Bilateral Cryptographic Data Exchange — a clean-room-native primitive for pharma audience data partnerships.

Squyr is a bilateral cryptographic protocol for exchanging hashed patient identifiers between two parties without requiring either party to share PII, depend on a shared infrastructure vendor, or trust the other unilaterally. **Standard primitives. Distinctive architecture. Healthcare-optimized.**

LIVE · PRODUCTION DEPLOYED | PATENT-PENDING | EXTERNAL AUDIT IN PROGRESS

VERSION	STATUS	AUDIENCE	BY
v1.0 · May 2026	Production deployed	Data partners · MLR · CTOs	Summit Audience Segments

01 What is Squyr?

Squyr is a protocol for exchanging hashed patient identifiers between two parties — typically a data licensor and a marketing or activation partner — **without requiring either party to share PII, depend on a shared infrastructure vendor, or trust the other unilaterally.**

The protocol uses standard cryptographic primitives (SHA-256, HKDF, AES-256-GCM) configured under a **bilateral salting model** so neither party can independently decrypt the exchanged tokens. The result is a data flow that ships clean, activates flexibly, and audits cleanly on both sides.

The problem Squyr solves

The problem Squyr solves is **operational, not theoretical**. Healthcare data partnerships have historically moved data through one of three patterns — each of which works, but each carries cost:

PATTERN 01 · DIRECT FILE TRANSFER

CSV over SFTP. **Fundamentally insecure and unauditable.** Defensible only by procedure, not architecture.

PATTERN 02 · VENDOR-MEDIATED

The data licensor's platform handles all matching and activation, **creating lock-in** and limiting buyer optionality.

PATTERN 03 · THIRD-PARTY CLEAN ROOM

A paid intermediary sits in every transaction. Per-record fees compound; **strategic neutrality is lost.**

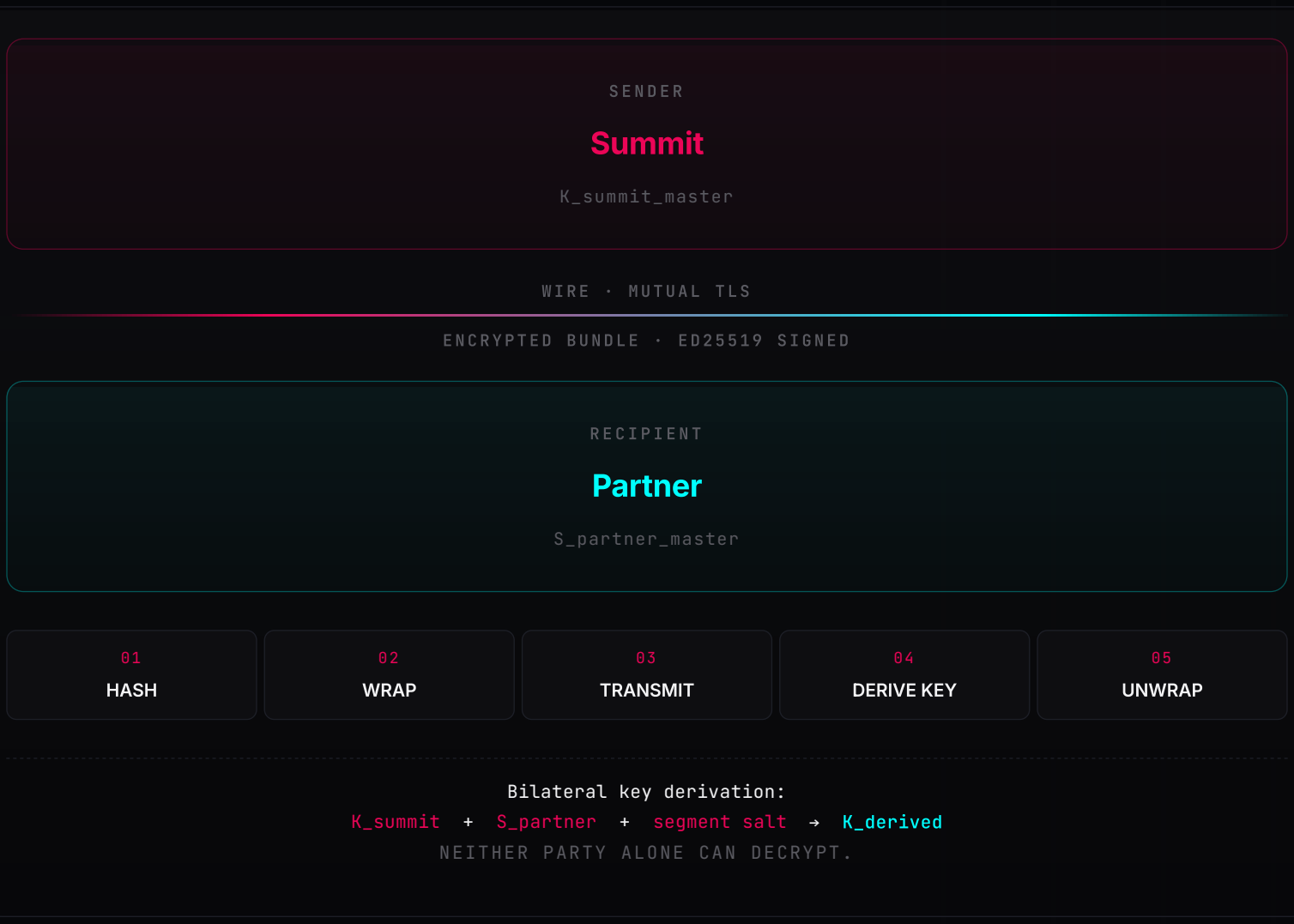
SQUYR COLLAPSES THE CHOICE

The data licensor controls how data is encoded. The buyer controls how data is activated. Neither party requires shared infrastructure beyond standard cryptographic primitives available in every modern browser and KMS.

For data buyers operating sophisticated identity stacks — combinations of identity graphs, internal clean rooms, and DSP relationships — the architecture matters because most data partnerships force a choice between losing flexibility and accepting vendor lock-in. **Squyr removes the choice.** Summit ships SHA-256 tokens compatible with every major identity platform; the buyer decides which stack handles which campaign.

02 Architecture at a glance

Five operations across two environments and one transit. Identifier hashing and token wrapping occur entirely in Summit's environment. Bundle transmission moves over mutual-TLS authenticated transport. Key derivation and token unwrapping occur entirely in the partner's environment. **At no point does PII cross any boundary.** At no point can either party act on the exchanged tokens without the other party's cryptographic contribution to the derived key.



What this enables

MULTI-STACK FLEXIBILITY

SHA-256 deterministic hashes are the universal identity-resolution input across major identity graphs, RampID translation, DSP-native identity stacks, and virtually every modern healthcare data activation platform. **Once unwrapped, the SHA hash list activates through whichever stack fits the campaign.**

ZERO PII MOVEMENT

Regulatory exposure on the partner side is dramatically reduced because **the partner never receives identifying information** — only encrypted tokens that resolve to hashes.

BILATERAL TRUST

The salt split protects both parties symmetrically. A breach of Summit's environment **does not enable decryption** of partner-side bundles. A breach of the partner's environment **does not enable derivation** of future Summit data.

AUDIT READINESS

Every operation logs on both sides with cryptographic signatures, producing a **mutually verifiable trail** for compliance review.

03 Protocol specification

The cryptographic foundation and the step-by-step protocol.

Cryptographic primitives

Squyr uses four standard primitives, all **FIPS 140-2 compliant** when implemented via the Web Crypto API (browser-native) or equivalent server-side KMS. Each primitive is documented in publicly available RFCs and has been peer-reviewed for decades. **The protocol does not introduce novel cryptography;** it composes proven primitives in a configuration that addresses healthcare-specific exchange requirements.

PRIMITIVE	STANDARD	ROLE IN SQUYR
SHA-256	RFC 6234	Deterministic identifier hashing
HKDF	RFC 5869	Bilateral key derivation
AES-256-GCM	NIST SP 800-38D	Authenticated token wrapping
Ed25519	RFC 8032	Bundle signature and verification

Key derivation

The cryptographic heart of Squyr is **bilateral key derivation**. The working key used to encrypt and decrypt tokens **cannot be derived by either party acting alone**. Both parties' secret contributions plus a segment-specific salt produce the derived key:

```
K_derived = HKDF-SHA256(
  ikm    = K_summit_master
        || S_partner_pubkey_commitment
        || S_segment,
  salt   = "squyr-v1-salt-2026",
  info   = "squyr-key-derivation-v1",
  length = 32 bytes
)
```

`K_summit_master` is held exclusively in Summit's KMS or HSM and never crosses environments. `S_partner_master` is held exclusively in the partner's KMS. Only **public commitments** — SHA-256 hashes of each master secret bound to a context string — are exchanged at contract signing. `S_segment` is a fresh 128-bit value generated by Summit per licensed audience, transmitted as part of the encrypted token bundle metadata.

DEFENSE IN DEPTH · THREE FAILURE MODES

A breach of either party's master secret reveals **no useful information** about the other party's master secret, because the cryptographic commitments are one-way hashes. A breach of the encrypted bundle without access to either master secret reveals only **opaque ciphertext**. A breach of one master secret plus the encrypted bundle reveals opaque ciphertext that **still cannot be decrypted**, because key derivation requires both master secrets as inputs.

Step-by-step protocol

01 SUMMIT

Hash

Each patient identifier is normalized (E.164 phone, lowercase email, deterministic name+address composite) and hashed with SHA-256. Result: 32-byte deterministic identifier.

02 SUMMIT

Derive segment key

Fresh 128-bit `S_segment` generated. `K_derived` computed per the formula above.

03 SUMMIT

Wrap tokens

Each identifier hash encrypted using AES-256-GCM with `K_derived`, a unique 96-bit nonce per token, and segment metadata as additional authenticated data.

04 SUMMIT

Sign bundle

Tokens assembled into Encrypted Token Bundle: bundle ID, segment ID, timestamp, record count, `S_segment`, wrapped tokens with nonces, and Ed25519 signature.

05 BILATERAL

Transmit

Bundle transmitted via mutual-TLS authenticated API or pre-signed S3 URL with short expiration. Both endpoints log the transmission.

06 PARTNER

Verify & derive

Partner verifies Ed25519 signature against Summit's public key commitment. `K_derived` computed using partner's master secret plus received `S_segment` plus Summit's commitment.

07 PARTNER

Unwrap tokens

Each wrapped token decrypted using AES-256-GCM with `K_derived` and associated nonce. Output: original SHA-256 hash.

08 PARTNER

Activate

Identifier hashes available for matching against any identity platform the partner uses. **Summit has no visibility into activation destination.**

04 Security model

Attack scenarios, compliance posture, and industry context.

Attack scenarios

Squyr is designed to **fail safely** under realistic threat models. The protocol is not unhackable in an absolute sense — no protocol is. What Squyr guarantees is that **no single breach produces a usable attack**, and that the attack surface for a multi-party breach is bounded by salt rotation policy and per-segment isolation.

SCENARIO	ATTACKER OBTAINS	MITIGATION
Summit breached	`K_summit_master + encrypted bundles`	Cannot derive `K_derived` without `S_partner_master`. Bundles remain opaque. Summit rotates `K_summit_master`.
Partner breached	`S_partner_master + previously unwrapped hashes`	Past matches remain valid (already used). **Future bundles unbreakable** after salt rotation.
Bundle intercepted	`AES-256-GCM ciphertext`	Cannot decrypt without both master secrets and mTLS transport key. AES-GCM remains cryptographically unbroken.
Both parties breached	`Full crypto context within rotation window`	Only viable attack vector. Bounded by **~90-day salt rotation**, per-segment isolation, anomaly detection.

Compliance posture

Squyr was designed against the operational requirements of healthcare data partnerships, **not retrofitted from consumer adtech**. The protocol is **HIPAA-safe by design**: identifiable information never crosses between parties, eliminating the most common BAA-violation patterns.

- **HIPAA Safe Harbor** — 45 CFR §164.514(b)(2). PII never crosses environments.
- **BAA-compatible** — when Summit and a partner execute a Business Associate Agreement, the protocol operates within the agreement's scope rather than requiring exceptions.
- **SOC 2 Type II** — both parties' infrastructure should be SOC 2 audited for the agreement to be defensible.
- **FIPS 140-2** — primitives compliant when implemented via the Web Crypto API or equivalent KMS.
- **State-level privacy** — California CMIA, Washington MHMDA, and the proliferating 2026 state framework. Squyr provides architectural defensibility on both sides: the data licensor can demonstrate that PII never left their environment; the activation partner can demonstrate they never possessed PII, only cryptographic tokens that resolved to deterministic identifiers already in their existing identity stack.

Industry context

Squyr composes the **same cryptographic primitives used by every major data collaboration platform** in production today. The differences are in the configuration and the optimization target, not the underlying math.

CAPABILITY	SQUYR	LIVERAMP	INFOSUM	AWS CR	SNOW CR
Bilateral encryption	✓	✓	✓	✓	✓
Threshold cryptography	✓	✓	✓	✓	✓
Healthcare-optimized	✓	—	—	—	—
No vendor lock-in	✓	—	—	~	~
Embedded in license	✓	—	—	—	—
Multi-stack activation	✓	✓	—	~	~

THE HONEST POSITIONING

Squyr is **not technically superior** to LiveRamp, InfoSum, AWS Clean Rooms, or Snowflake Clean Rooms in cryptographic terms. Those platforms operate at industrial scale. Squyr's contribution is **configuration, not invention**. The protocol is designed from the start for the specific shape of healthcare data partnerships — bilateral exchanges with audit and compliance as first-class concerns. The result is a primitive that fits inside data licensing agreements without requiring a separate vendor relationship.

FOR TECHNICAL QUESTIONS OR PARTNERSHIP DISCUSSIONS

info@summitaudiencesegments.com

Summit Audience Segments · summitaudiencesegments.com

• ENCRYPTED · AES-256-GCM • HIPAA SAFE HARBOR • BAA COMPATIBLE • SOC 2 TYPE II IN PROGRESS