# GOPEL Confidential Processing Extension (CPE) v1.1

**Extension to:** GOPEL Proof of Concept Specification v0.6.1; HAIA-RECCLIN Agent Architecture Specification, Appendix A

**Author:** Basil C. Puglisi, MPA

**Date:** March 2026

**Status:** Draft for Tier 0 human arbiter final signature

**Scope:** This extension addresses the privacy-during-computation gap in the GOPEL architecture. It specifies how GOPEL governs the exposure of sensitive data during AI platform inference without violating the non-cognitive constraint. It introduces a four-profile classification system that ensures every governed dispatch carries a deterministic privacy status, producing auditable evidence at every layer of protection. No dispatch passes through ungoverned. No scenario falls outside classification.

**Evidence tier:** Tier 2 (working concept specification). The extension applies established cryptographic and confidential computing technologies (Tier 1) to the GOPEL orchestration architecture in a novel integration pattern that has not yet been operationally tested within GOPEL.

**Version history:**

- v1.0: Initial draft. Submitted to six-platform CAIPR review (Gemini, Grok, DeepSeek, Kimi, ChatGPT, Claude informing prior analysis).
- v1.1: Incorporates all modifications and enhancements from CAIPR convergence analysis. Four targeted modifications (profile matrix logic, Profile 1 evidence split, compliance table language, record type count) and five enhancements (trust store HSM signing, tokenization coverage field, WebAssembly future note, Profile 2 advisory, "regulated" definition) integrated.

---

## 1. The Gap

GOPEL secures governed data at two points in the workflow. Before dispatch, data sits in append-only, hash-chained, digitally signed audit records. After collection, responses enter the same chain. Between those two points, custody transfers to an external AI platform. The platform receives the prompt in cleartext, processes it through its inference stack, and returns a response. GOPEL has zero visibility into what happens during that interval.

This is the maximum exposure window. The platform's inference stack, including CPU, GPU VRAM, attention key-value cache, speculative decoding buffers, and any internal logging or monitoring, operates entirely outside GOPEL's reach. GOPEL cannot determine whether the platform retained the prompt, logged it, exfiltrated it, or processed it in a protected environment.

The gap is structural, not incidental. GOPEL's seven deterministic operations (dispatch, collect, route, log, pause, hash, report) govern what moves between platforms and human authority. They do not govern what happens inside a platform's compute environment. That boundary exists by design: GOPEL performs zero cognitive work, and governing internal platform computation would require content evaluation that violates the non-cognitive constraint.

### 1.1 Why This Matters

GDPR Article 25 requires privacy by design during processing, not just at rest and in transit. A governance architecture that protects records before and after platform processing but produces no evidence of what happened during processing has a gap at the most vulnerable moment.

The EU AI Act Article 10 addresses data governance for high-risk AI systems, primarily focused on training, validation, and testing data. It is not the strongest direct citation for runtime prompt confidentiality during inference, but data governance completeness requires accounting for the inference pathway when governed data passes through a high-risk system.

DORA supports audit trail integrity and operational resilience through incident management and record-keeping requirements. It strengthens the case for recording what evidence exists about computation conditions but is not the primary legal anchor for privacy during processing.

The legal exposure is real. The strongest regulatory basis is GDPR Article 25. The gap matters most when governed workflows contain personal data, regulated financial data, health information, or classified material that passes through external AI platforms.

### 1.2 What Closing This Gap Requires

Any solution must meet four conditions:

1. **Preserve the non-cognitive constraint.** GOPEL performs zero semantic evaluation of content. It cannot inspect prompts, evaluate responses, or interpret platform claims about data handling. Any privacy mechanism must produce evidence verifiable through deterministic cryptographic checks only.

2. **Produce auditable evidence.** A regulator examining the audit trail must find concrete records documenting what privacy protections applied to each dispatch, what evidence the platform provided, and what governance decision the human arbiter made based on that evidence.

3. **Cover every dispatch scenario.** No governed dispatch can pass through without a privacy classification. Every scenario, from a fully attested confidential endpoint to an opaque third-party API, must produce a governance record with an explicit privacy status.

4. **State limitations honestly.** A specification that claims to close the gap when it cannot creates compliance misrepresentation risk worse than the gap itself. The specification must distinguish between verified confidential processing, minimized disclosure, and unverified computation, and report each status accurately.

---

## 2. The Architectural Position

**GOPEL cannot fully close the privacy-during-computation gap at the orchestration layer for opaque third-party AI platform APIs.**

This is the governing architectural position of this extension, not a caveat buried in a limitations section. When GOPEL dispatches a prompt to an external platform that does not expose remote attestation or equivalent verifiable evidence, the privacy status of that computation is unknown. GOPEL can govern what happens before and after. It cannot verify what happens during.

What GOPEL can do is make the gap auditable, enforceable, and visible. The Confidential Processing Extension ensures that every dispatch carries a deterministic privacy profile, every profile maps to a specific evidence standard, and every deviation triggers a governance checkpoint. The gap is managed through structural enforcement, not eliminated through technical proof.

For platforms that do expose confidential computing evidence, GOPEL can verify that evidence deterministically and produce audit records that support regulatory evidentiary requirements. For platforms that do not, GOPEL can enforce data minimization, document the residual risk, and ensure human arbiters make informed decisions about what data reaches unverified environments.

This extension converts an invisible gap into a governed, auditable, classified exposure with explicit human accountability at every decision point.

---

## 3. The Four-Profile Classification

Every GOPEL dispatch is assigned exactly one Confidential Processing profile before the Dispatch operation executes. There is no fifth option. No dispatch passes through unclassified.

In cases where the profile assignment matrix produces a mandatory Pause gate (see Section 4), the matrix assigns a provisional profile and the Pause operation triggers before Dispatch executes. The human arbiter then confirms, upgrades, or rejects the provisional assignment. The provisional profile and the final arbiter-approved profile are both logged as audit records. The "exactly one profile" invariant holds because the arbiter resolves the assignment before any data leaves GOPEL's custody.

### Profile 0: Opaque External Processing

**Description:** The target AI platform is a standard API endpoint with no remote attestation, no confidential computing evidence, and no signed inference receipts. GOPEL dispatches the prompt and collects the response with no evidence of what happened during computation.

**Allowed data:** Public data or data already transformed under Profile 2 tokenization. Raw sensitive data must not be dispatched under Profile 0 without explicit human arbiter override, documented in the audit trail with rationale.

**Report status:** "UNVERIFIED DURING COMPUTATION"

**Governance controls:** - GOPEL logs the absence of attestation evidence as an Unverified Processing Record with `attestation_present: false`. - If the data sensitivity classification (set by governance policy, not by GOPEL content evaluation) indicates sensitive, regulated, or confidential data, the Pause operation triggers and the checkpoint package routes to the human arbiter with the privacy gap explicitly flagged. - The human arbiter may override and authorize dispatch with documented rationale. The override is itself an audit record. - Report operation includes this dispatch in the "unverified computation" count.

**What this profile does:** It ensures that opaque dispatches are never silent. Every dispatch to an unverified endpoint produces a record. Every sensitive dispatch to an unverified endpoint requires human authorization. The gap exists, but it is governed, documented, and accountable.

**What this profile does not do:** It does not protect data during computation. It does not verify platform behavior. It documents the absence of protection and places the governance decision on the human arbiter.

---

### Profile 1: Attested Confidential Inference

**Description:** The target AI platform runs inference inside a hardware-enforced Trusted Execution Environment (TEE) and exposes remote attestation evidence to GOPEL. The platform provides a signed attestation quote proving that a specific, verified code image ran inside specific hardware with specific security configuration. The platform may additionally provide attested key release (where the decryption key for the prompt is released only after attestation succeeds) and signed inference receipts binding model identity to input and output hashes.

**Allowed data:** Raw sensitive data, regulated data, personal data, classified material. This is the only profile that permits raw sensitive data dispatch without transformation or human override.

**Report status (two evidence grades within Profile 1):**

- **“VERIFIED ATTESTED ENVIRONMENT”** — Attestation passed (four binary checks described below), but no signed inference receipt is present. The compute environment is verified against the pre-approved configuration. The binding between the attestation and the specific response rests on temporal correlation and session nonce. This grade confirms where computation occurred but does not cryptographically bind the specific transaction to the attested environment.
- **“VERIFIED CONFIDENTIAL PROCESSING”** — Attestation passed AND a signed inference receipt is present with a matching input hash (the hash of the dispatched prompt matches the input hash in the receipt). Full evidence chain from environment verification through transaction binding. This is the strongest evidence grade GOPEL can produce. It confirms both the environment and the specific transaction.

The evidence grade is determined by the evidence available from the platform, not by GOPEL content evaluation. Receipt present with matching hash produces the stronger grade. Receipt absent produces the weaker grade. Both grades are deterministic.

**Governance controls (Attestation-Gated Dispatch):**

**Step 1: Pre-dispatch attestation request.** Before dispatching the prompt, GOPEL requests a remote attestation quote from the target platform endpoint. The attestation quote is a cryptographic structure produced by the TEE hardware, signed by the hardware vendor’s attestation key. It contains the enclave measurement (hash of the code running inside the TEE), the platform security version number (PSVN or TCB status), and a freshness nonce provided by GOPEL.

**Step 2: Deterministic verification.** GOPEL performs four deterministic checks on the attestation quote. Each check is a binary pass/fail with no content evaluation:

- **Signature verification:** The attestation quote signature is verified against the hardware vendor’s root certificate stored in GOPEL’s trust store. Valid or invalid.
- **Measurement allowlist check:** The enclave measurement hash is compared against a pre-approved list of known-good platform images maintained as a signed configuration artifact. Match or no match.
- **Freshness verification:** The nonce embedded in the attestation quote is compared against the nonce GOPEL issued. Match or no match. This prevents replay of stale attestation quotes and mitigates attestation relay attacks where a platform forwards a valid quote from a different enclave.
- **TCB status check:** The platform security version is compared against the minimum acceptable version specified in the deployment configuration. Greater-than-or-equal or below minimum.

All four checks are deterministic. GOPEL does not evaluate what the platform does with the data. It verifies that the platform’s compute environment matches a pre-approved configuration.

**Step 3: Dispatch or pause.** If all four checks pass, the Dispatch operation proceeds. If any check fails, the Pause operation triggers and the checkpoint package routes to the human arbiter with the specific failure identified.

**Step 4: Post-processing evidence collection.** When GOPEL collects the platform's response (Collect operation), it also collects:

- A post-processing attestation or signed inference receipt, if the platform provides one.
- The receipt binds the platform identity, model identifier, input hash, output hash, attestation token hash, and a timestamp.
- GOPEL verifies the receipt signature and checks that the input hash matches the hash of the dispatched prompt. Both are deterministic checks.
- If the receipt is present and the input hash matches, the evidence grade is "VERIFIED CONFIDENTIAL PROCESSING." If the receipt is absent or the input hash does not match, the evidence grade is "VERIFIED ATTESTED ENVIRONMENT."

**Step 5: Audit logging.** GOPEL produces a Privacy Attestation Record containing:

| Field | Content |
|---|---|
| record_type | PRIVACY_ATTESTATION |
| profile | 1 (Attested Confidential Inference) |
| evidence_grade | VERIFIED_ATTESTED_ENVIRONMENT or VERIFIED_CONFIDENTIAL_PROCESSING |
| platform_id | Platform identifier |
| attestation_present | true |
| attestation_quote_hash | SHA-256 hash of the raw attestation quote |
| enclave_measurement | Measurement hash from the attestation quote |
| tee_type | AMD_SEV_SNP, INTEL_TDX, INTEL_SGX, ARM_CCA, NVIDIA_H100_CC, or equivalent |
| tcb_status | Platform security version or TCB status string |
| nonce | The freshness nonce GOPEL issued |
| nonce_match | true/false |
| measurement_allowlist_version | Version identifier of the allowlist configuration |
| measurement_match | true/false |
| signature_valid | true/false |
| tcb_sufficient | true/false |
| verification_result | PASS or FAIL |
| inference_receipt_present | true/false |
| inference_receipt_hash | SHA-256 hash of the signed inference receipt (if present) |
| input_hash_match | true/false (receipt input hash vs. dispatched prompt hash) |

| Field | Content |
|---|---|
| data_sensitivity_classification | Policy label applied to this dispatch |
| timestamp | ISO 8601 UTC, millisecond precision |
| previous_hash | Hash chain link |
| record_hash | SHA-256 of canonical record |
| signature | Deployment signing key signature |

**Attested key release (optional, strongest configuration):** In the strongest deployment configuration, GOPEL does not dispatch the prompt in cleartext. Instead, GOPEL encrypts the prompt to the enclave's public key. The decryption key is released to the enclave only after the attestation quote verifies against a key management service (such as AWS KMS with Nitro Enclave attestation conditions, or Azure Attestation with policy-driven key release). The platform never receives plaintext outside the enclave boundary.

**What this profile does:** It produces hardware-backed cryptographic evidence that the platform's compute environment matched a pre-approved configuration at the time of dispatch. At the stronger evidence grade, it additionally binds the specific transaction to the attested environment through a signed receipt with matching input hash.

**What this profile does not do:** It does not prove that the code inside the enclave handled the prompt correctly during inference. Attestation proves the environment, not the runtime behavior. GPU VRAM on most current architectures sits outside CPU enclave boundaries (though NVIDIA H100 Confidential Computing extends protection to GPU memory). The attestation proves code integrity, not that the specific prompt was never logged internally by the attested code. Model memorization of sensitive data from the prompt is outside the scope of TEE protection.

**Residual risks:** - Trust in the hardware vendor's TEE implementation (Intel, AMD, ARM, NVIDIA). Side-channel attacks against enclaves have been demonstrated in research (CacheOut, SGAxe) and require vendor patching. - Trust in the cloud provider's correct deployment of the TEE infrastructure. - Without a signed inference receipt, the binding between attestation and the specific response rests on temporal correlation. The weaker evidence grade reflects this. - Supply chain compromise of CPU microcode is a theoretical vector.

---

### Profile 2: Minimized External Processing

**Description:** GOPEL applies deterministic data minimization transformations to the prompt before dispatching to an opaque external endpoint. The platform receives a transformed prompt with sensitive fields tokenized, redacted, or replaced. The platform processes the minimized representation. After collection, GOPEL applies the reverse transformation if needed.

Profile 2 provides partial minimization of structured sensitive data, not comprehensive anonymization. Sensitive information that does not match a pre-defined pattern passes to

the platform in cleartext. Organizations should treat Profile 2 as a disclosure reduction measure, not an equivalence to confidential processing.

**Allowed data:** Sensitive data after transformation under an approved minimization policy. The platform never receives raw sensitive fields that match the tokenization ruleset. The minimized representation may still carry contextual sensitivity (see residual risks).

**Report status:** "DISCLOSURE MINIMIZED — CONFIDENTIAL PROCESSING NOT VERIFIED"

**Governance controls (Pre-Dispatch Tokenization):**

**Step 1: Policy-driven field classification.** GOPEL does not determine what is sensitive. Sensitivity classifications are established by governance policy, human controllers, or external classifiers operating outside GOPEL. GOPEL receives a data sensitivity label (public, internal, confidential, regulated) attached to each dispatch by the upstream governance workflow. The label is a metadata field, not a content evaluation.

**Step 2: Deterministic tokenization.** GOPEL applies a pre-compiled tokenization ruleset to the prompt before dispatch. The ruleset specifies pattern-based transformations:

- Format-matching patterns: Social security numbers, email addresses, phone numbers, account numbers, dates, IP addresses, and other structured PII formats are matched via regex patterns and replaced with consistent placeholder tokens.
- Named field replacements: If the prompt is structured (JSON, XML, or field-delimited), specific fields identified by the policy as sensitive are replaced regardless of content format.
- Consistent tokenization: The same input value always maps to the same token within a session, preserving referential consistency in the prompt without exposing the raw value.

The tokenization ruleset is a versioned, signed policy artifact. GOPEL does not decide which fields to tokenize. It applies the ruleset specified by governance policy. This is a deterministic string transformation, not content evaluation.

**Step 3: Token map management.** The mapping between raw values and tokens is stored as a signed, encrypted artifact within GOPEL's custody boundary. The token map never leaves GOPEL. The platform never receives the map. Detokenization of the response (if needed) occurs only within GOPEL's custody or within an attested enclave under Profile 1 controls.

**Step 4: Dispatch and collection.** GOPEL dispatches the tokenized prompt to the platform. The platform processes the minimized representation and returns a response containing tokens rather than raw values. GOPEL collects the response normally.

**Step 5: Audit logging.** GOPEL produces a Minimization Record containing:

| Field | Content |
|---|---|
| record_type | MINIMIZATION_RECORD |
| profile | 2 (Minimized External Processing) |

| Field | Content |
|---|---|
| platform_id | Platform identifier |
| tokenization_ruleset_version | Version identifier of the applied ruleset |
| tokenization_ruleset_hash | SHA-256 of the ruleset artifact |
| fields_tokenized_count | Integer count of fields transformed |
| tokenization_coverage_pct | Percentage of identified sensitive fields successfully tokenized (optional; supports DPIA filings) |
| data_sensitivity_classification | Policy label |
| attestation_present | false (this is an opaque endpoint) |
| timestamp | ISO 8601 UTC, millisecond precision |
| previous_hash | Hash chain link |
| record_hash | SHA-256 of canonical record |
| signature | Deployment signing key signature |

**What this profile does:** It reduces the sensitive data that reaches the platform. The audit trail documents exactly which ruleset governed the transformation, how many fields were affected, and what policy authorized the dispatch.

**What this profile does not do:** It does not verify what the platform did with the minimized data during processing. It does not protect against contextual inference, where the surrounding text allows reconstruction of tokenized values. It does not protect unstructured sensitive content that does not match a pattern rule.

**Residual risks:** - Pattern coverage is finite. Sensitive information that does not match a pre-defined pattern passes to the platform in cleartext. - Contextual inference: even with tokens, the surrounding text may allow a sophisticated adversary or the model itself to infer the redacted values. - Prompt quality degradation: tokenization may reduce the quality of the platform's response if the redacted information is necessary for reasoning. - The platform still processes in an opaque environment. Minimization reduces what is exposed but does not verify the computation conditions.

---

### Profile 3: Cryptographic Inference (Experimental)

**Description:** The target platform or custom inference stack supports computation on encrypted data via Fully Homomorphic Encryption (FHE) or distributes computation across multiple non-colluding parties via Secure Multi-Party Computation (SMPC). GOPEL dispatches encrypted or share-split inputs and collects encrypted or reconstructed outputs.

**Allowed data:** Research and experimental workloads only. This profile cannot be claimed as general deployable compliance infrastructure.

**Report status:** "EXPERIMENTAL CRYPTOGRAPHIC INFERENCE — NOT VALIDATED FOR PRODUCTION COMPLIANCE"

**Why this profile exists:** FHE and SMPC represent the theoretical ideal: computation where the platform never accesses plaintext. Current research shows real progress but neither is deployable at production scale against standard LLM APIs today. This profile exists to provide governance structure for organizations that operate custom inference stacks where these technologies are viable, and to establish the specification framework so GOPEL is ready when the technology matures.

**Governance controls:** GOPEL dispatches ciphertext or shares. GOPEL collects ciphertext or reconstruction receipts. GOPEL verifies cryptographic proof artifacts (if available) through deterministic signature and hash checks. All operations remain non-cognitive.

**What this profile does not do:** It does not validate that FHE or SMPC inference produces equivalent results to cleartext inference. It does not verify computational correctness beyond what the proof system provides. It does not claim production readiness. Zero production compliance claims may rely on undeployed cryptographic inference paths.

**Deployment status:** Tier 3 (proposed for future development). Minimum 5-year horizon for FHE-based LLM inference at production scale. SMPC for LLM inference is research-stage with no clear production timeline.

---

## 4. Catch-All Enforcement Rule

**Every GOPEL dispatch must carry an assigned CPE profile before the Dispatch operation executes.** Profile assignment is a deterministic lookup based on two inputs:

1. **Endpoint capability declaration:** Each platform endpoint registered in GOPEL's configuration carries a capability flag: `attested_confidential`, `opaque_external`, or `experimental_cryptographic`. This flag is set at deployment configuration time, not at runtime. GOPEL does not evaluate the endpoint to determine its capability. The configuration is a signed policy artifact maintained by the deploying organization's security team.

2. **Data sensitivity classification:** Each dispatch carries a sensitivity label assigned by governance policy: `public`, `internal`, `confidential`, or `regulated`. GOPEL does not assign this label. It is provided by the upstream governance workflow, the human controller, or an external classifier operating outside GOPEL. The "regulated" classification is deployment-specific and must be mapped to applicable legal frameworks (GDPR, HIPAA, SOX, GLBA, or other jurisdiction-specific requirements) in the deploying organization's data governance policy.

**Defaults fail closed:** If the endpoint capability flag is missing from the configuration, GOPEL treats the endpoint as opaque (Profile 0). If the data sensitivity label is missing from the dispatch, GOPEL treats the data as regulated (maximum restriction).

The profile assignment matrix:

| Data Classification | Attested Endpoint | Opaque Endpoint | Experimental Endpoint |
|---|---|---|---|
| Public | Profile 1 | Profile 0 | Profile 3 |
| Internal | Profile 1 | Profile 0 (with logging) | Profile 3 |
| Confidential | Profile 1 | Profile 2 (provisional) + mandatory Pause | Profile 3 |
| Regulated | Profile 1 only | Profile 0 (provisional) + mandatory Pause | Not permitted |

**Mandatory Pause semantics:** When the matrix produces a mandatory Pause, the process is:

1. The matrix assigns a provisional profile (Profile 2 for confidential-to-opaque, Profile 0 for regulated-to-opaque).
2. The Pause operation triggers before Dispatch executes. No data leaves GOPEL's custody.
3. The checkpoint package delivered to the human arbiter includes the provisional profile, the data sensitivity classification, the endpoint capability flag, and an explicit notation that no privacy-during-computation evidence is available for this endpoint.
4. The arbiter decides: confirm the provisional profile (authorizing dispatch under Profile 2 with tokenization or Profile 0 without), upgrade (e.g., redirect to an attested endpoint under Profile 1), or reject (no dispatch).
5. The provisional profile assignment and the arbiter's final decision are both logged as audit records. The final arbiter-approved profile becomes the profile of record for this dispatch.
6. Dispatch executes only after the arbiter resolves the assignment.

The "exactly one profile" invariant holds: every dispatch carries a single resolved profile before data leaves GOPEL's custody. Mandatory Pause ensures the arbiter resolves ambiguous assignments before execution, not after.

**Regulated data to an experimental endpoint is not permitted.** This combination does not receive a provisional profile or a Pause gate. It is blocked. Experimental cryptographic inference cannot be claimed as compliance infrastructure for regulated data.

---

## 5. Integration with GOPEL's Seven Operations

The CPE does not add new cognitive operations. It extends existing operations with deterministic checks and introduces three new audit record types: Privacy Attestation Record (Profile 1), Minimization Record (Profile 2), and Unverified Processing Record (Profile 0).

| Operation | CPE Extension |
|---|---|
| **Dispatch** | Before dispatch: (1) Look up endpoint capability flag (config lookup). (2) Look |

| Operation | CPE Extension |
|---|---|
| | up data sensitivity label (metadata read). (3) Assign CPE profile (matrix lookup). (4) If mandatory Pause: trigger Pause, await arbiter resolution. (5) If Profile 1: execute attestation verification (four binary checks). (6) If Profile 2: execute tokenization (pattern-based string transformation). (7) Dispatch only after profile requirements are met and any Pause is resolved. |
| **Collect** | If Profile 1: collect attestation evidence and inference receipt alongside response. Verify receipt signature and input hash match. Assign evidence grade. If Profile 2: response contains tokens; detokenization occurs within GOPEL custody. |
| **Route** | If attestation verification failed or receipt is inconsistent, route to human arbiter instead of Navigator. |
| **Log** | Write Privacy Attestation Record (Profile 1), Minimization Record (Profile 2), or Unverified Processing Record (Profile 0) to the audit chain. Log provisional profile and final arbiter-approved profile separately when mandatory Pause was triggered. |
| **Pause** | Triggered by: attestation failure, missing attestation for sensitive data, sensitive data routed to opaque endpoint (mandatory Pause from matrix), inconsistent pre/post attestation, or receipt signature failure. |
| **Hash** | All CPE records are hash-chained into the audit trail identically to existing record types. Attestation quotes, receipt hashes, tokenization ruleset hashes, and evidence grades are included in the chain. |
| **Report** | CPE metrics added to governance reports: count of dispatches by profile, count by evidence grade within Profile 1, percentage of sensitive dispatches under Profile 1, count of Pause events triggered by attestation failures, count of mandatory Pause events from matrix assignments, count of human overrides authorizing Profile 0 for sensitive data, tokenization coverage statistics for Profile 2 dispatches. |

## 6. Trust Store and Configuration Management

### 6.1 Attestation Trust Store

GOPEL maintains a trust store containing:

- Hardware vendor root certificates (Intel, AMD, ARM, NVIDIA) for attestation quote signature verification.
- An approved measurement allowlist: enclave measurement hashes corresponding to known-good, verified platform inference images.
- Minimum acceptable TCB/PSVN values per TEE type.
- A freshness nonce generation seed (cryptographic random, not content-derived).

The trust store is a signed configuration artifact. The trust store configuration artifact must be signed using the same HSM-backed key material used for audit record signing. Signature

verification of the trust store occurs at load time and on each configuration update. This ensures that compromise of the filesystem cannot modify the approved measurement allowlist without detection.

Updates to the trust store are themselves audit events producing Configuration Update Records in the chain. The trust store is maintained by the deploying organization's security team, not by GOPEL. GOPEL reads the trust store; it does not decide what belongs in it.

**Update frequency:** Trust store updates must not exceed 90 days. Hardware vendor security bulletins, enclave measurement changes from platform updates, and TCB version changes must trigger out-of-cycle reviews.

**Future consideration:** As WebAssembly-based verification components mature (demonstrated in research for AMD SEV-SNP and Intel TDX attestation evidence), GOPEL may adopt platform-specific verification logic bundled with evidence rather than maintained in GOPEL's codebase. This would reduce vendor-specific code in the core system while maintaining deterministic verification. Adoption depends on standardization and security audit of WebAssembly verification components.

### 6.2 Tokenization Ruleset

GOPEL maintains tokenization rulesets as versioned, signed policy artifacts. Each ruleset contains:

- Regex patterns for structured PII formats (SSN, email, phone, account number, date, IP address, etc.).
- Named field replacement rules for structured prompt formats.
- Consistent tokenization mappings (same input value produces same token within a session).

Rulesets are maintained by the deploying organization's data governance team. GOPEL applies the ruleset specified by governance policy. Ruleset updates produce Configuration Update Records.

### 6.3 Endpoint Capability Registry

Each platform endpoint registered in GOPEL's configuration carries:

- Endpoint identifier.
- Capability flag: `attested_confidential`, `opaque_external`, or `experimental_cryptographic`.
- TEE type (if attested): AMD_SEV_SNP, INTEL_TDX, INTEL_SGX, ARM_CCA, NVIDIA_H100_CC.
- Attestation API endpoint URL (if attested).
- Data processing agreement reference (if applicable).
- Last attestation verification timestamp.

The registry is a signed configuration artifact maintained by the deploying organization.

## 7. What This Extension Does

1. **Classifies every dispatch.** No governed data moves to any AI platform without an assigned privacy profile and a corresponding audit record.

2. **Verifies confidential computing evidence where available.** When platforms expose TEE attestation, GOPEL verifies it through four deterministic binary checks and produces hardware-backed evidence. When signed inference receipts are present, GOPEL binds the specific transaction to the attested environment for the strongest evidence grade.

3. **Reduces disclosure where confidential computing is unavailable.** When platforms are opaque, GOPEL applies deterministic tokenization to reduce the sensitive data that reaches the platform, documenting exactly what transformation governed the dispatch.

4. **Fails closed on sensitive data.** Confidential or regulated data dispatched to an unverified endpoint triggers a mandatory human governance checkpoint with a provisional profile assignment. The arbiter resolves the assignment before any data leaves GOPEL's custody. The default is deny, not permit.

5. **Reports honestly.** The report status for each profile and evidence grade states exactly what GOPEL verified and what it did not:

   - Profile 1 (with receipt): "VERIFIED CONFIDENTIAL PROCESSING"
   - Profile 1 (without receipt): "VERIFIED ATTESTED ENVIRONMENT"
   - Profile 2: "DISCLOSURE MINIMIZED — CONFIDENTIAL PROCESSING NOT VERIFIED"
   - Profile 0: "UNVERIFIED DURING COMPUTATION"
   - Profile 3: "EXPERIMENTAL — NOT VALIDATED FOR PRODUCTION"

6. **Preserves the non-cognitive constraint.** Every check is binary and deterministic. Signature valid or invalid. Measurement match or mismatch. Nonce match or mismatch. TCB sufficient or insufficient. Attestation present or absent. Pattern match or no match. Receipt present or absent. Input hash match or mismatch. GOPEL never evaluates what the data means, whether the platform's behavior was appropriate, or whether the attestation evidence is truthful.

## 8. What This Extension Does Not Do

1. **It does not guarantee privacy during computation.** No orchestration layer can guarantee what happens inside an external platform's compute environment. Profile 1 produces the strongest available evidence. It is not mathematical proof that the platform protected the data. It is hardware-backed evidence that the platform's environment matched a verified configuration, with optional transaction binding through signed receipts.

2. **It does not verify runtime behavior inside the enclave.** Attestation proves code integrity at the point of attestation. It does not prove the attested code handled the specific prompt correctly during the specific inference. A platform running verified code in a verified enclave could still log, retain, or mishandle data within the attested environment.

3. **It does not protect against model memorization.** An LLM that memorizes sensitive prompt data during inference can reproduce that data in future responses to other users. TEE protection prevents external access to the data during computation but does not prevent the model's weights from encoding the data.

4. **It does not solve the GPU VRAM boundary problem on all architectures.** CPU-based TEEs (SGX, SEV-SNP) protect main memory but not discrete GPU VRAM on most current architectures. NVIDIA H100 Confidential Computing extends protection to GPU memory with less than 5-7% performance overhead, but this requires specific hardware and deployment configuration.

5. **It does not eliminate trust in hardware vendors.** TEE attestation shifts the trust boundary from the AI platform to the hardware vendor. Side-channel attacks against enclaves have been demonstrated in research (CacheOut, SGAxe). The trust is better founded (hardware-backed rather than assertion-backed) but not absolute.

6. **It does not make opaque endpoints safe for sensitive data.** Profile 0 and Profile 2 do not verify computation privacy. They govern the dispatch decision, document the residual risk, and ensure human accountability. The gap remains for opaque endpoints. The extension makes it visible and governed, not closed.

7. **It does not prove data deletion.** TEE attestation proves an execution environment existed. It does not prove the platform deleted the prompt data after inference. Ephemeral enclaves mitigate this but the extension does not claim deletion assurance.

---

## 9. Regulatory Compliance Mapping

| Requirement | Profile 0 | Profile 1 (Attested Environment) | Profile 1 (Confidential Processing) | Profile 2 | Profile 3 |
|---|---|---|---|---|---|
| GDPR Art. 25: Privacy by design | Gap documented. Evidentiary posture depends on human override rationale. | Supports Art. 25 evidentiary posture through environment verification. | Strongest current technical support for Art. 25 evidentiary posture. | Supports Art. 25 evidentiary posture through documented minimization. | Experimental. Cannot support compliance claims. |

| Requirement | Profile 0 | Profile 1 (Attested Environment) | Profile 1 (Confidential Processing) | Profile 2 | Profile 3 |
|---|---|---|---|---|---|
| EU AI Act Art. 10: Data governance | Audit trail complete but privacy gap documented. | Audit trail complete with attestation evidence. | Audit trail complete with attestation and receipt evidence. | Audit trail complete with minimization records. | Audit trail complete. Research context only. |
| DORA Art. 9/12: Operational resilience and records | Audit records present for all dispatches. | Full evidence chain including attestation. | Full evidence chain including attestation and receipt. | Full evidence chain including minimization. | Full evidence chain. |

**Note on GDPR Article 25 compliance:** Article 25 requires controllers to implement appropriate technical and organisational measures, taking account of the state of the art and related factors. It is structured around design-time architectural choices, not per-inference cryptographic proof. An organization that implements Profile 1 as the default for sensitive data, documents the architectural measures in a DPIA, and maintains auditable attestation records for every governed dispatch is producing the strongest currently available evidence of privacy by design during processing. Whether that evidence constitutes sufficient compliance is a legal and regulatory interpretation, not a technical determination. GOPEL produces the evidence. Legal counsel determines its sufficiency.

**Note on EU AI Act Article 10:** Article 10 addresses data governance primarily for training, validation, and testing data in high-risk AI systems. Runtime prompt confidentiality during inference is not the direct subject of Article 10. The CPE supports data governance completeness by accounting for the inference pathway, but Article 10 should not be overclaimed as the primary basis for runtime privacy-during-computation requirements.

**Note on DORA:** DORA supports audit trail integrity through incident management (Article 17) and operational resilience requirements (Articles 9, 12). It strengthens the evidentiary value of GOPEL's audit records but is not the primary legal anchor for privacy during processing.

---

## 10. Deployment Path

**Phase 1 (2026, immediate):** Implement Profile 0 and Profile 2 for all governed dispatches. Every dispatch carries a classification. Sensitive data to opaque endpoints triggers mandatory Pause with provisional profile assignment. Tokenization rulesets reduce disclosure. This requires no platform cooperation and no API changes.

**Phase 2 (2026-2027, with enterprise agreements):** Implement Profile 1 for platforms operating in confidential computing environments. This requires platform cooperation: the platform must expose an attestation API endpoint or provide signed inference receipts. AWS Nitro Enclaves, Azure Confidential Computing, Google Confidential VMs, and NVIDIA H100 Confidential Computing support the underlying infrastructure today. Adoption depends on platform willingness to expose attestation to orchestrators.

**Phase 3 (2027-2028, industry standardization):** As confidential AI inference becomes standard practice (analogous to TLS becoming universal for web traffic), Profile 1 becomes the default for all sensitive data dispatches. GOPEL maintains an allowlist of CPE-compliant platform endpoints. Non-compliant platforms trigger automatic human arbitration.

**Phase 4 (2030+, cryptographic inference):** Profile 3 transitions from experimental to operational as FHE or SMPC technologies mature for LLM inference at production scale. The specification framework is already in place. Implementation depends on cryptographic and hardware advances outside GOPEL's control.

---

## 11. Decision Point for Human Arbiter

**Approve, modify, or reject the following:**

1. Adopt the four-profile CPE classification (Opaque External, Attested Confidential with two evidence grades, Minimized External, Cryptographic Experimental) as the governing framework for privacy during computation.
2. Adopt the catch-all enforcement rule: every dispatch must carry an assigned CPE profile resolved before Dispatch executes, with mandatory Pause and provisional profile assignment for ambiguous matrix cells, defaults fail closed.
3. Adopt the attestation-gated dispatch mechanism for Profile 1 (four deterministic binary checks, two evidence grades, Privacy Attestation Record).
4. Adopt pre-dispatch tokenization for Profile 2 (pattern-based deterministic transformation, Minimization Record with optional coverage percentage).
5. Adopt the profile assignment matrix mapping data sensitivity to endpoint capability, with mandatory Pause semantics for confidential/regulated data to opaque endpoints.
6. Adopt the governing architectural position: GOPEL cannot fully close the privacy-during-computation gap at the orchestration layer for opaque third-party APIs. The extension manages the gap through structural enforcement, honest reporting, and human accountability.
7. Adopt trust store HSM signing requirement and configuration management controls.
8. Adopt the corrected legal framing: GDPR Article 25 as primary anchor with evidentiary posture language, tightened AI Act Article 10 and DORA citations.
9. Classify the extension as Tier 2 (working concept specification) with Phase 1 deployable immediately.

**Recommendation:** Approve all nine items. This extension covers every dispatch scenario, produces auditable evidence at every layer, preserves the non-cognitive constraint, reports two distinct evidence grades for attested environments, and states its limitations honestly. It is publishable as prior art and establishes the specification framework before any competitor in the space.

---

## 12. CAIPR Review Record

This extension was informed by a six-platform CAIPR review (Claude, Gemini, Grok, DeepSeek, Kimi, ChatGPT).

### Initial Review (v1.0)

| Platform | Primary Contribution |
|---|---|
| Claude | Three-layer combination architecture. Binary check pattern (present/absent, valid/invalid, match/mismatch). "Platform can lie" residual risk. Provider plurality as defense. |
| Gemini | Attestation-gated dispatch with measurement allowlist against hardware vendor CA. GPU VRAM boundary limitation. |
| Grok | NVIDIA H100 Confidential Computing as production-ready TEE. ZKML as emerging verification layer. |
| DeepSeek | Most complete Appendix A.6 draft. Trust update mechanism. Compliance matrix. |
| Kimi | Fullest implementation specification. YAML config, JSON audit schema, HTTP header extension, three-phase deployment. Nonce binding for anti-replay. ATTEST operation. |
| ChatGPT | Four-profile CPE classification (adopted). IETF RATS/EAT/AIR evidence model. Attested key release. Legal framing correction. Normative rule: GOPEL cannot certify what it cannot verify. |

### Verification Review (v1.0 to v1.1)

| Platform | Disposition | Key Contributions |
|---|---|---|
| Gemini | Approve, no modifications | Confirmed architectural soundness |
| Grok | Approve, one optional enhancement | Tokenization coverage percentage field |
| DeepSeek | Approve, one enhancement | Trust store HSM signing requirement |
| Kimi | Approve, three minor items | WebAssembly future note, Profile 2 advisory, "regulated" definition, TEE technical verification against vendor docs |
| ChatGPT | Approve with targeted | Matrix logic fix (provisional profile + mandatory Pause), Profile 1 evidence grade split, compliance table |

| Platform | Disposition | Key Contributions |
|---|---|---|
| | modifications | evidentiary posture language, record type count correction |

All six platforms confirmed the architecture is sound across both review rounds. No platform rejected. The modifications and enhancements are incorporated in this version.

---

**Document version:** v1.1 Draft

**Requires:** Tier 0 human arbiter final signature before publication.