STAP
Smart Technologies Academic Press

Jordanian Journal of Informatics and Computing
https://www.jjic.thestap.com/

Secure Multi-Party Computation for Cross-Institutional Query Federation without Raw Data Exposure in Healthcare Databases

Haider H. Razzaq[1]

[1]*Computer Department, College of Science, Mustansiriyah University, Baghdad, Iraq*

ARTICLE INFO

Article History

Received: 07-05-2026
Revised: 02-06-2026
Accepted: 18-10-2026

Vol.2027, No.1

DOI:
https://doi.org/10.63180/jjic.thestap.2027.1.2

***Corresponding author.**
Email:
haideritsec@uomustansiriyah.edu.iq

Orcid:

This is an open access article under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

Published by STAP Publisher.

ABSTRACT

Cross-institutional healthcare research requires secure analysis of patient data distributed across hospitals while complying with the Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR). Existing federated query platforms restrict analyses to predefined aggregates and may expose institution-level statistics, whereas federated learning remains vulnerable to information leakage. This paper proposes a Secure Multi-Party Computation (SMPC) framework for privacy-preserving federated Structured Query Language (SQL) queries across multiple healthcare institutions without disclosing individual records or intermediate results. The framework introduces a federated query language and compiler that transforms queries into depth-bounded arithmetic circuits. It integrates verifiable secret sharing, Beaver-triple generation, and distributed key generation to support secure equality testing, comparison, record linkage, and aggregation. Protocol complexity and security guarantees are analyzed under semi-honest and malicious adversarial models, with malicious security approximately doubling communication requirements. An analytical cost model, calibrated using the SPDZ family of Multi-Party Computation (MPC) frameworks, evaluates scalability across 3–30 institutions and 1,000–500,000 patient records per institution. Results project that a representative cross-institutional linkage query requires less than one second and approximately 77 megabytes of communication for 15 institutions containing 10,000 records each. Four healthcare case studies demonstrate the framework's potential applicability to rare-disease cohort discovery, drug-safety surveillance, and comparative-effectiveness research. The proposed framework provides a scalable approach to privacy-preserving healthcare analytics while supporting regulatory compliance and enabling secure collaborative research without centralizing sensitive patient information.

Keywords: Secure multi-party computation, federated query processing, healthcare data privacy, secret sharing, privacy-preserving record linkage, verifiable secret sharing, malicious security, HIPAA, GDPR.

How to cite the article
Razzaq, H. H. (2026). Secure Multi-Party Computation for Cross-Institutional Query Federation without Raw Data Exposure in Healthcare Databases. Jordanian Journal of Informatics and Computing, 2027(1), 20. https://doi.org/10.63180/jjic.thestap.2027.1.2

1. Introduction

Multi-institutional collaboration is often essential in clinical research: rare diseases rarely present in numbers sufficient for statistical power within a single hospital, post-marketing drug-safety signals are only detectable across large and diverse patient populations, and comparative-effectiveness research requires patient counts that no single institution can supply alone. At the same time, the patient data needed to answer these questions is legally and ethically bound to the institution that collected it: in the United States, the Health Insurance Portability and Accountability Act (HIPAA) restricts disclosure of protected health information (PHI) without patient authorization or a qualifying exception [1]; in the European Union, the General Data Protection Regulation (GDPR) imposes comparable restrictions on the processing and cross-border transfer of personal health data [2]. The result is a structural tension between the scientific value of pooled, cross-institutional data and the legal requirement that raw patient records never leave the custody of the institution that holds them.

Federated query platforms built on the i2b2 data model, most notably the Shared Health Research Information Network (SHRINE), address this tension by keeping patient-level data at each site and returning only aggregate patient counts to a central query hub [3]; the OHDSI collaborative extends this idea across a much larger, internationally federated network of databases standardized on the OMOP common data model [4], and PCORnet and the FDA Sentinel System apply the same distributed-query principle to comparative-effectiveness research and drug-safety surveillance at national scale [5], [6]. Both approaches are effective for hypothesis-generating cohort counts, but share two structural limitations that motivate this work. First, the set of computable queries is restricted to counts and simple aggregates over a fixed common data model; none of these systems supports an arbitrary cross-institutional join, for example, identifying patients who received care at two or more participating sites without either a trusted honest broker or an out-of-band linkage step that itself discloses identifying information. Second, the per-site count returned to the coordinating hub is not itself protected: a curious hub operator, or a sequence of overlapping queries, can use these per-site statistics to infer information about small, potentially re-identifiable subgroups at a specific institution, the same differencing-attack concern that motivates differential privacy [7]. Federated learning offers a complementary paradigm in which institutions exchange model gradients rather than data or query answers [8], and secure aggregation protocols can cryptographically hide any single institution's gradient contribution from the coordinating server [9]. Federated learning, however, answers a different question, it trains a shared statistical model rather than executing an ad hoc query and gradient updates have been shown to leak substantial information about the underlying training examples absent additional protection.

Secure multi-party computation (SMPC) offers a cryptographically stronger primitive that is a closer structural match to the federated-query problem: a set of mutually distrusting parties jointly compute an agreed function of their private inputs such that no party learns anything beyond the function's output, a guarantee formalized under a simulation-based security definition rather than an empirical one [10]. Several recent systems apply SMPC directly to healthcare query workloads: VaultDB evaluates SQL queries over secret-shared electronic health records within the HIPAA-covered CAPriCORN clinical research network [12]; the HiGHmed consortium's SIMON protocol supports secure distributed feasibility queries across German university hospitals [13]; and an SMPC-based bucketization technique has been proposed for privacy-preserving query processing over horizontally partitioned electronic medical records arranged in a star topology [14]. These systems demonstrate that SMPC-based query federation is practical at real institutional scale, but each targets a specific topology or query shape — two-source joins, fixed feasibility-count templates, or star-mediated aggregation — and none couples query compilation to an explicit cross-institutional identity-resolution stage, a reusable query language, or a complexity model that lets an architect budget round count and communication volume before deployment. This is the gap the present framework targets. Table 1 summarizes how these existing approaches compare on data exposure, query scope, and reported scale before Section 2 examines each in depth.

Table 1. Federated healthcare query approaches at a glance.

Approach	Data Exposure	Query Scope	Reported Scale	Key Limitation
SHRINE / i2b2 [3]	Aggregate counts only	Fixed count queries	Regional/national networks	Per-site counts visible to hub
OHDSI / OMOP [4]	Local statistics only	Pre-validated analysis packages	300+ databases, hundreds of millions of records	No arbitrary cross-site joins
PCORnet / Sentinel [5], [6]	Aggregate statistics	Comparative-effectiveness, safety surveillance	300+ million person-years (Sentinel)	Per-site outputs not cryptographically protected

Approach	Data Exposure	Query Scope	Reported Scale	Key Limitation
Federated learning + secure agg. [8], [9]	Model gradients	Model training, not ad hoc queries	Multi-institution deployments reported	Gradient-inversion risk without added protection
SMPC query systems [12]–[14]	None (secret-shared)	Fixed topology or query template	Single-network pilots	No reusable language or cost model
This framework	None (secret-shared)	Arbitrary SQL-style joins, n-party	Evaluated analytically to n = 15	Cost is projected, not yet empirically measured

This research makes seven contributions, substantially extending a preliminary treatment of the problem. First, it presents a system and threat model for SMPC-based federated query processing across n healthcare institutions under semi-honest, covert, and malicious adversary models. Second, it defines a small federated query language and a compiler that lowers it to an arithmetic circuit suitable for secret-sharing-based MPC, including depth-minimizing optimization passes. Third, it describes the cryptographic building blocks threshold and verifiable secret sharing, offline triple generation, and distributed key generation needed to instantiate the framework without a trusted dealer. Fourth, it specifies secure operators for equality, comparison, privacy-preserving record linkage, and aggregation, each with an explicit algorithm. Fifth, it derives the round and communication complexity of the resulting protocol, gives a composed security proof sketch, and quantifies the cost of upgrading from semi-honest to malicious security. Sixth, it develops an analytical cost model calibrated against complexity results reported for the SPDZ family of protocols and uses it to project latency and communication volume across federation sizes, cohort sizes, and network conditions, illustrated with four healthcare case studies. Seventh, it discusses the framework's alignment with HIPAA and GDPR and the governance model such a deployment would require.

The remainder of this research is organized as follows. Sections 2-3 review related work and define the system and threat model; Sections 4-6 present the query language, compiler, and cryptographic building blocks behind the secure operators; Section 7 derives complexity and security results, and Section 8 turns them into a cost model evaluated against four case studies in Section 9. The paper closes with regulatory considerations, limitations, deployment architecture, and a conclusion.

2. Literature Review

2.1 Federated Query Platforms in Healthcare

SHRINE couples an i2b2 clinical data repository at each site with a central Query Aggregator Interface that dispatches a structured count query to every site and returns the summed result [3]; per-site counts are visible to the network operator. OHDSI standardizes participating databases on the OMOP Common Data Model and runs pre-validated analysis packages locally, aggregating only the resulting statistics [4], spanning several hundred databases and hundreds of millions of records worldwide. PCORnet connects clinical and patient-powered research networks for comparative-effectiveness research without physically pooling data [5], and the FDA's Sentinel System applies the same model to safety surveillance across more than 300 million person-years of data [6]. All four are, by design, count-and-aggregate systems: none evaluates arbitrary joins or record-level linkage, and all trust the coordinating hub and per-site outputs not to be misused.

2.2 Cryptographic Foundations of Secure Multi-Party Computation

Yao introduced two-party secure computation and the garbled-circuit protocol underlying constant-round MPC [10]. Goldreich, Micali, and Wigderson generalized this to the multi-party, dishonest-minority setting (GMW) [17], and Ben-Or, Goldwasser, and Wigderson (BGW) gave an information-theoretically secure honest-majority protocol based on Shamir's polynomial secret sharing [18], [19]. Beaver's circuit-randomization technique precomputing random correlated triples so each online multiplication needs only local computation and two masked broadcasts remains the standard building block for practical arithmetic MPC [20]; the SPDZ family extends this to the dishonest-majority, actively secure setting with online communication linear in the number of parties [15]. Paillier's homomorphic cryptosystem [21] and modern FHE schemes for approximate arithmetic [22] offer a non-interactive alternative at substantially higher per-operation cost;

the research adopts interactive secret sharing because it composes more directly with the join and comparison operations a federated query requires.

2.3 Practical MPC Frameworks and Compilers

MP-SPDZ implements more than thirty MPC protocol variants behind a common interface and is the framework underlying the complexity figures calibrated in Section 8 [16]. ABY combines arithmetic, Boolean, and garbled-circuit sharing with efficient conversions [23]; ABY3 extends this to the honest-majority three-party setting for privacy-preserving machine learning [24]. Evans, Kolesnikov, and Rosulek survey the resulting design space [11]. A separate line of work automates translation from a high-level program to an MPC circuit: HyCC compiles ANSI C into hybrid protocols that automatically partition computation across representations to minimize cost [25]. Section 4's query compiler follows the same philosophy lowering a declarative specification to an efficient circuit specialized to federated SQL rather than general-purpose programs.

2.4 Verifiable and Proactive Secret Sharing

Shamir's original scheme [19] assumes an honest dealer and honestly reported shares; neither assumption holds against an active adversary. Feldman's verifiable secret sharing (VSS) adds public commitments to the sharing polynomial's coefficients so every recipient can locally verify its share without revealing the secret [26]; Pedersen's variant hides the secret information-theoretically at the cost of a computationally binding commitment [27]. Section 5.2 uses Feldman's construction. Because a federation is long-lived, shares should not remain static indefinitely: Herzberg et al. introduced proactive secret sharing, periodically re-randomizing shares so an adversary compromising different institutions over time gains nothing by combining views [28]. Gennaro, Jarecki, Krawczyk, and Rabin give a distributed key generation (DKG) protocol letting parties jointly compute a key pair with no single party learning it, correcting a bias attack in earlier constructions [29]; Section 5.6 adopts this protocol.

2.5 Privacy-Preserving Record Linkage

Establishing that two records at different institutions refer to the same patient without disclosing raw identifiers is a prerequisite for any cross-institutional join. Gkoulalas-Divanis, Vatsalan, Karapiperis, and Kantarcioglu survey privacy-preserving record linkage (PPRL) techniques spanning Bloom-filter encoding, secure multi-party protocols, and differentially private variants, characterizing the recurring accuracy/privacy/scalability trade-off [30]. Schnell, Bachteler, and Reiher introduced Bloom-filter encoding as a scalable approximate-matching primitive [31]; MainSEL couples such encodings with an MPC back end to compute similarity without reconstructing filters in the clear [32], and is the closest prior work to Section 6.3.

2.6 SMPC Deployments in Healthcare and Genomics

Beyond the query-federation systems of Section 2.1, VaultDB [12], HiGHmed's SIMON protocol [13], and star-topology EMR processing [14] SMPC has been applied to genome-wide association studies, which share the same tension between statistical power and data custody. Cho, Wu, and Berger demonstrate a secure GWAS pipeline scaling to tens of thousands of individuals [33], and Bonte et al. compute the significance of a χ^2 association statistic via SMPC without revealing the statistic itself [34]. These results establish SMPC as practical for computationally intensive biomedical workloads at meaningful scale.

2.7 Alternative Privacy Paradigms: Federated Learning and Differential Privacy

Federated learning trains a shared model by exchanging parameter updates instead of data; secure aggregation cryptographically hides each participant's update from the server while tolerating client dropout [9], and Rieke et al. survey the resulting opportunities for digital health [8]. Differential privacy instead answers queries directly but perturbs the result with calibrated noise bounding how much any single record can change the output [7]. Both are complementary to SMPC rather than substitutes: federated learning does not support ad hoc SQL-style queries, and differential privacy trades accuracy for its guarantee an accumulating cost under repeated querying whereas SMPC returns an exact result and pays its cost in rounds and communication, quantified in Sections 7 and 8.

2.8 Complementary Hardware- and Retrieval-Based Approaches

Two further lines of work address adjacent problems. Trusted execution environments (TEEs) such as Intel SGX provide hardware-enforced isolation so even a compromised OS cannot read enclave memory [35]; a TEE-based system would avoid the construction's interaction rounds entirely, at the cost of trusting the hardware vendor and remaining exposed to

side-channel and speculative-execution attacks a different trust model than the purely cryptographic privacy of Section 7.3. Private information retrieval (PIR) instead hides which record a query touches, not its value [36], addressing the access-pattern gap of Section 3.5; combining PIR with the construction of Sections 5-6, rather than substituting for it, is the natural way to close that gap.

Neither line of work is a drop-in substitute for the cryptographic guarantee of Sections 5-7. A TEE-based deployment shifts the trust assumption from information theory to a specific hardware vendor's attestation chain, which a federation's governance body may or may not consider acceptable depending on the sensitivity of the data involved and the institutions' existing vendor relationships; several of the enclave attacks cited above were demonstrated against exactly the kind of multi-tenant cloud deployment a resource-constrained hospital IT department might otherwise find attractive. PIR, meanwhile, solves a narrower problem than the one this paper addresses: it is naturally suited to single-database record retrieval, not the multi-party joins and aggregations of Section 6, so a practical integration would layer PIR's access-pattern hiding on top of, rather than instead of, the constructions already described.

Table 2. Positioning against representative prior approaches

Approach	Ref.	Raw data exposed?	Arbitrary joins?	Formal proof?
Centralized warehouse	-	Yes	Yes	No
Federated count hub	[3],[5],[6]	No (counts)	No	No
Federated analytics network	[4]	No (stats)	No	No
Federated learning + secure agg.	[8],[9]	No (gradients)	N/A	Partial
Differential-privacy auditing	[7]	Noised answer	Yes*	Yes
SMPC record linkage	[31],[32]	No	Linkage only	Yes
SMPC SQL query federation	[12]–[14]	No	Yes (fixed top.)	Yes
This work	-	No	Yes (n-party)	Yes

Differential privacy supports arbitrary aggregate queries, but per-query accuracy decays as the cumulative privacy budget is spent; deployment scale is discussed in Sections 1-2: SHRINE/OHDSI/PCORnet/Sentinel operate nationally/internationally, SMPC query systems remain at the pilot stage, and this work is evaluated analytically.

3. Research Methodology

3.1 Entities, Network, and Deployment Model

The framework involves n healthcare institutions P_1, …, P_n, each holding a local relational database D_i over a schema agreed in advance across the federation (e.g., an OMOP-CDM-compatible schema, consistent with the OHDSI and PCORnet deployments of Section 2.1); a query coordinator that orchestrates protocol rounds but holds no patient data of its own and may itself be one of the P_i; and a requesting analyst R who receives only the final, disclosure-controlled query answer. The research assumes the primary case of horizontally partitioned data each institution holds disjoint patients under the same schema consistent with the SHRINE and OHDSI deployment model; vertically partitioned or schema-heterogeneous federations are discussed as future work in Section 11.3. Institutions communicate over pairwise-authenticated channels (e.g., mutually authenticated TLS) in a partially synchronous network: messages are eventually delivered, and the round-based complexity of Section 7 assumes a bound on per-round delay consistent with the wide-area round-trip times used in Section 8.

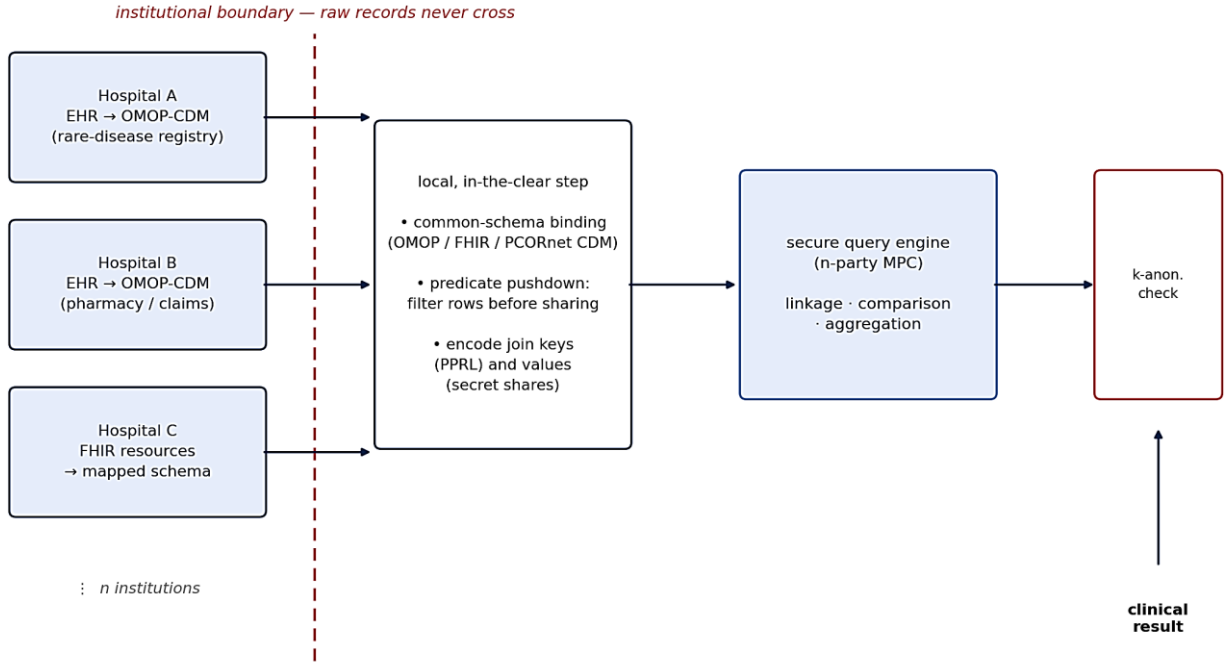

Figure 1. Data flow across the federation. Each hospital maps its local records to a common schema and filters or encodes them before anything leaves the institutional boundary; only secret shares and PPRL encodings reach the secure query engine, and the clinical result is released only after the disclosure-control check.

3.2 Adversary Models

The research considers three standard adversary models, summarized in Table 3, and uses the weakest that is adequate for a given deployment's trust relationships. A semi-honest (honest-but-curious) adversary follows the protocol specification exactly but attempts to infer additional information from the messages it receives; this is the default model for Sections 4-6 and the baseline cost model of Section 8. A covert adversary may deviate arbitrarily from the protocol but is deterred by a fixed probability of being caught if it does a realistic model for institutions with reputational or regulatory exposure that would follow from detected misbehavior. A malicious adversary may deviate arbitrarily with no such deterrent and must be cryptographically prevented, not merely detected probabilistically; Section 7.5 describes the upgrade path to this model using SPDZ-style information-theoretic message authentication codes. Across all three models, the research bounds the number of corrupted institutions by a threshold t; the default semi-honest construction of Sections 5-6 requires an honest majority, $t < n/2$, while the malicious-secure upgrade tolerates a dishonest majority, $t \leq n-1$, at the communication cost quantified in Section 8.5. Fig. 2 plots both tolerance curves as a function of federation size: the gap between them widens linearly with n, which is the quantitative reason Section 7.6 recommends matching the security model to a federation's actual, and not merely nominal, trust relationships rather than defaulting to the strongest available model regardless of size.

Table 3. Adversary models supported by the framework

Model	Deviation	Corruption bound	Mechanism	Rel. cost
Semi-honest	None (follows protocol)	$t < n/2$	Shamir/BGW (§5)	1× (baseline)
Covert	Arbitrary, deterred	$t < n/2$	+ spot-checks	≈1.2–1.5×
Malicious	Arbitrary	$t \leq n-1$	SPDZ MACs (§7.5)	≈2×

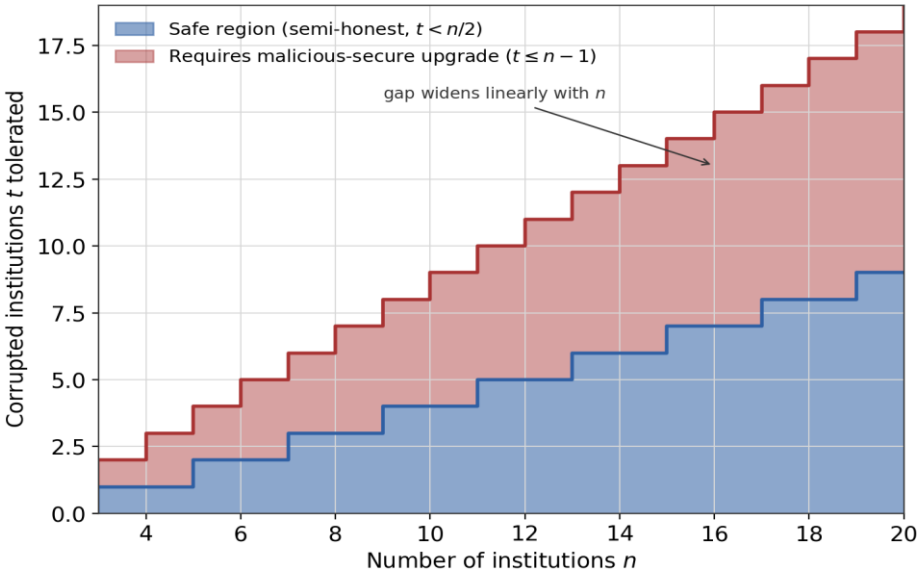

Figure 2. Corrupted-institution tolerance under the semi-honest vs. malicious-secure constructions, as a function of federation size.

3.3 Security Definitions

Security is defined in the standard real/ideal simulation paradigm [10], [11]: a protocol π securely computes a function f if, for every adversary A corrupting at most t parties in the real execution of π, there exists a simulator S, given only the corrupted parties' inputs and outputs under the ideal functionality F^f, whose output is computationally (or, for the information-theoretic constructions used here, statistically) indistinguishable from A's view of the real execution. Section 7.3 applies this definition to the core multiplication sub-protocol, and Section 7.4 extends it by composition to a full compiled query circuit.

3.4 Trust Assumptions and Setup Requirements

Before any query executes, the federation must complete a one-time (or periodically refreshed) setup phase: participating institutions agree on a common schema and a field $\mathbb{Z}_q$ of sufficient size; a threshold key pair is established via the distributed key generation protocol of Section 5.6, so that no single institution, including the coordinator, ever holds a private key capable of unilaterally decrypting or forging authenticated shares; and an initial batch of Beaver triples is generated via the offline protocol of Section 5.5. None of these steps requires a trusted third party, the guarantee is that no coalition of fewer than the relevant threshold can bias the setup, but they do require every institution to run the corresponding software, which the research treats as a governance and operational cost distinct from the cryptographic guarantees analyzed in Section 7.

3.5 Scope and Explicit Non-Goals

The construction hides the value of every intermediate computation and every input record from every party other than the institution that holds it. It does not hide: the fact that a query was executed; the structure of the query itself (which tables and predicates it references), unless additionally combined with private information retrieval, which the research treats as out of scope; or the final result's approximate cardinality, which is instead bounded below by the disclosure-control threshold k defined in Section 6.5. Denial of service through an institution's non-participation is likewise out of scope. None of these represent a regression relative to SHRINE, OHDSI, PCORnet, or Sentinel, all of which already disclose query structure and per-site participation to the coordinating hub.

3.6 Notation Summary

Table 4 collects the notation introduced across Sections 3-8 for reference; each symbol is defined at its first point of use in the corresponding section.

Table 4. Notation used throughout Sections 3-8

Symbol	Meaning	§
n, t	institutions; corruption threshold	III
P_i, D_i	institution i; its local database	III-A
q, F_q	field modulus; finite field of order q	V-A
f(z), t	Shamir sharing polynomial; its degree	V-A
g, C_j	VSS public generator; coefficient commitment	V-B
[[x]], x_i	secret-shared value; institution i's share	V-D
a, b, c	Beaver triple (c = ab)	V-E
d, e	opened Beaver masks (x−a, y−b)	V-E
ℓ	field/comparison bit-width	VI-B
τ, Dice(·,·)	PPRL match threshold; similarity score	VI-C
k	disclosure (cell-size) threshold	VI-E
d(Q), M(Q)	circuit depth; multiplication-gate count	VII-A
α, m_x	global MAC key; MAC on x	VII-E
R, B, RTT	records/site; bandwidth; round-trip time	VIII-A

3.7 Architecture Overview

Figure 3 shows the four stages the constructions of Sections 5-6 realize. Each institution first evaluates any purely local predicates against its own database in the clear and encodes the query-relevant attributes join keys via PPRL, and numeric or categorical values via secret sharing without transmitting any raw record; a secure query engine realized as an n-party MPC protocol among the participating institutions then executes the linkage, comparison, and aggregation operations the compiled query requires; and an output gate applies a minimum cell-size (k-anonymity) threshold before the final result is reconstructed and released to the requesting analyst. No raw patient-level value crosses an institutional boundary at any stage; only secret shares, PPRL encodings, and the final, threshold-checked result do. Section 12 maps each of these four stages onto a concrete software component.

4. Federated Query Language and Compilation Pipeline

4.1 Design Goals

A federated query language for this setting must satisfy three goals that pull in different directions. It must be expressive enough to state the cohort-discovery, safety-surveillance, and comparative-effectiveness queries of Section 9 selection, cross-institutional join, and aggregation without requiring the analyst to reason about circuits or secret sharing. It must be restrictive enough that every expressible query compiles to a circuit of bounded, predictable depth, since Section 7 shows depth is the dominant driver of interactive latency. And it must let the compiler distinguish predicates that are locally evaluable in the clear from those that require secure evaluation, since this distinction is the single largest lever on cost. The research therefore defines a small declarative subset of SQL rather than adopting general-purpose SQL wholesale; unsupported constructs (nested subqueries, window functions, user-defined functions) are rejected at compile time rather than silently mishandled.

Formal problem statement. Given n institutions with local relations $D_1, \ldots, D_n$ over a common schema and a query Q in the grammar of Section 4.2, the framework computes $f_Q(D_1, \ldots, D_n)$ a scalar or small tuple of aggregates such that (i) correctness: the released value equals f_Q over the true union under the LINK ON join semantics; (ii) privacy: for every institution and coalition of at most t others, the coalition's view is simulatable from its own inputs and the output alone; and (iii) disclosure control: the release is suppressed if contributing records fall below threshold k. Sections 5-7 construct a protocol meeting all three for Section 6's operators, and Section 4.3-4.4 show every grammar-expressible query compiles to a circuit built exclusively from them.

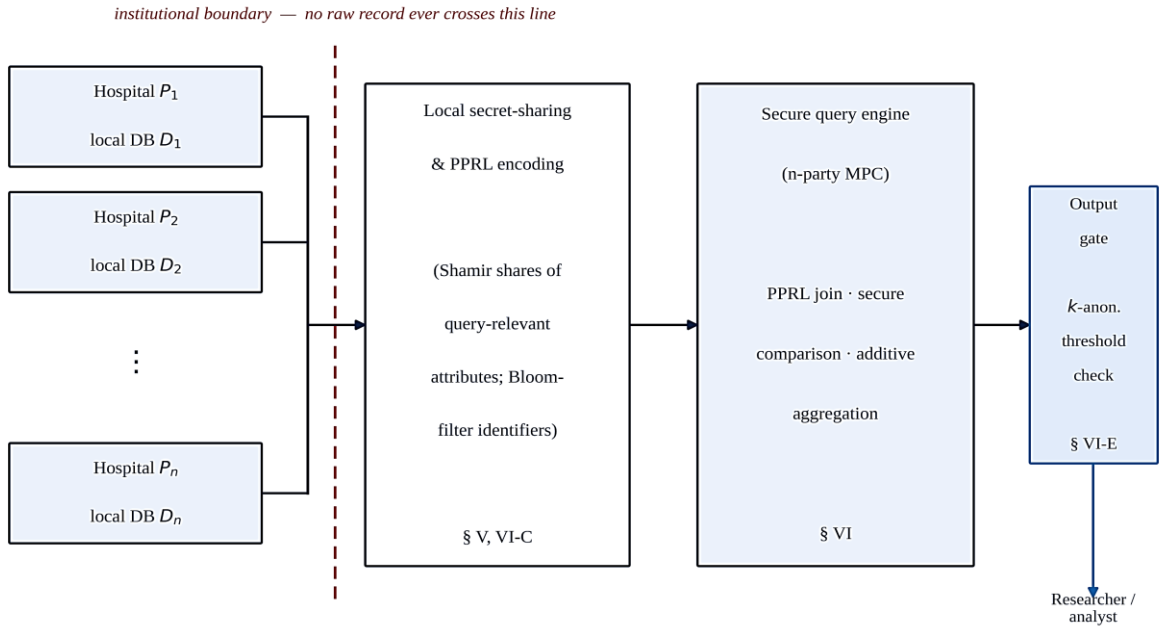

Figure 3. Framework architecture. Institutions retain custody of their local databases; only secret shares and PPRL encodings cross the institutional boundary, and the final result is released only after a disclosure-control check.

4.2 Query Grammar

The complete grammar is given below in EBNF; nonterminal not otherwise defined follow standard SQL lexical conventions (identifiers, numeric and string literals).

```
query      ::= SELECT agg-list FROM source-list
               [LINK ON attr-list]
               [WHERE pred] [HAVING pred]
agg-list   ::= agg (',' agg)*
agg        ::= agg-fn '(' [attr] ')' [AS ident]
agg-fn     ::= COUNT | SUM | AVG | VAR
source-list::= inst-ref (',' inst-ref)*
inst-ref   ::= ident [AS ident]
attr-list  ::= attr (',' attr)*
attr       ::= [ident '.'] ident
pred       ::= atom | pred AND pred | pred OR pred
             | '(' pred ')' | NOT pred
atom       ::= attr cmp-op operand
operand    ::= attr | literal
cmp-op     ::= '=' | '<>' | '<' | '<=' | '>' | '>='
literal    ::= number | string
```

Unsupported SQL constructs nested subqueries, window functions, user-defined functions, and outer joins are rejected at parse time. The bounded-depth guarantee this grammar exists to enforce can be stated as a lemma over the compilation passes of Section 4.3-4.4.

Lemma 1 (depth-boundedness). Let Q be any query accepted by the grammar above, and let α(Q) denote the number of atom productions and β(Q) the number of pred productions reachable in Q's parse tree. Then the circuit C(Q) emitted by the lowering pass has multiplicative depth

$$d(C(Q)) \leq \alpha(Q) \cdot d_{\text{cmp}} + \lceil \log_2 \beta(Q) \rceil + 1 \quad (1)$$

Where d_cmp = O(log ℓ) is the per-atom comparison depth of a comparison, or a single round for a LINK ON atom, and the ⌈log₂ β(Q)⌉ term is the depth-minimized combination cost of Section 6.5. Proof sketch: every atom lowers to exactly one SecureCompare or SecureLink sub-circuit (Table 6), each of bounded depth by construction (Algorithms 2–3); the balanced-tree combination pass combines β(Q) atom results in ⌈log₂ β(Q)⌉ rounds rather than β(Q)−1; and the aggregation stage adds at most 1 further round (VARIANCE, Table 6) or 0 (SUM/COUNT). Summing these three contributions gives the bound. Because α(Q) and β(Q) are determined entirely by Q's syntax, the grammar has no recursion through WHERE or HAVING beyond finite conjunctions/disjunctions of atoms d(Q) is computable statically, at parse time, before any secret sharing occurs; this is what lets Section 8's cost model (Eq. 26) be applied prospectively to a proposed query rather than only retrospectively to one already executed.

4.3 Compilation to Arithmetic Circuits

Compilation proceeds in three passes. The binding pass resolves attribute references against the schema and classifies each predicate atom as local (columns/constants already known to the row's institution) or joint (depends on a prior secure computation or spans LINK ON). The lowering pass emits one circuit operator per joint atom SecureEquals, SecureCompare, SecureLink, or a linear aggregation node composed per the query's Boolean structure via Section 6.5's secure-AND/OR identities. The scheduling pass assigns each operator a depth level, batching data-independent operators into the same round, and appends the output-gate check of Section 6.5.

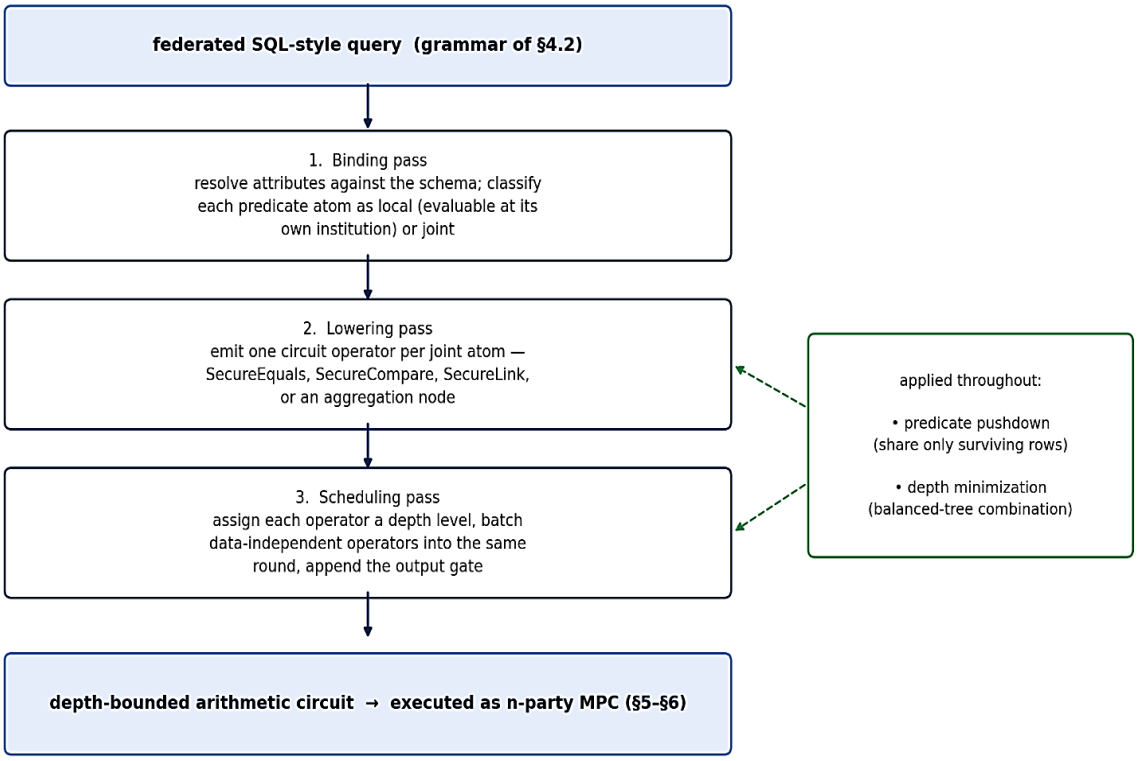

Figure 4. The three-pass query compilation pipeline. Predicate pushdown and depth minimization (Section 4.4) apply throughout the lowering and scheduling passes, keeping the compiled circuit's multiplicative depth as low as its query class allows.

4.4 Optimization Passes

Two optimizations dominate the resulting cost. Predicate pushdown evaluates local atoms at their owning institution before any sharing occurs, so only rows surviving the local filter are ever shared why Q1 in Table 5 has zero secure gates despite scanning a full local table. Depth minimization balances commutative/associative sub-expressions into a tree rather than a

left-deep chain: m independent comparisons cost $\lceil \log_2 m \rceil$ rounds rather than m−1. Both passes are purely structural rewrites that do not change the computed function, so Section 7.4's security argument applies unchanged to the optimized circuit.

4.5 Worked Compilation Example

Consider the query "SELECT COUNT(*) FROM A, B LINK ON patient_id WHERE diagnosis = 'X' AND age > 65," a simplified version of the linkage query underlying the case study of Section 9.1. The binding pass classifies diagnosis = 'X' as local (evaluable at each site before sharing) and age > 65 as joint only insofar as the compiled circuit must still secret-share the surviving rows' ages before comparing them, since the comparison result itself must remain hidden until aggregation. Fig. 5 shows the resulting compiled circuit: local filtering and Bloom-filter encoding happen in parallel at each site; a share-distribution round follows; the PPRL match and the age comparison proceed in parallel, each occupying its own depth budget; their results are combined with a secure AND; and the matched, qualifying count is aggregated by free addition before the output gate. The circuit's total depth is 6, the maximum of the PPRL branch (depth 1) and the comparison branch (depth 5), plus 1 for the final and matching the Q3 parameters of Table 5 and Section 8's evaluation.

4.6 Relationship to Existing Health Data Standards

The schema-agreement assumption of Section 3.1 is not a new requirement the framework introduces; it is already satisfied, for the horizontally partitioned case, by the common data models underlying the systems of Section 2.1. An institution already mapping its electronic health record system to the OMOP Common Data Model for OHDSI participation [4] or to the PCORnet Common Data Model [5] can reuse that same mapping as the schema against which Section 4's grammar is bound, so that adopting the framework does not require a separate data-modeling effort on top of a federation's existing standards work. HL7 FHIR resources, where an institution's operational systems expose them, provide a natural source for the local plaintext filtering step of Section 4.4, since FHIR's resource-level access control already distinguishes exactly the kind of locally-evaluable predicate the compiler needs to identify; mapping FHIR search parameters to the attr productions of Section 4.2's grammar is a straightforward, schema-specific configuration step rather than a change to the compiler itself. The research does not propose a new common data model the framework is deliberately agnostic to which one a federation already uses, so long as every participating institution maps to the same one. Table 5 summarizes three representative query classes, referenced throughout Sections 4-9 as running examples of the depth-dominated cost structure Section 7 derives.

Table 5. Representative query classes and their compiled circuit cost.

Class	Description	Depth d(Q)	Mult. gates / record
Q1	Local-predicate aggregate (COUNT/SUM over a locally evaluable filter)	0	0
Q2	Secure threshold aggregate (range predicate vs. a jointly computed secret threshold)	5	$\ell = 32$
Q3	Cross-institutional PPRL join + aggregate	6	$2\ell = 64$

Depth assumes $\ell = 32$-bit fixed-point field elements and the $O(\log \ell)$ comparison protocol of Section 6.2; Q3's per-record gate count reflects one Bloom-filter AND plus one Dice-threshold comparison, each $\approx \ell$ gates.

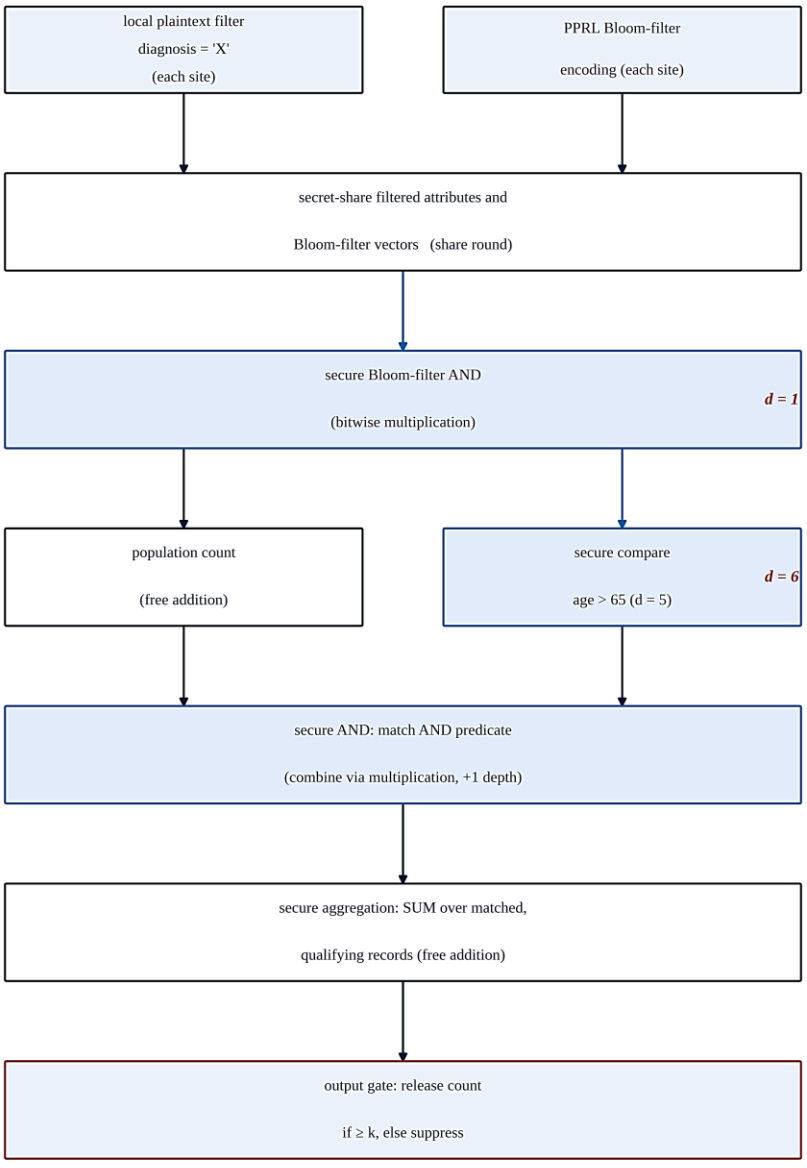

Figure 5. Compiled circuit for the worked example query of Section 4.5. Depth is annotated on the right; population count and secure aggregation are free (linear) operations that do not add to circuit depth.

5. Cryptographic Building Blocks

5.1 Shamir Secret Sharing and Resharing

Each query-relevant local value x is distributed using Shamir's (t+1, n) threshold scheme [19]: the holder chooses a random polynomial of degree t over a finite field F_q with constant term x,

$$f(z) = x + a_1 z + a_2 z^2 + \cdots + a_{t-1} z^{t-1} \pmod{q} \tag{2}$$

and sends $\text{share}_i = f(i)$ to institution i:

$$\text{share}_i = f(i), \quad i = 1, \ldots, n \tag{3}$$

Any t+1 shares reconstruct x by Lagrange interpolation at zero, while any t or fewer shares are information-theoretically independent of x:

$$x = f(0) = \sum_{i \in T} \lambda_i f(i), \quad \lambda_i = \prod_{j \in T \setminus \{i\}} \frac{j}{j - i} \tag{4}$$

Because a federation is long-lived, shares should be periodically refreshed without changing the underlying secret, following the proactive secret sharing approach of Herzberg et al. [28]: each holder of share f(i) generates a fresh degree-t polynomial δ with δ(0) = 0, distributes δ(j) to institution j, and every institution sums the increments it receives into its existing share,

$$f'(z) = f(z) + \delta(z), \quad \delta(0) = 0 \tag{5}$$

Which yields a new, statistically independent sharing of the same secret, so that an adversary who separately compromises different institutions in different refresh periods cannot combine the resulting partial views. Concrete numeric example (t = 1, n = 3, q = 17). To make Eqs. 2–3 tangible, suppose three institutions share the value x = 6 with a degree-1 (t = 1) polynomial over F_{17}, tolerating one corrupted party. The dealer picks a random coefficient $a_1 = 4$, giving f(z) = 6 + 4z (mod 17). Evaluating at z = 1, 2, 3 yields the three shares:

```
f(1) = 6 + 4(1) = 10  (mod 17)  -> P1
f(2) = 6 + 4(2) = 14  (mod 17)  -> P2
f(3) = 6 + 4(3) = 18 = 1 (mod 17) -> P3
```

Any two shares reconstruct x = 6 via Eq. 4; using P1 and P2's shares (10, 14) at z = 1, 2, the Lagrange weights are $\lambda_1 = 2/(2-1) = 2$ and $\lambda_2 = 1/(1-2) = -16 \equiv 1 \pmod{17}$, giving $x = 2 \cdot 10 + 1 \cdot 14 = 20+14 = 34 \equiv 6 \pmod{17}$, as expected. Any single share alone (e.g., P3's share, 1) is consistent with every possible secret in F_{17} under some choice of a_1, which is the concrete instantiation of the information-theoretic privacy claimed following Eq. 4.

Fig. 6 shows the same example geometrically, over the reals for clarity rather than modulo 17. Panel (a) plots the sharing polynomial f(z) = 6 + 4z: the three shares are points on this line, and the secret is the (hidden) y-intercept f(0) = 6 that any two of them jointly pin down via the line they determine. Panel (b) shows why a single share reveals nothing: infinitely many lines of the required degree pass through one point, each implying a different, equally plausible value of f(0) the geometric picture of the privacy argument formalized for the general case in Eq. 21 and Section 7.3.

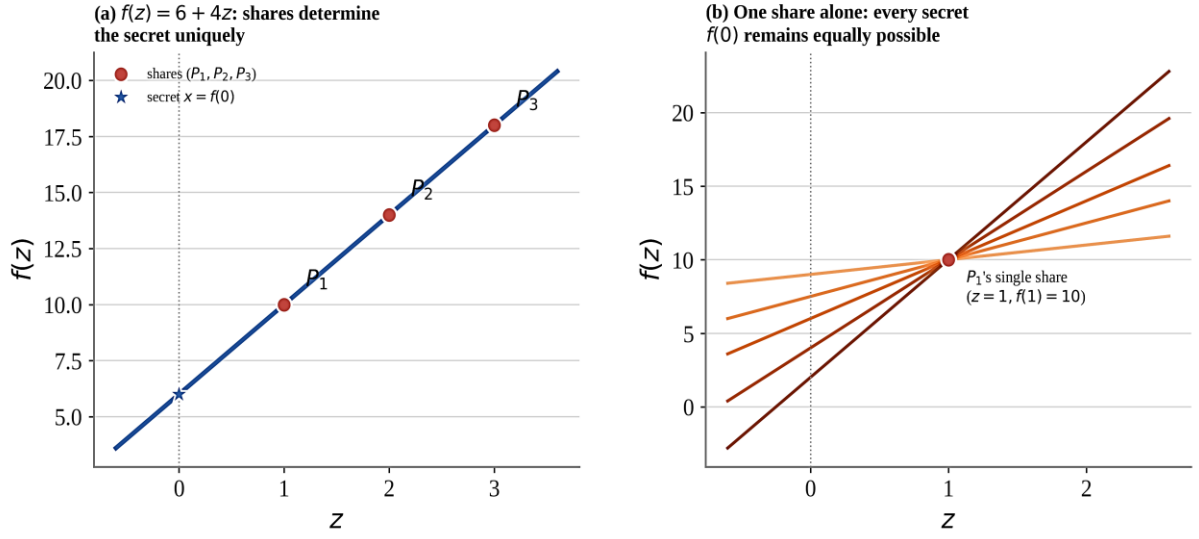

Figure 6. Shamir sharing of the running example x = 6 (§5.1). (a) Two or more shares reconstruct the line, and hence the secret, uniquely. (b) One share alone is consistent with every possible secret.

5.2 Verifiable Secret Sharing

The scheme of Section 5.1 assumes the dealer distributing shares is honest; under the covert or malicious adversary models of Section 3.2 this cannot be assumed. Feldman's VSS [26] makes the dealer accountable by publishing, alongside the

shares themselves, a commitment to each coefficient of f under a public generator g of a group in which discrete log is hard,

$$C_j = g^{a_j} \pmod{p}, \quad j = 0, \ldots, t \tag{6}$$

Every recipient of share$_i$ = f(i) can then verify it against the public commitments without learning any other party's share or the secret itself,

$$g^{f(i)} = \prod_{j=0}^{t} (C_j)^{i^j} \pmod{p} \tag{7}$$

A share that fails this check is provably inconsistent with the dealer's declared polynomial, and the recipient broadcasts a complaint rather than silently accepting corrupted input the mechanism the covert and malicious models of Section 3.2 rely on to detect (respectively, prevent) a cheating dealer. Pedersen's variant [27] achieves the same accountability while keeping the secret itself information-theoretically hidden even from a computationally unbounded adversary, at the cost of an additional random blinding factor per share; the research uses Feldman's simpler construction where only computational hiding is required and Pedersen's where a stronger guarantee is warranted, such as the key-generation setting of Section 5.6.

5.3 Comparison of Secret-Sharing Representations

Sections 5.1-5.2 and 5.4 below use two different secret-sharing representations for different purposes, and it is worth stating explicitly why neither alone suffices. Table 6 compares the two against a third alternative, replicated secret sharing, used by three-party frameworks such as ABY3 [24] but not adopted here because its efficiency gains are specific to the three-party setting and do not generalize to the arbitrary federation sizes Section 8 evaluates.

Table 6. Secret-sharing representations used across Sections 5-7

Scheme	Reconstruction threshold	Used for	Why
Shamir (t+1,n)	any t+1 of n	Initial distribution (§5.1)	tolerates n−t−1 offline parties
Additive (n,n)	all n shares	Online arithmetic (§5.4-6)	addition is free; matches Beaver
Replicated	varies by design	Not used (cf. [24])	gains specific to 3-party case

5.4 Additive Sharing and Share Conversion

Online-phase values are held as additive secret sharings over $\mathbb{Z}$_q,

$$x = \left(\sum_{i=1}^{n} x_i \right) \pmod{q}, \quad x_i \text{ random in } \mathbb{Z}_q \tag{8}$$

Under which addition and multiplication-by-a-known-constant are local, communication-free operations, the property Section 6.3 relies on for population counts and Section 6.4 relies on for aggregation. A Shamir sharing is converted to this representation, when needed, by a one-round local re-randomization that is standard in the BGW/SPDZ literature and does not affect the security argument of Section 7.

5.5 Beaver Triple Generation: The Offline Phase

Multiplying two secret-shared values needed for the Bloom-filter AND of Section 6.3 and internally within the comparison protocol of Section 6.2 requires interaction. Following Beaver [20], the parties pre-compute, independent of any specific query, a random multiplication triple

$$c = (a \cdot b) \pmod{q} \tag{9}$$

Shared identically to any other secret value. This precomputation is the “offline phase” of the SPDZ family of protocols and is the only part of the framework whose cost depends on the underlying cryptographic assumption rather than pure information theory: it is instantiated via a somewhat-homomorphic encryption scheme in the original SPDZ construction

or via oblivious transfer extension in later, faster variants implemented in MP-SPDZ. Because triple generation is independent of the query being evaluated, it can be run continuously in the background between queries, so that the online latency figures of Section 8 reflect only the interactive multiplication protocol of Section 6.2-6.3, not triple generation itself consistent with how SPDZ's own complexity claims are typically stated.

Given a precomputed triple, the online multiplication protocol proceeds as follows. Each party locally forms and publicly opens (broadcasts its share of) the two masked differences

$$d = x - a, \qquad e = y - b \quad \text{(opened)} \tag{10}$$

Because a and b are freshly random and unknown to any single party, d and e are individually uniform over $\mathbb{Z}_q$ and reveal nothing about x or y (Section 7.3 makes this precise). Writing x = d+a and y = e+b and expanding,

$$xy = de + d \cdot b + e \cdot a + c \tag{11}$$

Every term on the right is either a public constant (de) or a public scalar times a value each party already holds a share of, so the result is reconstructed purely by local computation on shares:

$$[\![xy]\!] = de + d[\![b]\!] + e[\![a]\!] + [\![c]\!] \tag{12}$$

Fig. 7 lays out the full protocol as a message-sequence diagram across three parties: the masking step (Eq. 10) happens locally and in parallel at every party with no communication, the broadcast step is the protocol's single interaction round, and the reconstruction step (Eq. 12) is again local and parallel. This is the only step in the framework that requires interaction proportional to the number of multiplications in the compiled query; Section 7.1 makes the resulting round and communication complexity explicit.

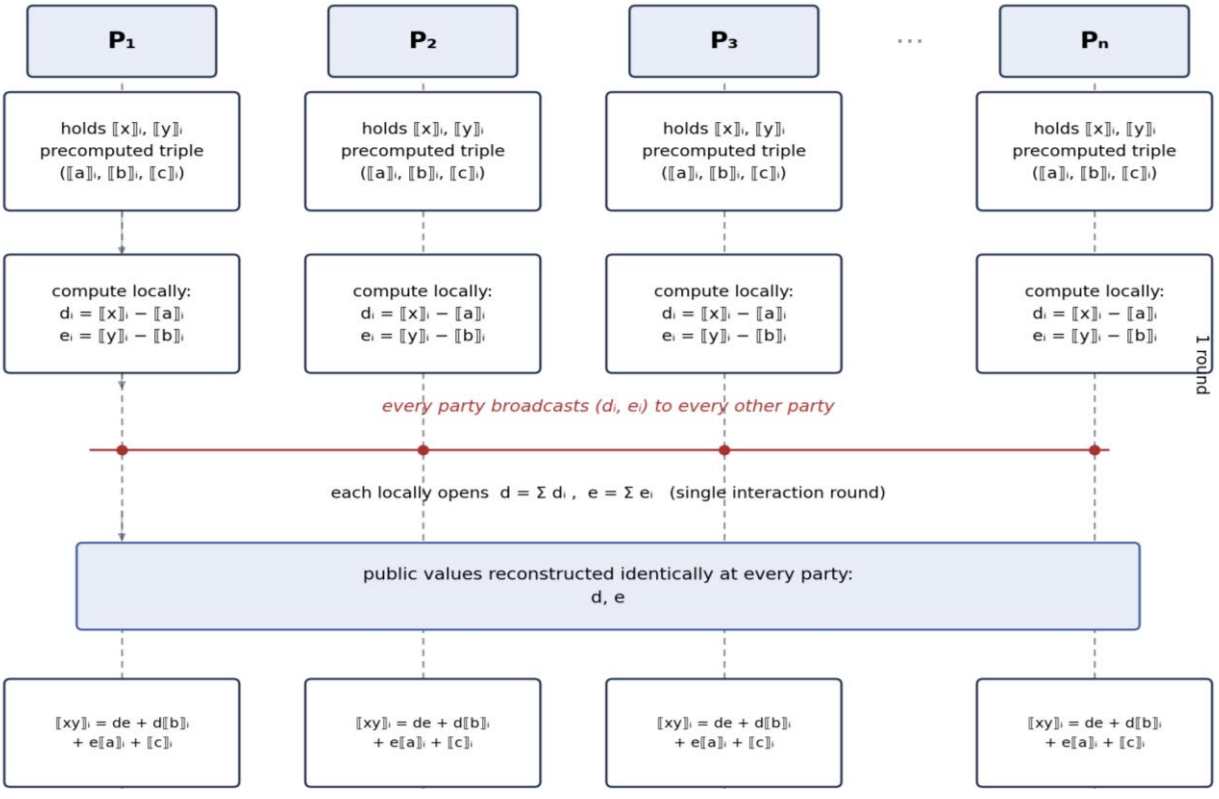

Figure 7. Message-sequence diagram for the Beaver multiplication sub-protocol (§5.5). Masking and reconstruction are local; the broadcast of d and e is the protocol's one interaction round.

5.6 Distributed Key Generation for Threshold Setup

Where the framework's setup or its malicious-security upgrade requires a shared public/private key pair, no single institution including the coordinator should ever hold the private key outright. Gennaro et al.'s DKG protocol [29] achieves this, summarized in Algorithm 1: each institution runs an independent instance of Pedersen VSS [27] to share a locally chosen random value, every institution locally sums the shares it receives from all n instances into a share of the joint secret x (structurally identical to the summation of Eq. 8), and the corresponding public key is published as

$$y = g^{x} = \prod_{i=1}^{n} g^{x_i} \pmod{p} \tag{13}$$

Algorithm 1 *DistributedKeyGen(n, t)*

Output: public key y; each P_i holds share x_i
of the joint private key x (unknown to all)
1: each P_i picks random z_i, runs Pedersen VSS
to share z_i as a degree-t polynomial among
all n institutions
2: each P_j verifies every received share against
its public commitments; broadcasts a complaint
for any P_i that fails verification
3: institutions failing >t complaints are excluded
from the 'qualified set' Z; repeat with Z if
any exclusions occurred
4: each P_j sets x_j = Σ_{i∈Z} (share of z_i from P_i)
5: y ← ∏_{i∈Z} g^{z_i,0} // public commitments
6: return y; each P_j retains x_j

Gennaro et al. show that a naïve version of this protocol, in which a party that fails Feldman verification is simply excluded, allows a small coalition to bias the distribution of the resulting x; their corrected protocol adds the complaint-and-reveal sub-round shown as steps 2–3 above, which restores uniformity even under active cheating [29], and is the version adopted for the setup of Section 3.4.

6. Secure Query Operators

6.1 Secure Equality Testing

The simplest joint predicate is equality between two secret-shared values, used both directly (e.g., diagnosis code equality when the code is itself the output of a prior secure step) and as a building block of the comparison protocol of Section 6.2. Equality reduces to a single secure multiplication once the difference is bounded: for values known to lie in {0,1} (as Bloom-filter bits, Section 6.3, always do), equality of two bits a and b is exactly $a \cdot b + (1-a)(1-b)$, computable with one Beaver multiplication.

6.2 Secure Comparison

WHERE-clause predicates involving an inequality age > 65, HbA1c above a jointly computed cohort average cannot be evaluated as a single multiplication and instead require a dedicated secure-comparison sub-protocol. The research adopts the bit-decomposition-free construction of Nishide and Ohta [37], which evaluates a secure less-than predicate over ℓ-bit field elements in $O(\log \ell)$ communication rounds rather than the $O(\ell)$ rounds required by naive bitwise comparison, at the cost of $O(\ell)$ Beaver multiplications per comparison. Algorithm 2 summarizes the protocol at the level of detail the complexity analysis of Section 7.1 requires.

Algorithm 2 *SecureCompare([[x]], [[y]], ℓ)*

Input: shares [[x]], [[y]] of ℓ-bit field elements
Output: share [[b]] with b = 1 iff x < y
1: [[z]] ← [[x]] − [[y]] + 2^ℓ // shift positive

2: parties jointly mask z with random r,
open c = z + r (r secret-shared bitwise)
3: recover the top bit of c ⊕ (top bit of r)
via O(log ℓ) rounds of carry propagation
using precomputed bitwise shares of r
4: [[b]] ← recovered top bit (= 1 iff x < y)
5: return [[b]]

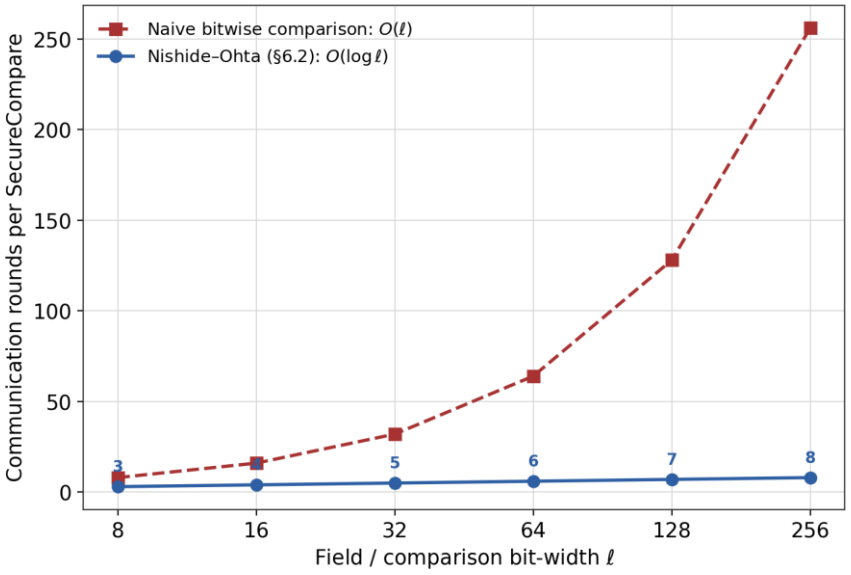

Figure 8. Communication rounds per SecureCompare (Algorithm 2) as a function of bit-width ℓ, vs. naive bitwise comparison.

The gap is decisive at the field sizes Table 5 uses: at ℓ = 32, Nishide–Ohta needs 5 rounds against 32 for the naive protocol, and the advantage only widens as ℓ grows. Independent comparisons over different records occupy the same circuit depth, so this round count is incurred once per query regardless of how many records are filtered in parallel a batching property Section 8.1 relies on directly.

6.3 Privacy-Preserving Record Linkage Protocol

A query such as "count patients treated at both Hospital A and Hospital B for condition X" requires resolving patient identity across institutions before any aggregation can occur. Following the Bloom-filter PPRL approach of Schnell et al. [31], each institution encodes its patients' quasi-identifiers (e.g., name q-grams, date of birth) into fixed-length Bloom filters using a keyed hash agreed across the federation. Rather than exchanging these filters in the clear which is itself subject to known cryptanalytic attacks on Bloom-filter encodings [30] the filters are secret-shared and their similarity is computed inside the same MPC engine used for the rest of the query, following the secure-linkage design of MainSEL [32]. Record-pair similarity is quantified by the Dice coefficient over the two Bloom filters A and B,

$$\mathrm{Dice}(A,B) = \frac{2\,|A \cap B|}{|A| + |B|} \tag{14}$$

The Bloom-filter length ℓ is itself a parameter to be chosen, trading matching accuracy against the per-comparison cost of Table 6. For a filter of m = ℓ bits populated by hashing each of a record's n constituent q-grams with k independent hash functions, the standard analysis of Bloom filters gives the probability that an unrelated pair collides on all k positions purely by chance a false-positive match and the filter-length-optimal choice of k for a fixed m, n:

$$p_{\mathrm{fp}}(m,k,n) = (1-e^{-kn/m})^k, \qquad k^* = \tfrac{m}{n}\ln 2 \tag{15}$$

A federation fixing a target false-positive rate p_0 (e.g., driven by the acceptable false-linkage rate identified in Section 11.2) therefore solves Eq. 15 for the minimum m satisfying $p_{fp}(m, k^*, n) \le p_0$ at the optimal k^*, which is why ℓ in Table 5 and Table 6 is a deployment-time constant fixed once during setup rather than something Section 7's complexity results need to treat as a free variable: it is pinned by the accuracy target before any query is compiled.

Algorithm 3 *SecureLink([[A]], [[B]], τ)*

Input: shares of ℓ-bit Bloom filters [[A]], [[B]],
match threshold τ
Output: share [[m]] with m = 1 iff Dice(A,B) ≥ τ
1: for k = 1..ℓ in parallel:
[[c_k]] ← [[A_k]]·[[B_k]] // 1 Beaver mult. each
(all ℓ done in a single round, depth 1)
2: [[I]] ← Σ_k [[c_k]] // free addition
3: [[S]] ← |A| + |B| // local constants
4: [[b]] ← SecureCompare(τ·[[S]], 2·[[I]], ℓ')
5: return [[b]]

A pair is declared a match once its secret-shared Dice score exceeds the agreed threshold; the resulting secret-shared match bits drive the join without either institution learning which of its patients matched, or with which counterpart record.

6.4 Secure Aggregation Operators

A federated query's aggregation stage computes SUM, COUNT, AVERAGE, or VARIANCE as a function of the (possibly match- and predicate-gated) per-record contributions. SUM and COUNT are linear combinations of secret-shared values and are therefore free under additive sharing, contributing no circuit depth beyond whatever gated the records being summed. Algorithm 4 gives the general aggregation procedure, including the variance case.

Algorithm 4 *SecureAggregate([[g_i]], [[x_i]], op)*

Input: gate bits [[g_i]] (match ∧ predicate), values
[[x_i]] for records i = 1..R; op ∈ {SUM,
COUNT, AVG, VAR}
Output: share of the requested aggregate
1: [[y_i]] ← [[g_i]]·[[x_i]] // 1 mult./record, depth 1
(skip this line entirely if op = COUNT)
2: [[S]] ← Σ_i [[y_i]] // free addition
3: [[N]] ← Σ_i [[g_i]] // free addition
4: if op = SUM or COUNT: return [[S]] or [[N]]
5: if op = AVG: return [[S]] / [[N]] (Newton–
Raphson secure division, O(1) extra rounds)
6: if op = VAR: [[z_i]] ← [[g_i]]·[[x_i]]²
(1 more mult./record, depth 1 additional)
return Σ[[z_i]]/[[N]] − ([[S]]/[[N]])²

AVERAGE requires one additional secure division by the (secret-shared) count, which the research implements via Newton–Raphson iteration over secret-shared values, a standard technique in the fixed-point MPC literature that adds a small constant number of multiplication rounds. VARIANCE reduces to sums of the record values and their squares,

$$\mathrm{Var}(X) = \frac{1}{N}\sum_{i=1}^{N} x_i^2 - \bar{x}^2 \tag{16}$$

So that a variance query requires exactly one additional multiplication round to compute each record's x_i^2 beyond whatever a corresponding SUM query would require, since Σx_i^2 and Σx_i are both free once the squares are formed.

6.5 Assembling Joins and Multi-Predicate Queries

A query's full WHERE clause is a Boolean combination of atoms, each evaluated by Section 6.1, 6.2, or VI-C. Conjunction (AND) of two secret-shared 0/1 results is a single Beaver multiplication; disjunction (OR) follows from $a \vee b = a+b-ab$, also one multiplication. Following the depth-minimization pass of Section 4.4, a conjunction of m independent atoms is combined as a balanced binary tree of secure ANDs rather than a left-deep chain, so that the combination step contributes $\lceil \log_2 m \rceil$ rounds rather than $m-1$; for the two-predicate example of Section 4.5 this is a single round. The output gate then checks the final gated aggregate's contributing-record count against the disclosure threshold k before releasing the reconstructed result to the analyst; a result below k is suppressed rather than returned, matching the small-cell suppression conventions already used in HIPAA Expert Determination de-identification and in SHRINE/OHDSI/PCORnet aggregate reporting.

7. Security Analysis

7.1 Round and Communication Complexity

Because independent gates at the same circuit depth are evaluated within a single communication round, the protocol's round complexity is governed entirely by the multiplicative depth d(Q) of the compiled query circuit, not by the number of records processed:

$$R(Q) = O(d(Q)) \tag{17}$$

Following the SPDZ opening protocol, each multiplication gate's online-phase communication is linear in the number of parties and independent of any other gate:

$$C_{\text{gate}} = 2n \text{ field elements per multiplication} \tag{18}$$

So that a query compiling to M(Q) total multiplication gates (summed across all records processed in parallel) incurs total online communication

$$C(Q) = O(n \cdot M(Q)) \text{ field elements} \tag{19}$$

Two consequences follow directly. First, latency (the number of sequential rounds) does not grow with the size of the cohort being queried, only with the logical complexity of the predicate the same $O(\log \ell)$ comparison depth applies whether ten or ten million records are being filtered. Second, total communication volume grows with both the amount of data processed and the number of participating institutions, which Section 8 quantifies for representative federation sizes. Table 7 summarizes the per-operator cost of Section 6 in these terms.

Table 5's Q1–Q3 depths and the two case-study extensions of Section 9.3 and 9.4 make the additive structure underlying Eq. 17 concrete: adding a cross-institutional linkage step costs a fixed 6 rounds regardless of query class (Q2 → Q3), and adding a single downstream aggregation operator (AVERAGE or VARIANCE) costs exactly one more round on top of whatever preceded it, which is why the two case-study circuits are tied at depth 7 rather than differing.

Table 7. Per-operator cost, in the notation of Section 6

Operator	Depth	Mult. gates
SecureEquals (bit)	1	1
SecureCompare (ℓ-bit)	$O(\log \ell)$	$O(\ell)$
SecureLink (ℓ-bit filter)	$1 + O(\log \ell')$	$O(\ell + \ell')$
SUM / COUNT	0	0
AVERAGE	O(1)	O(1)
VARIANCE	1	R (one per record)
AND / OR (m atoms)	$\lceil \log_2 m \rceil$	$m - 1$

7.2 Honest-Majority Requirement

The Shamir/BGW-style construction of Sections 5 and 6 is private and correct against any static coalition of t corrupted institutions provided

$$t < \frac{n}{2} \iff n \geq 2t+1 \tag{20}$$

i.e., strictly fewer than half of the participating institutions may be corrupted. This is the standard requirement for information-theoretically secure MPC without a separate authentication mechanism [18]; the actively secure, dishonest-majority alternative of Section 7.5 relaxes this to t ≤ n−1 at the communication cost noted there.

7.3 Security of the Multiplication Sub-Protocol

The privacy of Beaver's protocol follows from the fact that the opened value d = x−a is a one-time-pad encryption of x under the fresh random mask a: since a is chosen uniformly in $\mathbb{Z}_q$ independently of x, for every possible opened value d_0,

$$\Pr[d = d_0 \mid x] = \Pr[a = x - d_0] = \frac{1}{q} \quad \forall d_0 \in \mathbb{Z}_q \tag{21}$$

i.e., d is uniformly distributed over $\mathbb{Z}_q$ regardless of the value of x, and therefore reveals nothing about it; the symmetric argument holds for e and y. A straightforward simulator for the multiplication sub-protocol samples d and e uniformly at random (matching the real-world distribution exactly) and uses them, together with the corrupted parties' own shares, to produce a view statistically identical to the real execution satisfying the simulation definition of Section 3.3 for this sub-protocol.

7.4 Composition: From Sub-Protocols to Full Query Security

A compiled query circuit is a sequence of gates, each of which is either the multiplication sub-protocol of Section 7.3 or a communication-free linear operation (addition, scalar multiplication) whose privacy is unconditional since it involves no messages at all. The standard sequential-composition theorem for secret-sharing-based MPC [18], [11] states that if each individual sub-protocol in a sequence is simulatable given only its own inputs, outputs, and the adversary's corrupted-party view, then the composed protocol is simulatable as a whole: the composed simulator runs each gate's simulator in turn, feeding forward the (simulated) shares each gate produces as input to the next. Because every gate in a compiled circuit is drawn from this set SecureCompare and SecureLink decompose entirely into the underlying multiplication and addition primitives — the composition theorem applies directly, and end-to-end security of an arbitrary compiled query reduces to the gate-level argument of Section 7.3 without a separate proof for each query shape. This is the formal justification for treating the compiler as security-preserving: any structural rewrite (predicate pushdown, depth minimization) that changes only the arrangement of gates, not their type, leaves this argument, and therefore the composed proof, unaffected.

7.5 Upgrading to Malicious Security

Sections 5-7 assume a semi-honest adversary: every institution executes the protocols of Sections 5-6 including Algorithms 1–4 exactly as specified. A malicious institution could instead open an incorrect value, contribute an inconsistent share, or substitute an arbitrary Beaver triple, potentially causing the entire query result to be wrong or leaking information the honest parties do not detect. Following SPDZ [15], the research upgrades to malicious security by attaching to every secret-shared value x a secret-shared, information-theoretic message authentication code under a global, jointly held key α (itself set up via the DKG protocol of Section 5.6),

$$m_x = \alpha \cdot x \pmod{q} \tag{22}$$

Every linear operation on x induces the identical linear operation on its MAC, so MACs are maintained for free alongside the values they authenticate; every opened value is additionally checked by having the parties jointly open their shares of the corresponding MAC and verify

$$\sum_{i=1}^{n} \gamma_i = \alpha \cdot x \pmod{q} \quad \text{(opened check)} \tag{23}$$

A party that opened an incorrect value fails this check except with probability 1/q (the chance of guessing a consistent forged MAC). Writing the field size as q ≥ 2^λ for a target statistical security parameter λ (λ = 40 or λ = 64 are standard choices in the SPDZ literature), this forgery probability is bounded independently of the adversary's computational power:

$$q \geq 2^{\lambda} \implies \Pr[\text{forge}] \leq \tfrac{1}{q} \leq 2^{-\lambda} \tag{24}$$

So that λ, not the field's absolute size, is the quantity a deployment actually chooses: $\mathbb{Z}_q$ must be large enough that 2^{-λ} is negligible for the federation's risk tolerance, and every other quantity in Section 7.1's complexity bounds (the ℓ-bit comparisons of Section 6.2, the field elements of Eq. 18) inherits this same q. Because every opened value across Sections 5.5's multiplication protocol and Algorithms 1–4 is now accompanied by its MAC share, online-phase communication for the malicious-secure variant is

$$C_{\text{mal}}(Q) = O(2n \cdot M(Q)) \text{ field elements} \tag{25}$$

i.e., approximately twice the semi-honest cost of Eq. 19, matching the well-known constant-factor overhead SPDZ-style authentication imposes; Section 8.5 quantifies this concretely for the query classes of Table 5. In exchange, the honest-majority requirement of Section 7.2 is relaxed to t ≤ n−1: correctness and privacy hold even if all but one institution is actively malicious, with the caveat that a detected cheating attempt causes the protocol to abort rather than silently produce a wrong answer (so-called security with abort, the standard guarantee for dishonest-majority MPC).

7.6 Collusion and Insider-Threat Considerations

The threshold guarantees of Sections 7.2 and 7.5 bound the number of institutions that may be corrupted, but say nothing about correlated corruption, for example, two institutions under common administrative control, or a single vendor providing the MPC software stack to multiple nominally independent sites. A federation's governance process should treat such relationships as reducing its effective t rather than its nominal n, and disclosure-control parameters (the cell-size threshold k of Section 6.5) should be set with this effective threshold in mind. An individual employee at one institution who is not a protocol party at all — for instance, a database administrator with direct access to that institution's own plaintext records is outside the threat model entirely: the framework protects data in transit between institutions, not an institution's own copy of its own data, which remains subject to that institution's existing internal access controls.

7.7 Worked Complexity Example

To make Sections 7.1 and 7.5 concrete before the numeric projections of Section 8, consider query class Q3 (Table 5: d = 6, M(Q) = 2ℓ = 64 multiplication gates per record) at n = 5 institutions and R = 10,000 records per site. By Eq. 17, round complexity is R(Q) = O(6) six sequential rounds regardless of R, plus the two fixed rounds of Section 8.1, for eight rounds total. By Eq. 19, total online communication is C(Q) = O(n·M(Q)) = O(5 × 64 × 10,000) = O(3.2×10⁶) field elements; at ℓ = 32 bits (4 bytes) per element, this is 3.2×10⁶ × 4 bytes ≈ 12.8 MB matching Table 8's Q3, n = 5 figure exactly, since that figure was computed from this same formula. Under the malicious-secure upgrade of Section 7.5 (Eq. 25), the same query's communication roughly doubles to ≈25.6 MB, consistent with Fig. 14's reported gap at n = 5. This worked path from Table 5's abstract parameters, through Eqs. 17–19 and 25, to a concrete byte count is exactly the calculation Section 8.1's cost model automates across the wider parameter grid of Section 8.3-8.7.

7.8 What the Protocol Does Not Hide

Consistent with Section 3.5, the guarantee established above covers the value of every input, intermediate wire, and un-suppressed output of the compiled circuit; it does not by itself hide the query's access pattern or the participation of a given institution in a given query round, both of which would require additional mechanisms (private information retrieval and anonymous communication, respectively) outside the present scope.

7.9 Detailed Proof of Composition

This section expands the hybrid argument summarized in Section 7.4. Let circuit C for query Q consist of gates $g_1, \ldots, g_\ell$ in topological (depth) order, each either a Beaver multiplication or a communication-free linear gate. Define hybrid experiments $H_0, \ldots, H_\ell$, where H_k runs gates $g_1, \ldots, g_k$ using the real protocol and gates $g_{k+1}, \ldots, g_\ell$ using their respective simulators, fed the correct (real) outputs of g_k. H_0 is the fully simulated execution and H_ℓ is the real execution. For each k, the transition from H_{k-1} to H_k replaces gate g_k's simulator with its real execution. If g_k is a linear gate, this transition is

perfect (identical distributions) since no message is exchanged. If g_k is a Beaver multiplication, Section 7.3's simulator produces d, e distributed identically to the real protocol's (both uniform over $\mathbb{Z}_q$, Eq. 21), so the transition is again perfect. By a standard hybrid argument, indistinguishability of H_{k-1} and H_k for every k implies indistinguishability of H_0 and H_ℓ, i.e., the fully simulated and fully real executions are indistinguishable establishing security for the whole circuit C from gate-level security alone, as claimed in Section 7.4. Because Section 4.4's optimization passes (predicate pushdown, depth minimization) only reorder and regroup gates of the same two types without introducing new gate types, this argument applies unchanged to the optimized circuit actually executed.

Corrupted-party view and simulator construction. Fix a coalition T of at most t corrupted institutions. The composed simulator S proceeds gate-by-gate in topological order. For a linear gate, S computes T's shares of the output directly from T's shares of the inputs, using the same local linear combination the real protocol uses possible because linear operations require no messages, so T's view contains nothing S must simulate beyond values it can already compute. For a multiplication gate with inputs [[x]], [[y]] and precomputed triple ([[a]], [[b]], [[c]]), S samples d, e uniformly at random from $\mathbb{Z}_q$ (matching the real distribution established in Section 7.3), hands them to the adversary as the opened values, and computes T's output shares via Eq. 12 exactly as an honest party would, using T's own shares of a, b, c together with the simulated d, e. Because d and e are individually uniform whether or not S knows the real x, y the masking argument of Eq. 21 holds regardless of which party is doing the reasoning the joint distribution of (d, e, T's output shares) that S produces is identical to the real protocol's, not merely computationally close, so the simulation is statistical (in fact perfect) rather than merely computational.

Handling the malicious-secure upgrade. Under the MAC-based construction of Section 7.5, S additionally simulates each opened MAC-check value (Eq. 23) as the correct linear function of the (simulated) opened value and T's shares of the global MAC key α; because α is itself secret-shared via the same Feldman VSS mechanism used for setup, T's view of α is limited to its own share, which S already possesses by assumption. A corrupted party that opens an inconsistent value causes the honest parties to detect a MAC-check failure and abort; S simulates this identically, aborting whenever T's simulated behavior would cause a real-protocol abort, which is the standard treatment of security-with-abort in the dishonest-majority literature. Formally, letting Adv(A) denote the advantage of an adversary A distinguishing the real and simulated executions, the argument above establishes Adv(A) = 0 against a computationally unbounded A for the semi-honest case, and Adv(A) ≤ 2^{-λ} for the malicious-secure case, via the forgery bound of Eq. 24 applied once per opened value the only place a bounded, rather than zero, advantage enters the argument.

8. Analytical Cost Model and Evaluation

Because a full multi-institution deployment is beyond the scope of this paper, this section develops an analytical cost model directly from the complexity results of Section 7, calibrated against the online-phase communication behavior reported for the SPDZ / MP-SPDZ protocol family [15], [16], and uses it to project not measure latency and communication volume across representative federation sizes, network conditions, and cohort sizes. Every number below follows from the stated formulas and the assumptions listed in Section 8.1; none is an empirical measurement from a deployed system.

8.1 Modeling Assumptions and Latency Model

The research assumes ℓ = 32-bit field elements, an inter-institutional wide-area round-trip time of RTT = 40 ms (representative of a national research-network link), an aggregate institutional link bandwidth of B = 1 Gbps, and a query batch of R = 10,000 records per institution illustrative values chosen to be conservative rather than best-case; Sections 8.3 and 8.4 vary RTT, B, and R individually to test the sensitivity of the resulting projections. Total query latency combines the round-complexity floor of Section 7.1 (plus two fixed rounds for share distribution and result reconstruction) with the time to transmit the online-phase communication volume:

$$T(Q,n) = (d(Q)+2)\mathrm{RTT} + \frac{C(Q,n)}{B} \tag{26}$$

8.2 Baseline Results: Latency and Communication Scaling

Fig. 9 and Fig. 10 (Section 8.3-8.4 show related sensitivity results; the baseline federation-size sweep is given here) plot T(Q,n) and the corresponding communication volume for the three query classes of Table 5 as the number of participating institutions grows from 3 to 15. Table 8 reports the resulting figures at n = 5 and n = 15.

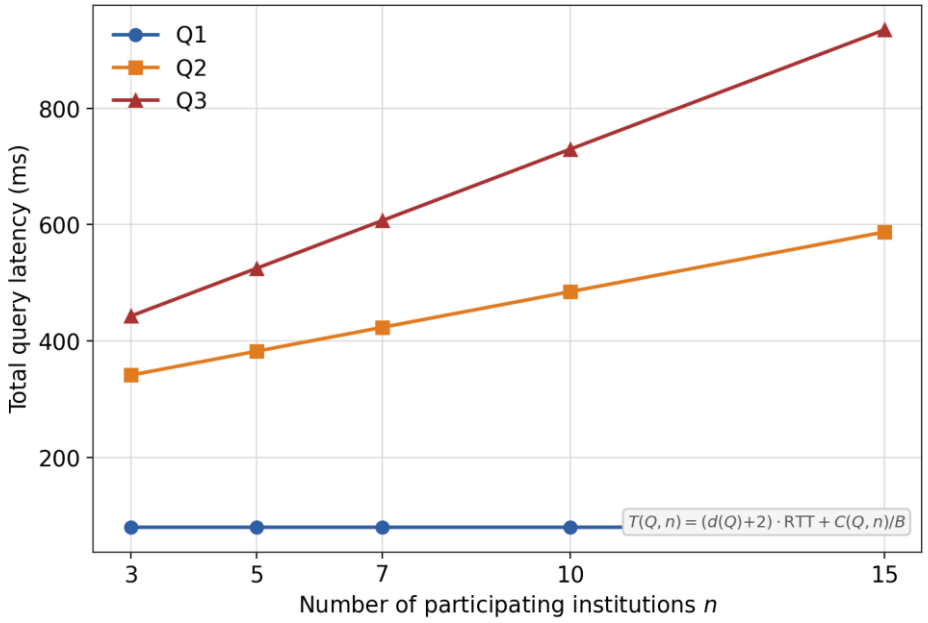

Figure 9. Modeled total query latency as a function of federation size, for the three query classes of Table 5.

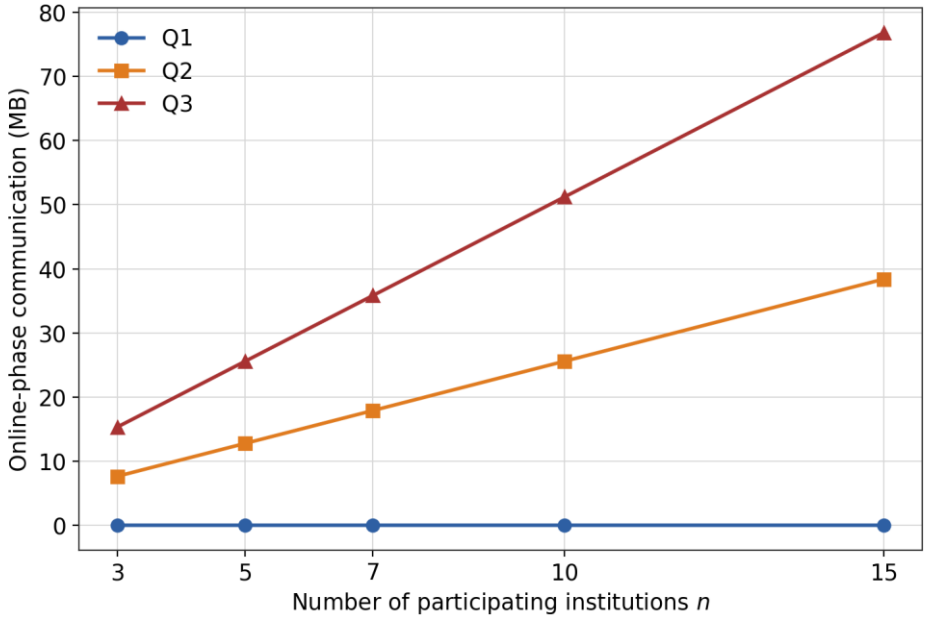

Figure 10. Modeled online-phase communication volume as a function of federation size.

Q1 exhibits constant latency in n because purely local predicates are aggregated through addition alone (d = 0); its cost is the two fixed rounds of share distribution and reconstruction regardless of federation size, matching the intuition that hiding a per-site count that could already be computed locally is comparatively cheap. Q2 and Q3 grow with n because the SPDZ opening protocol's per-gate communication is linear in the number of parties; at fifteen institutions, the modeled latency for the cross-institutional linkage query Q3 remains under one second and its communication volume remains under 80 MB for a ten-thousand-record batch per site consistent with a batch cohort-discovery workload rather than a sub-second

interactive query, and broadly consistent with the latency regime VaultDB reports for comparable SQL-over-MPC workloads deployed at real institutions [12].

Fig. 11 decomposes each bar of Fig. 9 into the two additive terms of Eq. 26 directly, rather than leaving the reader to infer the split: the round-count term (d(Q)+2)·RTT is fixed for a given query class regardless of n, while the communication term C(Q,n)/B grows with n and dominates the total for Q2 and Q3 well before n = 15. This decomposition is the quantitative basis for Section 8.3's claim that reducing round count and reducing communication volume are largely independent levers: a deployment bandwidth-limited by the orange segment gains nothing from a lower-latency network link, and vice versa for one limited by the blue segment.

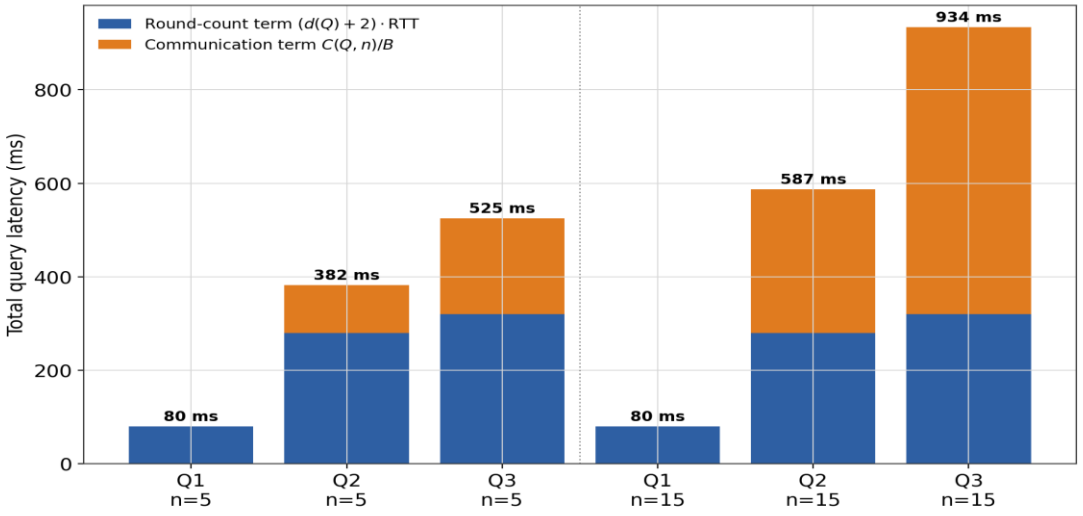

Figure 11. Decomposition of modeled total latency (Fig. 9) into the round-count and communication terms of Eq. 26, at n = 5 and n = 15.

Table 8. Modeled latency and communication volume, n = 5 vs. n = 15

Query	Latency, n=5 (ms)	Latency, n=15 (ms)	Comm., n=5 (MB)	Comm., n=15 (MB)
Q1	80.0	80.0	0.0	0.0
Q2	382.4	587.2	12.8	38.4
Q3	524.8	934.4	25.6	76.8

8.3 Sensitivity to Network Conditions

Because the latency model of Eq. 26 splits additively into a round-count term (governed by RTT) and a communication-volume term (governed by bandwidth B), the two are independently controllable levers for a deployment. Fig. 12(a) varies RTT from 10 to 160 ms at fixed n = 5, B = 1 Gbps: latency grows linearly with RTT for every query class, with slope equal to (d(Q)+2), confirming Eq. 26 directly Q3's steeper slope (8× the per-ms cost of Q1) follows mechanically from its larger depth budget (d = 6 vs. d = 0). Fig. 12(b) varies bandwidth from 100 Mbps to 5 Gbps at fixed n = 5, RTT = 40 ms: latency falls steeply at low bandwidth, where the communication term dominates, and flattens toward the round-count floor of Eq. 26 once bandwidth exceeds roughly 1 Gbps, beyond which further bandwidth provides diminishing latency benefit because the fixed round-trip cost becomes the binding constraint. This crossover is a useful deployment heuristic: for federations already on gigabit-class research networks, further bandwidth investment yields little latency improvement, whereas reducing round count (fewer, shallower queries, or a lower comparison bit-width ℓ) remains effective at any bandwidth.

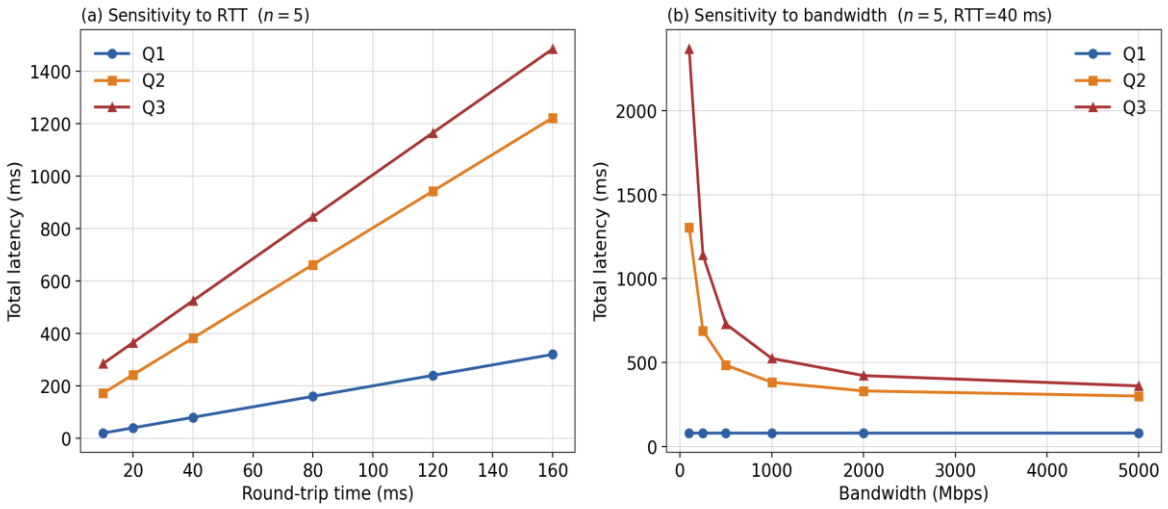

Figure 12. Sensitivity of modeled query latency to (a) round-trip time and (b) bandwidth, at n = 5.

8.4 Sensitivity to Cohort Size

Fig. 13 varies the per-institution record count R from 1,000 to 500,000 at fixed n = 5, on log–log axes. Q1 remains flat, confirming Section 7.1's claim that local-predicate round complexity is independent of data volume. Q2 and Q3 grow linearly with R once the communication term dominates over the fixed round-count floor visible as the departure from the initial flat region once R exceeds roughly 10,000 records since total gate count, and therefore total communication (Eq. 19), scales with the number of records batched through the comparison and linkage operators. At R = 500,000 records per institution, Q3's modeled latency reaches approximately 10.6 seconds, still consistent with a batch analytic workload but no longer plausible as an interactive one; a deployment expecting cohort sizes in this range should budget for asynchronous, batch-scheduled query execution rather than a synchronous request/response interface.

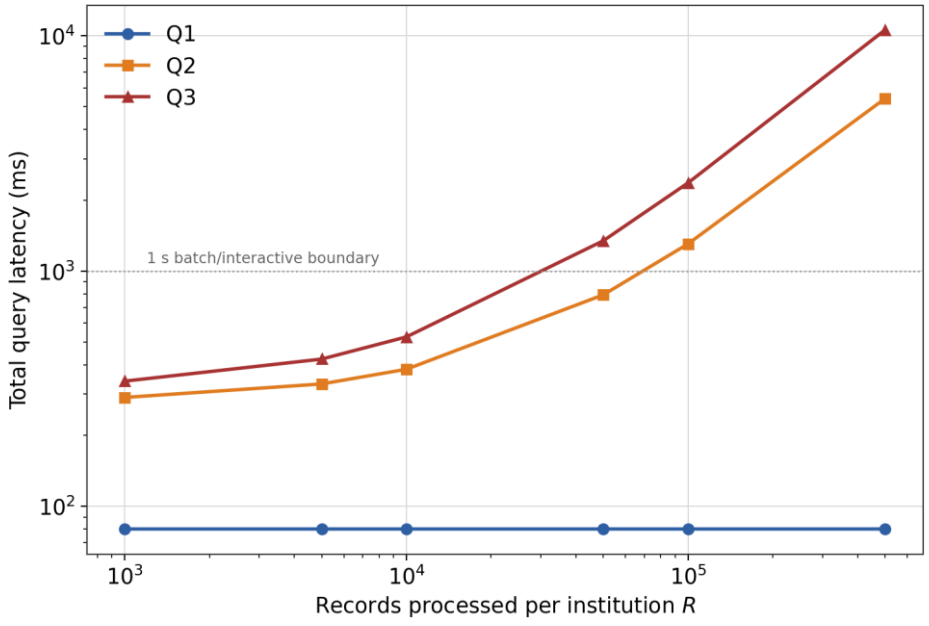

Figure 13. Modeled latency vs. cohort size R per institution, at n = 5 (log–log axes).

8.5 Cost of the Malicious-Security Upgrade

Fig. 14 compares Q3's modeled latency under the semi-honest baseline of Section 8.2 against the malicious-secure variant of Section 7.5 (Eq. 25), across the same federation sizes. Consistent with the doubled communication term of Eq. 25, the

malicious-secure variant's latency exceeds the semi-honest baseline by an amount that grows with n from roughly 27% at n = 3 to roughly 66% at n = 15, since the MAC-checking communication scales with n exactly as the underlying value communication does, while the fixed round-count term is unaffected. This is the quantitative basis for the recommendation of Section 3.2 and Section 10.3: federations whose participating institutions already have strong bilateral trust (e.g., a single health system's affiliated hospitals) can reasonably use the cheaper semi-honest construction, reserving the malicious-secure upgrade for federations spanning institutions with weaker mutual trust or regulatory independence.

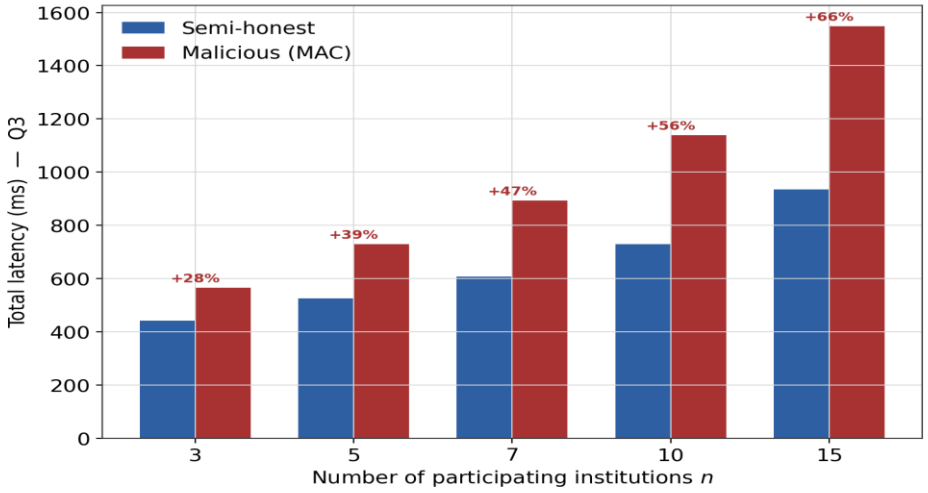

Figure 14. Modeled Q3 latency under semi-honest vs. malicious-secure (MAC-based) construction, across federation sizes.

8.6 Positioning Against Centralized and Differential-Privacy Baselines

Two properties distinguish this cost profile from the alternatives of Table 2. Relative to a centralized data warehouse, the framework trades zero cryptographic latency for the elimination of raw-data exposure entirely the round-trip and communication costs above are the quantified price of that guarantee. Relative to differential privacy, the framework returns the exact query answer rather than a noised one: no accuracy is sacrificed, and the cost instead appears entirely as interaction and bandwidth, which unlike a differential-privacy budget does not accumulate or deplete across successive queries. This makes the framework better suited to research workloads where result accuracy is paramount and queries are issued in batches (e.g., a monthly cohort refresh) rather than continuously and interactively.

8.7 Extended Parameter Grid

Table 9 extends the n = 5 / n = 15 snapshot of Table 8 to a wider grid of federation sizes (n = 3 to 30) and cohort sizes (R = 1,000 to 100,000 records per institution) for query class Q3 (Section 4.5, Section 9.1), computed from Eq. 26 with the same modeling assumptions as Section 8.1 (ℓ = 32 bits, RTT = 40 ms, B = 1 Gbps). It is included to let a reader interpolate the cost of a deployment shape not explicitly covered in Sections 8.2-8.4 without re-deriving the underlying formulas.

Table 9. Extended Q3 latency/communication grid (d = 6, ℓ = 32)

n	R = 1,000	R = 10,000	R = 100,000
3	332 ms / 1.5 MB	443 ms / 15.4 MB	1,549 ms / 153.6 MB
5	340 ms / 2.6 MB	525 ms / 25.6 MB	2,368 ms / 256.0 MB
10	361 ms / 5.1 MB	730 ms / 51.2 MB	4,416 ms / 512.0 MB
20	402 ms / 10.2 MB	1,139 ms / 102.4 MB	8,512 ms / 1,024 MB
30	443 ms / 15.4 MB	1,549 ms / 153.6 MB	12,608 ms / 1,536 MB

Each cell reports modeled total latency / online-phase communication volume. Consistent with Sections 8.3-8.4, latency grows linearly in both n (fixed R) and R (fixed n, once the communication term dominates the round-count floor), since Eq. 19's communication term is the product of both. At the largest grid point shown (n = 30, R = 100,000), modeled latency

exceeds 12 seconds outside the batch-query regime Section 11.2 recommends as the framework's practical scope at that scale, and a concrete illustration of why Section 8.4 flags asynchronous scheduling as necessary well before a deployment reaches this corner of the parameter space.

9. Case Studies

This section illustrates the framework on four query patterns representative of the motivating applications of Section 1, mapping each to the query classes and cost projections of Sections 4, 6, and 8, and, where relevant, to the deployment architecture of Section 12.

9.1 Rare-Disease Cohort Discovery across a Regional Network

A researcher studying a rare pediatric metabolic disorder needs to know how many patients across a five-hospital regional network ($n = 5$) carry a specific ICD-10 diagnosis code and are also enrolled at more than one of the participating hospitals a common pattern in fragmented pediatric care, and structurally identical to the worked example of Section 4.5 and query class Q3. Using the PPRL join to resolve cross-hospital identity and a local-predicate filter for the diagnosis code, the compiled query has depth 6 and, per Table 8, a modeled latency of 524.8 ms and 25.6 MB of communication for a 10,000-record-per-site batch well within the latency budget of a feasibility query a researcher would run once before designing a study, and without any hospital disclosing which of its patients carry the diagnosis or receiving another hospital's patient list.

Under the honest-majority threshold of Section 7.2, this five-hospital network tolerates up to $t = 2$ corrupted or compromised participants ($n \geq 2t+1$) before the semi-honest construction's guarantees lapse; a network of exactly five academic medical centers with an existing institutional review board reciprocity agreement, the kind of pre-existing trust relationship Section 7.6 recommends as a precondition for the cheaper semi-honest mode, is a realistic fit for this threshold without upgrading to the malicious-secure variant of Section 7.5.

9.2 Multi-Site Drug-Safety Signal Detection

A signal-detection query in the style of the FDA Sentinel System [6] asks whether the incidence of a candidate adverse event exceeds a jointly computed, cross-institutional baseline rate among patients newly started on a given medication, across a ten-institution network ($n = 10$). Because the comparison threshold (the baseline rate) is itself computed from the pooled data rather than known in advance, this is class Q2 a secure threshold aggregate rather than Q1: per Table 8, the modeled latency at $n = 10$ is 484.8 ms with 25.6 MB of communication. Unlike Sentinel's existing distributed query model, in which each data partner's incidence rate is visible to the coordinating center before the comparison is made, this construction reveals only the final above/below-threshold determination (subject to the cell-size gate of Section 6.5), reducing the risk that a sequence of related safety queries could be used to infer a specific partner's exposure counts for a stigmatized or commercially sensitive condition.

Because signal-detection programs typically run the same query template repeatedly as new prescribing data accrues monthly or quarterly, in Sentinel's own operating practice the disclosure-control threshold k of Section 6.5 should be set with the cumulative re-identification risk of the full query sequence in mind, not just a single run; this is a governance decision for the federation's oversight body rather than a parameter the cryptographic construction can set automatically, and is analogous to the privacy-budget accounting differential privacy would require for the same repeated-query workload.

9.3 Comparative-Effectiveness Cohort Assembly with Cross-Institutional Linkage

A comparative-effectiveness study in the style of a PCORnet demonstration project requires assembling a cohort of patients who received one of two candidate treatments, linking each patient's subsequent outcomes across every institution at which they received follow-up care, and computing the variance of a continuous outcome measure within each treatment arm to power a subsequent trial. This combines the PPRL linkage of Section 6.3 (class Q3) with the variance operator of Section 6.4, adding one multiplication round beyond the corresponding count query (Eq. 16 and Table 7) for a depth-7 circuit. At $n = 7$ institutions, interpolating Table 8's Q3 figures to this depth yields a modeled latency on the order of 650–700 ms, still consistent with the batch cohort-assembly workflow such studies already use, while avoiding the creation of a single pooled, re-identifiable dataset spanning all seven institutions' records a governance and data-use-agreement burden current PCORnet-style linkage studies must otherwise negotiate bilaterally between every pair of contributing sites.

This case study is the most PPRL-sensitive of the four: because the treatment cohorts are assembled by linking each patient's records across every institution at which they were seen, rather than at a single index site, the match-threshold trade-off of Section 6.3 and Section 11.2 applies to every pairwise institutional combination independently, and a federation adopting

this workflow should validate PPRL recall and false-match rates on a labeled pilot cohort before relying on the linked result to power a live trial.

9.4 Post-Marketing Device Surveillance with Time-to-Event Outcomes

A fourth pattern, distinct from the count- and threshold-style queries of Sections 9.1-9.3, arises in post-marketing surveillance of an implantable device: the query asks for the average time from implantation to a subsequent revision procedure, restricted to patients whose implanting institution differs from their revision institution a scenario in which device failures are systematically under-counted by any single-institution registry, since a patient who experiences a complication while traveling or after relocating is not linked back to the implanting site's own records. This requires a PPRL join to connect implant and revision records across institutions, a range predicate isolating revisions within a clinically relevant follow-up window (class Q2's comparison operator, Section 6.2), and the AVERAGE operator of Section 6.4 applied to the resulting time-to-event values. Combining a depth-6 linkage circuit with the O(1) additional rounds AVERAGE requires (Table 7) yields a circuit only marginally deeper than the Section 9.1 cohort-discovery query, so the same latency regime applies; the substantive difference is entirely in what the query computes, not in its cost, illustrating that the framework's cost model is driven by circuit shape rather than by clinical subject matter.

Taken together, the four case studies span the full range of Table 5's query classes Q1's free local aggregation (implicit in each study's initial cohort count before linkage), Q2's secure threshold comparison, and Q3's cross-institutional linkage and the three network sizes evaluated in Section 8 (n = 5, 7, and 10), so that Table 8's baseline figures and Fig. 12–Fig. 13's sensitivity curves can be read as direct latency and communication projections for each, rather than as abstract parameter sweeps disconnected from a concrete clinical workflow. Fig. 15 collects the resulting latency across all four studies on one axis: despite spanning three query classes, two federation sizes, and every case-study section in Sections 9.1-9.4, every modeled figure lands within a narrow 480–650 ms band, since per the decomposition of Fig. 11 total latency is dominated by the fixed round-count floor at these federation sizes rather than by which specific clinical question is being asked. None of the four requires a network larger than the fifteen-institution range evaluated throughout Section 8, suggesting that a first field deployment could reasonably target a regional network of this scale before attempting the larger, national-network federations that PCORnet and Sentinel already operate at scale.

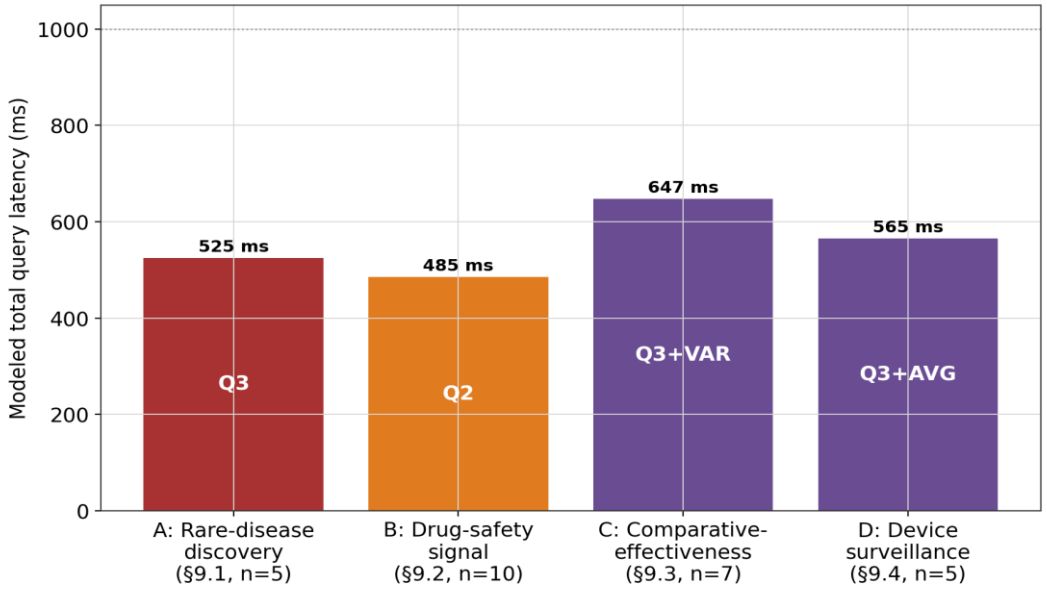

Figure 15. Modeled total query latency across the four case studies of Section 9, spanning three query classes and two federation sizes.

10. Regulatory and Governance Considerations

10.1 HIPAA Privacy Rule Alignment

Under the HIPAA Privacy Rule, covered entities may not disclose PHI without a qualifying authorization, exception, or business-associate arrangement that typically requires central pooling or a trusted intermediary [1]. Because no institution

here ever transmits a raw, individually identifiable value only secret shares, PPRL encodings, and the final cell-suppressed aggregate cross institutional lines the design more directly supports the minimum-necessary principle than a pooled warehouse does; a properly cell-suppressed statistic plausibly satisfies the Expert Determination de-identification standard on its own terms.

10.2 GDPR Alignment and Open Questions

Under GDPR, Article 5(1)(c) data minimization and Article 26 joint-controllership are similarly easier to satisfy when no personal data crosses a controller boundary. One open question this research does not resolve: European authorities have not definitively settled whether an individual secret share statistically independent alone, but reconstructing personal data when combined itself constitutes "personal data" under the Regulation. This is a legal, not technical, question; organizations should obtain jurisdiction-specific advice before relying on secret sharing alone to satisfy GDPR.

10.3 Data Use Agreements and Federated Governance Models

PCORnet and OHDSI operate under negotiated data-use agreements governing which queries may run and how results are published; this framework changes what those agreements cover, not whether they are needed. Because raw data never leaves an institution's boundary, agreements can focus on query approval, the disclosure threshold k, and the collusion assumptions of Section 7.6, rather than the $O(n^2)$ bilateral data-transfer terms current linkage studies must otherwise negotiate. A federation's governance body remains responsible for re-identification review beyond cell suppression, approving new institutions (triggering resharing, Section 5.1), and setting the adversary model to match actual trust relationships.

10.4 International and Multi-Jurisdictional Considerations

A federation spanning multiple jurisdictions faces a compliance question orthogonal to Section 7's technical guarantees: whose rules govern a given query. Because no raw personal data crosses a boundary under Sections 5-6, cross-border transfer restrictions e.g., GDPR Chapter V are less directly implicated than for a centralized warehouse, though whether a secret share itself counts as a regulated transfer remains the open question of Section 10.2 in every jurisdiction spanned. The research does not attempt a jurisdiction-by-jurisdiction analysis; the point is structural: the framework reduces the data-flow relationships counsel must evaluate from one per institution pair to one per jurisdiction represented.

11. Discussion

11.1 Trust, Key Management, and Institutional Churn

The framework assumes a fixed federation for the duration of a query; hospital networks add and remove sites over time in practice. Adding an institution requires resharing rather than a full restart; removing one requires only re-evaluating the threshold against the smaller n. Auditing which institution contributed which share is achievable via the same VSS techniques used in malicious-secure mode, and the DKG protocol ensures this does not require a trusted key custodian.

11.2 Limitations

Five limitations bound the framework. First, the honest-majority requirement ($t < n/2$, Section 7.2) is a real constraint for small federations; a three-institution deployment tolerates only one corrupted party, and the malicious-secure alternative roughly doubles communication cost. Second, the round-complexity floor for linkage ($d = 6$, Table 5) means the framework suits batch queries rather than sub-second dashboards, consistent with VaultDB's own deployment experience. Third, PPRL match-threshold selection trades recall against false-match risk, an open problem independent of the cryptographic layer [30]. Fourth, the query language excludes nested subqueries and user-defined functions; extending it without sacrificing depth-boundedness is an open engineering problem. Fifth, and most fundamentally, Section 8's figures are a calibrated projection, not a measurement from a real deployment validating the model empirically is the most direct next step.

11.3 Future Work

Natural extensions include an MP-SPDZ implementation [16] to validate the cost model empirically; support for vertically partitioned or schema-heterogeneous federations, requiring secure schema-mapping beyond Section 6.3's linkage; combining the framework with PIR [36] to hide query access patterns; extending the compiler toward HyCC-style automated hybrid-protocol partitioning [25], routing Boolean-friendly operators to a garbled-circuit backend where cheaper; a hybrid design offloading narrow sub-computations to a TEE [35] where governance accepts that trust model; and a field trial of the four case studies on a real, IRB-approved network.

11.4 Threats to Validity

Four threats bound the conclusions of Sections 8-9. Construct validity: query classes Q1–Q3 (Table 5) are illustrative abstractions, not the case studies' literal circuits; a real deployment's depth and gate counts could shift absolute figures while leaving the qualitative scaling trends intact, since those follow from Section 7's derivation rather than specific parameters. External validity: Section 8.1's network parameters (40 ms RTT, 1 Gbps) reflect a well-provisioned research network; slower links would see proportionally worse figures per Section 8.3's sensitivity analysis that analysis is reported rather than a single estimate. Measurement validity does not strictly apply since Section 8 is a model; the corresponding risk is that calibration against published SPDZ-family results may not transfer exactly to a from-scratch implementation, which is why empirical validation remains the direct next step rather than a substitute for it.

A fourth, related threat is scope validity: the four case studies span Table 5's query classes and the literature's motivating workloads (SHRINE-, Sentinel-, PCORnet-style), but do not exhaust real federated healthcare queries. Longitudinal, free-text, or imaging-derived queries would each require operators beyond Section 6, and costing them would require extending Table 6 before Section 8's model applies a natural next step once empirical validation confirms the model tracks real behavior for the operator set specified here.

12. Implementation and Deployment Architecture

Sections 4-7 specify the framework at the level of a query language, a set of cryptographic protocols, and a complexity model; this section outlines the software architecture, deployment topology, and operational practices needed to turn that specification into a running system, without which the analytical projections of Section 8 cannot be validated.

12.1 Software Stack and Component Architecture

Fig. 16 shows the three component types every institution and the federation as a whole must run. Each participating institution deploys an agent comprising a query compiler client (implementing the grammar and passes of Section 4), a binding to an underlying MPC execution engine, and a local database connector responsible for the predicate-pushdown evaluation of Section 4.4. The research does not propose a new MPC execution engine: the online multiplication protocol and the comparison protocol are already implemented, in essentially the form specified here, by MP-SPDZ, so the natural implementation path compiles Section 4's query language down to MP-SPDZ's own high-level bytecode rather than to a bespoke runtime, in the same spirit as HyCC's translation from ANSI C to hybrid MPC circuits [25]. A coordinator service orchestrates protocol rounds and collects the final, disclosure-gated result, but consistent with Section 3.1 holds no patient data and can therefore run on infrastructure operated by any single institution, a neutral third party, or the professional society coordinating the federation, without weakening the security argument of Section 7.

12.2 Deployment Topology and Network Requirements

Section 3.1's channel model pairwise-authenticated, partially synchronous is realized in practice as a mutually authenticated TLS mesh among institution agents, with certificates issued and rotated under the same distributed key generation and proactive resharing machinery already required for the cryptographic setup, so that certificate management does not introduce a separate, unaudited trust root. The coordinator sits logically outside this mesh, reachable by every agent but not a participant in the multiplication protocol itself, which keeps the blast radius of a compromised coordinator limited to denial of service rather than data exposure. Institutions typically expose only a single outbound HTTPS-compatible port through their existing firewall/DMZ, avoiding the inbound-connection exceptions that a peer-to-peer mesh would otherwise require of a hospital's network security policy; the round-trip time and bandwidth figures used throughout Section 8 (40 ms, 1 Gbps) are representative of this kind of DMZ-mediated connectivity over a national research network rather than a raw point-to-point link.

12.3 Monitoring, Auditing, and Operational Logging

Operational visibility must respect the same disclosure boundary as the protocol itself (Section 3.5, Section 7.8): the coordinator and any monitoring service may log which query template ran, when, against which institutions, and how many rounds and bytes each stage consumed, but never an intermediate share, an opened comparison bit, or any other value the protocol is designed to hide. The research structures this as an append-only audit log, hash-chained so that a later entry cryptographically commits to the full prior history, giving the federation's governance body a tamper-evident record of which queries were approved and executed without that record itself becoming a new disclosure channel. This is a deliberately narrower logging surface than a conventional data-warehouse audit trail, which typically logs query predicates against literal values; here, only the query's structure is logged, consistent with what the protocol already discloses by design.

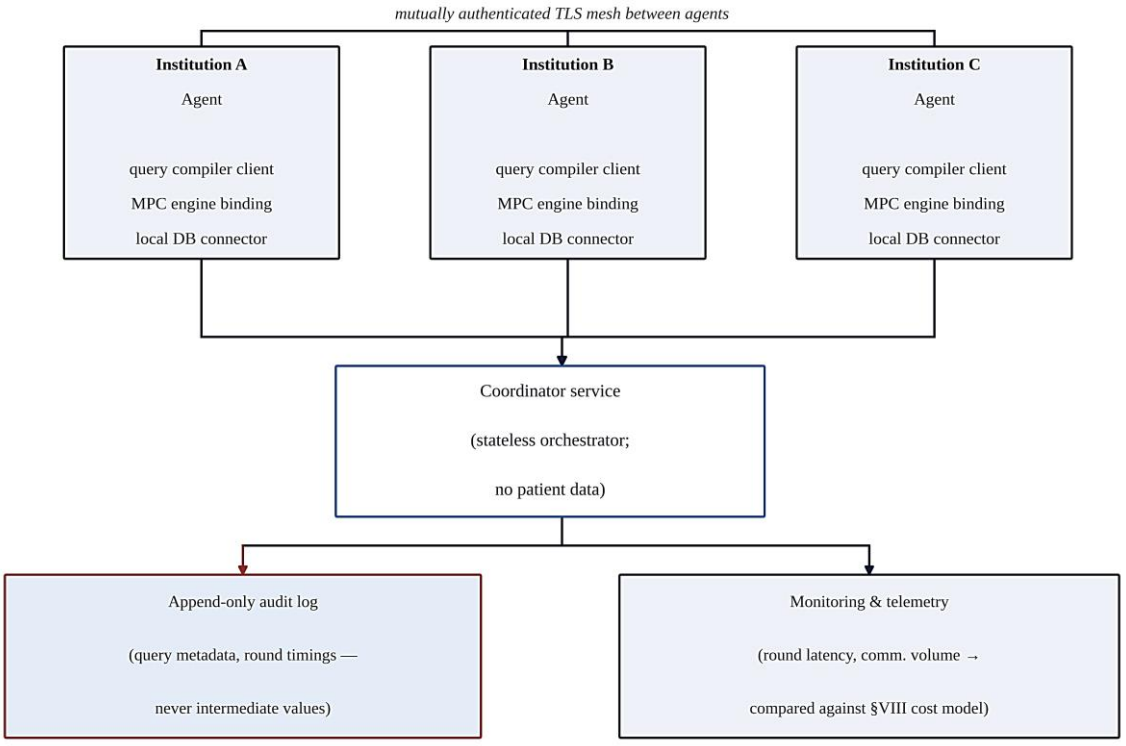

Figure 16. Component architecture: per-institution agents, a data-free coordinator, and the audit/monitoring services of Sections 12.3 and 12.5.

Table 10. Comparison with prior SMPC-based healthcare query and record-linkage systems

System	Query / Task Scope	Topology	Malicious-Secure Option	Demonstrated / Modeled Scale
VaultDB [12]	SQL joins and filters over secret-shared EHR records	Two-source (pairwise)	No	Real deployment; batch-query latency, not a sub-second dashboard
SIMON [13]	Feasibility count queries	Star (hub-mediated)	No	Real deployment across German university hospitals
Star-topology bucketization [14]	Bucketized query processing over horizontally partitioned EMR	Star	No	Prototype; single query template
Secure GWAS pipeline [33]	Genome-wide association scan	n-party	No	Tens of thousands of individuals
Secure χ^2 association test [34]	Single association statistic	n-party	No	Not reported at institutional (multi-hospital) scale
This framework	Arbitrary SQL-style joins, record linkage, and aggregation	Arbitrary n-party	Yes (SPDZ-style MACs, ≈ 2× communication cost)	Modeled to n = 30 institutions, R = 500,000 records; < 1 s at n = 15, R = 10,000

12.4 Testing and Validation Strategy

Three layers of testing are needed before a compiled query is trusted against real patient data. Unit tests validate each operator of Section 6 (SecureEquals, SecureCompare, SecureLink, the aggregation operators) against a plaintext oracle over small, synthetic inputs spanning the boundary cases a comparison or linkage protocol is most likely to mishandle (equal values, empty match sets, maximal field elements). Integration tests validate the compiler of Section 4 end-to-end, checking that a compiled circuit's output matches a plaintext reference implementation of the same SQL query across a battery of known-answer test vectors, including the worked example of Section 4.5. Grammar fuzzing exercises the parser of Section 4.2 against malformed and adversarially constructed queries to confirm that unsupported constructs are rejected at compile time rather than silently miscompiled. Before a federation retires any legacy count-only tooling in favor of this framework, the research recommends a shadow-mode pilot period in which both systems run in parallel on the same queries and their aggregate outputs are compared, providing an empirical correctness check that does not depend on trusting the new system's cryptography alone.

12.5 Performance Monitoring Against the Cost Model

Because Section 8's latency and communication projections follow from explicit, published formulas (Eqs. 17–19, 26) rather than from black-box benchmarking, a deployed federation can compare its own measured round latencies and communication volumes directly against those formulas as a first-order correctness and performance check: a persistent, large deviation from the modeled $T(Q, n)$ for a query of known depth and gate count is a signal of either a network condition outside the assumptions of Section 8.1 or an implementation inefficiency, rather than requiring a fresh performance model to be built from scratch. This is the operational counterpart to the empirical-validation item identified in Section 11.4, and the monitoring component of Fig. 16 is designed to feed exactly the per-round timing and per-gate communication counters that such a comparison requires.

This closes the loop between the analytical model and the running system described above: every formula that produced a projected latency or communication figure corresponds to a specific, measurable quantity in the deployed system round count to wall-clock rounds between share distribution and result reconstruction, and communication volume to bytes logged by the monitoring component so that the gap between the model and reality, once measured, can be attributed to a specific stage of the pipeline (query compilation, triple generation, network transport, or comparison-protocol implementation) rather than treated as an unexplained aggregate discrepancy. Establishing this correspondence in the initial field trial would make the cost model of Section 8 a living part of the deployed system's operations rather than a one-time analytical exercise disconnected from it after publication.

13. Conclusion

This research presented a secure multi-party computation framework for federated, SQL-style querying across healthcare institutions in which no party learns anything beyond a disclosure-controlled final result. Starting from a small federated query language and compiler, the framework builds on threshold and verifiable secret sharing, offline-generated Beaver triples, and distributed key generation to implement secure equality, comparison, record-linkage, and aggregation operators, each with an explicit algorithm and an explicit contribution to the circuit-depth-driven complexity derived in Section 7, which also includes a composed security proof sketch and quantified the cost of upgrading from semi-honest to malicious security. An analytical cost model calibrated to, but not measured from, existing MPC frameworks projected sub-second-to-low-second latencies for representative cohort-discovery, drug-safety, and comparative-effectiveness queries across federations of up to fifteen institutions, under network conditions and cohort sizes spanning two orders of magnitude, and Section 12 outlined a concrete software and deployment architecture for realizing it. The framework's regulatory alignment with HIPAA's minimum-necessary principle and GDPR's data-minimization requirement, together with its honestly stated complexity, threats to validity, and deployment limitations, position it as a candidate architecture for the next generation of cross-institutional healthcare query systems, pending the empirical validation identified as the most direct next step. The through line across Sections 4-12 is that every design choice traces back to a stated, checkable property rather than an unexamined default: the query language is restricted precisely because unrestricted SQL would break the depth-boundedness Section 7's complexity results depend on; the online arithmetic uses additive rather than Shamir sharing precisely because it makes addition free, which Sections 6.4 and 6.5's aggregation and join-assembly operators rely on directly; and the honest-majority threshold is stated as an explicit inequality (Eq. 20) rather than a qualitative assumption, so that a federation's governance body can check it against its own membership rather than take the framework's security on faith. That discipline is what separates an analytically credible architecture from an implementation that merely runs: this paper has aimed for the former, and identified concretely, in Sections 11 and 12, what remains to build the latter.

Acknowledgements
This work was supported by Name of funding agency under Grant BS123456.

Funding
No funding.

Contributions
All authors; Conceptualization, All authors; Investigation; All authors; Writing (Original Draft), All authors; Writing (Review and Editing) Supervision, All authors; Project Administration.

Data Availability Statement
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Use of Generative Artificial Intelligence
The authors declare that no generative artificial intelligence tools were used in the preparation of this manuscript.

Competing interests
All authors declare no competing interests.

References

[1] U.S. Department of Health and Human Services. (2002). Standards for privacy of individually identifiable health information (HIPAA Privacy Rule) (45 C.F.R. Parts 160 and 164, as amended). https://www.hhs.gov/hipaa/for-professionals/privacy/index.html
[2] European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). Official Journal of the European Union, L119, 1–88. https://eur-lex.europa.eu/eli/reg/2016/679/oj.
[3] Weber, G. M., Murphy, S. N., McMurry, A. J., MacFadden, D., Nigrin, D. J., Churchill, S., & Kohane, I. S. (2009). The Shared Health Research Information Network (SHRINE): A Prototype Federated Query Tool for Clinical Data Repositories. Journal of the American Medical Informatics Association, 16(5), 624–630. https://doi.org/10.1197/jamia.m3191
[4] Hripcsak George, Duke Jon D., Shah Nigam H., Reich Christian G., Huser Vojtech, Schuemie Martijn J., Suchard Marc A., Park Rae Woong, Wong Ian Chi Kei, Rijnbeek Peter R., van der Lei Johan, Pratt Nicole, Norén G. Niklas, Li Yu-Chuan, Stang Paul E., Madigan David, & Ryan Patrick B. (2015). Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers. MEDINFO 2015: eHealth-Enabled Health. https://doi.org/10.3233/978-1-61499-564-7-574
[5] Fleurence, R. L., Curtis, L. H., Califf, R. M., Platt, R., Selby, J. V., & Brown, J. S. (2014). Launching PCORnet, a national patient-centered clinical research network. Journal of the American Medical Informatics Association, 21(4), 578–582. https://doi.org/10.1136/amiajnl-2014-002747
[6] Behrman, R. E., Benner, J. S., Brown, J. S., McClellan, M., Woodcock, J., & Platt, R. (2011). Developing the Sentinel System — A National Resource for Evidence Development. New England Journal of Medicine, 364(6), 498–499. https://doi.org/10.1056/nejmp1014427
[7] Dwork, C., & Roth, A. (2014). The Algorithmic Foundations of Differential Privacy. Foundations and Trends® in Theoretical Computer Science, 9(3-4), 211–487. https://doi.org/10.1561/0400000042
[8] Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., Ourselin, S., Sheller, M., Summers, R. M., Trask, A., Xu, D., Baust, M., & Cardoso, M. J. (2020). The future of digital health with federated learning. Npj Digital Medicine, 3(1). https://doi.org/10.1038/s41746-020-00323-1
[9] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175–1191. https://doi.org/10.1145/3133956.3133982
[10] Yao, A. C. (1982). Protocols for secure computations. 23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982), 160–164. https://doi.org/10.1109/sfcs.1982.38
[11] Evans, D., Kolesnikov, V., & Rosulek, M. (2018). A Pragmatic Introduction to Secure Multi-Party Computation. Foundations and Trends® in Privacy and Security, 2(2-3), 70–246. https://doi.org/10.1561/3300000019
[12] Kamm, L., & Bogdanov, D. (2024). Overview of Secure Multi-Party Computation Applications in Health Research and Social Sciences. Handbook of Sharing Confidential Data, 242–256. https://doi.org/10.1201/9781003185284-19
[13] Wettstein, R., Kussel, T., Hund, H., Fegeler, C., Dugas, M., & Hamacher, K. (2022). Secure Multi-Party Computation Based Distributed Feasibility Queries – A HiGHmed Use Case. German Medical Data Sciences 2022 – Future Medicine: More Precise, More Integrative, More Sustainable! https://doi.org/10.3233/shti220802

[14] Tawfik, A. M., Sabbeh, S. F., & EL-Shishtawy, T. (2018). Privacy-Preserving Secure Multiparty Computation on Electronic Medical Records for Star Exchange Topology. Arabian Journal for Science and Engineering, 43(12), 7747–7756. https://doi.org/10.1007/s13369-018-3122-5
[15] Damgård, I., Pastro, V., Smart, N., & Zakarias, S. (2012). Multiparty Computation from Somewhat Homomorphic Encryption. Advances in Cryptology – CRYPTO 2012, 643–662. https://doi.org/10.1007/978-3-642-32009-5_38
[16] Keller, M. (2020). MP-SPDZ: A Versatile Framework for Multi-Party Computation. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 1575–1590. https://doi.org/10.1145/3372297.3417872
[17] Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play ANY mental game. Proceedings of the Nineteenth Annual ACM Conference on Theory of Computing - STOC '87, 218–229. https://doi.org/10.1145/28395.28420
[18] Ben-Or, M., & Wigderson, A. (1988). Completeness theorems for non-cryptographic fault-tolerant distributed computation. Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing-STOC '88, 1–10. https://doi.org/10.1145/62212.62213
[19] Shamir, A. (1979). How to share a secret. Communications of the ACM, 22(11), 612–613. https://doi.org/10.1145/359168.359176
[20] Beaver, D. Efficient Multiparty Protocols Using Circuit Randomization. Advances in Cryptology — CRYPTO '91, 420–432. https://doi.org/10.1007/3-540-46766-1_34
[21] Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Advances in Cryptology — EUROCRYPT '99, 223–238. https://doi.org/10.1007/3-540-48910-x_16
[22] Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic Encryption for Arithmetic of Approximate Numbers. Advances in Cryptology – ASIACRYPT 2017, 409–437. https://doi.org/10.1007/978-3-319-70694-8_15
[23] Demmler, D., Schneider, T., & Zohner, M. (2015). ABY - A Framework for Efficient Mixed-Protocol Secure Two-Party Computation. Proceedings 2015 Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2015.23113
[24] Mohassel, P., & Rindal, P. (2018). ABY3: A mixed protocol framework for machine learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 35–52). Association for Computing Machinery. https://doi.org/10.1145/3243734.3243760
[25] Büscher, N., Demmler, D., Katzenbeisser, S., Kretzmer, D., & Schneider, T. (2018). HyCC: Compilation of hybrid protocols for practical secure computation. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 847–861). Association for Computing Machinery.
[26] Feldman, P. (1987). A practical scheme for non-interactive verifiable secret sharing. 28th Annual Symposium on Foundations of Computer Science (Sfcs 1987), 427–438. https://doi.org/10.1109/sfcs.1987.4
[27] Pedersen, T. P. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. Advances in Cryptology — CRYPTO '91, 129–140. https://doi.org/10.1007/3-540-46766-1_9
[28] Herzberg, A., Jarecki, S., Krawczyk, H., & Yung, M. (1995). Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. Advances in Cryptology — CRYPT0' 95, 339–352. https://doi.org/10.1007/3-540-44750-4_27
[29] Gennaro, R., Jarecki, S., Krawczyk, H., & Rabin, T. (1999). Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Advances in Cryptology — EUROCRYPT '99, 295–310. https://doi.org/10.1007/3-540-48910-x_21
[30] Gkoulalas-Divanis, A., Vatsalan, D., Karapiperis, D., & Kantarcioglu, M. (2021). Modern Privacy-Preserving Record Linkage Techniques: An Overview. IEEE Transactions on Information Forensics and Security, 16, 4966–4987. https://doi.org/10.1109/tifs.2021.3114026
[31] Schnell, R., Bachteler, T., & Reiher, J. (2009). Privacy-preserving record linkage using Bloom filters. BMC Medical Informatics and Decision Making, 9(1). https://doi.org/10.1186/1472-6947-9-41
[32] Stammler, S., Kussel, T., Schoppmann, P., Stampe, F., Tremper, G., Katzenbeisser, S., Hamacher, K., & Lablans, M. (2020). Mainzelliste SecureEpiLinker (MainSEL): privacy-preserving record linkage using secure multi-party computation. Bioinformatics, 38(6), 1657–1668. https://doi.org/10.1093/bioinformatics/btaa764
[33] Cho, H., Wu, D. J., & Berger, B. (2018). Secure genome-wide association analysis using multiparty computation. Nature Biotechnology, 36(6), 547–551. https://doi.org/10.1038/nbt.4108
[34] Bonte, C., Makri, E., Ardeshirdavani, A., Simm, J., Moreau, Y., & Vercauteren, F. (2018). Towards practical privacy-preserving genome-wide association study. BMC Bioinformatics, 19(1). https://doi.org/10.1186/s12859-018-2541-3
[35] Costan, V., Lebedev, I., & Devadas, S. (2017). Secure Processors Part II: Intel SGX Security Analysis and MIT Sanctum Architecture. https://doi.org/10.1561/9781680833034
[36] Ostrovsky, R., & Skeith, W. E. (2007). A Survey of Single-Database Private Information Retrieval: Techniques and Applications. Public Key Cryptography – PKC 2007, 393–411. https://doi.org/10.1007/978-3-540-71677-8_26
[37] Nishide, T., & Ohta, K. Multiparty Computation for Interval, Equality, and Comparison Without Bit-Decomposition Protocol. Public Key Cryptography – PKC 2007, 343–360. https://doi.org/10.1007/978-3-540-71677-8_23