

MINIMIZING THE ATTACKABLE WORLD

Ephemeral Execution, Temporary Authority, and the Next Cybersecurity Boundary

Cybersecurity White Paper

October 2026

THE ATTESTORR PRINCIPLE

Minimize what can be attacked.

Minimize how long it can be attacked.

Minimize what remains after the attack window closes.

Less persistence. Less exposure. Less opportunity.

AttesTorr, Inc.

Layer ZERO Concept and Architecture

Executive Summary

Cybersecurity has spent decades making persistent systems harder to penetrate. Firewalls protect networks. Encryption protects data. Identity systems protect accounts. Endpoint security protects machines. Monitoring attempts to identify attackers after they arrive.

All of these remain necessary. But they share an underlying assumption: the thing being protected continues to exist.

The best way to reduce the attack surface is to reduce what exists to be attacked.

AttesTorr Layer ZERO introduces execution environments in which authority, working data, credentials, identities, keys, AI interactions, and other sensitive state can be bounded to a particular purpose and lifetime.

A fresh Ephemeral Trust Room is admitted, and identity and purpose are bound to it. Governed execution occurs within a declared trust boundary. Nothing leaves merely because an application calls it permitted: any export must satisfy its authorization conditions after the room closes. At closure, execution-scoped authority and identity are invalidated. Evidence can remain for independent verification without requiring the sensitive execution environment itself to remain.

ATTACK-SURFACE REDUCTION

Less persistent state is available to target.

ATTACK-WINDOW REDUCTION

Sensitive state and authority remain available for less time.

The objective is not to claim that a system can never be attacked. The objective is to give an attacker less to attack, less time to attack it, less authority to steal, and less useful state to recover afterward.

The verifiable claim is narrower than disappearance. AttesTorr is designed to make it independently verifiable that execution-scoped authority and identity ended, with no admissible way to continue them inside the declared trust boundary. It does not claim to prove that every copy of sensitive data has ceased to exist, or to control copies outside that boundary. Reducing what persists lowers exposure; it is not itself the guarantee.

Architecture Snapshot



Independent verification evaluates closure evidence after CLOSE. It is not a lifecycle phase.

Layer ZERO governs a bounded execution lifecycle in which continuity is not assumed. A subsequent execution must establish its own authority rather than silently inheriting the prior Ephemeral Trust Room's authority.

1. The Persistence Problem

Modern computing was built around persistence. Applications retain state. Servers accumulate data. Credentials remain valid. Logs multiply. Caches survive. Sessions leave artifacts. Backups reproduce information. Biometric systems may retain identity information. AI systems can create still more copies, embeddings, histories, prompts, telemetry, and derived information.

Every persistent element can become part of the attackable environment. Traditional cybersecurity therefore faces an expanding problem: the more valuable digital infrastructure becomes, the larger the collection of things that must continuously be defended.

AttesTorr changes the premise. Not everything that must exist during computation must continue to exist afterward.

2. Cybersecurity Through Existence Minimization

**Minimize what exists. Minimize how long it exists.
Minimize the authority it possesses. Verify that it
ended.**

AttesTorr's principle changes the questions security asks.

Instead of asking only how to defend data, Layer ZERO also asks whether that data needs to remain. Instead of asking only how to protect a credential, Layer ZERO asks whether that authority needs to remain valid after the execution ends. Instead of asking how to continually secure an execution environment, Layer ZERO asks why the execution environment should continue existing at all.

The difference is fundamental.

3. The Ephemeral Trust Room

ADMIT

BIND

EXECUTE

CLOSE

SUCCESSOR

Independent verification of closure evidence happens separately, after CLOSE.

STEP	ROLE
ADMIT	A designated admission function determines whether the human, machine, AI, application, device, data, or other participant may enter the execution, and only that function issues the execution's identity. Arriving with input or a connection does not by itself establish authority.
BIND	Bind identity, purpose, authority, scope, and relevant constraints to that execution.
EXECUTE	Perform the authorized work within the declared boundary.
CLOSE	Terminate the execution environment and execution-scoped authority according to the applicable closure profile.
SUCCESSOR	Where a subsequent execution is requested, the prior execution's closure evidence is checked, and the new execution is admitted separately with a new identity. No live authority is inherited from the prior execution.
INDEPENDENT VERIFICATION (outside the lifecycle)	Not a lifecycle phase. After closure, an independent verifier evaluates the complete closure evidence and reaches its own conclusion. It does not issue a finality certificate, and its conclusion does not grant authority to any successor execution.

An Ephemeral Trust Room is a purpose-bound execution environment created for a governed lifecycle.

The important property is that continuity is not assumed. The next execution must earn its own authority.

4. Reducing the Attack Surface

Ephemerality changes what an attacker can find. A persistent environment may expose long-lived credentials, cached information, application state, session tokens, sensitive temporary files, AI histories, biometric working data, standing permissions, machine identities, encryption keys, residual memory, databases, logs, telemetry, abandoned services, and historical execution state.

An ephemeral architecture seeks to eliminate unnecessary persistence across these categories. This does not eliminate vulnerabilities during execution. It changes what remains available before, during, and especially after execution.

Less persistence creates less residual attack surface.

5. Reducing the Attack Window

Attack surface is only one dimension. Time is another.

A credential available for a year presents a different risk than authority valid for a single execution. An environment operating continuously presents a different opportunity than one that closes immediately after completing its authorized purpose. A session that remains active provides an attacker time to explore. A short-lived execution compresses that opportunity.

AttesTorr therefore treats closure time as a security property. The objective is a rapid transition from active execution to a safe closed state.

For implementations capable of extremely rapid safe closure, the usable attack window can approach the duration of the authorized execution itself rather than the lifetime of the underlying infrastructure.

We reduce not only the attack surface. We reduce the attack time.

6. Temporary Authority

Persistent systems frequently accumulate standing authority. Users remain logged in. API keys remain valid. Machine credentials remain available. Services retain privileges long after a particular task has ended. AI agents may receive permissions broader or longer-lived than the work actually requires.

AttesTorr instead supports execution-scoped authority. Authority can be purpose-bound, identity-bound, time-bound, environment-bound, non-transferable, revocable, and extinguished at closure.

When the governed execution ends, the corresponding authority does not need to survive.

An attacker cannot indefinitely exploit authority that no longer exists.

7. Biometrics Without Permanent Biometric Authority

BIOMETRIC**VERIFY****TEMP
AUTHORITY****EXECUTE****CLOSE****TERMINATE**

Biometrics prove presence. They should not automatically become permanent authority.

Verify the biometric. Do not warehouse the encounter.

Biometrics are becoming increasingly important to digital identity. Fingerprints, facial characteristics, iris patterns, voice, and other biological characteristics can provide powerful evidence that a particular human is present.

But biometrics present a unique cybersecurity problem: a password can be changed; a human face or fingerprint cannot easily be replaced.

AttesTorr therefore separates biometric proof from persistent machine authority.

The protected application or server need not receive or warehouse the raw biometric. Instead, it can receive the result required for the authorized operation.

Biometric data that legitimately must persist requires appropriate protection. AttesTorr does not imply that every biometric reference can or should disappear. The distinction is between necessary identity infrastructure and the unnecessary persistence of every biometric interaction.

Enrollment, template storage, capture-device integrity, matching accuracy, and copies held by identity providers remain the responsibility of those systems unless they are explicitly included, and evidenced, within the declared trust boundary.

8. AI Containment

AI expands cybersecurity beyond human accounts. An AI system can act, retrieve, infer, call tools, generate code, access databases, communicate with services, and operate at machine speed.

That creates a new question: how much standing authority should an AI possess?

Layer ZERO provides an architectural answer: summon the intelligence for the task rather than grant perpetual authority to the intelligence.

An AI execution can be admitted for a defined purpose, constrained to authorized resources, given temporary identity, prevented from unrestricted persistence, restricted in export, closed after completion, and required to requalify for subsequent execution.

This creates a distinction between AI capability and AI authority. Powerful intelligence does not require permanent access.

Layer ZERO limits whether an AI execution is admitted, how long its authority lasts, and what evidence it must leave. It does not prove that a model's outputs are correct, safe, fair, unbiased, or appropriate for consequential use. Model evaluation, policy governance, and application security remain necessary.

9. Gateway Enforcement

HUMAN / AI / DEVICE	ADMISSION + BINDING	GOVERNED EXECUTION	CLOSE	CLOSURE EVIDENCE	AUTHORIZED EXPORT
------------------------	------------------------	-----------------------	-------	---------------------	----------------------

Layer ZERO defines the governing constraints; each box is an assigned enforcement or evidence responsibility. Export, where authorized, occurs only after closure. Independent verification is a separate, read-only evaluation after closure.

Where an implementation places a gateway in front of protected infrastructure, the traditional connection model changes. That gateway enforces Layer ZERO constraints; it is not itself an admission authority. Layer ZERO is the governing architecture, not a gateway, proxy, or service in the network path, and admission remains with the architecture's designated admission function.

In this model, possession of a conventional credential alone need not imply unrestricted server authority. The system can determine who or what is requesting execution, why execution is requested, what resources may be touched, what information may enter, what information may persist, what information may leave, how long authority exists, and whether closure was successfully completed.

The gateway therefore becomes an enforcement point for execution authority, not merely a traffic filter. It does not become a new source of admission or finality, and it does not treat a conventional credential, network location, successful connection, or operational status as sufficient for admission.

10. Fail Closed

An ephemeral security architecture cannot depend solely on assertions that deletion occurred. Closure must be enforceable and verifiable.

AttesTorr's architecture therefore favors fail-closed behavior: if required authorization cannot be established, do not admit; if required execution conditions cannot be maintained, do not continue; if export conditions are not satisfied, do not export; and if closure evidence is incomplete or cannot be verified, a verifier does not recognize finality.

For claim-bearing closure, monitoring, logs, shutdown messages, and operator statements are not substitutes for complete, current, correctly signed closure evidence. Finality is reached only through an independent verifier's own evaluation of that evidence; incomplete or conflicting evidence means the claim is denied.

This changes evidence from an observational function into part of the security architecture.

11. Evidence Without Retaining the Event

A fundamental AttesTorr distinction is that evidence can survive without requiring the underlying execution to survive.

An organization may need proof that an authorized identity was present, required conditions were satisfied, a transaction occurred, a specific output was permitted, closure was initiated, or expected teardown conditions were satisfied.

That does not necessarily require retaining every sensitive input, intermediate state, biometric sample, AI interaction, or session environment.

The desired result is proof without unnecessary persistence. This can reduce the tension between cybersecurity, privacy, accountability, and audit.

Proof without unnecessary persistence.

12. Quantum-Era Security

Protect what must persist. Minimize what must persist.

Quantum computing is frequently described as if it will make all cryptography obsolete. That is incorrect.

Large-scale quantum computers pose a serious threat to several widely deployed public-key cryptographic systems, which is why post-quantum cryptographic standards are already being deployed.

AttesTorr does not replace cryptography. It complements it. Post-quantum cryptography addresses how to protect information and cryptographic relationships that must survive. Ephemeral architecture addresses another question: what information and authority need to survive in the first place?

Anything that legitimately survives - including long-lived credentials, signatures, permitted exports, receipts, or other evidence - should use appropriate cryptographic protection, including post-quantum-resistant mechanisms when required.

AttesTorr's contribution is not a claim of quantum-proof execution. Its architectural contribution is different: a future cryptanalytic breakthrough cannot decrypt sensitive information that was never unnecessarily retained for it to attack. That does not protect ciphertext already intercepted, prior or external copies, authorized exports, or the keys, signatures, and evidence that must persist; those still require cryptographic protection, including post-quantum mechanisms where required.

13. Cybersecurity as a Function of Time

Traditional cybersecurity frequently treats data primarily as an asset. AttesTorr adds another dimension: duration.

The security exposure associated with a piece of information or authority is affected not only by its value, but by how long it remains available. Conceptually, exposure is related to attackable state multiplied by time.

This is an explanatory rule of thumb, not a validated risk formula. Risk also depends on accessibility, privilege, vulnerabilities, controls, adversary capability, and consequences. Any quantified effectiveness claim requires a stated measurement method, environment, and test evidence.

Reducing either can reduce opportunity. Reducing both changes the architecture.

Persistence is not neutral. Persistence creates exposure time. Therefore, lifecycle control itself becomes a cybersecurity primitive.

14. What AttesTorr Does Not Claim

AttesTorr does not claim that ephemeral systems cannot contain vulnerabilities; attackers cannot act during active execution; all information should disappear; every persistence mechanism is harmful; cryptography is unnecessary; biometrics are inherently secure; quantum computing ceases to matter; hardware is perfectly trustworthy; every byte can always be proven physically absent from every possible location; or any architecture can be declared unhackable.

The proposition is narrower and more defensible: a system can reduce cyber exposure by deliberately limiting the existence, lifetime, authority, retrievability, and continuity of sensitive execution state.

Those properties must ultimately be demonstrated through implementation, testing, independent verification, and clearly declared trust boundaries.

In particular, reducing the attack surface and the attack window are design objectives, not guarantees that vulnerabilities, copies outside the boundary, or other persistence are absent. Execution duration is one dimension of exposure, not a standalone risk measure: no universal reduction ratio is claimed, and no immunity is claimed while execution is active.

Only evidence the architecture authorizes to persist may survive, and that evidence must not itself carry sensitive execution state. The existence of a receipt does not by itself establish that a claim is valid; completeness, freshness, emission authority, and independent validation still apply. Claim-bearing guarantees are limited to a declared, versioned, cryptographically hash-bound trust boundary. Provider assumptions, persistence outside that boundary, and other residual risks are disclosed rather than treated as proven absent.

15. The Layer ZERO Cybersecurity Model

The AttesTorr cybersecurity model can be compressed into six principles.

1 MINIMIZE EXISTENCE

Do not persist sensitive state merely because infrastructure makes persistence easy.

2 MINIMIZE AUTHORITY

Grant only the authority required for the execution.

3 MINIMIZE DURATION

Terminate execution state and temporary authority when their purpose ends.

4 MINIMIZE EXPORT

Only authorized information crosses the execution boundary.

5 VERIFY CLOSURE

Do not equate a software message saying "deleted" with independently supportable finality.

6 REQUIRE NEW AUTHORITY

Do not allow one successful execution to silently become permanent authority.

Together:

ADMIT	BIND	EXECUTE	CLOSE	SUCCESSOR
-------	------	---------	-------	-----------

Independent verification of closure evidence follows a completed lifecycle. It is not an execution phase, a source of successor authority, or a portable finality result.

Conclusion

The cybersecurity industry has become extraordinarily sophisticated at protecting things that remain. AttesTorr asks whether all of those things should remain.

That question becomes increasingly important as AI systems gain machine authority, biometrics become identity infrastructure, servers accumulate ever more data, and quantum computing forces organizations to reconsider long-lived cryptographic assumptions.

The next cybersecurity frontier may therefore be not simply a stronger wall around an ever-growing digital estate. It may be a smaller estate. A shorter-lived one. An environment where authority is temporary, sensitive execution is bounded, persistence requires justification, closure is part of the architecture, and the attacker continually confronts a shrinking target.

Minimize what can be attacked. Minimize how long it can be attacked. Minimize what remains after the attack window closes.

Less persistence. Less exposure. Less opportunity.

The next cybersecurity boundary is existence itself.

Reference Note

National Institute of Standards and Technology (NIST), Post-Quantum Cryptography Project. NIST has standardized post-quantum cryptographic algorithms and recommends migration planning for systems that rely on cryptography vulnerable to sufficiently capable quantum computers.

<https://www.nist.gov/pqc>

Maturity and Claim Boundary

This white paper describes an architecture thesis and product direction. Security properties depend on implementation, deployment profile, hardware and software assumptions, integration choices, verification methods, and declared trust boundaries. Claims must be earned through reference builds, testing, independent verification, and operational evidence.