The Stateful Isolation Law

Formal Contamination Bounds for Multi-Principal AI Systems

PUBLIC VERSION

Full specifications, threshold derivations, and implementation constants available under commercial license from the Auburn Patent Family.

Ryan Fields

Auburn Patent Family

UncleBroFields@proton.me

February 2026

License: Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0). Academic and non-commercial use permitted with full attribution. Commercial use strictly prohibited without license. See IP Declaration (Section 12) for details.

Abstract

Modern AI inference systems transform large language models from stateless functions into probabilistic databases through the KV cache—a persistent state structure that, at 128K context with FP16 precision, constitutes approximately 16 GB of continuously updated per-session data. The economics of cloud inference demand that this state be shared: continuous batching, PagedAttention, prefix caching, and FlashAttention's shared-SRAM tiling create physical entanglement between the computational states of co-resident principals.

We demonstrate that this entanglement constitutes a new class of security failure that existing isolation frameworks—Bell-LaPadula, Biba, Goguen-Meseguer noninterference, database serializability, and separation kernels—cannot address. We catalog 25+ demonstrated attacks achieving up to 99% prompt reconstruction from timing alone, 181 MB of leaked GPU memory per query, 82.4% inter-agent compromise rates, and exact gradient inversion with ROUGE > 0.99.

We introduce the **Stateful Isolation Law**: a unified formal framework based on a contamination functional parameterized by Rényi divergence that recovers differential privacy, mutual information bounds, and classical noninterference as special cases. The law comprises a General Principle, six enforceable clauses covering every contamination surface in the inference stack (KV drift, spatial isolation, timing channels, context integrity, memory hygiene, and gradient isolation), a Composability Theorem establishing when clause-level compliance propagates to system-level guarantees, four graduated compliance tiers (Development through Sovereign), and a cryptographic enforcement protocol.

This public version presents the complete framework architecture, threat landscape, clause definitions, theorem statements, compliance tiers, and regulatory mapping. Threshold derivations, explicit constants, scaling laws, proof bodies, and implementation specifications are held under the Auburn Patent Family and available under commercial license.

Contents

1 Introduction

1.1 The KV Cache as a Probabilistic Database

A large language model at inference time is not a function. It is a stateful system. The mechanism that creates this state is the key-value (KV) cache: a data structure that stores the intermediate representations of all previously processed tokens, enabling the model to attend to its full context without recomputing attention from scratch at each generation step.

For a model with N layers, H attention heads per layer, head dimension d, and cached sequence length L, the KV cache size is:

$$\text{KV size} = 2 \times N \times H \times d \times L \times \text{sizeof(dtype)} \tag{1}$$

For a 70-billion parameter model (e.g., Llama-3-70B: $N = 80$, $H = 64$, $d = 128$) at 128K context with FP16 precision:

$$\text{KV size} = 2 \times 80 \times 64 \times 128 \times 131{,}072 \times 2 \text{ bytes} \approx 16 \text{ GB per session} \tag{2}$$

This is not a cache in the traditional sense—a performance optimization that can be invalidated without correctness impact. The KV cache *is* the model's memory of the conversation. It determines what the model attends to, how it weights different parts of the context, and ultimately what it generates. Corrupting the KV cache corrupts the model's output. Leaking the KV cache leaks the conversation.

In this sense, the KV cache is a *probabilistic database*: a structured store of information that is queried (through the attention mechanism), updated (through each new token's key-value computation), and persisted (across the generation of a response). Unlike a traditional database, the queries are continuous (softmax-weighted sums over all entries), the updates are append-only (new tokens are added but old entries are not modified), and the "transactions" are probabilistic (the model's output is sampled from a distribution conditioned on the entire cache).

1.2 The Economics of Shared State

The economics of AI inference are brutal. A single H100 GPU costs approximately $30,000 and consumes 700W of power. Serving a 70B model requires at least 4 H100s for the model weights alone, plus additional capacity for KV cache storage. At these costs, leaving GPU capacity idle is economically untenable.

This creates overwhelming pressure to share GPU resources between principals—the users, tenants, or organizations whose data is being processed. The sharing mechanisms that have become standard practice include:

Continuous batching (Orca, 2022): Rather than processing one request at a time, the GPU processes a batch of requests simultaneously, with new requests joining and completed requests leaving the batch dynamically. This increases GPU utilization from ∼30% (static batching) to ∼90%+. The consequence: multiple principals' computations execute concurrently on the same GPU, sharing the same CUDA streams, the same SRAM, and—critically—the same scheduling decisions.

PagedAttention (vLLM, 2023): The KV cache is managed as a set of fixed-size blocks (analogous to virtual memory pages), allocated on demand and freed when no longer needed. Blocks are content-addressed: if two principals have identical prefixes (e.g., the same system prompt), their KV cache blocks can be *aliased*—both principals' attention computations read from the same

physical memory. This reduces memory consumption by 30–60% for workloads with shared prefixes. The consequence: the physical memory backing one principal's KV cache may be shared with another principal, creating a direct channel for state contamination.

Prefix caching (SGLang, RadixAttention): Extends PagedAttention's sharing across time. If a new request has a prefix that matches a previously computed KV cache (even from a different principal), the cached prefix is reused. This avoids redundant computation. The consequence: a principal's inference depends on state computed for a prior principal. Cache hits and misses create a binary timing signal (TTFT difference of 10–200ms) that can be exploited for prompt reconstruction.

FlashAttention: The standard attention implementation tiles the computation across GPU SRAM (shared memory) for efficiency. When multiple requests are batched, their attention computations share the same SRAM banks. The consequence: bank conflicts create timing-dependent interference between co-batched principals, and the SRAM contents from one principal's attention computation may be readable by the next computation on the same SRAM (the LeftoverLocals vulnerability class).

Each of these optimizations is individually sound engineering. Collectively, they create a regime in which the computational states of multiple principals are physically entangled at every level of the memory hierarchy—from GPU registers to SRAM to HBM to the scheduling decisions that determine which computations execute when.

1.3 Contamination: A Taxonomy

We define *state contamination* as any information flow between principals that is not authorized by the system's security policy. In the context of AI inference, contamination manifests through five channels:

Hidden-prompt contamination. A principal's system prompt, which is intended to be confidential, leaks to another principal through shared KV-cache blocks, timing side-channels, or cache-hit oracles. The PROMPTPEEK attack achieves 99% reconstruction accuracy through TTFT measurement alone.

Ghost tokens. Residual KV-cache entries from a prior principal's session persist in memory and are attended to by a subsequent principal's computation. The prior principal's tokens act as "ghost tokens"—invisible to the user but influential on the model's output. The attention mechanism cannot distinguish between the current principal's tokens and ghost tokens left in uninitialized memory.

Silent carry-over. In continuous batching, the completion of one principal's request and the initiation of another occur within the same batch. The model's internal state (hidden representations, attention patterns) retains traces of the completed request that influence the initiated request. This is especially acute at attention sink positions (position 0), where all subsequent tokens attend with disproportionate weight.

Timing side-channels. The latency of a principal's inference depends on the state and behavior of co-resident principals: batch size affects scheduling latency, cache hits/misses affect prefill time, and memory pressure affects KV-cache eviction patterns. Each of these creates an observable signal that leaks information about co-resident principals.

Hardware memory leakage. GPU local memory (SRAM), shared memory, and registers are not zeroed between kernel invocations on certain GPU architectures. A kernel launched after an LLM inference kernel can read the complete contents of the prior kernel's local memory, recovering KV-cache fragments, attention weights, and intermediate activations.

1.4 Why Classical Models Fail

The security community has developed powerful formal frameworks for reasoning about isolation, each successful within its domain. None addresses the AI inference setting:

Bell-LaPadula (1973) assumes discrete, deterministic state transitions with binary access decisions (read/write). The transformer attention mechanism is a continuous, probabilistic read over the entire cached state. There is no binary "read" or "write"—there is a softmax-weighted sum that touches every cached position with nonzero weight. BLP's lattice structure is valuable, but its access control model is fundamentally incompatible with continuous computation.

Biba (1977) addresses integrity rather than confidentiality, with the same discrete-state assumptions. Applied to LLM inference, shared prefix corruption (a low-integrity input contaminating a high-integrity cache) and agentic confused deputy attacks (low-integrity tool outputs directing high-integrity tool invocations) are integrity violations that Biba's framework can classify but not bound.

Goguen-Meseguer noninterference (1982) provides the strongest classical guarantee: the outputs observed by one principal are identical whether or not another principal exists on the system. This is the ideal that Tier 3 of the Stateful Isolation Law instantiates. But perfect noninterference is unachievable in shared-resource systems with continuous computation, approximate arithmetic, and timing-dependent scheduling. What is needed is not perfect noninterference but *bounded* noninterference—a quantitative relaxation that permits controlled, measurable leakage below a defined threshold.

Database isolation (Berenson et al., 1995; Adya et al., 2000) defines graduated isolation levels (Read Uncommitted through Serializable) based on forbidden anomalies in transaction interleavings. This graduated approach is the inspiration for the Stateful Isolation Law's compliance tiers. But database isolation assumes symbolic state, discrete transactions, and deterministic conflict detection—none of which apply to the continuous, probabilistic, attention-weighted computation of transformer inference.

seL4 (Klein et al., 2009) provides formally verified noninterference at the OS kernel level, with 200,000 lines of Isabelle/HOL proof. But seL4's noninterference proof explicitly excludes timing channels and assumes hardware correctness. Spectre and Meltdown demonstrated that the hardware correctness assumption is violated by every modern processor. For GPU inference, the situation is worse: no GPU architecture has a formal specification, a formal threat model, or a formal proof of any isolation property.

1.5 Scope and Structure of This Document

The Stateful Isolation Law addresses the gap between the isolation requirements of multi-principal AI systems and the isolation guarantees provided by existing frameworks. It comprises:

1. A **General Principle** (Section 4): A contamination functional based on Rényi divergence that provides a unified, parameterized measure of isolation, recovering differential privacy, mutual information bounds, and classical noninterference as special cases.

2. **Six Clauses** (Section 5): Enforceable bounds on every contamination surface in the AI inference stack:

   - **AI-1:** KV Drift Bound (numerical fidelity)
   - **AI-2:** Non-Interference of Attention (spatial isolation)
   - **AI-3:** Temporal Isolation (timing channels)

- **AI-4:** Context Integrity (agentic noninterference)
- **AI-5:** Memory Hygiene (hardware isolation)
- **AI-6:** Gradient Isolation (federated bound)

3. A **Composability Theorem** (Section 6): Formal conditions under which clause-level compliance propagates to system-level guarantees, with lemmas for sequential (agentic chains), parallel (batch inference), and cross-layer (hardware to application) composition.

4. **Compliance Tiers** (Section 7): Four graduated levels from Development (Tier 0, non-compliant) through Sovereign (Tier 3, classical noninterference), with regulatory mapping to GDPR, EU AI Act, HIPAA, FedRAMP, PCI DSS, and 15+ additional frameworks.

5. An **Enforcement Protocol** (Section 8): A four-phase (Detect $\rightarrow$ Flush $\rightarrow$ Hash $\rightarrow$ Attest) framework with cryptographic audit trails, incident classification, and SOC-2/ISO alignment.

This public version presents the complete framework architecture, threat landscape, clause definitions, theorem statements, compliance tiers, and regulatory mapping. Threshold derivations, explicit constants, scaling laws, proof bodies, and implementation specifications are held under the Auburn Patent Family and available under commercial license.

2 Threat Landscape: Demonstrated Attacks on Multi-Tenant AI Systems

This section catalogs demonstrated attacks against production and research AI inference systems, organized by contamination surface. Each attack is mapped to the clause(s) it violates and the compliance tier at which it is mitigated. All attacks cited here have been published in peer-reviewed venues, disclosed through CVE processes, or documented by the affected vendors.

2.1 KV-Cache Side-Channel Attacks

PROMPTPEEK (Wu et al., NDSS 2025). Achieves 99% prompt reconstruction accuracy by measuring Time-to-First-Token (TTFT) against the SGLang serving framework with prefix caching enabled. The attacker probes the inference endpoint with candidate prompt prefixes and observes whether the TTFT indicates a cache hit (short latency, prefix already cached from the victim's session) or miss (long latency, prefix must be computed). By extending the matching prefix one token at a time, the attacker reconstructs the victim's full prompt. The attack requires no model access, no gradient access, and no special privileges—only API access to the same inference endpoint.

Early Bird (arXiv 2409.20002, 2024). Validates PROMPTPEEK-class timing attacks against production services: Claude (Anthropic), DeepSeek, and Azure OpenAI. Achieves 89% prefix recovery accuracy and 95.4% semantic recovery accuracy through latency-based fingerprinting. Demonstrates that production rate limiting and network jitter are insufficient defenses: the timing signal (10–200ms differential) is large relative to network noise (∼1–5ms variance).

InputSnatch (arXiv 2411.18191, 2024). Extends timing-based extraction to privacy-sensitive domains: medical queries processed by healthcare AI systems. Achieves 62% recovery of medical data from KV-cache timing patterns.

KV-Cloak (arXiv 2508.09442, 2025). Demonstrates three attack classes against KV caches: (1) *inversion*—reconstructing plaintext from cached key-value tensors; (2) *collision*—crafting inputs that alias to the same cache blocks as the victim, enabling cache-based side channels; (3) *injection*—manipulating cached values to alter the model's output for subsequent users.

2.2 Hardware Memory Leakage

LeftoverLocals (Trail of Bits, CVE-2023-4969, 2024). On AMD, Apple, and Qualcomm GPUs, local memory (the fast on-chip SRAM used by GPU kernels) is not zeroed between kernel invocations. A reader kernel launched after an LLM inference kernel can read approximately 181 MB of data per query, including KV-cache fragments, attention weights, and intermediate activations. The vulnerability is documented behavior: the CUDA/OpenCL/Metal programming models specify that local memory contents are "undefined" at kernel launch, meaning they may contain data from prior kernel invocations. NVIDIA data center GPUs (A100, H100) were not affected in testing, but this is an implementation choice, not a formal guarantee.

CUDA Leaks (Di Pietro et al., 2013). Demonstrates information leakage through GPU shared and global memory, including recovery of AES encryption keys from prior kernel invocations.

GPU Cache Contention (Naghibijouybari et al., CCS 2018). Demonstrates covert channels through GPU texture cache contention, enabling cross-application information flow on shared GPUs.

MOLE Attack (CCS 2025). Demonstrates bypass of NVIDIA's GPU Trusted Execution Environment (TEE) through exploitation of the GPU-embedded microcontroller (MCU), enabling extraction of encryption keys and falsification of attestation responses.

EM Emanation (Maia et al., USENIX Security 2022). Demonstrates reconstruction of model weights and intermediate activations from electromagnetic emanations of GPU computation.

2.3 Cloud Multi-Tenant Incidents

Google Vertex AI Response Misrouting (September 2025). Direct cross-tenant response delivery: responses generated for one organization's request were delivered to a different organization. Attributed to a proxy-layer error, but the underlying architecture permits cross-tenant contamination at the inference level.

Microsoft 365 Copilot EchoLeak (CVE-2025-32711). Zero-click prompt injection through cached email content: an attacker sends an email containing hidden instructions; when the victim's Copilot processes their inbox, the cached instructions activate and exfiltrate data.

OpenAI GPT-4 Turbo System Message Exposure (November 2023). System messages from one user's session were observed in another user's responses, attributed to a KV-caching error during a high-load period.

2.4 Agentic System Vulnerabilities

Inter-Agent Trust Exploitation (arXiv 2507.06850, 2025). 82.4% of LLMs in multi-agent configurations execute malicious payloads when requested by a peer agent, even when they successfully resist identical attacks delivered as direct user input. The trust asymmetry—models treat peer agents as more trusted than external sources—creates a relay vulnerability: compromise one agent, use it to compromise all connected agents.

Magentic-One Hijack (OpenReview, 2025). Microsoft's Magentic-One multi-agent framework executes arbitrary malicious code 97% of the time when an agent encounters a malicious local file during task execution.

MCP Amplification (arXiv 2601.17549, 2025). The Model Context Protocol (MCP) amplifies prompt injection attack success by 23–41% because tool responses from one MCP server influence invocations on another server through the shared context window. Each server's response enters the agent's context, creating cross-server propagation paths for injected instructions.

Amazon Bedrock Agent Memory Poisoning (Palo Alto Networks Unit 42, 2025). Indirect prompt injection causes an Amazon Bedrock Agent to store malicious instructions in its persistent memory. In subsequent sessions, the agent retrieves and executes the stored instructions, enabling persistent cross-session data exfiltration.

LangChain Serialization RCE (CVE-2025-68664). LangChain's serialization framework uses type keys in JSON objects to instantiate Python classes. An attacker who can influence the model's output (through any injection vector) can cause the serialized output to contain a malicious type key, achieving Remote Code Execution when the output is deserialized.

2.5 Federated Gradient Attacks

DAGER (Petrov et al., NeurIPS 2024, ETH Zurich). Achieves *exact* text reconstruction from gradients of decoder-only transformer models, with ROUGE-1 and ROUGE-2 scores exceeding 0.99. The attack exploits the low-rank structure of transformer embedding gradients: nonzero rows of the embedding gradient correspond exactly to tokens present in the training batch, and the rank structure enables separation of individual sequences from batch gradients. Effectiveness *increases* with model size (larger embedding dimensions provide more degrees of freedom for sequence separation). Scales to batch sizes of 128 sequences.

Secure Aggregation Evasion (Pasquini et al., CCS 2022). Demonstrates that a malicious server can completely elude secure aggregation by sending different model parameters to different participants. The server crafts per-participant parameter perturbations such that individual gradients are extractable from the aggregate through projection, rendering secure aggregation ineffective as a privacy mechanism.

2.6 Numerical Drift and Quantization

FlashAttention Deviation (Golden et al., 2024). FlashAttention's tiled computation introduces up to 10× numerical deviation compared to standard attention at BF16 precision, due to the non-associativity of floating-point addition and the tiling order's effect on accumulator precision.

KVTuner Token Flipping (arXiv 2502.04420, 2025). Demonstrates that aggressive KV-cache quantization (2-bit, KIVI-2) causes cascading token prediction errors (divergence from FP16 baseline), while 4-bit quantization (KIVI-4) matches FP16 output. The gap between 2-bit and 4-bit is not gradual—it is a phase transition from acceptable to catastrophic drift.

2.7 The Isolation Gap

The following table summarizes the gap between demonstrated attacks and existing formal defenses:

Layer	Attack Exists?	Formal Bound?	Gap
KV-Cache state	Yes	No	No framework for KV drift tolerance
Timing channels	Yes	No	No capacity bounds for LLM timing
GPU memory	Yes	No	No formal GPU isolation spec
Cloud multi-tenant	Yes	No	No inference-level isolation standard
Agentic contexts	Yes	No	No formal IFC for LLM agents
Federated gradients	Yes	Partial (DP)	DP exists but not mandated; SecAgg insufficient
Attention computation	Yes	No	No formal contamination bound
Hardware TEE	Yes	No	No GPU TEE formal verification

The conclusion is stark: **the entire AI inference stack operates without provable isolation guarantees.** Contamination has been demonstrated at every layer, from silicon to application. Formal bounds exist only partially in the federated learning setting (differential privacy) and in classical computing (seL4 for CPU isolation, with the timing channel exclusion). The Stateful Isolation Law addresses this gap.

3 Classical Isolation Models and Their Limitations

The Stateful Isolation Law does not replace classical security models—it extends them into a domain they were not designed for. This section provides a precise analysis of each classical framework's contributions and limitations when applied to AI inference, establishing the formal foundation from which the General Principle is derived.

3.1 Bell-LaPadula: Confidentiality in Lattice Models

The Bell-LaPadula (BLP) model defines confidentiality through two properties on a lattice of security levels $(\mathcal{L}, \leq)$:

- **Simple Security (no read up):** A subject at level ℓ_s may read an object at level ℓ_o only if $\ell_s \geq \ell_o$.
- **Star Property (no write down):** A subject at level ℓ_s may write to an object at level ℓ_o only if $\ell_s \leq \ell_o$.

The Basic Security Theorem proves that any system satisfying both properties in every state transition preserves confidentiality as an invariant.

Applied to LLM inference: The attention mechanism reads every cached position with nonzero weight—it trivially satisfies simple security (it reads everything at or below its level) but also reads everything *above* its level if higher-level content is present in the shared cache. The star property is violated by design: the system prompt (high level) influences the user's output (lower level). Prompt injection is precisely a write-up violation: low-level injected content alters the system's behavior as if it were high-level instruction.

Contribution to the Stateful Isolation Law: BLP's lattice structure informs the Information Flow Control (IFC) labeling scheme in Clause AI-4, where context tokens are labeled from a trust lattice and the tool invocation gate enforces label-based access control.

3.2 Biba: Integrity as the Dual Problem

The Biba model inverts BLP for integrity:

- **Simple Integrity (no read down):** A subject may not read objects of lower integrity.
- **Star Integrity (no write up):** A subject may not write to objects of higher integrity.

Applied to LLM inference: Shared prefix corruption is a read-down violation: a principal's attention reads from cache blocks potentially contaminated by a lower-integrity prior occupant. Agentic confused deputy attacks are write-up violations: low-integrity tool outputs direct high-integrity tool invocations.

Contribution: Biba's integrity dimension complements BLP's confidentiality dimension. The Stateful Isolation Law treats both as aspects of the contamination functional: confidentiality violations (leakage from B to A) and integrity violations (corruption of A by B) are both measured by the Rényi divergence between A's output distribution with and without B's presence.

3.3 Goguen-Meseguer Noninterference: The Ideal

Goguen-Meseguer noninterference (1982) provides the strongest classical guarantee: the outputs observed by principal u when the system processes trace tr are identical to the outputs when all of principal v's actions are removed from the trace:

$$\text{run}(s_0, \text{tr}) =_u \text{run}(s_0, \text{purge}(\text{tr}, v)) \tag{3}$$

Extensions include intransitive noninterference (controlled declassification through intermediaries) and probabilistic noninterference ($P[O_u \mid I_u, I_v] = P[O_u \mid I_u]$).

Applied to LLM inference: Perfect noninterference is unachievable in shared-resource systems with approximate arithmetic and timing-dependent scheduling. The seL4 microkernel achieves noninterference for CPU computation but explicitly excludes timing channels from its proof—an exclusion that Spectre and Meltdown proved consequential. For GPU inference, the gap is far wider: no GPU architecture has a formal specification, threat model, or proof of any isolation property.

Contribution: Goguen-Meseguer noninterference is the *ideal*—the $\epsilon = 0$ limit of the contamination functional, instantiated as Tier 3 (Sovereign) of the compliance hierarchy. Practical tiers (1 and 2) are bounded relaxations of this ideal, with the relaxation parameterized by Rényi order α and contamination bound ϵ.

3.4 Database Transaction Isolation: The Graduated Model

Berenson et al. (SIGMOD 1995) identified ambiguities in ANSI SQL isolation levels. Adya, Liskov, and O'Neil (ICDE 2000) resolved these through Direct Serialization Graphs (DSGs) with three dependency types: write-read (wr), write-write (ww), and read-write (rw). Isolation levels are defined by which cycle types are forbidden in the DSG:

Level	Forbidden Cycles	AI Analog
Read Uncommitted	None	Naive prefix sharing
Read Committed	G1a (dirty read)	Salted cache (no aliasing)
Repeatable Read	G1a, G1b, G2-item	Deterministic generation
Serializable	All cycles	Dedicated instance

Contribution: The graduated isolation spectrum directly inspires the compliance tier structure. Database isolation demonstrates that a single framework can support multiple assurance levels, each with specific guarantees and costs. The Stateful Isolation Law extends this approach from symbolic, discrete transactions to continuous, probabilistic inference.

Limitation: No published formalism defines a serialization graph for AI inference—what it means for inference operations to be "isolation-equivalent to serial execution" is undefined. The contamination functional fills this role.

3.5 seL4: The Cross-Layer Verification Requirement

The seL4 microkernel provides formally verified noninterference through 200,000 lines of Isabelle/HOL proof. This is the gold standard for verified isolation.

Caveat: seL4's proof assumes hardware correctness—that the CPU faithfully executes instructions as specified. Spectre and Meltdown demonstrated that this assumption is violated by

speculative execution in every modern processor. The proof is valid at the specification level but does not hold at the implementation level when the hardware deviates from its specification.

For GPU inference, the gap is categorical: no GPU architecture has a formal specification (only engineering documentation), no formal threat model (only informal security notes), and no formal proof of any isolation property. NVIDIA's Multi-Instance GPU (MIG) provides engineering-level hardware partitioning but lacks formal verification. The entire GPU computing stack—from silicon design through driver implementation to the CUDA programming model—operates without formally verified isolation guarantees.

Contribution: seL4 establishes the cross-layer verification requirement formalized in the Composability Theorem's Lemma 3 (Section 6.4): software-level isolation is meaningless without hardware-level isolation, and verification must span the full stack from silicon to application.

3.6 Summary: What Is Needed

The classical models collectively provide:

- Lattice-based access control (BLP) → IFC labeling for agentic systems
- Integrity dimension (Biba) → Bidirectional contamination measurement
- The noninterference ideal (Goguen-Meseguer) → Tier 3 target
- Graduated isolation (database) → Compliance tier structure
- Cross-layer verification (seL4) → Composability requirements

What they do not provide is a *unified, quantitative framework* for measuring and bounding contamination in continuous, probabilistic, approximately-computed, timing-dependent, multi-layered AI inference systems. This is the General Principle.

4 The General Principle: The Contamination Functional

4.1 Design Requirements

A formal measure of contamination in AI inference must satisfy five requirements:

1. **Generality:** It must apply to any AI system architecture—any model, any serving framework, any hardware.
2. **Parameterization:** It must support graduated assurance levels, from statistical bounds to worst-case guarantees to perfect noninterference.
3. **Classical recovery:** It must reduce to established security metrics (differential privacy, mutual information, noninterference) as special cases.
4. **Composability:** It must compose—component-level bounds must combine into system-level bounds through well-defined composition rules.
5. **Measurability:** It must be estimable from observable quantities (latencies, norm differences, output distributions) without requiring access to internal model state.

4.2 Rényi Divergence as Foundation

The Rényi divergence of order $\alpha > 0$, $\alpha \neq 1$, between distributions P and Q is:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \sum_x P(x)^\alpha Q(x)^{1-\alpha} \tag{4}$$

Key limits and properties:

- $D_1(P\|Q) = \mathrm{KL}(P\|Q)$ (Kullback-Leibler divergence, the $\alpha \to 1$ limit)
- $D_\infty(P\|Q) = \log \max_x \frac{P(x)}{Q(x)}$ (max-divergence, the $\alpha \to \infty$ limit)
- Non-negativity: $D_\alpha \geq 0$ with equality iff $P = Q$
- Monotonicity: $\alpha_1 \leq \alpha_2 \implies D_{\alpha_1} \leq D_{\alpha_2}$
- Data-processing inequality: post-processing cannot increase divergence

The Rényi family is the unique divergence family that simultaneously satisfies all five design requirements.

4.3 The Contamination Functional

Definition 4.1 (Contamination Functional — General Principle)**.** Let A and B be principals sharing an AI inference system with state S_t at time t. The contamination of A by B at Rényi order α is:

$$\boxed{\mathcal{C}_\alpha(A, B, S_t) = D_\alpha(P(O_A \mid S_t, I_A, I_B) \parallel P(O_A \mid S_t, I_A)) \leq \epsilon(\alpha, \mathrm{Tier})} \tag{5}$$

where O_A is A's output, I_A and I_B are A's and B's inputs, and $\epsilon(\alpha, \mathrm{Tier})$ is the tier-dependent contamination bound.

The functional measures how much A's output distribution changes when B is present on the system. Zero divergence means B's presence has no effect on A's output—perfect noninterference. Nonzero divergence quantifies the contamination.

4.4 Recovery of Classical Frameworks

The Rényi parameterization recovers all classical isolation metrics:

- $\alpha = 1$ **(Mutual Information):** $\mathcal{C}_1 = \mathrm{KL}(P(O_A \mid I_A, I_B) \| P(O_A \mid I_A))$. Taking the expectation over I_B yields $I(O_A; I_B)$—the mutual information between A's output and B's input. This is the Tier 1 metric: statistical leakage in expectation.
- $\alpha = \infty$ **(Differential Privacy):** $\mathcal{C}_\infty = \log \max_{O_A} \frac{P(O_A|I_A,I_B)}{P(O_A|I_A)} = \epsilon_{\mathrm{DP}}$. This is the ϵ-differential privacy guarantee: no single output is more than e^ϵ times more likely when B is present. This is the Tier 2 metric: worst-case leakage.
- $\alpha \to \infty$, $\epsilon = 0$ **(Noninterference):** $P(O_A \mid I_A, I_B) = P(O_A \mid I_A)$ for all O_A, I_A, I_B. The distributions are identical—Goguen-Meseguer noninterference. This is the Tier 3 requirement.

The monotonicity of Rényi divergence ($\alpha_1 \leq \alpha_2 \implies \mathcal{C}_{\alpha_1} \leq \mathcal{C}_{\alpha_2}$) creates a strict hierarchy:

$$\text{Tier 3 compliance} \implies \text{Tier 2 compliance} \implies \text{Tier 1 compliance} \tag{6}$$

4.5 Composition Property

Rényi divergence composes linearly: for n independent contamination events with bounds $\epsilon_1, \ldots, \epsilon_n$:

$$\mathcal{C}_\alpha^{\text{total}} \leq \sum_{i=1}^{n} \epsilon_i \tag{7}$$

This converts to (ϵ, δ)-differential privacy via:

$$\epsilon_{\text{total}} \leq \sum_i \epsilon_i + \frac{\log(1/\delta)}{\alpha - 1} \tag{8}$$

This composition property is the foundation of the Composability Theorem (Section 6).

4.6 Temporal Invariant

The contamination bound must hold at every time step, not just in expectation:

$$\sup_t \mathcal{C}_\alpha(A, B, S_t) \leq \epsilon \tag{9}$$

This is critical because numerical drift accumulates over the context length: the worst-case contamination occurs at the maximum context length, not at the average.

The estimation methods for the contamination functional—including direct paired inference, proxy estimation via norm monitoring with Lipschitz bounds, and timing-based mutual information estimation—are specified in the private version. These methods enable runtime monitoring of clause compliance without access to internal model state. Available under commercial license from the Auburn Patent Family.

5 The Six Clauses

Each clause of the Stateful Isolation Law instantiates the General Principle for a specific contamination surface in the AI inference stack. This section presents the formal statement, threat model, and rationale for each clause. Threshold derivations, explicit constants, and implementation specifications are held under the Auburn Patent Family.

5.1 Clause AI-1: KV Drift Bound (Numerical Fidelity)

5.1.1 Threat Model

The KV cache accumulates numerical drift from three sources: (1) *passive drift* from floating-point approximation in FlashAttention (up to 10× deviation at BF16), quantization artifacts, and accumulator truncation; (2) *stale state contamination* from PagedAttention block aliasing and LeftoverLocals-class memory leaks; and (3) *adversarial manipulation* through malicious token injection and KV-cache inversion attacks.

5.1.2 Formulation

Clause AI-1 establishes separate bounds on Key and Value drift, reflecting their distinct roles in the attention mechanism:

Definition 5.1 (Clause AI-1: KV Drift Bound)**.** For each layer ℓ of the model, the drift between cached and freshly computed KV tensors must satisfy:

$$\boxed{\left\|\mathbf{K}^{(\ell)}_{\text{cached}} - \mathbf{K}^{(\ell)}_{\text{fresh}}\right\|_2 \leq \lambda^{(\ell)}_K} \tag{10}$$

$$\boxed{\left\|\mathbf{V}^{(\ell)}_{\text{cached}} - \mathbf{V}^{(\ell)}_{\text{fresh}}\right\|_F \leq \lambda^{(\ell)}_V} \tag{11}$$

where $\|\cdot\|_2$ denotes the spectral norm (operator 2-norm) and $\|\cdot\|_F$ denotes the Frobenius norm.

5.1.3 Rationale for Split Norms

Keys and Values serve fundamentally different roles in the attention mechanism. Keys appear inside the softmax (attention routing): a perturbation $\Delta\mathbf{k}$ aligned with the query vector $\mathbf{q}$ produces a logit change $\Delta z = \mathbf{q}^\top \Delta\mathbf{k}/\sqrt{d}$, which is amplified nonlinearly by the softmax. The spectral norm $\|\Delta\mathbf{K}\|_2$ captures the worst-case directional sensitivity—the maximum perturbation in any direction.

Values appear in a weighted sum (content delivery): $\Delta\mathbf{o} = \sum_i \alpha_i \Delta\mathbf{v}_i$, which propagates linearly. The Frobenius norm $\|\Delta\mathbf{V}\|_F$ captures the aggregate perturbation magnitude across all cached positions.

Empirical evidence supports the split: Key matrices exhibit higher spectral norms than Value matrices (outlier channels 10–100× median), and quantization-induced errors in Keys produce qualitatively different failures (token prediction flips) than errors in Values (output quality degradation).

The derivation of $\lambda_K^{(\ell)}$ from decision margin analysis, $\lambda_V^{(\ell)}$ from output fidelity requirements, layer-wise drift budget allocation accounting for multiplicative amplification through the transformer stack, quantization sub-bounds yielding minimum bit-width requirements, and attention sink position thresholds with amplification factors are specified in the private version. These constants are model-specific and include explicit safety factors for each compliance tier. Available under commercial license.

5.2 Clause AI-2: Non-Interference of Attention (Spatial Isolation)

5.2.1 Threat Model

Spatial contamination occurs through: (1) PagedAttention block aliasing (content-addressed hashing shares physical blocks between principals with identical prefixes, with `cache_salt` optional and disabled by default); (2) FlashAttention shared-SRAM side channels (bank conflicts and tiling order create timing-dependent interference); (3) continuous batching cross-talk (batch position determines memory layout and thread assignment); and (4) prefix cache contamination (shared prefix recomputation creates discontinuity artifacts).

5.2.2 Formulation

Definition 5.2 (Clause AI-2: Non-Interference of Attention)**.** The mutual information between principal A's attention weights and principal B's KV state must be bounded:

$$\boxed{I(W_{\text{attn}}^{A}\,;\,\mathbf{KV}^{B}) \le \mu(\text{Tier})} \tag{12}$$

where $I(\cdot\,;\,\cdot)$ denotes mutual information and μ is the tier-dependent threshold.

This connects to the contamination functional through the data-processing inequality: $I(O_A;\mathbf{KV}^B) \le I(W_{\text{attn}}^A;\mathbf{KV}^B) \le \mu$. Bounding mutual information at the attention weight level provides a tighter control point than bounding it at the output level.

The derivation of μ thresholds for each tier, PagedAttention aliasing information capacity analysis, FlashAttention SRAM bank conflict channel bounds, existence channel entropy calculations, canary-based estimation methods, and architectural requirements for achieving $\mu = 0$ (Tier 2+) are specified in the private version. Available under commercial license.

5.3 Clause AI-3: Temporal Isolation (Timing Channel Bound)

5.3.1 Threat Model

Timing is the dominant covert channel in production AI systems. PROMPTPEEK achieves 99% prompt reconstruction from TTFT alone. Early Bird validates timing attacks against production Claude, DeepSeek, and Azure OpenAI services. The attacks exploit inherent performance optimizations: prefix caching creates a cache hit/miss binary signal (10–200ms differential), continuous batching creates load-dependent latency, and KV-cache eviction creates observable latency spikes.

5.3.2 Formulation

Definition 5.3 (Clause AI-3: Temporal Isolation)**.** The expected latency perturbation caused by the presence of principal B must be bounded per-token:

$$\boxed{\Delta T(I_B) = \left|\mathbb{E}[T_A^{(k)} \mid I_B] - \mathbb{E}[T_A^{(k)}]\right| \leq \tau_{\text{safe}} \quad \forall k} \tag{13}$$

where $T_A^{(k)}$ is the generation time for A's k-th token.

The information capacity of the timing channel is bounded by:

$$C = \frac{1}{2}\log_2\left(1 + \frac{\text{Var}[\Delta T]}{\text{Var}[\text{noise}]}\right) \text{ bits per observation} \tag{14}$$

The PROMPTPEEK mechanism creates a high-SNR binary oracle: cache hit ∼10ms, cache miss ∼200ms, yielding ∼16.8 effective bits per token—nearly the full vocabulary entropy (17 bits for 128K vocabulary). Compliance requires reducing this channel capacity to levels where reconstruction becomes infeasible within any practical number of observations.

The derivation of τ_{safe} from channel capacity targets, constant-time serving architecture specifications (TTFT normalization, fixed-rate token delivery, batch padding), jitter calibration formulas, scheduling constraint analysis (request-independent scheduling, preemption isolation), formal defeat analysis of PROMPTPEEK and Early Bird under compliant systems, and performance cost models are specified in the private version. Available under commercial license.

5.4 Clause AI-4: Context Integrity (Agentic Noninterference)

5.4.1 Threat Model

In agentic AI systems, the model possesses tool-use capabilities granted by the user's authorization. The context window conflates the instruction plane and the data plane: user instructions, system constraints, tool outputs, external data, and peer agent messages are concatenated into the same token sequence with no architectural mechanism to distinguish between them. This creates the *confused deputy* vulnerability: untrusted data in the context can alter the agent's control flow (which tools it invokes and in what order).

The empirical consequences: 82.4% inter-agent compromise rates, 97% malicious code execution in Magentic-One, 23–41% MCP amplification, persistent memory poisoning in Amazon Bedrock, and Remote Code Execution through LangChain serialization.

5.4.2 Formulation

Definition 5.4 (Clause AI-4: Context Integrity)**.** Let $\mathcal{P} : \mathcal{C} \to \mathcal{A}^*$ denote the agent's planning function and $\text{purge}(\mathcal{C},\mathcal{U})$ denote the context with all untrusted content replaced by neutral placeholders. The system satisfies Clause AI-4 if:

$$\boxed{\text{tools}(\mathcal{P}(\mathcal{C})) = \text{tools}(\mathcal{P}(\text{purge}(\mathcal{C},\mathcal{U})))} \tag{15}$$

The tool selection sequence must be identical whether or not untrusted content is present. Untrusted content may influence tool *arguments* (the data plane) but not tool *selection* (the control plane).

This is a direct instantiation of Goguen-Meseguer noninterference applied to the agent's planning function, relaxed to require noninterference of control flow only (not data flow). The relaxation preserves the utility of the agent (processing untrusted data is its purpose) while eliminating the confused deputy vulnerability (untrusted data cannot redirect the agent's authority).

5.4.3 The Confused Deputy Taxonomy

Context integrity violations fall into four categories:

Type 1: Direct Prompt Injection. Adversarial instructions placed directly in the user input. Addressed by safety training and system prompt enforcement, not by Clause AI-4 (the user input is trusted content in the control plane).

Type 2: Indirect Prompt Injection. Adversarial instructions embedded in content the agent retrieves or processes (web pages, documents, emails, API responses). The core target of Clause AI-4: if the tool selection is independent of untrusted content, embedded instructions in the data plane cannot redirect the agent's behavior.

Type 3: Inter-Agent Trust Exploitation. A compromised peer agent relays adversarial instructions to other agents, exploiting the trust asymmetry (agents treat peers as more trusted than external sources). Defense: peer messages must be classified as untrusted unless the peer's own context integrity is verified—a recursive application of Clause AI-4 formalized in the Composability Theorem.

Type 4: Memory Poisoning. Adversarial instructions stored in the agent's long-term memory during one session activate in future sessions. Defense: memory writes must satisfy the same noninterference requirement as tool invocations, and memory reads must be treated as data-plane content.

The Information Flow Control (IFC) architecture—including the trust lattice labeling scheme, dynamic taint propagation rules with entropy-adaptive thresholds, tool invocation gate specification, quarantined execution protocol, MCP cross-server propagation formal analysis with amplification bounds, serialization vulnerability formalization (non-executability of data), and the connection between Clause AI-4 and the contamination functional at $\alpha = \infty$—are specified in the private version. Available under commercial license.

5.5 Clause AI-5: Memory Hygiene (Hardware Isolation)

5.5.1 Threat Model

Every isolation guarantee in Clauses AI-1 through AI-4 rests on the assumption that hardware faithfully executes memory management instructions. LeftoverLocals (CVE-2023-4969) proved this assumption false: on AMD, Apple, and Qualcomm GPUs, local memory is not zeroed between kernel invocations, leaking approximately 181 MB of prior computation data per query. The MOLE attack demonstrated GPU TEE bypass through microcontroller exploitation. No amount of software-level isolation provides security if the hardware leaks the complete state of prior computations.

5.5.2 Formulation

Definition 5.5 (Clause AI-5: Memory Hygiene)**.** For any addressable memory region m (including GPU local memory, shared memory, registers, cache lines, and HBM pages) used by principal A and subsequently made available to principal B:

$$\boxed{P(\text{read}(m, t_2) \neq \mathbf{0} \mid \text{free}(m, t_1),\ t_2 > t_1) = 0} \tag{16}$$

The probability of reading nonzero data from a freed memory region is exactly zero. This is a deterministic requirement with no ϵ tolerance.

This is the foundation clause. Unlike Clauses AI-1 through AI-4, which permit bounded contamination, Clause AI-5 requires *zero* residual data. The zeroing must be cryptographic (indistinguishable from fresh memory) and must occur before any subsequent principal's computation can access the region.

5.5.3 The Hardware Isolation Gap

NVIDIA's Multi-Instance GPU (MIG) provides the strongest available hardware-level partitioning (dedicated SMs, L2 cache banks, memory controllers per instance) but has no formal specification, no formal threat model, no formal proof, and no independent security audit. Shared global resources (scheduler, PCIe interface, power delivery, thermal management) create potential side-channel surfaces even with MIG active. For Tier 2 compliance, MIG is necessary but not sufficient. For Tier 3, dedicated physical GPUs are required.

The Trusted Execution Environment (TEE) chain of trust specification, attestation chain cryptographic protocol, CPU-GPU handoff vulnerability analysis, MOLE attack defense architecture, cross-GPU channel analysis (L2 contention, NVLink congestion, EM emanation), performance cost models for deterministic zeroing with amortization strategies, and the canary verification kernel specification are provided in the private version. Available under commercial license.

5.6 Clause AI-6: Gradient Isolation (Federated Bound)

5.6.1 Threat Model

The DAGER attack (NeurIPS 2024) achieves *exact* text reconstruction from transformer gradients with ROUGE > 0.99, exploiting the low-rank structure of embedding gradients. The attack's effectiveness *increases* with model size. Secure aggregation is defeated by malicious server attacks (Pasquini et al., CCS 2022). The fundamental premise of federated learning—that gradients are less sensitive than data—has been conclusively invalidated.

5.6.2 Formulation

Definition 5.6 (Clause AI-6: Gradient Isolation)**.** The mutual information between a participant's data D_i and any shared model update ∇W_{shared} must be bounded:

$$\boxed{I(D_i\ ;\ \nabla W_{\text{shared}}) \leq \epsilon_{\text{fed}}(\text{Tier})} \tag{17}$$

This connects to the contamination functional at $\alpha = 1$ (mutual information, Tier 1) and at $\alpha = \infty$ (differential privacy, Tier 2). Differential privacy through DP-SGD (gradient clipping + calibrated Gaussian noise) is the minimum acceptable mechanism. Secure aggregation is a useful additional defense but does not reduce the required DP budget, because a malicious server can elude it.

The connection between ϵ_{fed} and the DP parameter ϵ_{DP}, Rényi DP accounting formulas for multi-round training, group privacy scaling for multi-record participants, noise calibration tables for each compliance tier, utility preservation analysis, practical approaches for Tier 2 compliance (federated distillation, homomorphic encryption), and the privacy accountant specification are provided in the private version. Available under commercial license.

5.7 Clause Summary

Clause	Surface	Metric	Bound Type
AI-1	KV-cache drift	Split-norm ($\|\cdot\|_2$, $\|\cdot\|_F$)	Per-layer, per-norm threshold
AI-2	Attention cross-talk	Mutual information	Per-tier threshold
AI-3	Timing channels	Latency perturbation	Channel capacity target
AI-4	Agentic control flow	Planning noninterference	Deterministic (control plane)
AI-5	Hardware memory	Residual data probability	Zero (deterministic)
AI-6	Federated gradients	Mutual information / DP	Per-tier threshold

6 The Composability Theorem

6.1 Why Clause-Level Compliance is Insufficient

A system can satisfy every individual clause and still fail to provide system-level isolation. Consider: Clause AI-1 is satisfied (KV drift within bounds), Clause AI-2 is satisfied (attention spatially isolated), Clause AI-3 is satisfied (timing channels bounded)—yet the *composition* of these three channels produces a correlated signal observable through the API that exceeds the system-level bound.

The MCP amplification finding (23–41% increase in attack success) is direct empirical evidence of composability failure: individual tool invocations satisfy their bounds, but the chain of invocations creates a propagation path that amplifies contamination through interaction terms.

The Composability Theorem establishes the conditions under which clause-level compliance *does* compose into system-level guarantees.

6.2 Main Theorem: The Small Gain Condition

Theorem 6.1 (Compositional Isolation)**.** Let a multi-principal AI system be decomposed into n components $\{G_1, G_2, \ldots, G_n\}$. Let γ_i denote the **contamination gain** of component G_i—the ratio of output contamination to input contamination:

$$\gamma_i = \sup_{\mathcal{C}_\alpha^{\text{in}} > 0} \frac{\mathcal{C}_\alpha^{\text{out}}(G_i)}{\mathcal{C}_\alpha^{\text{in}}(G_i)} \tag{18}$$

If each component satisfies its relevant clauses and:

$$\boxed{\prod_{i=1}^{n} \gamma_i < 1} \tag{19}$$

then the composed system satisfies the Stateful Isolation Law with system-level bound:

$$\mathcal{C}_\alpha^{\text{system}} \leq \frac{\max_i \epsilon_i}{1 - \prod_{i=1}^{n} \gamma_i} \tag{20}$$

where ϵ_i is the per-component contamination bound from the relevant clause.

The product condition has an intuitive interpretation: contamination that propagates through the full system loop must be *attenuated*, not amplified. A component that amplifies contamination ($\gamma_i > 1$) can be tolerated if other components attenuate sufficiently to compensate. When the condition fails ($\prod \gamma_i \geq 1$), the system exhibits *contamination resonance*: small initial contamination is amplified through feedback to produce unbounded system-level contamination.

The proof adapts the Small Gain Theorem from H^∞ robust control theory (Zhou, Doyle, and Glover, 1996) to the contamination functional, modeling the system as a feedback interconnection where contamination propagates through components and returns (amplified or attenuated) to the origin.

The complete proof of Theorem 6.1, including the feedback interconnection model, the operator norm formulation in the Rényi metric, the geometric series convergence argument, and the steady-state bound derivation, is provided in the private version. Available under commercial license.

6.3 Lemma 1: Sequential Composition (Agentic Chains)

Lemma 6.2 (Sequential Composition)**.** Let an agentic system perform k actions $(a_1, \ldots, a_k)$ with per-action context integrity bounds δ_i and contamination gains γ_i^{seq}. The end-to-end contamination is bounded by:

$$\boxed{\mathcal{C}_\alpha^{\text{chain}} \leq \sum_{i=1}^{k} \delta_i \prod_{j=i+1}^{k} \gamma_j^{\text{seq}} + \sum_{1 \leq i < j \leq k} \Xi_{i,j}} \tag{21}$$

where $\Xi_{i,j}$ is the **interaction term** between actions a_i and a_j: the additional contamination at a_j caused by the presence of a_i's output in the shared context.

The interaction terms $\Xi_{i,j}$ are the formal explanation of MCP amplification. They scale combinatorially: for k actions, there are $\binom{k}{2}$ interaction terms, producing $O(k^2)$ growth in total contamination. If Clause AI-4 is perfectly enforced ($\mathcal{C}_\alpha^{\text{AI-4}} = 0$ for each action), all interaction terms vanish—this is the formal justification for prioritizing context integrity enforcement in agentic systems.

The proof of Lemma 1, including the recurrence relation derivation, the interaction term bound in terms of per-action Clause AI-4 violations and cross-action sensitivity, and the formal analysis of MCP amplification as a composability failure, is provided in the private version. Available under commercial license.

6.4 Lemma 2: Parallel Composition (Batch Inference)

Lemma 6.3 (Parallel Composition)**.** Let b principals be processed concurrently in a shared batch, with pairwise contamination $\mathcal{C}_\alpha(i, j, S_t) \leq \epsilon_{ij}$. The system-wide isolation holds if:

$$\boxed{\max_{i \neq j} \mathcal{C}_\alpha(i, j, S_t) \leq \epsilon(\alpha, \text{Tier})} \tag{22}$$

The per-pair threshold scales with batch size:

$$\epsilon_{ij} \leq \frac{\epsilon_{\text{system}}}{b - 1} \tag{23}$$

Larger batches require proportionally tighter per-pair isolation: a batch of 32 with system bound 0.01 nats requires per-pair bound $\leq 3.2 \times 10^{-4}$ nats. This creates a fundamental tension between throughput (larger batches) and isolation (tighter bounds).

The proof of Lemma 2, including the chain rule argument for Rényi divergence, the distinction between correlated and independent pairwise contamination ($\sqrt{b-1}$ scaling for independent cases), and the batch size scaling table with specific per-pair thresholds, is provided in the private version. Available under commercial license.

6.5 Lemma 3: Cross-Layer Composition (Hardware to Application)

Lemma 6.4 (Cross-Layer Composition)**.** Let the system be organized into L layers (hardware, inference engine, application), each with contamination gain γ_k and per-layer contamination ϵ_k. The system-level contamination is:

$$\boxed{\mathcal{C}_\alpha^{\text{system}} \leq \sum_{k=1}^{L} \epsilon_k \prod_{j=k+1}^{L} \gamma_j} \tag{24}$$

A violation at any layer breaks the chain:

$$\epsilon_k = \infty \implies \mathcal{C}_\alpha^{\text{system}} = \infty \quad \text{regardless of other layers' bounds} \tag{25}$$

This is the **weakest link property**: software isolation is meaningless without hardware isolation. The seL4 experience is a concrete instance: seL4 proves $\epsilon_{\text{software}} = 0$ but Spectre/Meltdown create $\epsilon_{\text{hardware}} > 0$, yielding nonzero system contamination despite perfect software proofs. For GPU inference: even if the serving software perfectly isolates principals at the logical level, LeftoverLocals contamination at the hardware level propagates through the inference engine to the application output.

The proof of Lemma 3, the cross-layer gain analysis with numerical values for hardware (γ_1), inference engine (γ_2), and application (γ_3) layers, the attention sink amplification factor, the system-level bound computation for specific hardware configurations, and the formal derivation of the Spectre/seL4 lesson are provided in the private version. Available under commercial license.

6.6 Compositional Architecture Summary

Composition Type	Lemma	Bound Structure	Key Insight
Sequential (agentic)	Lemma 1	$\sum \delta_i \prod \gamma_j + \sum \Xi_{i,j}$	Interaction terms dominate; AI-4 eliminates them
Parallel (batch)	Lemma 2	$(b-1) \cdot \max \epsilon_{ij}$	Linear scaling; larger batches need tighter per-pair bounds
Cross-layer (HW to app)	Lemma 3	$\sum \epsilon_k \prod \gamma_j$	Weakest link; hardware isolation is foundational

7 Compliance Tiers

The Stateful Isolation Law defines four compliance tiers mapping the continuous contamination bound to discrete assurance levels. The tier structure follows database transaction isolation levels: isolation is graduated, each level provides specific guarantees against specific anomaly classes, and higher levels impose higher performance costs.

7.1 Tier 0: Development (Non-Compliant)

Tier 0 is explicitly defined as **non-compliant**. It exists for two reasons: (1) acknowledgment that the vast majority of deployed AI inference systems currently operate at this level, and (2) internal development environments processing only synthetic data may legitimately operate without isolation guarantees.

Tier 0 must never be used in production with real user data.

Current industry state: All major cloud inference providers—AWS Bedrock, Azure OpenAI, Google Vertex AI, Anthropic, and open-source frameworks (vLLM, SGLang)—currently operate at Tier 0 for the majority of their offerings. This is not a criticism of engineering quality; these systems are optimized for throughput and cost. It is a factual characterization: the isolation guarantees provided do not meet any formal standard.

7.2 Tier 1: Standard

Assurance level: Statistical isolation. The expected information leakage between any two principals is bounded at the Tier 1 threshold per interaction.

What it protects against: Opportunistic cross-tenant leakage, accidental state carry-over, naive timing attacks, casual adversaries. Sufficient for general-purpose commercial SaaS where data is not subject to specific regulatory requirements.

What it does not protect against: Persistent, targeted adversaries with thousands of controlled queries. Nation-state actors. Physical hardware access.

Rényi order: $\alpha = 1$ (mutual information metric).

Key mechanisms: KV drift monitoring, cache salt enabled, timing jitter injection, IFC labeling with tool invocation gate, software memory zeroing, DP for federated learning.

Estimated overhead: 5–10% latency increase; 10–30% memory increase (from cache salt); $< 2\%$ throughput reduction.

Migration from Tier 0: Primarily software configuration. Estimated 2–4 weeks of engineering time. No hardware changes required.

Specific contamination bounds (ϵ_1), per-clause threshold values (λ_K, λ_V, μ_1, τ_{safe}, ϵ_{fed}), monitoring intervals, safety factors, and implementation parameters for Tier 1 compliance are specified in the private version. Available under commercial license.

7.3 Tier 2: Regulated

Assurance level: Worst-case isolation. The maximum information any adversary can extract is bounded regardless of computational power, number of observations, or auxiliary information. This is the differential privacy guarantee.

What it protects against: All attacks documented in Section 2, including PROMPTPEEK, Early Bird, LeftoverLocals, KV-cache inversion, and inter-agent trust exploitation. Persistent adversaries with API-level access. Side-channel attacks through timing, cache contention, and batch scheduling.

What it does not protect against: Physical hardware access (EM emanation, power analysis). TEE firmware compromise (MOLE-class). Hardware supply chain attacks.

Rényi order: $\alpha = \infty$ (worst-case / differential privacy metric).

Key mechanisms: Dedicated KV memory pools, separate batch processing, constant-time serving, TEE chain of trust, hardware memory zeroing with attestation, CUDA deterministic mode, full IFC with dynamic taint thresholds.

Estimated overhead: 50–200 ms additional latency; 2–3× memory consumption; 40–60% throughput reduction. Comparable to encrypted database systems in regulated financial services.

Migration from Tier 1: Architectural changes required. Estimated 2–6 months of engineering time. Hardware procurement may be needed (CC-capable GPUs, TEE-capable CPUs).

Specific contamination bounds (ϵ_2), per-clause threshold values, architectural specifications for dedicated KV pools and constant-time serving, TEE deployment requirements, deterministic computation configuration, and the full parameter table for Tier 2 compliance are specified in the private version. Available under commercial license.

7.4 Tier 3: Sovereign

Assurance level: Classical noninterference ($\epsilon = 0$). The probability distribution over any principal's output is exactly identical whether or not any other principal exists on the system.

What it protects against: All documented and theoretical attacks, including hardware side-channels, EM emanation, and firmware compromise.

Key mechanisms: Dedicated GPU per principal, air-gapped network segments, full memory wipe between sessions, EM shielding, physical access control, supply chain verification, independent model copies.

Estimated overhead: 5–10× cost relative to Tier 0 (dedicated hardware eliminates all sharing).

Appropriate for: National security, critical infrastructure, classified intelligence, and any application where the cost of a single contamination event exceeds the cost of dedicated infrastructure.

Migration from Tier 2: Primarily hardware and facilities. Estimated 6–18 months, comparable to establishing a new classified computing facility.

7.5 Regulatory Mapping

The compliance tiers align with existing regulatory frameworks. No existing regulation explicitly addresses cross-user state contamination through shared KV-cache or GPU memory; the mappings below apply the *intent* of existing regulations to the contamination vectors identified by the Stateful Isolation Law.

Regulation / Standard	Min. Tier	Rationale
GDPR Art. 25 (Data Protection by Design)	Tier 2	Requires "appropriate technical measures" for purpose limitation. Cross-tenant contamination violates purpose limitation.
GDPR Art. 5(1)(f) (Integrity / Confidentiality)	Tier 2	Requires protection against "unauthorised or unlawful processing." Stateful contamination constitutes unauthorized processing.
EU AI Act Art. 15 (Robustness)	Tier 2	Requires "appropriate levels of accuracy, robustness and cybersecurity" for high-risk AI. KV-cache contamination directly affects accuracy and robustness.
CCPA/CPRA ADMT Regulations (eff. Jan 2027)	Tier 2	Requires "segregation of environments" for automated decision-making. Closest existing regulation to inference-level isolation.
HIPAA Security Rule §164.312	Tier 2	Requires "access controls" and "audit controls" for ePHI. Cross-tenant contamination constitutes unauthorized access.
FedRAMP High	Tier 2	Requires "strong isolation" for high-impact systems.
FedRAMP Moderate	Tier 1	Requires "adequate isolation" for moderate-impact systems.
SOC-2 Type II (Security)	Tier 1	Requires "logical access controls." Tier 1 provides minimum formal mitigation.
ISO 42001 (AI Management System)	Tier 1	Requires "management of AI system risks." Cross-tenant contamination is a documented risk.
PCI DSS v4.0	Tier 2	Requires "strong access control" and "network segmentation" for cardholder data.
NIST AI 600-1 (Gen AI Profile)	Tier 1	Identifies "data leakage" as a risk category. Tier 1 addresses the identified risk class.
Executive Order 14110 (Safe AI)	Tier 2	Requires dual-use foundation models to demonstrate safety.

Regulation / Standard	Min. Tier	Rationale
ITAR / EAR (Export Control)	Tier 3	Requires protection of controlled technical data. Zero cross-contamination is the only defensible standard.
CNSSI 1253 (National Security)	Tier 3	Requires "high confidentiality" controls. Classical noninterference is the established standard for classified computing.
UK Data Protection Act 2018	Tier 2	Mirrors GDPR requirements.
Japan APPI (Amended 2022)	Tier 1	Requires "necessary and appropriate measures." Tier 1 provides a defensible baseline.
Singapore PDPA	Tier 1	Requires "reasonable security arrangements."
Brazil LGPD	Tier 2	Mirrors GDPR structure. Data protection by design maps to Tier 2.
South Korea PIPA	Tier 1	Requires appropriate safety measures for personal information.
Australia Privacy Act (APP 11)	Tier 1	Requires "reasonable steps" to protect personal information.
Canada PIPEDA (Principle 7)	Tier 1	Requires safeguards appropriate to sensitivity of information.

7.6 The Economic Case

The cumulative effect of Tier 2 compliance mechanisms is approximately 40–60% throughput reduction and 2–3× memory consumption relative to Tier 0. At current GPU prices, this translates to approximately $1–2 per GPU-hour of additional cost.

For a provider serving 1 million requests per day on 1,000 GPUs:

$$\text{Annual Tier 2 cost} \approx 1{,}000 \times \$1.50/\text{hr} \times 8{,}760\ \text{hr/yr} \approx \$13\text{M/year} \tag{26}$$

This is significant but modest compared to GDPR fines (up to 4% of global revenue) and the reputational cost of a public cross-tenant contamination incident. The economic case for compliance strengthens as the regulatory landscape matures.

8 Enforcement Protocol

The Stateful Isolation Law is only as strong as its enforcement. A bound that is defined but not monitored provides no security; a bound that is monitored but not enforced provides only detection. The enforcement protocol specifies the complete chain from detection through remediation, audit, and attestation.

8.1 The Four-Phase Protocol

Every clause shares a common enforcement structure:

$$\textbf{Detect} \rightarrow \textbf{Flush} \rightarrow \textbf{Hash} \rightarrow \textbf{Attest} \tag{27}$$

Detect: Continuous monitoring of clause-specific metrics identifies when a bound is violated or at risk.

Flush: Contaminated state is invalidated—zeroed or recomputed to restore compliance.

Hash: A cryptographic hash of the pre-remediation state is committed to an immutable audit ledger, preserving forensic evidence without retaining contaminated data.

Attest: The system produces a cryptographic attestation that post-remediation state satisfies the relevant bounds, signed by a hardware-rooted attestation key chained to the manufacturer's root certificate.

Each phase is necessary. Without detection, violations go unnoticed. Without flushing, violations persist. Without hashing, violations leave no forensic trail. Without attestation, there is no cryptographic proof for auditors that remediation was successful.

8.2 Per-Clause Monitoring

Clause	Metric	Method
AI-1 (K)	$\|\Delta\mathbf{K}\|_2$	Power iteration (3 iterations per check)
AI-1 (V)	$\|\Delta\mathbf{V}\|_F$	Single-pass norm computation
AI-2	$I(W^A_{\text{attn}}; \mathbf{KV}^B)$	Canary principal correlation test
AI-3	$\Delta T(I_B)$	TTFT distribution analysis
AI-4	Taint label on tool arguments	IFC gate check at every invocation
AI-5	Memory content post-free	Canary kernel read
AI-6	Cumulative $\hat{\epsilon}(\alpha)$	Rényi privacy accountant

Each metric has two thresholds: a **warning threshold** (0.8× violation threshold) that triggers increased monitoring frequency and preparation for flush, and a **violation threshold** that triggers the full four-phase protocol immediately.

Specific monitoring intervals per tier, overhead budgets (compute, memory, latency), minimum detection probabilities, flush ordering dependencies for multi-clause violations, attestation chain cryptographic specifications, ledger entry structure, and the service resumption gate protocol are specified in the private version. Available under commercial license.

8.3 Incident Classification

Level	Condition	Example	Response
SEV-4	Warning threshold crossed	KV drift at 85% of bound	Log, increase monitoring
SEV-3	Single-clause violation, single session	One session exceeds λ_K	Full protocol, affected session
SEV-2	Multi-clause or multi-session violation	Cross-tenant contamination	Full protocol, all co-residents
SEV-1	Hardware-level violation or TEE compromise	LeftoverLocals-class leak	GPU offline, breach notification
SEV-0	Systemic failure or confirmed exfiltration	Response misrouting, PII exposure	System halt, regulatory notification

8.4 SOC-2 and ISO 42001 Alignment

The enforcement protocol maps directly to existing audit frameworks:

Audit Criterion	**SIL Mapping**	**Evidence**
SOC-2 CC6.1 (Access)	Clauses AI-2, AI-4	Cache salt config, IFC labels, gate logs
SOC-2 CC6.3 (Access Removal)	Clause AI-5	Zeroing attestation, canary logs
SOC-2 CC7.2 (Monitoring)	All clauses	Continuous metrics, alert history
SOC-2 CC8.1 (Change Mgmt)	Enforcement protocol	Flush logs, attestation chain
ISO 42001 6.1 (Risk)	Tier selection	Threat landscape, tier rationale
ISO 42001 8.4 (Documentation)	This document	Full framework specification
ISO 42001 9.1 (Monitoring)	Enforcement protocol	Monitoring architecture
ISO 42001 10.1 (Nonconformity)	Breach response	Violation records, root cause

An auditor conducting a SOC-2 or ISO 42001 assessment of a system claiming SIL compliance verifies: appropriate tier selection for the data sensitivity and regulatory environment; monitoring coverage of all six clauses at the specified intervals; threshold calibration for the specific model and hardware; ledger integrity (hash chain intact, no gaps); complete four-phase execution for any logged violations; valid attestation signatures chaining to trusted hardware roots; and composability verification (Small Gain condition satisfied, batch scaling applied).

9 Conclusion

9.1 Five Theses

Thesis 1: Stateful inference has created a new class of security failure that existing frameworks cannot address. The KV cache transforms the LLM into a probabilistic database. Classical models assume discrete state, deterministic transitions, and binary access. The contamination functional bridges this gap with a continuous, parameterized measure recovering all classical frameworks as special cases.

Thesis 2: Cross-principal contamination is not theoretical but demonstrated and growing. Section 2 catalogs 25+ attacks spanning every layer: 99% prompt reconstruction (PROMPTPEEK), 181 MB hardware leaks (LeftoverLocals), 82.4% inter-agent compromise, exact gradient inversion with ROUGE > 0.99 (DAGER). The entire AI inference stack operates without provable isolation guarantees.

Thesis 3: Isolation must be continuous, measurable, and enforceable—not binary. The compliance tiers operationalize this: $\alpha = 1$ (mutual information, Tier 1), $\alpha = \infty$ (differential privacy, Tier 2), $\epsilon = 0$ (noninterference, Tier 3). Monotonicity ensures a strict hierarchy.

Thesis 4: Composability is the hardest requirement and the one most likely to fail. The Small Gain condition ($\prod \gamma_i < 1$) is the central requirement. MCP amplification (23–41%) is direct evidence of composability failure. The cross-layer lemma formalizes the Spectre/seL4 lesson: software isolation is meaningless without hardware isolation.

Thesis 5: The regulatory framework for inference-level isolation does not exist, but the legal basis for requiring it does. GDPR, the EU AI Act, HIPAA, FedRAMP, and numerous other frameworks contain provisions that, properly interpreted, require the isolation guarantees formalized here. The CCPA/CPRA ADMT framework (effective January 2027) comes closest. The Stateful Isolation Law provides the technical specification that regulators need.

9.2 The Path Forward

Formal verification. Machine-checked verification in Coq or Isabelle/HOL, following the seL4 precedent. The modular structure facilitates incremental formalization.

Empirical calibration. Threshold calibration against specific model families and hardware platforms to tighten worst-case bounds.

Hardware specification. Engaging GPU manufacturers to produce formal specifications of memory isolation properties—the prerequisite for Tier 2+ hardware compliance.

Regulatory engagement. Providing regulators with the technical framework to write explicit inference-isolation requirements.

Standardization. The structure mirrors existing security standards (SOC-2, ISO 27001, NIST 800-53) to facilitate integration.

9.3 Closing Statement

The economics of shared inference have created a regime in which the computational states of millions of principals are physically entangled at every level of the stack. The attacks documented in this paper demonstrate that this entanglement is exploitable, practical against production systems, and consequential.

The Stateful Isolation Law provides the formal framework to bound, measure, and enforce isolation in this regime. The contamination functional unifies decades of security research into a single parameterized measure. The six clauses instantiate it for every contamination surface.

The Composability Theorem establishes when component guarantees propagate to system-level assurance. The compliance tiers translate mathematical bounds into engineering practice aligned with regulatory requirements.

The physics is complete. The engineering is specified. The regulatory mapping is drawn. What remains is implementation—and the will to prioritize the security of the people whose data flows through these systems over the throughput of the systems that process it.

A Attack Catalog

Attack	Clause(s)	Tier	Mechanism	Reference
PROMPTPEEK	AI-3	1	TTFT cache-hit oracle	Wu et al., NDSS 2025
Early Bird	AI-3	1	Production timing channel	arXiv 2409.20002
InputSnatch	AI-1, 3	1	Medical query KV timing	arXiv 2411.18191
KV-Cloak (inv.)	AI-1	2	KV tensor reconstruction	arXiv 2508.09442
KV-Cloak (coll.)	AI-1, 2	2	Crafted cache collisions	arXiv 2508.09442
KV-Cloak (inj.)	AI-1	2	Adversarial KV manipulation	arXiv 2508.09442
LeftoverLocals	AI-5	2	GPU memory not zeroed	CVE-2023-4969
CUDA Leaks	AI-5	2	Shared/global mem leakage	Di Pietro et al., 2013
GPU Cache	AI-5	3	Texture cache side-channel	CCS 2018
MOLE	AI-5	3	GPU TEE MCU exploit	CCS 2025
EM Emanation	AI-5	3	Electromagnetic channel	USENIX Sec. 2022
Vertex Misroute	AI-1, 2	1	HTTP proxy cross-routing	Google, Sept 2025
EchoLeak	AI-4	2	Zero-click injection	CVE-2025-32711
GPT-4 Sys Leak	AI-1, 2	1	Caching error cross-session	OpenAI, Nov 2023
Inter-Agent	AI-4	2	Peer agent relay injection	arXiv 2507.06850
Magentic-One	AI-4	2	Malicious file → code exec	OpenReview 2025
MCP Amplify	AI-4	2	Cross-server propagation	arXiv 2601.17549
Bedrock Poison	AI-4	2	Persistent indirect injection	Unit 42, 2025

Attack	**Clause(s)**	**Tier**	**Mechanism**	**Reference**
LangChain RCE	AI-4	2	Serialization → RCE	CVE-2025-68664
DAGER	AI-6	1	Exact gradient inversion	NeurIPS 2024
SecAgg Evasion	AI-6	2	Model inconsistency	CCS 2022
FA Drift	AI-1	1	10× numerical deviation	Golden et al., 2024
KVTuner Flip	AI-1	1	Quantization cascade	arXiv 2502.04420
Token Injection	AI-1	2	Adversarial KV perturbation	MTI, 2025
PagedAttn Alias	AI-2	1	Block sharing	Kwon, SOSP 2023

B Regulatory Cross-Reference Table

Regulation	Provision	Tier	SIL Clause Mapping
GDPR	Art. 25 (DPbD)	2	AI-1 through AI-5 (technical measures)
GDPR	Art. 5(1)(f)	2	AI-2, AI-5 (confidentiality)
GDPR	Art. 32 (Security)	2	All clauses
GDPR	Art. 34 (Breach)	—	Enforcement protocol (SEV-1+)
EU AI Act	Art. 15 (Robustness)	2	AI-1, AI-2, AI-3
EU AI Act	Art. 9 (Risk Mgmt)	1	Tier selection
EU AI Act	Art. 12 (Records)	1	Audit ledger
CCPA/CPRA	ADMT Regs	2	AI-1 through AI-5
CCPA	§1798.82 (Breach)	—	Enforcement (SEV-1+)
HIPAA	§164.312(a)	2	AI-2, AI-4
HIPAA	§164.312(b)	2	Audit ledger
HIPAA	§164.312(c)	2	AI-1, AI-5
FedRAMP	High	2	All clauses
FedRAMP	Moderate	1	AI-1, AI-2, AI-3, AI-5
PCI DSS v4.0	Req. 1 (Network)	2	AI-2, AI-5
PCI DSS v4.0	Req. 7 (Access)	2	AI-4
PCI DSS v4.0	Req. 10 (Logging)	2	Audit ledger
SOC-2	CC6.1	1	AI-2, AI-4
SOC-2	CC7.2	1	Enforcement protocol
SOC-2	CC8.1	1	Enforcement protocol
ISO 42001	6.1	1	Tier selection
ISO 42001	8.4	1	This document
ISO 42001	9.1	1	Enforcement protocol
NIST AI 600-1	MS-2.6, MS-2.7	1	All clauses
EO 14110	Sec. 4.2	2	All clauses
ITAR/EAR	22 CFR 120-130	3	All clauses

Regulation	Provision	Tier	SIL Clause Mapping
CNSSI 1253	High Confid.	3	All clauses
UK DPA 2018	Sch. 2, Pt. 1	2	Mirrors GDPR
Japan APPI	Art. 23	1	AI-1, AI-2, AI-5
Singapore PDPA	Pt. VI	1	AI-1, AI-2, AI-5
Brazil LGPD	Art. 46	2	Mirrors GDPR
S. Korea PIPA	Art. 29	1	AI-1, AI-2, AI-5
Australia PA	APP 11	1	AI-1, AI-2, AI-5
Canada PIPEDA	Principle 7	1	AI-1, AI-2, AI-5

Intellectual Property (IP) Declaration

The methods, logic structures, contamination bounds, compliance tier specifications, and enforcement protocols contained in this work and its associated private specification are the sole property of Ryan Fields.

Public License (Non-Commercial)

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.

- **Academic Use:** Researchers may share and use this framework for non-commercial academic purposes, provided full attribution is given to Ryan Fields.
- **No Derivatives:** No modifications or "remixes" of the clause definitions, theorem statements, or compliance tier specifications are permitted without express written consent.

Commercial Prohibition

Commercial use of this framework is strictly prohibited without license. This includes, but is not limited to:

- Integration into proprietary AI inference serving platforms.
- Use within commercial AI governance, compliance, or audit software.
- Implementation of the compliance tiers or enforcement protocol in production systems serving paying customers.
- Use by cloud providers to market "SIL-compliant" inference services.

What This Public Version Provides

This document provides the complete framework architecture: threat landscape, classical analysis, contamination functional definition, clause statements, theorem statements, compliance tier descriptions, regulatory mapping, and enforcement protocol structure. It is sufficient for:

- Understanding the crisis and the solution architecture.
- Assessing regulatory exposure against the tier mapping.
- Evaluating whether the framework addresses an organization's isolation requirements.
- Academic citation and non-commercial research.

What Requires Commercial License

Implementation of the Stateful Isolation Law requires the private specification, which includes:

- Explicit threshold constants for all six clauses, calibrated per compliance tier.
- Complete derivations with scaling laws for model size, context length, batch size, and hardware configuration.

- Proof bodies for the Composability Theorem and all three lemmas.
- Monitoring intervals, overhead budgets, and implementation parameters.
- Attestation chain cryptographic specifications.
- Cross-layer gain numerical values and system-level bound computation procedures.

Contact

UncleBroFields@proton.me

References

[1] D.E. Bell and L.J. LaPadula, "Secure Computer Systems: Mathematical Foundations," MITRE Corp., MTR-2547, 1973.

[2] K.J. Biba, "Integrity Considerations for Secure Computer Systems," MITRE Corp., MTR-3153, 1977.

[3] J.A. Goguen and J. Meseguer, "Security Policies and Security Models," *IEEE S&P*, 1982.

[4] J. Rushby, "Noninterference, Transitivity, and Channel-Control Security Policies," SRI International, 1992.

[5] D.D. Clark and D.R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *IEEE S&P*, 1987.

[6] H. Berenson et al., "A Critique of ANSI SQL Isolation Levels," *ACM SIGMOD*, 1995.

[7] A. Adya, B. Liskov, and P. O'Neil, "Generalized Isolation Level Definitions," *IEEE ICDE*, 2000.

[8] G. Klein et al., "seL4: Formal Verification of an OS Kernel," *ACM SOSP*, 2009.

[9] T. Murray et al., "seL4: From General Purpose to a Proof of Information Flow Enforcement," *ACM TOCS*, 2013.

[10] P. Kocher et al., "Spectre Attacks: Exploiting Speculative Execution," *IEEE S&P*, 2019.

[11] M. Lipp et al., "Meltdown: Reading Kernel Memory from User Space," *USENIX Security*, 2018.

[12] T. Di Pietro et al., "CUDA Leaks: Information Leakage in GPU Architectures," arXiv:1305.7383, 2013.

[13] H. Naghibijouybari et al., "Rendered Insecure: GPU Side Channel Attacks are Practical," *ACM CCS*, 2018.

[14] S. Maia et al., "Can one hear the shape of a neural network?: Snooping the GPU via Magnetic Side Channel," *USENIX Security*, 2022.

[15] T. Hickey et al., "LeftoverLocals: Listening to LLM Responses Through Leaked GPU Local Memory," Trail of Bits, CVE-2023-4969, 2024.

[16] MOLE Attack, "Breaking NVIDIA GPU TEE," *ACM CCS*, 2025.

[17] T. Dao et al., "FlashAttention: Fast and Memory-Efficient Exact Attention," *NeurIPS*, 2022.

[18] T. Dao, "FlashAttention-2," *ICLR*, 2024.

[19] J. Shah et al., "FlashAttention-3," *NeurIPS*, 2024.

[20] W. Kwon et al., "Efficient Memory Management for LLM Serving with PagedAttention," *ACM SOSP*, 2023.

[21] Y. Yu et al., "Orca: A Distributed Serving System for Transformer-Based Generative Models," *USENIX OSDI*, 2022.

[22] Y. Wu et al., "PROMPTPEEK," *NDSS*, 2025.

[23] Early Bird, arXiv:2409.20002, 2024.

[24] InputSnatch, arXiv:2411.18191, 2024.

[25] KV-Cloak, arXiv:2508.09442, 2025.

[26] D. Golden et al., "Is Flash Attention Stable?," arXiv:2405.02803, 2024.

[27] KVTuner, arXiv:2502.04420, 2025.

[28] Inter-Agent Trust Exploitation, arXiv:2507.06850, 2025.

[29] MCP Security Analysis, arXiv:2601.17549, 2025.

[30] Amazon Bedrock Agent Memory Poisoning, Unit 42, 2025.

[31] LangChain Serialization RCE, CVE-2025-68664, 2025.

[32] EchoLeak, CVE-2025-32711, 2025.

[33] FIDES, Microsoft Research, arXiv:2505.23643, 2025.

[34] CaMeL, Google DeepMind, arXiv:2503.18813, 2025.

[35] K. Petrov et al., "DAGER: Exact Gradient Inversion for LLMs," *NeurIPS*, 2024.

[36] D. Pasquini et al., "Eluding Secure Aggregation," *ACM CCS*, 2022.

[37] M. Abadi et al., "Deep Learning with Differential Privacy," *ACM CCS*, 2016.

[38] K. Bonawitz et al., "Practical Secure Aggregation," *ACM CCS*, 2017.

[39] I. Mironov, "Rényi Differential Privacy," *IEEE CSF*, 2017.

[40] G. Barthe and B. Köpf, "Information-Theoretic Bounds on Flows," *IEEE CSF*, 2011.

[41] P. Cuff and L. Yu, "DP as a Mutual Information Constraint," *IEEE Trans. IT*, 2016.

[42] K. Zhou, J.C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice Hall, 1996.

[43] Google Vertex AI Response Misrouting, September 2025.

[44] OpenAI GPT-4 Turbo System Message Exposure, November 2023.

[45] Regulation (EU) 2016/679 (GDPR).

[46] Regulation (EU) 2024/1689 (EU AI Act).

[47] CCPA/CPRA ADMT Regulations, effective January 2027.

[48] HIPAA, 45 CFR Part 164.

[49] NIST AI 600-1, 2024.

[50] ISO/IEC 42001:2023.

[51] Executive Order 14110, 2023.

[52] CNSSI 1253.

[53] NVIDIA H100 Confidential Computing Architecture, 2023.

[54] NVIDIA Multi-Instance GPU (MIG) User Guide, 2023.

Auburn Patent Family

Full specifications available under commercial license.

Contact: UncleBroFields@proton.me

PUBLIC VERSION — CC BY-NC-ND 4.0