SoK: Systematizing Verifiable and Private Federated SPARQL

Jesse Wright · the sparq project

A systematization-of-knowledge contribution. It asserts no *security, privacy, soundness, or attestation property of any system it describes. The open estate that anchors the systematization is research-grade: its single-prover zero-knowledge layer is* not *externally audited (the external cryptographer review, gate* `sq-qhy4`*, is* ***open****), its MPC layer operates semi-honest honest-majority by default, and its collaborative-proof path is* ***unbuilt*** *— every entry point fails closed. "BUILT" in the capability matrix means* implemented and tested in that research-grade codebase*, never* assured*. The unsolved frontier this SoK isolates — verifying issuer signatures over a secret-shared witness inside one collaborative proof — is reported as of a stated survey date under a stated search protocol (the method section) and is a claim about the* absence of publications*, which time can falsify.*

Abstract

Standard federated SPARQL assumes every endpoint discloses its solution mappings. A growing body of work asks what federated query answering can still mean when that assumption fails — when the data are private, the answer must be *verifiable* by a relying party, and the inputs must be *attributable* to their issuers. The resulting literature is scattered across verifiable outsourced databases, MPC-based collaborative analytics, collaborative zero-knowledge proving, anonymous credentials, and semantic-web data integrity, with no shared frame. We systematize it — under a stated method: explicit inclusion criteria and a dated search protocol (§2) — along three *independent* axes: **cryptographic mechanism** (single-prover zero knowledge over committed graphs; MPC evaluation over secret-shared graphs; attested-input binding), **prover topology** (one witness holder vs. $N \geq 2$ mutually distrusting holders), and **adversary model** (semi-honest, covert, malicious — crossed with output guarantees under Cleve's impossibility bound). The axes are assembled from established MPC and collaborative-proof taxonomy; the frame's value is their joint application to federated SPARQL and the empty cells it exposes. Within this frame we contribute: an operator-level capability matrix over SPARQL, cross-referenced to the published systems, exposing a *two-regime split* — operators over disclosed terms are recomputed by the verifier crypto-free, while only hidden-regime operators pay for cryptography — and the structural observation that *global IRIs are public cross-credential join keys*, a cost advantage the RDF data model holds over relational and property-graph MPC systems that carries a privacy price we analyze rather than discount: the cleartext join discloses exactly the cross-source linkage that the end-state goal of source-unlinkability protects; a catalogue of *settled negatives* (fairness without honest majority, the post-quantum boundary of discrete-log attestations and proof systems, blank-node cross-graph joins, in-circuit entailment) carefully distinguished from cells that are merely *vacant* in the literature; and the isolation of the open frontier — producing one source-unlinkable collaborative proof in which each holder's issuer signature is verified *inside* the circuit over a secret-shared witness, a composition of separately solved ingredients that, as of the stated survey date (re-checked at revision), no publication in the peer-reviewed venues or preprint archives we searched instantiates. The systematization is anchored in one open research-grade estate implementing the single-prover lane and a semi-honest MPC operator suite; its capability tiers are self-reported from that estate's own records (a stated independence limitation, §2), we report its capabilities and its gaps with equal precision, and we claim no security property: the estate is not externally audited and its collaborative-proof path is unbuilt and fail-closed.

1. Introduction

SPARQL 1.1 Federated Query standardizes a world in which federation is a *disclosure* arrangement: the planner decomposes a query, ships subqueries to remote endpoints, and every endpoint returns

its solution mappings in cleartext. Three decades of database security research make the obvious next question precise. What happens when the endpoints hold data they cannot disclose (confidentiality), when the querier cannot trust the endpoints to evaluate honestly (correctness), and when the answer is only meaningful if the underlying data are attributable to accountable issuers (attestation)?

Each of those three requirements has its own mature literature — secure multi-party computation, verifiable/outsourced query processing, and digital signatures with anonymous credentials respectively — and each literature has begun to touch query languages. But the combined problem, *verifiable and private federated SPARQL*, sits at an awkward intersection: the systems that hide inputs do not prove answers, the systems that prove answers do not federate witnesses, and the systems that verify signatures assume a single message holder. Papers from the database, security, and semantic-web communities describe fragments of the same design space in mutually unintelligible vocabulary.

This SoK imposes one frame on that space. Our starting observation is that the design space is three-dimensional, not two-dimensional: the *cryptographic mechanism* (how a statement is proven or a value is hidden), the *prover topology* (who holds the witness), and the *adversary model* (what deviation the protocol tolerates) vary independently, and conflating any two of them — as informal discussion routinely does — produces category errors. A single-prover zero-knowledge system and an MPC evaluation engine are not competing points on one axis; they answer different questions, and a complete federated design needs both, plus a third ingredient (attested-input binding) that is orthogonal to each.

Our second observation is that SPARQL is not merely another query language to which existing relational results port unchanged. The RDF data model gives federated cryptography a structural gift: *global IRIs*. Cross-credential joins in the common case are equi-joins on IRI-valued terms that are already public, so they need no cryptography at all — a "disclosed-key join" regime with no analogue in the relational MPC systems, whose join keys are private values, or the property-graph systems, whose node identifiers are store-local. The gift is priced, not free: executing that join in cleartext discloses the cross-source linkage it computes — in a credential federation, that two sources hold credentials about the same subject — and §4 analyzes that price against the source-unlinkability end-goal instead of discounting it. The flip side is a structural *negative* with no relational analogue either: blank nodes cannot serve as cross-graph join keys at all, by the semantics of per-graph canonicalization, and a correct system must *exclude* such joins rather than approximate them.

Our third observation is methodological. In a domain where the gap between "designed", "implemented", "tested", and "audited" is precisely where past overclaims have lived, a systematization is only useful if its capability vocabulary is honest. We therefore anchor the matrix in one open estate whose specifications carry per-section implementation-status labels and whose unimplemented surfaces fail closed with gate-naming errors, and we adopt its discipline: BUILT means merged and tested (not audited); KNOWN means a published protocol exists but is not integrated; OPEN means no off-the-shelf protocol addresses the cell; and a settled NEGATIVE means the cell is closed by proof or by semantics, which is a different thing from a cell nobody has published into.

Contributions. (1) A systematization method and a three-axis frame for the space (§2, §3). We position the frame honestly: the axes themselves are assembled from *established* taxonomy — the adversary × threshold × output-guarantee grid is classical MPC vocabulary, and the single-vs-collaborative prover distinction comes from the collaborative-SNARK line — so the frame's contribution is the joint application to federated SPARQL and the empty cells it makes visible, not the axes (§12 engages the adjacent SoK lineage). The observations we believe are genuinely new are (2) and (3): (2) the two-regime split, the disclosed-key-join observation, *and an explicit analysis of its privacy price* (§4); (3) an operator-level capability matrix over SPARQL, cross-referenced to the published systems at the granularity their publications support (§5); (4) an analysis of each mechanism class (§6–§8); (5) a catalogue of settled

negatives distinguished from vacant cells, and a method-bounded statement of the unsolved frontier with the reasons it is hard (§9–§10); (6) an honest evaluation of the anchor estate — what exists, what fails closed, and what no one may yet claim (§11).

2. Systematization method and scope

An SoK whose headline frontier claim is an *absence* claim owes the reader its method before its taxonomy: an absence claim is only as strong as the search that failed to find the thing.

Search protocol. The survey underlying this SoK was conducted as four thematic sweeps — (i) relational MPC query engines and MPC primitives; (ii) cryptographic graph, SPARQL, and property-graph query systems; (iii) collaborative zero-knowledge proving, authenticated-input MPC, and signature verification inside and outside circuits; (iv) anonymous credentials and semantic-web data integrity — over the public indices IACR ePrint, arXiv, and dblp and the venue proceedings they index (IEEE S&P, ACM CCS, USENIX Security, NDSS, PVLDB/VLDB, SIGMOD, NSDI/SOSP/EuroSys, EuroCrypt/Crypto/Asiacrypt, WWW/ISWC/ESWC, and W3C TR), seeded from the known systems in each literature (SMCQL, Conclave, ZKSQL, the Ozdemir–Boneh collaborative SNARKs, GOOSE, Dutta et al.) and from the Bontekoe et al. survey of verifiable MPC (arXiv 2309.08248), and expanded by forward- and backward-citation snowballing. The sweeps carry stated dates: the distributed-signature feasibility sweep completed **2026-06-13** — the date every absence claim in this paper is indexed to — and the operator-capability reconciliation on **2026-06-16**. At revision time (**2026-07-02**) the frontier absence claim was re-checked with the queries "collaborative zk-SNARK signature verification over a secret-shared witness" and "MPC federated SPARQL attested inputs issuer signature" over the same indices; the re-check surfaced one further adjacent line (collaborative commit-and-prove NIZKs, §10) and no instantiation of the composition.

Inclusion criteria. A system enters Table 1 if it (i) evaluates a declarative query language (SQL, SPARQL, Cypher, or graph traversal) or proves general circuit relations directly usable for such evaluation; (ii) derives its guarantee from cryptographic assumptions among the participants — not from a hardware trust root and not from access-control policy; and (iii) is published at an identifiable peer-reviewed venue or maintained preprint with enough protocol detail to place it on all three axes. One exception is declared: the TACEO co-snark tooling is included despite having no venue publication because it targets the anchor estate's exact proof stack; it is marked unaudited in the table. Patent and product literature is excluded by scope — apparatus claims without security analyses or artifacts cannot be placed on the adversary axis — so every absence claim here is a claim about peer-reviewed venues and open preprint archives, not about all disclosure channels.

Exclusions, with justification. Four adjacent lines are engaged in §12 but excluded from the axes. *TEE/enclave query processing* (Opaque; ObliDB; Cipherbase) replaces cryptographic assumptions among holders with a hardware vendor trust root — a different object than axis 1 classifies, and one whose attestation soundness is empirically eroding under physical attack. *Client–server encrypted-database querying* (Blind Seer) is cryptographic but answers a two-party problem — a querier's privacy against one database owner — not federation among N mutually distrusting data holders. *Encrypted-at-rest RDF and access-control query rewriting* enforce policy against a known querier and offer no query-time cryptographic guarantee among holders. *Differential privacy for query answering* is an output-side leakage mitigation, not an evaluation mechanism; it composes with the mechanisms systematized here and appears in the matrix's result-size mitigations.

Provenance and independence — the self-sourcing caveat, stated before the matrix uses it. The operator-level tiers for the anchor estate (§5) trace to the estate's own specification corpus and reconciled capability review — documents authored *inside the same project as this SoK*. They are self-reported and independently unverified: no external audit of any lane has occurred (the single-prover

review gate `sq-qhy4` is open). The estate's discipline mitigates but does not remove this: per-section implementation-status labels, fail-closed `NotYetImplemented` stubs, and build-time evidence gates make overclaim structurally difficult, and every tier names its artifact. The reader should therefore weigh the paper's two evidentiary registers separately: *for the anchor estate* this SoK has the status of a system report with self-reported capability (negative space included); *across the published systems* it is a documentary systematization from their publications. The field-level contributions — the frame, the regime split and its price, the negative/vacant separation, the frontier isolation — stand on the published literature; the matrix's BUILT tiers stand only on the estate's records.

What this SoK is not. Not a security evaluation or audit (no property below is claimed proven); not a performance study (no wall-clock measurement appears — the few counts cited are deterministic structural facts injected at build time from a gated evidence file, and published systems' costs are described qualitatively); and not a claim that any system, including the anchor estate, offers production-grade confidential federated SPARQL today — none does, and the estate's own documentation says so plainly.

3. The systematization framework: three axes

3.1. Axis 1 — cryptographic mechanism: how the statement is proven

(a) Single-prover zero knowledge. One party holds the full witness — here, committed RDF graphs — evaluates a SPARQL fragment itself, and emits a non-interactive proof manifest that a verifier checks against externally supplied trust anchors without seeing the witness. In the anchor estate this is the zkSPARQL lane: graphs are canonicalized with RDFC-1.0, committed via Poseidon2 hashes, and statements are proven in Noir circuits under the Barretenberg UltraHonk backend over BN254. The mechanism answers *correctness plus confidentiality against the verifier*, for one holder.

(b) MPC evaluation (input-confidential). $N \geq 2$ mutually distrusting holders secret-share their private RDF inputs — Shamir sharing over the Mersenne field $\mathbb{F}_p$, $p = 2^{61} - 1$, chosen for native 64-bit arithmetic — run SPARQL operators collaboratively over shares, and reconstruct only an agreed disclosed output. The mechanism answers *confidentiality among the holders*. It does not, by itself, produce a transferable proof: in the anchor estate the pipeline driver assembles a `ProofStatement` shape, but the collaborative prove/verify entry points are honest `NotYetImplemented` placeholders.

(c) Attested-input binding. Orthogonally to (a) and (b), each contributing graph commitment must be bound to an *issuer signature* — in the estate, Schnorr over Baby Jubjub with a Poseidon2 challenge — so that the statement is not merely "some data satisfy the query" but "data *vouched for by these issuers* satisfy the query". In the single-prover setting this binding is enforced verifier-side (`bind_issuer_attestations`) against an external trusted key-set K. In the federated setting, binding N holders' MPC inputs to their respective issuer signatures *inside one proof* is the unsolved frontier of §10.

The three mechanisms are complements, not competitors: (a) and (b) trade off on topology, and (c) composes with either. A federation design that picks only one mechanism silently drops one of the three requirements (correctness, confidentiality, attestation).

3.2. Axis 2 — prover topology: who holds the witness

(a) Single prover. One holder controls all committed graphs; the witness is unified. Everything in the zkSPARQL lane's implemented fragment lives here — including its cross-*credential* joins, which join graphs held by the *same* prover.

(b) Collaborative / federated. $N \geq 2$ mutually distrusting holders each hold a private graph; no party may see the union witness. Evaluation must be MPC over shares, and a transferable proof of the

federated result would require a *collaborative* zk-SNARK over the shared witness. In the anchor estate this is the MPC-SPARQL lane: the evaluation tier (M0–M3) is implemented; the collaborative-proof tier (M4 and beyond) is open.

The axis matters because mechanism results do not transfer across it. Signature-verification gadgets, anonymity-set proofs, and proof-of-knowledge tiers designed for topology (a) assume the message is held by one party; under topology (b) the message itself is secret-shared and every non-linear step becomes multi-party communication (§10).

3.3. Axis 3 — adversary model: what the protocol tolerates

(a) Semi-honest (passive). All parties follow the protocol but may learn what they can from their view. This is the default and the implemented tier in the anchor estate's MPC layer: honest-majority Shamir with BGW degree reduction.

(b) Covert. Cheating is detected with some probability ε; the guarantee deters rather than prevents. Present in the estate's security-model vocabulary (`AdversaryModel::Covert`) and unimplemented — an honest vocabulary slot, not a capability.

(c) Malicious (active). Arbitrary deviation is tolerated. In the estate this tier is *partially* built: robust Berlekamp–Welch reconstruction and IT-MAC authenticated sharing with MAC-checked-before-open disclosure exist, while the full honest-majority malicious-with-abort suite remains open work. The operating default remains semi-honest.

The adversary axis is crossed with an *output-guarantee* sub-axis (abort, fairness, guaranteed output delivery) and a *corruption-threshold* sub-axis, and the cross is constrained by a theorem: Cleve's impossibility (STOC 1986) implies fairness and guaranteed output delivery require an honest majority. The estate encodes this as a type-level invariant — the output-guarantee constructors for fairness and guaranteed output return nothing under a dishonest-majority threshold — so no implementation can silently downgrade the theorem to a configuration option. We highlight this as a systematization lesson: *impossibility results belong in the type system*, where a refactor cannot argue with them.

3.4. Why three axes, and the cost of conflating them

None of the three axes is our invention, and we do not present them as one. Axis 3 — adversary model crossed with corruption threshold and output guarantee — is the classical MPC security vocabulary; axis 2 is the single-vs-collaborative prover distinction the collaborative-SNARK line introduced; axis 1′s mechanism classes are each a mature literature. What the field lacks, and what this frame supplies, is their *joint* application to federated SPARQL — the discipline of naming every system's position on all three coordinates at once, in the SoK tradition that Hastings et al. and Viand et al. applied to MPC and FHE compilers (§12).

Published systems routinely collapse the grid. "MPC query engine" conflates mechanism (b) with adversary (a) — most graph MPC systems are semi-honest only, and their guarantee evaporates, not degrades, under an active adversary. "Verifiable database" conflates mechanism (a) with topology (a) — the ZKSQL lineage proves answers over a *single* owner's commitment. "Collaborative proof" names a topology while leaving attestation unaddressed — the collaborative zk-SNARK line proves relations over shared witnesses that nobody has signed. The three-axis frame makes such gaps visible as *empty cells* rather than letting adjacent-cell results borrow credibility. Table 1 places the systems we survey.

System	Data model	Mechanism	Topology	Adversary	Guarantee surface
IntegriDB (CCS'15)	SQL	authenticated data structures	single (outsourced)	malicious server	answer correctness; *no* record hiding

System	Data model	Mechanism	Topology	Adversary	Guarantee surface
vSQL (S&P'17)	SQL	verifiable computation	single (outsourced, dynamic)	malicious server	answer + update correctness; *no* record hiding
ZKSQL (PVLDB'23)	SQL	single-prover ZK	single	malicious prover	answer correctness with records hidden
SMCQL (PVLDB'17)	SQL fragment	MPC compilation	2-party federation	semi-honest	input confidentiality
Conclave (EuroSys'19)	SQL	MPC + cleartext pre/post	federated	semi-honest	input confidentiality; disclosed/hidden routing
Senate (USENIX Sec'21)	SQL analytics	MPC	n-party federation	malicious	input confidentiality among parties
Cerebro (USENIX Sec'21)	analytics platform	MPC + policy/audit layer	federated	configurable	platform governance around the computation
Secrecy (NSDI'23)	SQL analytics	MPC, composable oblivious operators	federated (3 compute servers)	semi-honest, honest-majority 3PC	input confidentiality via oblivious-operator plans
ORQ (SOSP'25)	SQL (full TPC-H)	MPC, oblivious operators	federated	semi-honest and malicious variants	input confidentiality at relational scale
GOOSE (DBSec'20)	SPARQL UCRPQ fragment	MPC w/ honest broker	federated	honest-but-curious	confidentiality; no aggregates
SMPG / PPMQ	Cypher (Neo4j)	Shamir via JIFF	federated	semi-honest	confidentiality; conjunctive SPJ only
GORAM (VLDB'25)	property graph, egocentric	ORAM on 3PC (ABY3)	federated (3 servers)	semi-honest, honest-majority	confidentiality at billion scale; *no* correctness or attestation
Ozdemir–Boneh (USENIX Sec'22)	circuit relation	collaborative zk-SNARK	collaborative	up to malicious minority / $N-1$	one proof over a shared witness
Scalable coZK (USENIX Sec'25)	circuit relation	collaborative ZK, delegated	one client's witness, many helpers	semi-honest helpers	proof delegation at scale
TACEO co-snarks (coNoir)	Noir circuit relation	collaborative UltraHonk	collaborative	stated per protocol; *unaudited*	tooling; matches the estate's stack
Dutta et al. (Asiacrypt'24)	circuit relation	authenticated-input MPC	federated	honest-majority LSSS	inputs bound to attestations; *not* a public proof
Artemis (arXiv 2409.12055)	circuit relation	commit-and-prove SNARK	single prover	malicious prover	proof bound to external commitments
zkSPARQL lane (anchor estate)	RDF / SPARQL fragment	single-prover ZK + attested inputs	single	malicious prover, unaudited verifier	*targets* correctness + hiding + issuer binding over its implemented tiers only — the holder-PoK and holder-set tiers are self-labeled **not yet sound** and excluded from this cell; *research-grade, external audit gate open, nothing proven*

System	Data model	Mechanism	Topology	Adversary	Guarantee surface
MPC-SPARQL lane (anchor estate)	RDF / SPARQL operators	MPC evaluation	federated	semi-honest default, honest majority	input confidentiality; *no* collaborative proof (fail-closed stubs)

Table 1: The surveyed systems (selected per the inclusion criteria of §2) placed on the three axes. The systematization verdict is the *pattern of empty cells*: every surveyed graph/SPARQL cryptographic system is semi-honest and/or confidentiality-only and/or conjunctive-fragment-only; within the stated search scope no published system occupies the cell (full-SPARQL federation) × (attested inputs) × (security against active adversaries), and the collaborative-proof systems prove relations over witnesses that carry no issuer attestation. Cost remarks are qualitative by design; this SoK cites no wall-clock number.

4. The two-regime split and the disclosed-key advantage

The single most consequential engineering observation in the space is not cryptographic but data-model-shaped. SPARQL operators divide into two regimes:

- **Disclosed regime.** Operators whose inputs are public — or become public by policy — are recomputed by the verifier (or evaluated in cleartext by the federation) *crypto-free*. No proof is needed for what the checker can redo.
- **Hidden regime.** Operators over values that must stay hidden pay for cryptography: MPC rounds, oblivious data structures, or in-circuit constraints.

The routing of a query plan across this split — pushing maximal work into the disclosed regime, as Conclave did for relational MPC with annotated pre/post-processing — dominates end-to-end cost long before any cryptographic optimization matters. The anchor estate adopts exactly this split, with disclosed/hidden operator routing ratified against a dual leakage envelope and a default-deny privacy-aware source-selection stage in front of it.

RDF sharpens the split in a way relational and property-graph data models do not. The common federated join — "this credential's subject is that credential's subject" — is an equi-join on a *global IRI*, and global IRIs are public identifiers by construction. The join key is already disclosed even when everything else about both graphs is hidden, so the join is a cleartext hash join over commitments' disclosed keys: the *disclosed-key join*. In the relational MPC systems the join key is a private attribute and every equi-join is a hidden join; in the property-graph systems (Cypher on Neo4j; ego-centric traversal stores) node identifiers are store-local and cannot serve as cross-store keys at all. The capability matrix below (§5) records the consequence: the flagship federated SPARQL capability — joining W3C Verifiable Credentials across holders on their subject IRIs — is BUILT and crypto-free, while the hidden-value join that relational systems must always pay for is the expensive fallback, not the common case. That is the cost claim; it is not a privacy claim, and the distinction needs the same precision as the saving.

The privacy price of the disclosed key. In a credential federation the join key is typically the data subject's identifier — the join *is* "this credential's subject is that credential's subject". Executing it in cleartext therefore discloses, to every party that observes the join, three things: (i) the key values themselves — which subjects occur in each source; (ii) the *cross-source linkage fact* — that two sources each hold a credential about the same subject — which can be the sensitive datum even when every attribute stays hidden (that one holder's graph and an oncology clinic's graph share a subject is a disclosure in itself); and (iii) row-level source provenance — which holder contributed which row. The estate's own leakage taxonomy names (iii) as its L4 channel and is explicit that the only known mitigation is in-circuit unlinkable attested-source membership — precisely the unsolved frontier of

§10. The tension must be stated flatly: the disclosed-key join's leakage is *exactly the linkage* that the end-state goal of a source-unlinkable federated proof exists to protect. The two-regime split is therefore a *leakage-priced routing decision*, not a cryptographic free lunch. The disclosed-key join is the right default only under a policy in which the subject identifiers and the linkage relation are themselves disclosable to the federation — the anchor estate ratifies its routing against a dual leakage envelope with default-deny source selection in front for this reason — and where the linkage is the secret, the disclosed-key join is not an optimization but a violation, and the hidden-value join (or the unbuilt in-circuit composition) is the only sound routing. The honest summary of RDF's structural gift: global IRIs move the *cost* of the common federated join out of the cryptographic regime *when its linkage disclosure is acceptable by policy*; they do not dissolve the conflict between cleartext joins and unlinkability — they make it legible enough to route on.

The same data model also supplies the corresponding *negative*. Blank nodes are scoped to their graph; per-graph RDFC-1.0 canonicalization does not and cannot align blank-node labels across separately committed graphs. A join solution binding a shared variable to a blank node in more than one committed graph is therefore *excluded from the semantics* — `eval(Join(P1, P2))` simply does not contain it — rather than being a missing feature (§9).

5. A capability matrix over SPARQL

We tier every cell as **BUILT** (implemented and tested in the anchor estate's crates on its main branch — research-grade, *not* audited), **KNOWN** (a published protocol exists; not integrated), **OPEN** (no off-the-shelf protocol addresses the cell), or **NEGATIVE** (settled: by theorem, by semantics, or by an explicit not-yet-sound label). The matrix reflects the estate's reconciled capability review (2026-06-16) and the digested specification corpus; its provenance is documentary *and self-sourced* (§2), and no cell is a security claim. To make the matrix a cross-system instrument rather than a single-system report, the final column places the published systems of Table 1 against each operator — at the granularity their publications support, which is usually a query-suite or language fragment rather than an audited operator: that granularity mismatch is the honest ceiling of a cross-system matrix built from publications, and it is listed as a limitation (§13).

SPARQL operator (hidden-regime unless noted)	Regime	Tier	Basis (anchor estate, self-reported)	Published systems in the cell
BGP single-pattern equality FILTER	hidden	BUILT	equality-to-zero: one multiplication + one open; only the match bit is opened	equality predicates are standard in the relational MPC engines (SMCQL, Conclave, ORQ); GOOSE matches patterns via an honest broker
Inner JOIN on a disclosed global-IRI key	disclosed	BUILT	crypto-free cleartext hash join (`DisclosedKeyJoin`) — the flagship federated case; leaks key values, cross-source linkage, and row provenance (L4) — priced in §4	no published analogue — relational join keys are private attributes; Conclave's annotated cleartext routing is the nearest relative
Inner JOIN on a hidden value	hidden	BUILT / KNOWN	naive all-pairs $O(\|L\| \cdot \|R\|)$ built, incl. a full-obliviousness variant that never opens per-pair match bits; $O(n \log n)$ sort-merge and	the relational cost center: oblivious joins in SMCQL, Senate, Secrecy, ORQ; two-party key-on-key PSI (circuit-

SPARQL operator (hidden-regime unless noted)	Regime	Tier	Basis (anchor estate, self-reported)	Published systems in the cell
			circuit-PSI are published but not integrated	PSI, VOLE-PSI) does not compose into multi-way BGPs
SUM / COUNT (linear aggregate)	hidden	BUILT	zero multiplication rounds (local share addition); only a threshold verdict bit is disclosed, not the aggregate	standard across relational MPC (SMCQL, Conclave, Senate, ORQ); GOOSE excludes aggregates
FILTER range comparison (less-than / greater-than / BETWEEN)	hidden	BUILT	semi-honest secure comparison; full-field Rabbit-style bit decomposition; only the verdict bit opens; malicious hardening in flight	secure comparison is standard in the relational engines; Rabbit-style protocols published
Aggregate threshold over a hidden SUM	hidden	BUILT	in-MPC decomposition of the sum (no reconstruct); no party learns the mask; range bounded below 2^{60}	verdict-only disclosure is a policy atop standard comparison; the surveyed engines disclose the aggregate value itself
UNION (oblivious shuffle)	hidden	BUILT (primitive)	Waksman/Beneš permutation network built; the concat-and-pad UNION wrapper not yet wired	oblivious shuffles are standard components of oblivious relational plans (Secrecy, ORQ)
DISTINCT over hidden keys	hidden	BUILT	oblivious sort under a never-opened secure comparator, adjacent-equality keep-bits, oblivious compaction — no key is ever opened	oblivious sort-then-dedup is standard relationally (Secrecy, ORQ)
ORDER BY over hidden keys	hidden	KNOWN (primitives BUILT)	the sort network and secure comparator exist as primitives; the secret-key ORDER BY operator integration does not	oblivious sort networks are standard relationally (Secrecy, ORQ)
OPTIONAL (left join) / MINUS (anti-join)	hidden	KNOWN (primitives BUILT)	degree reduction landed; awaits the sort-merge join integration	outer and anti joins appear in ORQ's full-TPC-H coverage; no SPARQL-side instance
Bounded property paths, disclosed keys	disclosed	BUILT	bounded-length path evaluation over disclosed global-IRI keys	no published analogue — property-graph systems have no global keys to disclose
Bounded property paths, hidden edges	hidden	KNOWN (primitives BUILT)	primitives exist; operator integration remaining	GORAM traverses bounded ego-neighborhoods on 3PC; GOOSE covers a UCRPQ path fragment via its broker
Unbounded property paths over a secret edge set	hidden	OPEN / NEGATIVE at scale	transitive closure leaks structure at every fixpoint iteration; scoped to bounded length by design	no published system offers hidden-edge unbounded fixpoints; GORAM re-

SPARQL operator (hidden-regime unless noted)	Regime	Tier	Basis (anchor estate, self-reported)	Published systems in the cell
				stricts to ego-centric bounded traversal by design
String functions, regex FILTER	hidden	OPEN / KNOWN	needs a Boolean-circuit backend (GMW or garbled circuits) — the wrong family for a Shamir arithmetic core	Boolean-circuit MPC covers string predicates in the two-party SQL compilers' backend family; no graph-side instance
GROUP BY on a hidden key	hidden	KNOWN (primitives BUILT)	shuffle + comparator primitives built; operator integration remaining	sort-based oblivious grouping relationally (ORQ); Shrinkwrap pads group cardinalities under differential privacy

Table 2: The operator-level capability matrix for MPC evaluation of SPARQL over secret-shared RDF. The anchor-estate tiers are self-reported documentary provenance (§2) — BUILT means merged-and-tested research code under an open audit posture, never an assured property. The final column is the cross-system placement, at publication granularity. The two-regime split is visible structurally: the disclosed rows are crypto-free (and leakage-priced, §4), and the expensive rows are exactly the hidden-regime ones.

The single-prover fragment. The zkSPARQL lane proves, in zero knowledge against a committed witness: BGP scans over committed graphs; value FILTER constraints bucketed by datatype lane (integer, `xsd:double`, signed integer, `xsd:decimal`, and a value-dictionary lane); and a single-prover equality JOIN across distinct committed graphs. OPTIONAL, UNION, property paths, aggregation, and subqueries are *outside* the proven fragment. RDFS/OWL entailment is not proved in-circuit; the verifier re-checks entailment over disclosed bases only (`bind_entailment`) — simple entailment is the only regime proved in zero knowledge. The implementing circuit family is `Scan`, `FilterInt`, `FilterF64`, `FilterSignedInt`, `FilterDecimal`, `FilterValueDl`, `RevokeUnset`, `HiddenIssuer`, `HolderPok`, `HolderSet`, and `JoinEq`; circuits are authored in Noir and proved with Barretenberg (pinned at bb 5.0.0-nightly.20260324 / nargo 1.0.0-beta.21). The pinning is itself a conformance caveat the estate states plainly: the public-input byte layout is determined empirically by the toolchain rather than specified independently of it, so cross-version interoperability is not promised.

The blank-node negative, and its enforcement seam. As §4 noted, cross-graph joins on blank nodes are excluded by the semantics, and the estate encodes the exclusion structurally (the "Q6" guard). The estate's own composition review then found a namespace ambiguity worth systematizing as a lesson: the guard keys on a *scan-local graph index* rather than the canonical commitment identifier, so for cross-scan joins assembled from separate single-graph scans the gate is *inert*; the live backstop is a separate salt-separation audit gate (`SaltReused`) in attestation binding, and a hardening item is tracked. The lesson generalizes: *a settled semantics negative still needs an audited enforcement seam* — proving that a case is excluded on paper does not make the runtime check that excludes it well-scoped.

6. Mechanism class I: single-prover zero-knowledge SPARQL

The single-prover lane is the most completely realized mechanism class, and its architecture is worth systematizing because it is the shape any comparable system converges on.

Commit. Each RDF graph is canonicalized (RDFC-1.0) and committed with a circuit-friendly hash (Poseidon2). Canonicalization-then-commit is what lets the commitment track graph *isomorphism* classes rather than serializations — with a qualification that must be stated, because the property is soundness-flavored: canonical labeling of graphs with rich blank-node automorphisms is hard in general, and RDFC-1.0 handles adversarial "poison" inputs by complexity-bounded *rejection* rather than by canonicalizing them, so the well-defined-over-isomorphism-classes property holds for the inputs RDFC-1.0 accepts within its bounds, not unconditionally. The same per-graph scoping of blank-node identity is precisely what produces the join negative above.

Prove. The holder evaluates the query fragment and produces a proof manifest from the circuit family over BN254 (UltraHonk). The fragment is deliberately small (scans, laned value filters, equality join); everything else stays out of the circuit rather than being approximated inside it.

Verify, fail-closed. The verifier is a twelve-obligation pipeline (the `bind_*` obligations) with an extensive fail-closed error taxonomy: public-input reconstruction, external-anchor attestation binding against a trusted key set, entailment re-check over disclosed bases, revocation and freshness discipline, and a single-use burn-on-mismatch nonce. Two verifier tiers — holder proof-of-knowledge and holder set membership (`HolderPok`, `HolderSet`) — are explicitly labeled *not yet sound* in the implementation and documentation, and are opt-in only; we record this as a NEGATIVE-tier label the estate applies to itself.

What the class cannot do. Its witness is one party's. The moment two mutually distrusting holders must contribute private graphs to one statement, the class's gadgets stop applying (§10). Notably, the estate has no single-prover Schnorr-over-Baby-Jubjub verification gadget in-circuit either — its `Scan` circuit takes the commitment as a *public input* and never sees an issuer key; issuer binding lives verifier-side (§8). That design choice is defensible single-prover and becomes the crux federated.

The honest status line for the whole class: research-grade, internally reviewed, *not externally audited* — the external cryptographer review is the open gate `sq-qhy4`, and until it closes no soundness-adjacent property of this lane may be claimed, including by this SoK.

7. Mechanism class II: MPC evaluation over secret-shared graphs

The MPC lane implements the hidden-regime rows of the capability matrix over honest-majority Shamir sharing in $\mathbb{F}_p$, $p = 2^{61} - 1$, with BGW degree reduction, CSPRNG masking (ChaCha20), robust Reed–Solomon reconstruction (Berlekamp–Welch), and an IT-MAC authenticated-sharing foundation with MAC-checked-before-open disclosure. Three design decisions are systematization-worthy:

- **Verdict-bit disclosure discipline.** Wherever the use case permits, the protocol discloses a *decision bit*, not a value: the range FILTER opens only its verdict; the aggregate threshold decomposes the hidden sum in-MPC and discloses only the comparison outcome. The leakage envelope is thereby stated per-operator and kept minimal by construction rather than by post-hoc argument.
- **The security model as a first-class vocabulary.** Every operator reports its position on the three-axis sub-grid (adversary model × output guarantee × corruption threshold), and the Cleve constraint is a type-level invariant. A federation-facing registry *fail-closes* on requests for cells no protocol supports (e.g., dishonest-majority malicious evaluation), rather than serving the nearest weaker cell.
- **Simulation-tier transport, said plainly.** The wire layer is a star-coordinator, length-prefixed-frame protocol with field elements as 64-bit big-endian integers, run over a loopback network transport that measures real bytes and rounds; the estate's own documentation states the star coordinator *must not be deployed* and that pairwise authenticated channels are unspecified. Costs are tracked with a deterministic communication counter (rounds and bytes), which is the only

benchmark tier the estate treats as canonical evidence; wall-clock numbers from the loopback tier are labeled indicative, and this SoK cites neither.

An end-to-end federated pipeline driver exists (holder → share → join → secure threshold → `ProofStatement`), exercised by a running four-party example — the "four flatmates" scenario: the parties learn whether their combined salaries clear a threshold, and nothing else. What does *not* exist is any strong proof for that pipeline's output: the `ProofStatement` names the claim, and the collaborative prove/verify behind it are fail-closed stubs, with the `Proof` and `AttestationShare` types deliberately left without fields and the proof system unchosen. We regard this as the honest architecture for the current state of the art: the shape of the missing object is declared, and every path to overclaiming it is a compile error or a gate-naming runtime error.

8. Mechanism class III: attested-input binding

Attestation is the least systematized of the three mechanisms in the literature, and the one federated SPARQL cannot do without: a federated answer over unattributed data is an answer over data someone may have minted for the query. The estate's realized form is verifier-side: each graph commitment carries an issuer signature (Schnorr over Baby Jubjub, Poseidon2 challenge), and `bind_issuer_attestations` checks every contributing commitment against an externally supplied trusted key set K, with salt-separation auditing among its gates. Two properties of this design deserve emphasis.

First, *trust anchors are exogenous*. The verifier takes K from the relying party rather than from the proof, so the attestation statement is "signed under a key *you* trust", not "signed under a key the prover likes". Second, *binding composes with either evaluation mechanism in principle, but at different costs*: verifier-side binding is cheap for the single-prover lane (the commitments are public inputs) and remains available to the MPC lane only by *moving the check outside the collaborative computation*, which is exactly the interim M4 design below — at the price of source-unlinkability. Pulling the check *inside* a collaborative proof is the frontier.

The anonymous-credentials literature (CL signatures; BBS/BBS+; the bbs-2023 / ecdsa-sd-2023 Data-Integrity cryptosuites) solves an adjacent problem — selective disclosure of attribute subsets under issuer signatures — and the estate's relation to it is generalization, not replacement: instead of disclosing a subset of signed attributes, the holder proves that a SPARQL query over the signed data evaluates as claimed. Credential-level selective disclosure remains useful below the query layer, and BBS-2023 ingest into the estate is explicitly deferred (no in-repo BBS verifier). A VC cryptosuite bridge for eddsa-rdfc-2022 and ecdsa-rdfc-2019 ingest exists on a feature branch, unmerged and opt-in.

9. Settled negatives

A SoK in this space earns its keep by separating three things the literature blurs: cells *closed by proof or semantics*, cells *closed by an honest self-label*, and cells that are merely *vacant*. The distinction matters because vacant cells invite research, while closed cells invite type-level enforcement.

Closed by theorem. Fairness and guaranteed output delivery without an honest majority are impossible (Cleve, STOC 1986). Recorded in the estate as a type-level invariant, not a configuration default (§3).

Closed at the post-quantum boundary — stated per component, because the boundary cuts through the stack. Every signature scheme in scope — Schnorr over Baby Jubjub, EdDSA, BBS+ — rests on discrete-log hardness and falls to a Shor-capable adversary; the estate records these as honest negative facts (`PostQuantumForgery`, `PostQuantumSnooping`) in its security-annotation vocabulary. The proof layer falls with the signatures: UltraHonk over BN254 is a pairing-based, discrete-log

polynomial-commitment proof system, so a Shor-capable adversary can forge proofs — and it is *this*, not the graph commitment, that voids the *retrospective* soundness of previously accepted proofs. The graph commitment itself is a survivor: Poseidon2 is a hash, its binding rests on collision resistance, and quantum search speedups (Grover-family) degrade its security level rather than break it — a parameter-sizing question, not a Shor break. Beneath both, the Shamir MPC core is information-theoretic: its confidentiality rests on no computational assumption at all. The boundary is thus asymmetric three ways — the sharing layer survives unconditionally, the hash-commitment layer survives up to parameter sizing, and the exposure concentrates in exactly two places: the attestation signatures and the SNARK proof layer (the present single-prover one, and any future collaborative one that inherits the same stack).

Closed by semantics. Blank-node cross-graph joins (§4, §5): excluded from `eval(Join(P1,P2))` by definition under per-graph canonicalization; the open item is enforcement-seam hardening, not semantics.

Closed by honest self-label. The hidden-holder verifier tiers (`bind_holder_pok`, `bind_holder_set`) are labeled not yet sound and opt-in only; in-circuit RDFS/OWL closure is explicitly deferred, with simple entailment the only regime proved in zero knowledge; and unbounded property paths over a secret edge set are scoped out (every fixpoint iteration leaks structure — bounded length is the honest offer).

Vacant, not closed. Dishonest-majority malicious correctness for SPARQL/graph evaluation at usable performance has *no published instance*; the estate's registry fail-closes on the request. We deliberately file this under vacant-with-a-fail-closed-guard rather than impossible: no theorem closes the cell, and the honest statement is that nobody has published into it.

10. The unsolved frontier: attestation inside a collaborative proof

The end-state federated design is easy to state and, as of the survey date this paper's absence claims are indexed to (2026-06-13, re-checked 2026-07-02 with the stated queries; protocol and scope in §2), absent from every peer-reviewed venue and preprint archive we searched: verify each holder's issuer signature over their graph commitment *inside* the collaborative circuit over a secret-shared witness, yielding one source-unlinkable proof that simultaneously establishes (i) the federated SPARQL evaluation is correct, (ii) every input graph is signed under the relying party's trusted key set, and (iii) no verifier or subset of holders learns which holder contributed which graph.

Three ingredients exist; the composition does not.

- *Collaborative proving over a shared witness* exists: the Ozdemir–Boneh line (USENIX Security 2022; eprint 2021/1530) lifts standard SNARK provers into MPC; scalable delegation-flavored coZK exists (USENIX Security 2025; eprint 2024/940); TACEO's co-snarks (coNoir / UltraHonk) match the anchor estate's exact toolchain; and — surfaced by this paper's revision-time re-check — collaborative *commit-and-prove* NIZKs exist (Alghazwi, Bontekoe, et al.; eprint 2024/1209), lifting LegoSNARK-style linking glue to the collaborative setting so distributed witnesses can be bound to commitments across composed proofs. None of these instantiates the proven relation with a signature verification over a secret-shared message.
- *"Signed under one of a set K" exists single-prover*: the ZKPVS / CDLS / ZKAttest line (Di Stefano et al.; NDSS 2024). But the message is held by one party; nothing in this line addresses a multi-prover setting where the message itself is shared.
- *Signing over data the signers do not see* exists (Coconut, NDSS 2019; BBS+ blinded issuance) — but that is the *issuance* side, blinded from issuers, not multi-prover *verification*.

Why the composition is hard. Signature verification is heavily non-linear — elliptic-curve scalar multiplication plus hash-to-field — so evaluating it under MPC over a secret-shared message costs thousands of multiplication gates on shared values, each a communication round in a BGW/SPDZ-style protocol. The hidden-key-set requirement layers an OR/ring structure with cost proportional to $|K|$ on top. Every anonymity-set proof in the literature was designed single-prover; and the anchor estate has no in-circuit Schnorr gadget even in its single-prover lane to lift (its circuits bind commitments as public inputs and never touch issuer keys). The estate's collaborative-proof types are honest `NotYetImplemented` stubs with fieldless `Proof` and `AttestationShare` types; the collaborative proof system is unchosen.

A soundness caveat any attempt must inherit. Garg, Goel, Jain, Roberts, and Sekar (CRYPTO 2025; eprint 2025/1026) show the standard coZK template — a semi-honest MPC prover hardened with an off-the-shelf compiler for malicious security — has exploitable pitfalls: proving over an invalid or inconsistent secret-shared witness can leak honest provers' private inputs. Pre-proof witness validation is therefore a *security obligation*, not an optimization; the estate's federated specification makes it a MUST, and the estate's separate re-audit encodes it as a build-time test obligation. The nearest usable third-party stack (TACEO) is explicitly unaudited and its documentation does not reference the CRYPTO 2025 result — a dependency-risk fact any adopter must weigh.

The honest interim (M4 v1). The estate's designed-but-unbuilt interim sidesteps the unsolved problem by moving attestation checking to the *verifier*: holders provide authenticated inputs after Dutta, Ganesh, Patranabis, and Singh (eprint 2022/1648; Asiacrypt 2024 — the only surveyed work addressing MPC verifiability and input authentication together, honest-majority LSSS only), and the verifier checks a commit-and-prove anchor in the style of Artemis (arXiv 2409.12055) tying the MPC inputs to issuer-signed commitments. The design gives correctness plus attestation — and gives up, explicitly: source-unlinkability (the verifier learns which commitment is whose), a single succinct proof, and malicious security against $N - 1$ corruptions; it further requires out-of-circuit freshness binding. A federated public-input byte layout spanning multiple holders' commitments — prerequisite even for this interim — does not yet exist. We systematize the interim as the honest pattern for the field: *when the composition is unsolved, decompose the guarantee and label what each piece drops*, rather than approximating the end-state and hoping.

11. Honest evaluation: what exists, what is claimed, and what is not

The evaluation discipline of this SoK is inherited from its anchor estate and is, we argue, a contribution in itself: every number below is injected at build time from a gated evidence file whose records carry an environment label, and a build-level gate refuses to render a non-canonical (indicative, machine-dependent) number in any headline position. This SoK's headline counts are deterministic structural facts only; it cites no wall-clock measurement anywhere.

What is implemented and tested (research-grade, merged; *not* audited): the honest-majority Shamir backend and BGW degree reduction; disclosed-key and hidden-value joins including a full-obliviousness variant; verdict-bit-only secure comparison and zero-round linear aggregation with secure thresholds via full-field bit decomposition; oblivious shuffle and sort networks (as primitives); hidden-key DISTINCT (duplicate *secret* endpoint pairs collapsed under a never-opened secure comparator — the operator, not just its primitives; the matching hidden-key ORDER BY *operator* is not built, see below); bounded property paths over disclosed keys; result-size protection; robust reconstruction and IT-MAC authenticated sharing with MAC-checked-before-open disclosure; the three-axis security-model vocabulary with the Cleve invariant enforced in types; default-deny source selection and dual-envelope disclosed/hidden routing; the simulation-tier wire protocol and loopback transport with a

deterministic communication counter; the end-to-end federated pipeline driver; and, on the single-prover side, the full circuit family listed in §5, the twelve-obligation fail-closed verifier, external trust-anchor enforcement, single-use nonce discipline, the in-circuit hidden cross-credential equality join, and the security-annotation vocabularies with an ODRL admissibility profile.

What is designed or proposed only (nothing here may be described as existing): the collaborative proof over a secret-shared witness (fail-closed stubs; proof system unchosen); distributed issuer-signature attestation over a shared witness (placeholder types; unsolved in the literature); the verifier-side authenticated-input attestation gate (M4 v1) and the federated public-input layout it needs; the full honest-majority malicious-with-abort suite; any dishonest-majority backend (registry-refused); constant-round WAN comparison; the $O(n \log n)$ sort-merge join; the hidden-key ORDER BY operator (its sort network and secure comparator exist as primitives; the operator integration does not — the matrix's KNOWN tier, reconciled here so the two statements cannot be read apart); a deployable party-mesh transport and pairwise authenticated channels; verifier–federation handshake byte formats; canonical query normalization (the verbatim query string is digested today); a normative leakage vocabulary and differential-privacy budget format; multi-source revocation and federation-level freshness windows; BBS-2023 ingest; the nonce-issuance/manifest-submission wire protocol; in-circuit entailment closure; and portable conformance test vectors for the single-prover lane.

Committed structural fact (build-injected, canonical)	Count
Collaborative-proof / attestation entry points that fail closed with a gate-naming `NotYetImplemented`	6
Failure-mode lenses from the CRYPTO 2025 re-audit of the intended collaborative path — every one assigned RE-OPEN, none CLOSED	4
Clauses in the witness-validation-before-proving test obligation encoded as a build-time gate against the future collaborative prover	5
Confirmed findings of the prior *single-prover* verifier audit, under the open external audit gate	12

Table 3: The committed structural counts this SoK reuses from the estate's paper-bound evidence file — deterministic scans over committed code and audit records, not measurements. They quantify the *negative* space: how much of the collaborative path fails closed, and how much of the single-prover foundation awaits external review. No count is a security property.

evidence: `crates/sparq-mpc/src/proof.rs` — the `CollaborativeProof::{prove, verify}`, `Attestation::attest_source` trait methods plus their stub-test counterparts, each returning `MpcError::NotYetImplemented` with a named gate; the deferred-prover meta-test `gate_state::deferred_prover_never_proves_over_any_witness` in `witness_validation_tests.rs` pins the fail-closed posture (environment: canonical)

Audit posture, stated exactly. The external cryptographer review of the single-prover ZK layer is the open gate `sq-qhy4`. When it closes, it clears *single-prover claims only*: the MPC evaluation layer and any collaborative-proof toolchain require separate, additional review scopes, and no such audit is currently scheduled. Until then the honest sentence — the one the estate's own security documentation uses, and the one this SoK repeats — is that the estate provides *no* production privacy or soundness guarantee to a relying party. Readers should treat every BUILT cell in §5 accordingly.

12. Related work

Verifiable databases (the ZKSQL lineage). IntegriDB (Zhang, Katz, and Papamanthou; ACM CCS 2015) proves SQL answers correct against a committed outsourced database via algebraic hash trees and polynomial evaluation, without hiding records; vSQL (Zhang, Genkin, Katz, Papadopoulos, and Papamanthou; IEEE S&P 2017) extends the guarantee to dynamic databases with verifiable updates, still without hiding; ZKSQL (Li, Weng, Xu, Wang, and Rogers; PVLDB 2023, 16(8)) completes the line by proving answers while the records stay hidden. ZKSQL is the closest SQL-side analog of the

single-prover lane systematized here, with three structural differences the estate's specification itself identifies: the graph data model (blank nodes; per-graph canonicalization), trust rooted in *many issuers'* attestation signatures over per-graph commitments rather than one owner's commitment (so proofs compose across many small signed graphs), and policy-controlled admissibility of proof methods via an ODRL profile.

Federated SPARQL (the planner layer). SPARQL 1.1 Federated Query (Prud'hommeaux and Buil-Aranda; W3C Recommendation 2013) defines the non-confidential model whose disclosure assumption this whole space negates. FedX, ANAPSID, and CostFed are the classic planning pipelines; FedUP (WWW 2024) is the state-of-the-art planner shift to result-aware plans via provenance over quotient summaries, with order-of-magnitude gains over the classics on the FedShop benchmark. The planner layer is complementary to everything in this SoK: a confidential federation still needs source selection and join ordering, but its planner must be treated as *untrusted* (the anchor estate isolates it behind an untrusted-planner seam) and its source selection must be privacy-aware and default-deny.

Cryptographic graph systems. GOOSE (honest-broker MPC over a SPARQL UCRPQ fragment, honest-but-curious, no aggregates), SMPG/PPMQ (Cypher on Neo4j over Shamir sharing via JIFF, semi-honest, conjunctive select-project-join only, with co-located performance numbers the estate's review declines to treat as evidence), and GORAM (VLDB 2025; arXiv 2410.02234; ego-centric traversal on ABY3, semi-honest honest-majority 3PC at billion scale, confidentiality-only). The systematization verdict of §3 stands on these: each is semi-honest and/or confidentiality-only and/or fragment-only, and none binds inputs to issuers.

Relational MPC (the cost reference points). SMCQL (PVLDB 2017) compiled a SQL fragment onto two-party secure computation; Conclave (EuroSys 2019) scaled MPC queries by routing work into annotated cleartext pre/post-processing — the direct ancestor of the disclosed/hidden routing systematized in §4 Senate (USENIX Security 2021) achieved genuinely malicious n-party collaborative analytics; Cerebro (USENIX Security 2021) built the policy/audit platform layer; Secrecy (NSDI 2023) composed oblivious relational operators on honest-majority 3PC with plan-level cost reduction; ORQ (SOSP 2025; eprint 2025/1657) ran the full TPC-H suite under MPC with $O(n \log n)$ join–aggregate fusion, with an honest cost envelope in the minutes-to-tens-of-minutes range per query on a LAN and a modest multiplicative WAN overhead — the cost center being joins, since obliviousness forces worst-case padding at every intermediate. These systems calibrate expectations for the hidden regime; the disclosed-key advantage of §4 is precisely what lets federated SPARQL escape those costs in its common case — at the linkage price stated there.

Private set intersection — the hidden equi-join, named as its own literature. The hidden-key equi-join at the center of the matrix *is* the PSI problem, and a systematization owes that literature explicit placement. The OT-extension line of Pinkas, Schneider, and colleagues made two-party PSI practical at scale; circuit-PSI (Pinkas et al., EuroCrypt 2019) matters most here because it emits *secret-shared* intersection outputs that compose with downstream MPC instead of disclosing the intersection — exactly the shape a hidden SPARQL join pipeline needs; VOLE-PSI (Rindal and Schoppmann, EuroCrypt 2021; eprint 2021/266) is the malicious-model cost frontier for key-on-key intersection; and deployed private intersection-sum (Ion et al., IEEE EuroS&P 2020 — the private join-and-compute line) is the industrial existence proof. The systematization placement: these are two-party, single equi-join, key-on-key protocols — none composes for free into an N-party multi-way BGP with unbound variables, which is why the matrix tiers circuit-PSI as KNOWN-not-integrated rather than solved. And the disclosed-key join of §4 is precisely the degenerate PSI whose intersection is public by policy — which is both why it is crypto-free and why its leakage is exactly the concession the PSI literature exists to avoid.

Differential privacy over federated queries — the lineage behind the matrix's result-size mitigation. Shrinkwrap (Bater et al., PVLDB 2019) pads intermediate cardinalities inside an MPC federation to differentially private bounds instead of worst-case obliviousness; SAQE (Bater et al., PVLDB 2020) trades approximation error against both MPC cost and privacy budget; elastic sensitivity (Johnson, Near, and Song, PVLDB 2018) bounds the sensitivity of real SQL joins for practical DP query answering. We cite this line explicitly because the result-size protection the matrix records — and the estate's own leakage review recommends — is an instance of it, with the honest caveats the estate's review states: DP padding still leaks a noised size, and the budget composes across repeated queries, so this is a leakage-shaping tool, not an information-theoretic guarantee.

TEE/enclave and encrypted-database query processing — scoped out of the axes, and why. A completeness-minded reader will ask where the enclave line sits: Cipherbase (SQL over encrypted data with trusted hardware), Opaque (NSDI 2017; oblivious distributed SQL on SGX), and ObliDB (PVLDB 2019; oblivious enclave query processing) deliver confidential query evaluation with production-adjacent performance, and Blind Seer (IEEE S&P 2014) achieved private querying over an encrypted index with garbled circuits a decade ago. They are excluded from the mechanism axis by the criterion of §2, not by oversight: the enclave systems root their guarantee in a hardware vendor and its attestation chain rather than in cryptographic assumptions among the holders — a different trust object, and one whose attestation soundness is empirically eroding under physical attack, which is why the estate's own corpus prices TEEs as a latency escape hatch that needs a ZK integrity layer, strictly weaker on trust minimality. Blind Seer is cryptographic but client–server: one database owner, one querier, querier privacy — not federation among mutually distrusting holders. Both remain relevant as engineering references for oblivious operators.

Collaborative ZK and attested-input MPC. The proof-system layer (Ozdemir–Boneh; scalable coZK; TACEO co-snarks) and its CRYPTO 2025 caveat, plus the authenticated-input work of Dutta et al. and the commit-and-prove anchoring of Artemis, are systematized in §10 we do not repeat them here beyond noting the headline numbers in those papers come with material qualifications (fast-LAN settings, preprocessing excluded, proof-of-concept delegation topologies) that any adopting system must re-derive for its own deployment envelope.

Anonymous credentials and semantic-web data integrity. CL signatures (SCN 2002), BBS/BBS+ (CRYPTO 2004), and the W3C Data-Integrity cryptosuites (bbs-2023, ecdsa-sd-2023; 2025) provide credential-level selective disclosure that the query-level approach generalizes but does not replace (§8).

Encrypted RDF and access-controlled SPARQL — the semantic-web-native adjacent line. The RDF anchoring obliges placement of the semantic-web work on confidential RDF itself: encrypted RDF with self-enforcing access control and its compressed successor line (Fernández, Kirrane, Polleres, and Steyskal; ESWC 2017, and HDTcrypt) protect graph data at rest under partial-access policy, and the access-control literature over RDF (Kirrane et al.'s survey; enforcement by SPARQL query rewriting) filters what a *known, authenticated* querier may see. Neither provides verifiability of answers nor confidentiality among mutually distrusting holders during evaluation — the two guarantees systematized here — so the line is complementary below the cryptographic query layer, exactly as credential-level selective disclosure is.

Adjacent SoKs, and where the axes come from. Hastings, Hemenway, Noble, and Zdancewic (IEEE S&P 2019) systematized general-purpose MPC compilers; Viand, Jattke, and Hithnawi (IEEE S&P 2021) did the same for FHE compilers. We follow their method — a capability grid grounded in an honest artifact review — and we are explicit about taxonomic provenance so the frame is not mistaken for invention: axis 3 (adversary × threshold × output guarantee) is the classical MPC security vocabulary those SoKs also inherit, axis 2 is the collaborative-SNARK line's distinction, and axis 1′s mechanism

classes are each a mature literature. What we stake as new is the joint application to federated SPARQL, the disclosed-key regime split with its priced leakage (§4), and the operator-level SPARQL matrix (§5).

The estate's own lineage, declared for citation integrity. The security-properties vocabulary the estate extends originates in the SEC-PROP work of Wright, Shadbolt, Zhao, Zhao, and Braun ("Zero-Knowledge Proof of Correct SPARQL Evaluation over Verifiable Credentials"; the `sec-prop` vocabulary at `https://w3id.org/zkp-sparql/`, vendored under MIT), whose admissibility layer the estate's specification derives from and declares plainly. The research program's framing — a single-graph ZK question and a federated MPC+ZK question — appears in Wright's doctoral-consortium paper (ISWC 2025 Doctoral Consortium; CEUR-WS Vol-4085, paper 19). Braun, Wright, and Käfer ("Proving Soundness of SPARQL Query Results Using Selective Disclosure of RDF Datasets and Zero-Knowledge Proofs"; The Semantic Web, Springer, 2026; DOI 10.1007/978-3-032-25156-5_16) contribute the complementary mechanism of keeping some bases *disclosed* and proving derivations over them, where the lane systematized here keeps source graphs hidden and proves algebra evaluation in-circuit — complementary, not competing, approaches.

13. Limitations of this systematization

- **Single-estate anchoring.** The capability matrix is calibrated against one open estate (the only one, to our knowledge, that implements both a single-prover SPARQL ZK lane and a SPARQL MPC operator suite with an honest status vocabulary). This risks mistaking that estate's design choices — a Shamir arithmetic core, a Noir/UltraHonk proof stack — for boundaries of the field. We flag every cell where the tier is stack-relative (e.g., string functions are OPEN *for an arithmetic-sharing core*; a Boolean-circuit system would tier them KNOWN).
- **Absence claims are time-indexed and method-bounded.** The frontier statement (§10) and the vacant-cell verdicts rest on the survey protocol of §2, with stated dates (2026-06-13; re-checked 2026-07-02). They are claims about what that search failed to find in peer-reviewed venues and open preprint archives — the weakest kind of negative, and scoped: patent and product literature is excluded, a single new paper falsifies them, and this SoK should be read with that decay rate in mind.
- **Documentary, self-sourced provenance.** The anchor estate's capability tiers trace to documents authored inside the same project as this SoK (§2): the specification corpus, reconciled capability review, and design records — with per-section implementation-status labels and fail-closed stubs that make overclaim structurally difficult — but this SoK is not itself a code audit, its statements about implemented behavior are secondhand to those records, and no independent party has verified any tier. For the anchor estate, read this paper as a system report with a field survey attached, not as third-party evaluation.
- **Cross-system granularity mismatch.** The published systems are placed in the operator matrix at the granularity their publications support — usually a query suite or language fragment, not a per-operator artifact review — so their cells are coarser than the anchor estate's self-reported ones. A uniform operator-level cross-audit would require artifact evaluation of every surveyed system, beyond this SoK's scope.
- **No performance evaluation.** By design, no wall-clock number appears; hidden-regime costs are characterized qualitatively via the relational MPC literature. A cost model for the disclosed/hidden routing split over realistic federated SPARQL workloads is missing from the literature and from this SoK alike.
- **The matrix covers the algebra, not deployments.** Transport, key distribution, revocation federation, freshness windows, and the verifier handshake are all in the designed-or-proposed column; a deployed system would surface failure modes no operator-level matrix captures.
- **Judgment in tiering.** KNOWN vs OPEN cells rest on protocol-literature search and engineering judgment about integration fit; they are refutable classifications, not theorems, and we have kept

every theorem-grade statement (Cleve; the semantics exclusion) explicitly separated so the two kinds of "no" cannot be confused.

14. Conclusion

Verifiable and private federated SPARQL is not one problem but a three-axis space — mechanism, topology, adversary — and most confusion in the literature comes from naming a point by only one of its coordinates. Systematized on all three, the space shows a clear shape. The single-prover zero-knowledge lane is architecturally settled (canonicalize, commit, prove a small fragment, verify fail-closed against exogenous anchors) and awaits external audit rather than invention. The MPC evaluation lane is an operator-engineering program with a structural gift unique to RDF — disclosed global-IRI join keys — that moves the common federated case out of the cryptographic *cost* regime wherever the join's linkage disclosure is acceptable by policy: a priced advantage, not a free one, since the cleartext join reveals exactly the cross-source linkage that source-unlinkability protects (§4), alongside a set of verdict-bit disclosure disciplines that keep the hidden regime's leakage envelope minimal and explicit. The negatives divide cleanly into theorem-closed, semantics-closed, self-labeled, and merely vacant — and the honest systems encode the closed ones in their type systems. And one cell holds the field's future: a single source-unlinkable collaborative proof whose circuit verifies each holder's issuer signature over a secret-shared witness. Its three ingredients — collaborative proving, set-membership signature proofs, blind issuance — are each solved alone; their composition is, as of our stated survey dates and within the search scope of §2, published nowhere we could find, is hard for stateable reasons (non-linear signature verification under MPC; ring structure over the key set; a proven leakage pitfall when witness validation is skipped), and has an honest interim (verifier-side authenticated-input binding) that names exactly what it gives up. Until that cell is filled and independently audited, no system — including the estate anchoring this SoK — can honestly offer verifiable, attested, source-unlinkable federated SPARQL; what a system can do today is what we have systematized: state its cell precisely, fail closed outside it, and make its negative space as legible as its capabilities.

---

sparq project · this SoK asserts *no* proven security, privacy, soundness, or attestation property of any system it describes. The anchor estate is research-grade: its single-prover ZK layer is not externally audited (open gate `sq-qhy4`), its MPC layer is honest-majority semi-honest by default, and its collaborative-proof path is unbuilt and fail-closed (re-audit gate `sq-9hrn`). Capability tiers trace to the estate's specification corpus (zkSPARQL Revision 2, 2026-07-01; the MPC-SPARQL draft), the reconciled capability review (2026-06-16), and the distributed-signature feasibility survey (2026-06-13; frontier absence claim re-checked 2026-07-02 per the method section) — self-sourced documentary provenance, stated as a limitation. Counts in this document are injected at build time from the paper-bound evidence file; see the provenance stamp on the published page.