Specification for: Gateway-Enforced Symmetric Intent Authentication (GESIA)

Version: 1.1
Design Philosophy: mRNP-Inspired, Payload-Blind, Symmetric Security, Builder-Enforced

---

1. The Core Architecture

The protocol follows a **Three-Tier Security Model** based on a shared secret (Ks) established via a secure handshake:

- **Secure Tunnel (Transport):** Standard TLS provides foundational encryption.
- **The Interceptor (Client-Side):** A programmable proxy (e.g., Service Worker) that signs request metadata at the point of creation.
- **The Validator (Server-Side):** High-performance gateway that verifies the signature **before any business logic or routing**, enforcing all invariants.

---

2. Technical Implementation Details

A. The Handshake (Key Exchange)

- **Generation:** ECDH (P-256) is used to derive a unique symmetric key (Ks).
- **Persistence:** Secret stored in Local Cache (Client) and Global Cache (Server), mapped to Key ID (KID).
- **Reactive Key Refresh:** On 401 Unauthorized, the Interceptor pauses traffic, triggers a new ECDH exchange, and retries pending requests.
- **Session Scoping:** Keys are short-lived (recommended 5–15 minutes TTL) with overlapping validity to prevent race conditions.

B. The Header Structure

A Base64-encoded JSON object containing:

Field	Description
kid	Key ID

Field	Description
v	Protocol version (v1.1)
ts	Unix timestamp (seconds)
nonce	Random high-entropy string (≥128-bit recommended)
sig	HMAC-SHA256 signature

Example Header (JSON → Base64):

```
{
  "kid": "abc123",
  "v": "v1.1",
  "ts": 1738502400,
  "nonce": "7b9f2c1a8e3f4d6b",
  "sig": "f2a7b1c4d3e8..."
}
```

C. The Metadata Signature (Stable + Entropic Fingerprint)

The signature is calculated on **metadata only** to maintain performance, Java 8 compatibility, and multi-instance determinism.

Fingerprint Formula:

```
Fingerprint = HMAC(Ks, Method + "|" + ContentLength + "|" + Timestamp + "|" + Nonce)
Signature   = HMAC(Ks, Fingerprint)
```

Implementation Standards:

Field	Requirement
Method	Uppercase (e.g., GET, POST)
ContentLength	"0" if no body is present
Timestamp	Unix epoch time in seconds
Nonce	Unique, high-entropy per request

Key Differences from Previous Canonical Path Binding:

- No dependency on URL/path representation; avoids canonicalization fragility.
- Ks-derived HMAC provides entropy and tamper resistance.
- Deterministic across all instances; multi-instance safe.
- Optional logical operation IDs can be included, but verification no longer relies on mutable application-provided identifiers.

D. The Payload-Blind Trade-off

GESIA intentionally signs **request metadata only** (Method, ContentLength, Timestamp, Nonce) rather than the full request body. This reflects the **“streaming-first” philosophy**:

- **Performance First:** Avoids body hashing to prevent stream exhaustion, enabling high-throughput, low-latency systems.
- **Compatibility:** Works with environments such as Java 8 where streaming bodies cannot be fully materialized efficiently.
- **Intent-Focused Security:** Ensures that the **client’s intent and structural integrity** are verifiable by the server, while allowing optional body hashing for high-sensitivity routes.
- **Extensible:** Body hash profiles can be added in future versions for routes that require full content integrity.

Rationale: By separating **intent from content**, GESIA maintains a lightweight, scalable verification mechanism while keeping the protocol flexible for future enhancements.

---

3. Comprehensive Security & Threat Analysis

3.1 Cryptographic Foundation

GESIA v1.1 relies on well-studied cryptographic primitives:

- ECDH (P-256) for key agreement
- HMAC-SHA256 for authentication
- TLS for transport confidentiality

Risk Surface: Limited to protocol semantics, enforcement, and operational correctness.

3.2 Threat Scenarios & Mitigations

Scenario	Mitigation / Notes
Equal-Length Payload Mutation	Enforces structural intent (method, content length, fingerprint). Bit-level payload mutation is out-of-scope. Optional body hash profiles for high-sensitivity routes.
Replay Flooding	±60s timestamp window, nonce tracking, bounded caches, rate-limiting per KID. Distributed Bloom filters recommended for global deployments.
Clock Desynchronization	Requires strict NTP synchronization. Server-time bootstrap recommended.
Header Stripping	Hard reject any request missing X-GESIA header before application logic.
Key Rotation Race	Overlapping TTL for old/new Ks; reactive key refresh handles expired keys. Monitor high-volume traffic to avoid handshake storms.

3.3 Stable + Entropic Fingerprint

Input -> HMAC(Ks, Metadata) -> Fingerprint -> HMAC(Ks, Fingerprint) -> Signature

- Deterministic and available to all runtime instances.
- Entropy provided via Ks + nonce, ensuring tamper resistance.
- Signature verification occurs **before any routing, middleware, or framework mutation**.
- Optional logical operation IDs may be included for auditing but are not required for multi-instance safety.

Rationale: Eliminates fragility and ambiguity in path normalization while preserving binding between client intent and server verification.

3.4 Why Not Signed URLs?

In modern cloud-native architectures, the URL or request path is a **mutable transport detail**, not a reliable security anchor.

- **Environment Variability:** Reverse proxies, API gateways, or serverless routing can modify paths (trailing slashes, redirects, rewriting).
- **Multi-Instance Safety:** Multiple instances serving the same application may see different path representations.
- **Operational Robustness:** Avoiding path binding eliminates fragile invariants that could lead to false negatives or unnecessary request rejection.

Rationale: Instead of signing the path, GESIA uses a **Stable + Entropic Fingerprint**, derived from Ks and high-entropy metadata, ensuring tamper resistance and determinism across all runtime instances.

---

4. Extensions & Self-Healing

- **Stateless Replay Defense:** Validator rejects requests with reused nonces within the valid timestamp window.
- **Reactive Key Refresh:** On 401 Unauthorized, the Interceptor triggers a new ECDH handshake and retries the request.
- **Strict Structural Enforcement:** Validator enforces Length Boundary Check; physical payload size must match ContentLength in signature.

---

5. Non-Negotiable Invariants

1. Version Binding: Signature MUST include protocol version; downgrade attempts rejected.
2. Hard Rejection: Missing headers or nonce reuse results in immediate failure.
3. Timestamp Signing: Timestamp is always part of HMAC.
4. Verification Priority: Signature verification occurs **before application logic**.
5. Session Scoping: Keys are short-lived (5–15 min TTL recommended).
6. Stable + Entropic Fingerprint Binding: Signature binds request metadata + Ks; independent of mutable identifiers or URL/path canonicalization.

6. Evaluation & Impact Matrix

Category	Assessment	Verdict / Notes
Performance	High	Metadata-only signing; hardware-accelerated HMAC; no body hashing by default
Streaming	Excellent	Zero stream exhaustion; compatible with Java 8
Scalability	Moderate	Shared cache required for KID and nonce tracking; distributed Bloom filters recommended for global scale
Integrity	Structural	Ensures intent, method, length; bit-level payload integrity optional via future profiles
Overall Quality	Advanced	Coherent, disciplined, enforceable primitive; aligns with builder-enforced philosophy

7. Strategic Positioning & Roadmap

GESIA is a sovereign application-layer authentication primitive:

- Provides structural and intent integrity
- Does **not** replace TLS
- Does **not** provide non-repudiation

Future Evolution:

Feature	Purpose
X25519 support	Modern curve alignment for future cryptographic agility
Optional Body Hash Profile	Integrity for high-sensitivity financial routes
Distributed Bloom Filters	Global replay defense across horizontal clusters

Operational Guidance:

- Ensure strict NTP synchronization
- Implement bounded nonce caches

- Enforce TTL and overlapping key windows
- Provide audit logging for rejected requests (KID, ts, nonce, failure reason)
- All optional features must be explicitly declared and version-bound in signature

---

✅ GESIA v1.1 Summary

- Fully **user-controlled, builder-enforced** primitive
- **Payload-blind, metadata-signed** for high performance
- Enforces **timestamp, nonce, version, and Stable + Entropic Fingerprint invariants**
- **Self-healing, reactive, and operationally resilient**
- Extensible in a disciplined, auditable manner

---

8. Appendix: Known Risks & Implementation Guidance

- TLS provides **confidentiality and integrity**; GESIA does not replace TLS.
- Bit-level payload mutations are **outside scope**; optional body hash profiles recommended.
- Fingerprint security depends on **strong entropy** from Ks + nonce.
- Excluding URL/path entirely is safer; optional invariants may be added.
- Nonce collisions are theoretically possible under extreme load; increase entropy if needed.
- Distributed deployments must **synchronize nonce caches**; misconfiguration can block requests or allow replay.
- Short TTLs reduce risk but increase handshake load. Reactive key refresh must handle pending requests **atomically**.
- Verification must occur **before routing, middleware, or framework mutation**.
- Audit logging of rejections is critical for monitoring and investigation.
- Metadata-only signing ensures high performance; optional body hashing can be route-specific.

- Future invariants or body hash profiles can be added **without breaking backward compatibility**.