SEMAFORE DOCS

Architecture and Threat Model

VERSION 1.0 · REVISED 2026-05-17

SemaFore's threat model, cryptographic architecture, infrastructure posture, and multi-device wire format — engineering reference.

SECURITY CONTACT: SECURITY@ATTOMUS.COM

Architecture and Threat Model

VERSION 1.0 REVISED 2026-05-17 TECHNICAL-REVIEWER, CISO, SECURITY-AUDITOR

LAST UPDATED: 29 MAY 2026

This document summarises positions formally recorded in numbered Architecture Decision Records (ADRs) maintained in the SemaFore source repositories. ADR numbers are cited inline; the ADRs themselves are the canonical authority.

Status of this document

This is an engineering reference, not a marketing document. It records what SemaFore protects, what it does not protect, and why. It is written for readers who will pull apart any absolute claim and prefer precise scoping to confident assertion.

Where this document and one of the cited ADRs disagree, the ADR is correct.

---

1. Threat model

1.1 In scope

SemaFore is designed to provide cryptographic guarantees against the following adversaries and risks:

- **Server-side compromise.** Compromise of the SemaFore server-side implementation, the underlying database (MongoDB, Redis), the storage backend (SeaweedFS), and any infrastructure-tier credential. The compromised party cannot read historical or future message contents.
- **Network adversary.** A capable in-path adversary observing TLS-protected traffic between client and server. The adversary observes traffic-level metadata (timing, volume, peer addresses) but cannot read encrypted message bodies.
- **Insider exfiltration.** An Attomus employee, contractor, or compromised production-system operator. Such an actor cannot recover message content; their access is limited to operational metadata necessary for the platform to function, redacted per the operational-logs policy described in Section 2.
- **Lawful subpoena targeting Attomus.** A subpoena compelling Attomus to produce customer message content. Attomus cannot produce content it never had access to. This is a deliberate property, not an

emergent one.

- **Endpoint compromise of a single device.** Compromise of one user device exposes that device's content. Compromise of one device does not expose other devices' content owned by the same user or by their correspondents.

1.2 Out of scope

SemaFore does not claim cryptographic guarantees against:

- **Compromise of all endpoint devices belonging to a user.** Per-device encryption protects against partial endpoint compromise; total endpoint compromise yields total content exposure for that user. This is fundamental to any end-to-end encrypted messaging system.
- **Endpoint operating system telemetry.** Apple iOS and Google Android send diagnostic, crash, and usage telemetry to their respective vendors. SemaFore does not control these flows. Customers whose threat model requires endpoint-OS sovereignty must deploy SemaFore onto hardened mobile platforms of their own choosing (de-Googled Android distributions, MDM-locked iOS devices, dedicated hardware).
- **Push notification routing through APNs and FCM.** Message body content does not transit APNs or FCM; the notification payload is data-only with no message content (see Section 4). The fact that a notification was sent at a given time to a given device is visible to Apple or Google by virtue of operating the push infrastructure.
- **Traffic analysis by a global passive adversary.** SemaFore is not a traffic-analysis-resistant overlay. Customers whose threat model includes traffic-analysis resistance against nation-state adversaries should pair SemaFore with appropriate anonymising network infrastructure or use a product specifically designed for that threat model.
- **Availability against sustained nation-state-grade denial-of-service.** SemaFore's availability posture is described in Section 3. Reasonable resilience is provided through multi-site deployment, edge protection, and capacity planning. SemaFore does not claim invulnerability to a determined high-volume DDoS adversary.
- **Compromise of cryptographic primitives themselves.** SemaFore depends on the security of X25519, Ed25519, AES-256-GCM, HKDF-SHA256, and SHA-256. A future cryptographic break against any of these would affect SemaFore as it would affect every system built on them.

1.3 Threat-model honesty

The list of out-of-scope adversaries is non-trivial because honest threat-model scoping is the most useful service this document can render to a reviewer. SemaFore is purpose-built for the in-scope threats: enterprise-grade insider risk, cloud-jurisdictional subpoena risk, server-side compromise, and content interception. For these threats, SemaFore provides strong cryptographic guarantees backed by the architectural decisions recorded in the rest of this document.

---

2. Metadata and operational logging

2.1 Position

Message content is end-to-end encrypted. Operational metadata is audited and made available to customer organisations as a compliance, supervision, and retention capability. These are distinct properties; pretending otherwise is the imprecision worth eliminating from any positioning of the product.

2.2 What "metadata" means in SemaFore

The categories of data that the server processes and may record include:

- **Routing identifiers.** Sender and recipient identifiers, device identifiers, message identifiers, delivery identifiers, group identifiers, organisation identifiers.
- **Timing.** When events occurred, including message send, message delivery, message-read receipt, and session-establishment events.
- **Volume.** Counts of messages, sessions, and attachments.
- **Endpoint context.** Connecting client IP address (necessary for connection delivery and rate limiting), client user-agent (where the client provides one), TLS session metadata.
- **Content-existence proofs.** Cryptographic envelope hashes and delivery confirmations sufficient to prove a specific encrypted message existed and transited the platform.

The server does not process message body plaintext, attachment plaintext, or any content that would defeat the end-to-end encryption guarantee.

2.3 Operational-log identifier hashing

Operational logs (the slog-format JSON logs that operators consult for debugging and incident response) underwent a deliberate hardening pass in May 2026 to minimise the operational-metadata exposure on a server-compromise scenario. The work is recorded in ADR-0019 and in the audit at `sf-shared-docs/docs/security/findings/2026-05-12-006-operational-logs-raw-identifiers.md`.

After this pass, the following identifiers appear in operational logs only as 8-hex SHA-256 prefixes under field names suffixed with `_h8`:

- `sub` (JWT subject / user identifier)
- `subject`
- `device_id`
- `message_id`
- `delivery_id`
- `group_id`
- Composite identity markers

The identifier `org_id` / `organisation_id` is intentionally retained in raw form. The operator-facing surface needs to know which customer organisation is being investigated; hashing this identifier would prevent the operations team from doing their job and would not provide a meaningful adversary-resistance benefit.

The result of this hardening: a server-side adversary inspecting operational logs sees substantially less than the analogy to typical SaaS logging would suggest. The compromise surface for routing-graph metadata against an internal adversary is the database tier, not the log tier.

2.4 Customer-facing audit and compliance

Customer organisations receive metadata visibility through deliberately-designed audit and compliance surfaces:

- **SIEM export.** Per-organisation audit-event export to customer-controlled SIEM platforms (Splunk, Sentinel, ElasticSearch, et al). The audit-event format is documented and stable. Per-organisation access controls govern who can subscribe to or query the SIEM export.
- **Organisation-level retention policy.** Per-organisation retention policy for both content (where the customer's regulatory regime supports it) and metadata. ADRs 0093, 0094, 0095, and 0097 describe the retention enforcement model.
- **Audit log queries.** Authenticated admin-facing audit log queries surfacing organisation-scoped events: signup, subscription change, role change, membership change, screenshot reports, broadcast events, et al.

These surfaces are deliberate features. Customers in regulated industries — government, defence, policing, financial services, healthcare — have legal and procedural obligations to maintain communication metadata for supervision, audit, FOIA response, and similar regulatory functions. A messaging product that did not provide these surfaces would be unusable in these environments.

2.5 What this posture does not protect against

The metadata-visibility posture is meaningful to a nation-state adversary capable of compromising the SemaFore server and reading the database tier directly. Such an adversary would observe communication-graph metadata (who messaged whom, when, at what cadence) for the organisations housed on the compromised server. SemaFore does not claim resistance to this adversary class, and customers whose threat model includes it should not rely on SemaFore as their sole defence.

The fact that customers in defence, intelligence, and policing knowingly deploy SemaFore alongside this caveat reflects that, for their threat model, server-side compromise by a nation-state adversary is one risk among many, not the sole criterion of evaluation. The alternative — communicating over consumer

messaging products with weaker enterprise-control surfaces — typically presents a worse aggregate posture.

---

3. Infrastructure architecture and scaling

3.1 Sovereign hosting

SemaFore's backend operates on UK-resident, Attomus-controlled hardware. The decision is recorded in the storage-migration ADR-0155 and the deployment-shape ADR-0162. The substantive guarantee is jurisdictional: customer data does not transit infrastructure under foreign jurisdiction at runtime or at rest.

Sovereign hosting is not synonymous with single-rack hosting or non-scalable hosting. The two properties are independent. SemaFore is sovereign-hosted; SemaFore is also designed to scale.

3.2 Storage architecture

Object storage uses SeaweedFS deployed in a containerised single-node topology, recorded in ADR-0162. The same ADR documents the multi-node scale-out path: the runbook at `sf-shared-docs/docs/runbooks/seaweedfs-docker-install-ops-steps.md` describes the three-node promotion procedure as "provisioning two more hosts running the same compose definition." Horizontal scale-out is an operational evolution, not an architectural rewrite.

For the current customer base, the single-node topology is sized appropriately. Capacity planning is performed per customer engagement; multi-node deployment is the natural promotion path as customer base or per-customer usage warrants.

3.3 Database architecture

MongoDB is the primary data store; Redis serves real-time delivery primitives. Both support multi-node, replicated, geographically-distributed deployment within UK-sovereign jurisdiction. Database-tier scaling is a procurement-and-deployment exercise, not a customer-visible architectural choice.

3.4 Application-tier scaling

The Go server implementation is designed to be deployed in horizontally-scaled replicas behind a load balancer with shared state in MongoDB and Redis. Single-replica deployment is the current production topology for operational simplicity at the current customer base; replica-scale deployment is a configuration change, not an engineering project.

3.5 Availability posture

SemaFore's availability posture is addressed through procurement-time capacity planning, multi-site sovereign deployment for customers whose contracted SLA requires it, professional edge DDoS mitigation, and customer-aware crisis-mode redundancy. The posture is not addressed through cloud-elastic auto-scaling; this is a deliberate trade against jurisdiction-of-convenience hosting.

Customers whose contracted availability requirements include resistance to sustained nation-state-grade denial-of-service should size their engagement accordingly. SemaFore's contracted SLA is documented per-engagement and reflects the customer's specific operational requirements.

3.6 Comparison to hyperscale cloud

The implicit comparison "concentrated infrastructure cannot match AWS auto-scale resilience" deserves scrutiny. AWS, Azure, and Google Cloud have all experienced multi-hour customer-affecting outages from causes including BGP misconfiguration, control-plane failures, regional incidents, and DDoS. Cloud-scale is not synonymous with availability resilience. The relevant question is operational resilience, regardless of topology.

SemaFore's operational-resilience posture is a feature of the deployment shape, the multi-site capacity, and the edge protection — not a property of being or not being on AWS. Customers should evaluate the actual posture, not the topology label.

---

4. Multi-device cryptography

4.1 Position

SemaFore implements the X3DH key agreement and Double Ratchet ratcheting protocols from the Signal Protocol literature. The implementation is purpose-designed for multi-device, with per-device session state and per-device cryptographic envelopes throughout. Session desynchronisation, a known failure mode of naive implementations, is mitigated by the wire format design and is regression-tested across platforms.

4.2 Per-device wire format

Each user device — iOS phone, iPad, Android phone, Android tablet, GitHub Actions integration runner, future macOS / Windows / Linux clients — carries its own X25519 identity key, Ed25519 signing key, signed prekey, and one-time prekeys. Each pairwise correspondence between two devices is its own X3DH-bootstrapped Double Ratchet session.

A message sent from one user to another with multiple devices is encrypted N times, once per recipient device, and dispatched as N envelopes to the server's fan-out primitive. The wire-format ADRs are:

- ADR-0053 (dual identity key) — X25519 for ECDH, Ed25519 for signed prekey verification; both required in the key bundle.
- ADR-0054 (the file_id receive-side schema).
- ADR-0055 (broadcast receive contract — per-recipient envelopes for broadcast messages).
- ADR-0157 (attachment contract — ciphertext-only attachment uploads).
- ADR-0158 (broadcast contract — per-recipient envelopes wholesale).
- ADR-0159 (screenshot contract — hash-only forensic metadata).

The pattern is consistent: per-recipient envelopes throughout. The server never sees plaintext content; the wire format treats per-device envelopes as a first-class concept rather than an afterthought.

4.3 The canonical SMX1 prekey header

The X3DH bootstrap envelope uses a layout designated SMX1 (the magic four bytes `0x53 0x4D 0x58 0x31`). The layout is defined and pinned at byte-level:

```
Offset    Length   Field
--------  -------  ------------------------------------------------
0         4        Magic: 0x53 0x4D 0x58 0x31  ("SMX1")
4         1        Flags: 0x01 = OPK used; 0x00 = OPK absent
5         32       EK_A: sender ephemeral X25519 public key (raw bytes)
37        2        SPK key_id length L1 (big-endian uint16)
39        L1       SPK key_id (UTF-8)
39+L1     2        OPK key_id length L2 (big-endian uint16); 0 if no OPK
41+L1     L2       OPK key_id (UTF-8, if L2 > 0)
41+L1+L2  12       Nonce (random, 12 bytes)
53+L1+L2  …        AES-256-GCM ciphertext + 16-byte tag
```

HKDF parameters used during the X3DH bootstrap are canonicalised in ADR-0169: `salt = zeros(32)` (per RFC 5869 §2.2), `info = "SemaFore-X3DH-v1"` UTF-8, output length 32 bytes. The signed prekey signature must be verified (Ed25519) before any Diffie-Hellman operation.

4.4 Cross-platform interop test vectors

Wire-format drift between platforms — the iOS Swift, Android Kotlin, Go server, and JavaScript implementations producing slightly different byte outputs for the same logical operation — is the classic failure mode that leads to session desynchronisation. SemaFore mitigates this through pinned cross-language regression vectors:

- `sf-shared-docs/docs/test-vectors/dr-v1-interop.json` — the Double Ratchet continuation-path vectors. Pinned input → expected output for the ratcheting state machine.

- `sf-shared-docs/docs/test-vectors/dr-v1-live-result.json` — pinned outputs from production-style runs, cross-checked.
- `sf-shared-docs/docs/test-vectors/x3dh-prekey-v1.json` — the canonical X3DH bootstrap vector, byte-pinning the SMX1 envelope shape.

Each platform's test suite runs against these vectors. A platform that produces output diverging from the vector fails CI. The vector files are versioned alongside the wire-format ADRs; advancing the wire format requires advancing both the ADR and the vector.

4.5 Production hardening evidence

The May 2026 work designated SF-007 promoted the Double Ratchet release flag to true on iOS (ADR-0068) and Android, with a coordinated cross-platform interop test required before promotion to staging. The promotion ritual is documented at `ai-local/2026-05-16-mobile-sf-007-pseudo-release-handoff/`. The interop gate required live iOS-to-Android and Android-to-iOS DR_V1 frame exchange against the staging server, with plaintext-absence verification on both sides of the relay.

This is the operational discipline that prevents the desync class of bug. The protocol is implementable poorly; SemaFore's implementation reflects deliberate engineering against the known failure modes.

---

5. Endpoint-side cryptographic key material

5.1 Storage

Per-device cryptographic keys are held in the platform's hardware-backed key store: iOS Secure Enclave on Apple devices, Android Keystore (with hardware-backed StrongBox where available) on Android devices. Key material does not leave the secure element in plaintext; cryptographic operations are performed inside the secure element where the platform supports it.

The implications are recorded across multiple ADRs and are intentional design properties:

- The server cannot read message content even if it wanted to; it does not have the keys.
- A user who loses access to all their devices loses access to their decryptable content. SemaFore has no centralised key escrow and no centralised content recovery; cloud key escrow would defeat the end-to-end encryption guarantee.
- A user who compromises one device exposes that device's content but not other devices' content owned by the same user or by their correspondents.

These properties are described in the customer-facing material at `semafore.io/faq` (the Enterprise Value FAQ); this document records the engineering position.

5.2 Lawful retention compatibility

Customer organisations with lawful retention obligations are served through:

- Metadata and envelope-existence retention via the SIEM/audit export (Section 2.4).
- Organisation-level retention policy enforcement on the server side (ADRs 0093, 0094, 0095, 0097).
- Content retention on user devices, where the user is the legitimate custodian of their own communications.

For content retention that requires centralised cloud recovery — a specific regulatory regime that not all customers face — SemaFore is not the right product. The customer-fit boundary is honest and customers in regulated industries can determine it during procurement.

---

6. Cited ADRs

The following Architecture Decision Records are referenced in this document. Each is canonical for the position it describes; this document is a summary, not a substitute.

- ADR-0019 — Portal client-side analytics posture (no client-side analytics).
- ADR-0053 — Dual identity key contract.
- ADR-0054 — file_id receive-side schema.
- ADR-0055 — Broadcast receive contract.
- ADR-0068 (iOS client) — SF-007 Double Ratchet release flag.
- ADR-0093 / 0094 / 0095 / 0097 — Retention policy and enforcement.
- ADR-0155 — Attomus-controlled storage migration.
- ADR-0157 — Attachment contract ciphertext-only.
- ADR-0158 — Broadcast contract per-recipient envelopes.
- ADR-0159 — Screenshot contract hash-only.
- ADR-0160 — Self-hosted runner hardening.
- ADR-0161 — TOTP and switch-org hardening.
- ADR-0162 — SeaweedFS deployment shape, containerised single-node.
- ADR-0163 — Access-token refresh bounded.
- ADR-0164 — Test-OTP bypass scoping.
- ADR-0169 — Canonical X3DH HKDF parameters.

---

7. Document maintenance

This document is maintained by Attomus engineering. Material changes to the cited ADRs trigger a review of the relevant section. Substantive criticism, expert review feedback, and security-researcher engagement is

welcomed at `security@attomus.com`; see the Vulnerability Disclosure Policy at `attomus.com/security/` for the disclosure path.

Where this document is wrong, it should be corrected. Where customer requirements identify gaps in the threat model or the architectural position, those gaps should be addressed in subsequent revisions.