of FeCaD aligns with the directions advocated by recent positions on FHE-based medical-data sharing under the GDPR and HIPAA [11], [65], [66]. Specifically, providers observe only ciphertext during predicate evaluation, and the proxy holds no FHE secret key (cf. Section 5.2). The status of processed records as non-personal in a given jurisdiction nevertheless hinges on case-specific controllership arrangements [65]. Removing the platform operator from the trust base further suppresses an operator-side audit trail that ethics committees and data-access committees expect, so FeCaD complements organizational accountability through signed query provenance, periodic data-use reporting, and the institutional review pathways established by federated networks [16], [33].

Clinical Integration. Interoperability may build on established record formats such as ICD-10, LOINC, and SNOMED-CT for clinical terms, openEHR for longitudinal records, and FHIR for document exchange [67], [68]. FeCaD is agnostic to the source format because the column-chunk layout (cf. Section 6.1) converts and stores the corresponding schema at data-ingestion time.

Applications Beyond Patient Discovery. The combination of decentralized FHE evaluation with per-subject consent enforcement may generalize to other settings in which multiple custodians hold structured records about individuals, whose authorization conditions a predicate-based discovery step that precedes downstream analysis. Within the clinical domain, drug-trial pre-screening and cross-biobank genomic variant queries are further candidate applications that replace Beacon-style advertisements (cf. Section 3.1) with consent-aware predicates.

8. Conclusion

We presented FeCaD, a decentralized homomorphically encrypted database that, for the first time, couples secure cross-institutional patient discovery with cryptographic per-record consent enforcement. FeCaD positions itself between centralized discovery hubs, anytrust federated analytics, and single-server FHE engines by combining decentralization with expressive predicates and fine-grained consent enforcement, none of which prior work delivered jointly. Our prototype evaluation across three representative datasets indicates that completion time grows near-linearly in per-provider records, where consent enforcement dominates, and sub-linearly in federation size, so the architecture extrapolates toward the 10^7-record regime envisaged for biobank-scale discovery. The most consequential open directions are the integration of oblivious access primitives that further suppress the residual access-pattern channel and an empirical study of per-class consent-policy formulation in collaboration with clinical partners. These directions move homomorphically encrypted decentralized discovery toward replacing trusted-coordinator hubs with cryptographic guarantees.

Acknowledgments

We thank Michael Herwig for an initial exploration of decentralized consent and Jonas Hermsen for exploring HE^{3}DB's interface.

References

[1] S. Welten et al., "A Privacy-Preserving Distributed Analytics Platform for Health Care Data," *Methods Inf Med*, vol. 61, S 01 2022.

[2] A. J. Robertson et al., "It Is in Our DNA: Bringing Electronic Health Records and Genomic Data Together for Precision Medicine," *JMIR Bioinform Biotech*, vol. 5, 2024.

[3] I. S. Kohane et al., "A translational engine at the national scale: Informatics for integrating biology and the bedside," *J Am Med Inform Assoc*, vol. 19, no. 2, 2012.

[4] D. Moynihan et al., "Analysis and visualisation of electronic health records data to identify undiagnosed patients with rare genetic diseases," *Sci Rep*, vol. 14, no. 1, 2024.

[5] the eMERGE Team et al., "The eMERGE Network: A consortium of biorepositories linked to electronic medical records data for conducting genomic studies," *BMC Med Genomics*, vol. 4, no. 1, 2011.

[6] L. R. Kalankesh and E. Monaghesh, "Utilization of EHRs for clinical trials: A systematic review," *BMC Med Res Methodol*, vol. 24, no. 1, 2024.

[7] G. Hripcsak et al., "Characterizing treatment pathways at scale using the OHDSI network," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 113, no. 27, 2016.

[8] S. Courbier et al., "Share and protect our health data: An evidence based approach to rare disease patients' perspectives on data sharing and data protection - quantitative survey and recommendations," *Orphanet J Rare Dis*, vol. 14, no. 1, 2019.

[9] Y. Mou et al., "Distributed Privacy-Preserving Data Analysis in NFDI4Health With the Personal Health Train," *Proc Conf Res Data Infrastr*, vol. 1, 2023.

[10] "HHS Breach Portal," US Department of Health and Human Services.

[11] C. M. L. Zegers et al., "Mind Your Data: Privacy and Legal Matters in eHealth," *JMIR Form Res*, vol. 5, no. 3, 2021.

[12] A. M. Arellano et al., "Privacy Policy and Technology in Biomedical Data Science," *Annu. Rev. Biomed. Data Sci.*, vol. 1, no. 1, 2018.

[13] J. Lohmöller et al., "Complementing organizational security in data ecosystems with technical guarantees," in *Proc. 1st Conf. Build. Secure Empower. Cyberspace BuildSEC 24*, IEEE, 2024.

[14] Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nat Rev Genet*, vol. 15, no. 6, 2014.

[15] J. Pennekamp et al., "PatDiscover: Privacy-preserving discoverability of patients," in *Proc. 2025 Annu. Comput. Secur. Appl. Conf. Workshop*, IEEE, 2025.

[16] D. J. Froelicher, "Privacy-Preserving Federated Analytics using Multiparty Homomorphic Encryption," EPFL Lausanne, 2021.

[17] R. Geva et al., "Collaborative privacy-preserving analysis of oncological data using multiparty homomorphic encryption," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 120, no. 33, 2023.

[18] K. Caine and R. Hanania, "Patients want granular privacy control over health information in electronic medical records," *J. Am. Med. Inform. Assoc.*, vol. 20, no. 1, 2013.

[19] H. Kim et al., "iCONCUR: Informed consent for clinical data and bio-sample use for research," *J. Am. Med. Inform. Assoc.*, vol. 24, no. 2, 2017.

[20] G. Barazzetti et al., "Broad consent in practice: Lessons learned from a hospital-based biobank for prospective research on genomic and medical data," *Eur J Hum Genet*, vol. 28, no. 7, 2020.

[21] J. Kaye et al., "Dynamic consent: A patient interface for twenty-first century research networks," *Eur J Hum Genet*, vol. 23, no. 2, 2015.

[22] S. Bian et al., "HE3DB: An Efficient and Elastic Encrypted Database Via Arithmetic-And-Logic Fully Homomorphic Encryption," in *Proc. 2023 ACM SIGSAC Conf. Comput. Commun. Secur.*, ACM, 2023.

[23] S. Bian et al., "Engorgio: An arbitrary-precision unbounded-size hybrid encrypted database via quantized fully homomorphic encryption," in *Proc. 34th USENIX Secur. Symp.* USENIX Association, 2025.

[24] D. Froelicher et al., "UnLynx: A Decentralized System for Privacy-Conscious Data Sharing," *Proc. Priv. Enhancing Technol.*, vol. 2017, no. 4, 2017.

[25] J. Yuan et al., "Towards a privacy preserving cohort discovery framework for clinical research networks," *J. Biomed. Inform.*, vol. 66, 2017.

[26] J. Lohmöller et al., *FeCaD GitHub repository*, https://github.com/COMSYS/fecad, Online, 2026.

[27] J. Lohmöller et al., *FeCaD artifact*, https://doi.org/10.5281/zenodo.23563317, Zenodo, 2026.

[28] L. Guo and W.-C. Yau, "Efficient Secure-Channel Free Public Key Encryption with Keyword Search for EMRs in Cloud Storage," *J Med Syst*, vol. 39, no. 2, 2015.

[29] C. Bycroft et al., "The UK Biobank resource with deep phenotyping and genomic data," *Nature*, vol. 562, no. 7726, 2018.

[30] R. L. Fleurence et al., "Launching PCORnet, a national patient-centered clinical research network," *J. Am. Med. Inform. Assoc.*, vol. 21, no. 4, 2014.

[31] R. J. Ludwig et al., "A comprehensive review of methodologies and application to use the real-world data and analytics platform TriNetX," *Front. Pharmacol.*, vol. 16, 2025.

[32] G. Gürsoy, "Genome Privacy and Trust," *Annu. Rev. Biomed. Data Sci.*, vol. 5, no. 1, 2022.

[33] J. L. Raisaro et al., "MedCo: Enabling Secure and Privacy-Preserving Exploration of Distributed Clinical and Genomic Data," *IEEE/ACM Trans. Comput. Biol. and Bioinf.*, vol. 16, no. 4, 2019.

[34] Z. Zhang et al., "ArcEDB: An Arbitrary-Precision Encrypted Database via (Amortized) Modular Homomorphic Encryption," in *Proc. 2024 ACM SIGSAC Conf. Comput. Commun. Secur.*, ACM, 2024.

[35] S. Welten et al., "PASTA-4-PHT: A pipeline for automated security and technical audits for the personal health train," *BMC Med Inform Decis Mak*, vol. 26, no. 1, 2026.

[36] J. R. Popovic, "Distributed data networks: A blueprint for Big Data sharing and healthcare analytics," *Ann. N. Y. Acad. Sci.*, vol. 1387, no. 1, 2017.

[37] R. Haber et al., "Core Concepts in Pharmacoepidemiology: Multi-Database Distributed Data Networks," *Pharmacoepidemiol. Drug Saf.*, vol. 34, no. 7, 2025.

[38] M. Lablans et al., "Strategien zur Vernetzung von Biobanken: Klassifizierung verschiedener Ansätze zur Probensuche und Ausblick auf die Zukunft in der BBMRI-ERIC," *Bundesgesundheitsbl.*, vol. 59, no. 3, 2016.

[39] M. Fiume et al., "Federated discovery and sharing of genomic data using Beacons," *Nat Biotechnol*, vol. 37, no. 3, 2019.

[40] S. S. Shringarpure and C. D. Bustamante, "Privacy Risks from Genomic Data-Sharing Beacons," *Am. J. Hum. Genet.*, vol. 97, no. 5, 2015.

[41] Dawn Xiaoding Song et al., "Practical techniques for searches on encrypted data," in *Proc. 2000 IEEE Symp. Secur. Priv.*, IEEE Comput. Soc, 2000.

[42] D. Boneh and X. Boyen, "Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles," in *Proc. EUROCRYPT 2004*, red. by T. Kanade et al., vol. 3027, Springer Berlin Heidelberg, 2004.

[43] M. Iezzi, "Practical Privacy-Preserving Data Science With Homomorphic Encryption: An Overview," in *Proc. 2020 IEEE Int. Conf. Big Data*, IEEE, 2020.

[44] K. Munjal and R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," *Complex Intell. Syst.*, vol. 9, no. 4, 2023.

[45] I. Amorim and I. Costa, "Leveraging Searchable Encryption through Homomorphic Encryption: A Comprehensive Analysis," *Mathematics*, vol. 11, no. 13, 2023.

[46] A. Acar et al., "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Comput Surv*, vol. 51, no. 4, 2018.

[47] M. S. Islam et al., "Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation," *NDSS*, 2012.

[48] G. Kellaris et al., "Generic Attacks on Secure Outsourced Databases," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, ACM, 2016.

[49] P. Grubbs et al., "Pump up the Volume: Practical Database Reconstruction from Volume Leakage on Range Queries," in *Proc. 2018 ACM SIGSAC Conf. Comput. Commun. Secur.*, ACM, 2018.

[50] S. Oya and F. Kerschbaum, "Hiding the access pattern is not enough: Exploiting search pattern leakage in searchable encryption," in *Proc. 30th USENIX Secur. Symp.*, USENIX Association, 2021.

[51] HEW, *Belmont report*, PDF, 1979.

[52] B. M. Knoppers et al., "From genomic databases to translation: A call to action," *J Med Ethics*, vol. 37, no. 8, 2011.

[53] C. Dotter et al., "Sharing health data for research purposes: Results of a population survey in Germany," *BMC Health Serv Res*, vol. 25, no. 1, 2025.

[54] A. K. Conduah et al., "Data privacy in healthcare: Global challenges and solutions," *Digit. Health*, vol. 11, 2025.

[55] I. Budin-Ljøsne et al., "Dynamic Consent: A potential solution to some of the challenges of modern biomedical research," *BMC Med Ethics*, vol. 18, no. 1, 2017.

[56] C. Mouchet et al., "Multiparty Homomorphic Encryption from Ring-Learning-with-Errors," *Proc. Priv. Enhancing Technol.*, vol. 2021, no. 4, 2021.

[57] D. Boneh et al., "Short Signatures from the Weil Pairing," in *Proc. ASIACRYPT 2001*, red. by G. Goos et al., vol. 2248, Springer Berlin Heidelberg, 2001.

[58] M. Albrecht et al., "Homomorphic Encryption Standard," in *Protecting Privacy through Homomorphic Encryption*, Springer International Publishing, 2021.

[59] A. E. W. Johnson et al., "MIMIC-IV, a freely accessible electronic health record dataset," *Sci Data*, vol. 10, no. 1, 2023.

[60] HCUP, *HCUP National Inpatient Sample*, 2024.

[61] Z. Wang et al., "Chameleon: An Efficient FHE Scheme Switching Acceleration on GPUs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 36, no. 11, 2025.

[62] E. Stefanov et al., "Path ORAM: An Extremely Simple Oblivious RAM Protocol," *J. ACM*, vol. 65, no. 4, 2018.

[63] A. Tucker et al., "Generating high-fidelity synthetic patient data for assessing machine learning healthcare software," *npj Digit. Med.*, vol. 3, no. 1, 2020.

[64] L. Pilgram et al., "Protecting patient privacy in tabular synthetic health data: A regulatory perspective," *npj Digit. Med.*, vol. 8, no. 1, 2025.

[65] J. Scheibner et al., "Revolutionizing Medical Data Sharing Using Advanced Privacy-Enhancing Technologies: Technical, Legal, and Ethical Synthesis," *J Med Internet Res*, vol. 23, no. 2, 2021.

[66] A. Cohen and K. Nissim, "Towards formalizing the GDPR's notion of singling out," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 117, no. 15, 2020.

[67] A. Telenti and X. Jiang, "Treating medical data as a durable asset," *Nat Genet*, vol. 52, no. 10, 2020.

[68] A. Appenzeller et al., "Towards Distributed Healthcare Systems – Virtual Data Pooling Between Cancer Registries as Backbone of Care and Research," in

Proc. 2021 IEEE/ACS 18th Int. Conf. Comput. Syst. Appl., IEEE, 2021.

[69] E. B. Habermann et al., "Impact of Availability of Immediate Breast Reconstruction on Bilateral Mastectomy Rates for Breast Cancer across the United States: Data from the Nationwide Inpatient Sample," *Ann Surg Oncol*, vol. 21, no. 10, 2014.

[70] T. K. Chung et al., "Transoral robotic surgery for oropharyngeal and tongue cancer in the United States," *Laryngoscope*, vol. 125, no. 1, 2015.

[71] E. M. George et al., "Safety and Tolerance of Radical Hysterectomy for Cervical Cancer in the Elderly," *Gynecol. Oncol.*, vol. 134, no. 1, 2014.

Appendix A. Ethical Considerations

FeCaD replaces plaintext exchange in cohort discovery with cryptographically organized evaluation, to the benefit of patient privacy. We discuss the ethical aspects of this research along the following axes.

Minimizing Potential Harm. Our evaluation operates on previously collected datasets and on simulated patient identities generated within the prototype (cf. Section 6.1). Moreover, our cryptographic encapsulation contributes a strict boundary throughout the protocol: providers have only encrypted counts, the proxy operates over ciphertexts only, and the requester receives an encrypted indicator vector with indications controlled by given consent. The disclosure footprint of cohort discovery therefore shrinks well below the plaintext federated baselines surveyed in Section 3.

Impact on Third Parties. Our use of the source datasets remained limited to credentialed secondary use. The prototype performs no intervention, observation, or biological measurement, and exposes only cohort cardinalities to the requester. Reidentification attempts are prohibited by Data Use Agreements and lie outside the scope of our research questions.

Ethics Committee Oversight. Oversight is inherited from the source datasets. We obtained MIMIC-IV [59] under the PhysioNet Credentialed Health Data License following CITI Program "Data or Specimens Only Research" training and a signed Data Use Agreement; the records are already de-identified per HIPAA Safe Harbor. The HCUP National Inpatient Sample [60] is distributed by AHRQ as a de-identified, public-release dataset under the HCUP Data Use Agreement, which mandates the HCUP DUA Training Course and prohibits any attempt to identify discharges or hospitals. The nuclear-medicine cohort [15] is not derived from real patient records but synthesized by their open-source generator, so its use is unencumbered by a clinical access control. We thus regard the work as exempt from separate institutional review: the conditions attached to the credentialed datasets already govern usage, and the synthesized cohort raises no human-subjects exposure.

Releasable Artifacts. We release the FeCaD prototype, including the synthetic record generators and evaluation scripts used, on GitHub [26] and Zenodo [27]. Redistribution of MIMIC-IV and HCUP NIS remains with the originating data custodians (PhysioNet and AHRQ, respectively).

Appendix B. Discovery Protocol

We instantiate the discovery protocol of Fig. 3 along the setup phases (Ⓐ–Ⓒ) and the per-query steps (①–⑦) of that figure; Table 3 summarizes the key material per entity. The key manager may go offline after Ⓐ. We write sk_u for the user and sk_h for the data holder.

Key-switching Primitive. For two RLWE secret keys sk_A, sk_B, the key-switching key $\mathrm{ksk}_{A\to B} \leftarrow \mathrm{KSKGen}(sk_A, sk_B)$ encrypts the coefficients of sk_A under sk_B [56]. Applying it to ct_A yields ct_B with $\mathrm{Dec}_{sk_B}(ct_B) = \mathrm{Dec}_{sk_A}(ct_A)$ up to a controlled noise growth. FeCaD invokes this primitive twice per data holder: $\mathrm{ksk}_{u\to h}$ on inbound literals and $\mathrm{ksk}_{h\to u}$ on outbound results.

B.1. Setup Phase

Ⓐ **Setup and key distribution.** The key manager generates the user FHE keypair (sk_u, pk_u) and, per data holder, both the FHE keypair (sk_h, pk_h) and the two key-switching keys $\mathrm{ksk}_{u\to h}, \mathrm{ksk}_{h\to u}$. It distributes sk_u to the user, sk_h and the corresponding evaluation key to the data holder, the evaluation key alone to the evaluator, and both KSKs to the proxy, before erasing its copies. The user additionally generates a BLS12-381 signing keypair $(sk_u^{\mathrm{BLS}}, pk_u^{\mathrm{BLS}})$, and the optional review board may generate $(sk_{\mathrm{REV}}^{\mathrm{BLS}}, pk_{\mathrm{REV}}^{\mathrm{BLS}})$.

Ⓑ **Patient enrollment, blinding and cancellation.** The data holder packs records into ciphertexts of slot count N over plaintext modulus t and assigns patient p the row-index set $R_p \subseteq \{0, \ldots, N-1\}$ of canonical slot vectors $\{e_i\}_{i\in R_p}$. On enrollment, a patient generates a BLS12-381 keypair $(sk_p^{\mathrm{BLS}}, pk_p^{\mathrm{BLS}})$, samples a fixed blinding scalar $r_p \in \mathbb{Z}_t^*$, ships the blinding ciphertext $ct_{B,p} = \mathrm{Enc}_{pk_h}(\sum_{i\in R_p} r_p \cdot e_i)$ to the data holder, and registers pk_p^{BLS} with the proxy. The matching cancellation ciphertext $ct_{C,p} = \mathrm{Enc}_{pk_h}(\sum_{i\in R_p}(-r_p)\cdot e_i)$ is released by the patient inside the per-query envelope of Section B.3, and the proxy may cache previously seen $ct_{C,p}$ across queries to save bandwidth, although it is not required to.

Ⓒ **Local data encryption and upload to evaluator.** The data holder encrypts each record under sk_h and uploads the resulting ciphertext store, together with the evaluation key and the per-patient blinding ciphertexts $\mathrm{Enc}_{pk_h}(+r_p)$, to the evaluator. Records never leave the data holder in plaintext and the evaluator has no access to sk_h, so offloading evaluation to e.g., a cloud service is feasible.

B.2. Query Execution

① **Query Review (optional).** The user composes plaintext query Q, splits it into a public predicate template T and literal vector $\vec{v}$, samples randomness $\vec{r}$, and computes $\mathrm{ct}_{\vec{v}} \leftarrow$

$\mathrm{Enc}_{pk_u}(\vec{v};\vec{r}), \sigma_u \leftarrow \mathrm{BLS.Sign}(sk_u^{\mathrm{BLS}}, H(T \,\|\, \mathrm{ct}_{\vec{v}} \,\|\, qd))$, where the public query descriptor $qd = H(Q, \nu, D)$ binds the query, the per-query nonce, and the purpose description, as composed in Section 4.3.

If a review board is deployed, the user additionally submits $(Q, \vec{r}, \mathrm{ct}_{\vec{v}}, qd, \sigma_u)$. The board then verifies σ_u, recomputes $\mathrm{Enc}_{pk_u}(\vec{v};\vec{r})$ to confirm equality with $\mathrm{ct}_{\vec{v}}$, evaluates the admission policy on Q, signs $\sigma_{\mathrm{REV}} \leftarrow \mathrm{BLS.Sign}(sk_{\mathrm{REV}}^{\mathrm{BLS}}, H(\mathrm{ct}_{\vec{v}} \,\|\, qd))$, and erases $(Q, \vec{r})$.

② **Encrypted Query Submission.** The user (or board) ships $(T, \mathrm{ct}_{\vec{v}}, qd, \sigma_u, \sigma_{\mathrm{REV}})$ to the proxy, which verifies σ_u and may verify σ_{REV}.

③ **Consent Collection.** The proxy publishes qd to patients. A patient (or a trustworthy delegate) polls, evaluates its policy, and on a match returns the per-query envelope of Section B.3: the cancellation ciphertext $ct_{C,p}$ of step Ⓑ together with the fresh signature $\sigma_p \leftarrow \mathrm{BLS.Sign}(sk_p^{\mathrm{BLS}}, H(qd \,\|\, ct_{C,p}))$ binding it to the current qd (a proxy that already cached $ct_{C,p}$ from an earlier query may accept the signature alone). The proxy verifies each σ_p against the registered pk_p^{BLS}, admits the patient into the consenting set $C \subseteq \mathcal{P}$ iff verification succeeds, and then aggregates the admitted $\{\sigma_p\}_{p\in C}$ into a single multi-signature σ_{qd} over the message set $\{H(qd \,\|\, ct_{C,p})\}_{p\in C}$ that proves consent freshness to users. This step may run asynchronously with steps ④ and ⑤.

④ **Key Switching.** The proxy applies $\mathrm{ksk}_{u\to h}$ to obtain $\mathrm{ct}_{\vec{v}}^{(h)} = \mathrm{KS}(\mathrm{ksk}_{u\to h}, \mathrm{ct}_{\vec{v}})$ under the data holder's key and forwards $(T, \mathrm{ct}_{\vec{v}}^{(h)}, qd, \sigma_{\mathrm{REV}})$ to the evaluator.

⑤ **Homomorphic Evaluation and Blinding.** The evaluator verifies σ_{REV} (if present), compiles T against $\mathrm{ct}_{\vec{v}}^{(h)}$, and evaluates the predicate row-by-row over the ciphertext store to obtain the per-row indicator ct_{match}. It blinds each row by the persistent blinding ciphertext $ct_{B,p}$ of the row's owning patient and returns $ct_B = ct_{\mathrm{match}} + \sum_{p\in\mathcal{P}} ct_{B,p}$ to the proxy (cf. Section B.3).

⑥ **Cancellation and Key Switching.** The additive blinding $+r_p$ from step ⑤ already renders the non-consenting slots statistically independent of the predicate after decryption; to also hide the consent map from the user, the proxy masks the slots $\bigcup_{p\notin C} R_p$ to a public zero (or drops them, depending on the packing layout), adds the cancellation $ct_{C,p}$ for every $p \in C$, and applies $\mathrm{ksk}_{h\to u}$ to obtain $ct_R = \mathrm{KS}\big(\mathrm{ksk}_{h\to u}, \mathrm{Filter}_C(ct_B) + \sum_{p\in C} ct_{C,p}\big)$ under the user's key.

⑦ **Result Return and Consent Receipt.** The proxy ships ct_R together with the consent receipt $(\sigma_{\mathrm{qd}}, qd)$ to the user, who decrypts under sk_u to read $q(row_i) \in \{0, 1\}$ on the rows of consenting patients and may retain the receipt as an audit record; the consenting set C itself stays at the proxy. A regulator or ethics board with access to C and to a patient public-key registry can then verify, in a single pairing-product check, that the $|C|$ patients in C each signed consent for the exact qd admitted at ②. The user receives no identifying information about C and learns only the result cardinality from the ciphertext layout.

TABLE 3. KEY MATERIAL PER ENTITY.

Entity	Holds
User	(sk_u, pk_u) FHE; $(sk_u^{\mathrm{BLS}}, pk_u^{\mathrm{BLS}})$
Data holder	(sk_h, pk_h) FHE; records under sk_h; $\mathrm{Enc}_{pk_h}(+r_p)$ per patient
Evaluator	evaluation key of sk_h; ciphertext store and blinding ciphertexts
Proxy	$\mathrm{ksk}_{u\to h}, \mathrm{ksk}_{h\to u}$; (enrollment-time) $\mathrm{Enc}_{pk_h}(-r_p)$ per patient
Patient	$(sk_p^{\mathrm{BLS}}, pk_p^{\mathrm{BLS}})$; r_p
Key manager (offline)	all FHE secret keys during setup; deletable after distribution
Review board (opt.)	$(sk_{\mathrm{REV}}^{\mathrm{BLS}}, pk_{\mathrm{REV}}^{\mathrm{BLS}})$; ephemeral $(Q, \vec{r})$ during admission

B.3. Per-Query Consent Envelope

The per-query envelope $(ct_{C,p}, \sigma_p)$ of step ③ conditions each cancellation, as defined in step Ⓑ, on a fresh BLS signature; we summarize the scope and cost of the construction here.

Scope. Binding σ_p to $H(qd \,\|\, ct_{C,p})$ pins each cancellation jointly to the consented query and to the specific ciphertext, prohibiting cross-query replay and ciphertext substitution by the proxy. A malicious privacy proxy therefore cannot apply those cancellations to other queries without forging σ_p. The envelope does *not* by itself defend against a malicious data-holder/evaluator coalition, which can homomorphically negate the additive blinding $ct_{B,p}$ to pre-cancel records of non-consenting patients. The Filter_C of step ⑥ bounds that attack: the proxy zeroes non-consenting slots before the result leaves it, i.e., an unconsented record still reaches the user as a public zero. The role split of Section 7.2 is the defense against the literal disclosure such a coalition would otherwise obtain.

Patient-side Cost. Patients perform one public-key FHE encryption per enrollment, a fresh full-level ciphertext of ≈3 MiB for BGV at $\mathrm{LogN} = 15$, which the proxy stores modulus-switched down to level 1 at 1.0 MiB (≈100 KiB for BFV, ≈4 KiB for TFHE, ≈116 MiB for CKKS at the Engorgio parameter set) plus one 48-byte BLS signature *per query*. When the proxy caches $ct_{C,p}$, only the signature needs to be transmitted per query.

Lifecycle. A witness is scoped by the signed descriptor: $qd = H(Q, \nu, D)$ commits to the query, the reviewed purpose description, and a per-query nonce (Section 4.3), i.e., a witness is bound to one query instance and is not transferable across users, queries, or purposes. Replay is therefore excluded by the nonce, and the proxy additionally admits a witness only inside the consent window it opened for that descriptor. Revocation withdraws the patient's participation in all future queries, but cannot retract evaluations already in flight or result ciphertexts already delivered (Section 7.2).

Appendix C. Extended Evaluation

Setup Cost Breakdown. Figure 7 reports the one-time setup cascade of Section 6.3 per backend, decomposed into

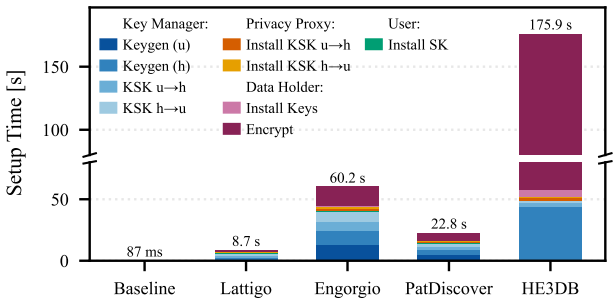


Figure 7. One-time setup cascade per backend at the reference point of 10 000 MIMIC-IV records and one data provider. Patient enrollment runs independently per patient and is not included.

the key-manager, user, privacy-proxy, and data-holder phases at the same reference point (10 000 MIMIC-IV records, one data provider, Lattigo at LogN = 15). Key generation at the key manager together with bidirectional key switching totals 6.0 s on Lattigo, 40 s on Engorgio, and 14 s on PatDiscover; HE3DB derives its switching key lazily and concentrates the cost in 44 s of key generation for sk_h. The data-provider ingestion phase ranges from 1.2 s on Lattigo to 118 s on HE3DB, with PatDiscover at 6.2 s and Engorgio at 16 s. Per-patient enrollment, which the figure omits, installs one BGV cancellation ciphertext of approximately 1.0 MiB per patient at the privacy proxy (Lattigo), i.e., 9.8 GiB for the cohort of 10 000 patients. At deployment time, the per-query consent envelope references this ciphertext by index, transmits one BLS12-381 signature of 48 B and 140 B of JSON-framed metadata to the privacy proxy, and does not retransmit cryptographic material (cf. Section 4.3).

Appendix D. Dataset Characteristics and Queries

The schemas below are minimal yet representative slices of their source datasets, chosen to exercise the different attribute types at least once. Runtime scales linearly in the number of attributes (Section 6.4), so wider real-world schemas do not alter these results. Continuous attributes are quantized to fixed-width unsigned integers during preprocessing, and query literals are stated in the same quantized units, so each value occupies one slot of the $t = 65\,537$ plaintext modulus shared by our Lattigo BGV and the BFV-based PatDiscover backend (cf. Section 6.1). We cap quantized attributes at 8 bits: the indicator polynomial has degree $maxVal$ with coefficients bounded by t, so its noise growth leaves the result exact up to a 9-bit domain and unusable from 10 bits upward under our parameters.

MIMIC-IV. We use one row per `subject_id` from the `hosp` module of MIMIC-IV v3.1 [59]: `age` (7 b, from `anchor_age`), `sex` (1 b, derived from `gender`), four diagnosis flags `icd_{sepsis,aki,pneumonia,heart_fail}` (1 b each, prefix-matched against `diagnoses_icd` over both ICD-10 and ICD-9 codes), and two quantized `labevents` values `creatinine_q` (8 b, mg/dL×10) and `lactate_q` (8 b, mmol/L×10).

Nuclear Medicine. We reuse the schema of Pennekamp et al. [15]: `whoGrade` and `tumorType` (2 b each, four enum values per attribute); three molecular markers `mgmtPromoterMethylation`, `codeletion1q19q`, `idhWildType` (1 b each); the 3-D tumor position `tumorPos_{x,y,z}` (6 b each, ×10 quantization of the normalized $[0, 4]$ coordinate, used for the distance variant); eight therapy/diagnostic flags `biopsy`, `fetPet`, `resection`, `ceTegProtocol`, `radioTherapy`, `chemoTherapy`, `boostTherapy`, `tmzTherapy` (2 b each, Yes/No/Unknown enums); two outcome flags `activeTumorTissue` and `tumorProgression` (1 b each); and `age` (7 b).

HCUP NIS. One row per discharge from the AHRQ NIS 2020 Core file [60]: `age` (7 b), `female` (1 b), and eight Boolean cohort flags derived from the 40 `I10_DXi` and 25 `I10_PRi` ICD-10-CM/PCS slots (ICD-9 codes from the original publications crosswalked via HCUP GEM), reproducing three published procedure-outcome studies [69], [70], [71]: `dx_cervical_ca`/`pr_radical_hyst` (George et al.); `dx_oroph`/`pr_pharyn` and `dx_tongue`/`pr_gloss`, the four raw flags that combine into the Chung et al. Q2 cohort (dx_oroph ∧ pr_pharyn) ∨ (dx_tongue ∧ pr_gloss) at query time via SQL precedence (∧ > ∨); and `dx_breast_ca`/`pr_mastectomy` (Habermann et al.).

Query Suite. The queries *Q1–Q4* instantiate the predicate classes of Section 4.2 at increasing complexity: *Q1* is a low-complexity filter (one to three attribute conditions), isolating fixed proxy/HTTP overhead from per-row evaluation cost; *Q2* adds one or two focused equality or range conditions; and *Q3* forms a conjunction of two to three equalities and ranges. *Q4*, finally, extends *Q3* with an additional equality literal that triggers the encrypted-literal path (cf. Section 6.1); HCUP NIS has no *Q4*. Predicate shapes and query goals per dataset differ because the schemas expose different attributes, as listed in Table 4.

TABLE 4. PER-DATASET ADAPTATIONS OF THE FOUR QUERIES USED IN SECTION 6.3.

Dataset	Slot	SQL predicate
MIMIC-IV	Q1	`age >= 0`
	Q2	`creatinine_q >= 15`
	Q3	`creatinine_q >= 15 AND age >= 60`
	Q4	`creatinine_q >= 15 AND age >= 60 AND sex = 1`
Nuclear med.	Q1	`whoGrade >= 1`
	Q2	`tumorType = 3 AND idhWildType = 1`
	Q3	`tumorType = 3 AND idhWildType = 1 AND age <= 60`
	Q4	`tumorType = 3 AND idhWildType = 1 AND age <= 60 AND biopsy = 1`
HCUP NIS	Q1	`female = 1 AND dx_cervical_ca = 1 AND pr_radical_hyst = 1`
	Q2	`dx_oroph = 1 AND pr_pharyn = 1 OR dx_tongue = 1 AND pr_gloss = 1`
	Q3	`female = 1 AND dx_breast_ca = 1 AND pr_mastectomy = 1`