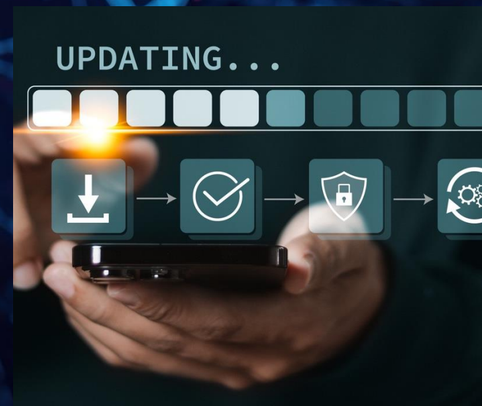
# APPLIED AI FIELD NOTES - EXECUTIVE BRIEF

*AN OVERVIEW OF NSA ZERO TRUST PILLARS FOR ARCHITECTURE DESIGN*

Tom M. Gomez

May 28, 2026




Luminity Digital
INTELLIGENCE, ENGINEERED. DELIVERED.

# KEY AREAS COVERED

1. Introduction to Zero Trust Architecture
2. The Seven Pillars Explained
3. Integration and Implementation Phases
4. Conclusion

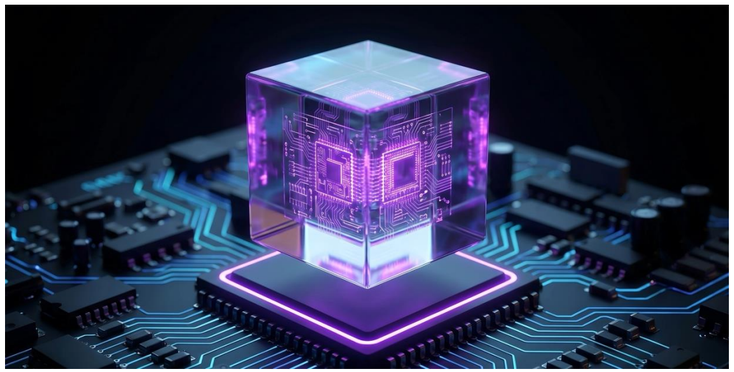
# WHY ARCHITECTURE CHANGES

## Perimeter assumptions no longer hold

- Cloud and SaaS dissolve “inside vs outside,” making location-based trust unreliable.
- VPN and flat network patterns expose weaknesses in risk reviews.

## Architect for containment, not perfection

- Design as if adversaries are already present and credentials will be abused.
- Limit lateral movement with enforcement near resources, not the edge.



## Shift to context-driven access

- Make each request depend on identity, device posture, and current context.
- Enable continuous verification with telemetry and policy-based decisions.


# CORE ZT ACCESS MODEL

### Per-request decisioning

- Evaluate each access attempt using user identity, device posture, and context.
- Stop treating network location as proof of trustworthiness.

### Least privilege per action

- Grant only the minimum permissions needed for this specific request.
- Keep enforcement close to the application, data, or workload.

### Continuous verification architecture

- Re-check sessions as conditions change, not just at login time.
- Make telemetry and decision logging **load-bearing** for audit and tuning.

# PILLARS AS ARCHITECTURE

## Resource pillars define access

- User and Device establish who is requesting and from what posture.
- Application/Workload and Data define what is accessed and protected.
- Network/Environment constrains paths, limiting lateral movement after compromise.

## Cross-cutting pillars make it operable

- Automation and Orchestration turns policy into repeatable, auditable enforcement.
- Visibility and Analytics provides decision telemetry and behavioral signal.

## Architectural takeaway

- Design as a coordinated control fabric, not isolated point solutions.

# SEVEN PILLARS EXPLAINED

# IDENTITY AS CONTROL PLANE

## Build an authoritative identity layer

- Unify workforce, partner, admin, and service identities in one source.
- Define clear ownership for identity lifecycle and access entitlements.

## Enforce per-request access decisions

- Require strong authentication and risk-based conditional access each interaction.
- Use privileged access controls to limit standing admin permissions.

## Make identity the architecture’s input

- Design apps, data, and networks to consume **identity context** for policy.
- Log allow/deny outcomes so decisions remain traceable and auditable.

# DEVICE TRUST SIGNALS

## Establish device authority

- Maintain a complete device inventory with clear ownership and lifecycle state.
- Require enrollment so every endpoint has a managed identity anchor.

## Use posture as an access input

- Gate access on **attestation** signals: integrity, configuration, and encryption status.
- Continuously evaluate health and security telemetry, not one-time compliance.

## Design implication

- Unmanaged devices remove reliable context, weakening identity, data, and segmentation controls.

# APPS AND WORKLOADS

## Know what you run

- Maintain an authoritative application inventory with external exposure mapped.
- Treat each app as a protected resource with explicit access boundaries.

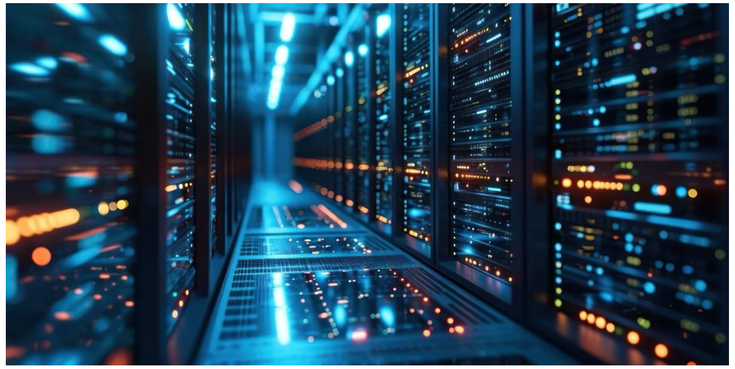
## Protect at runtime

- Add runtime protection and monitoring to detect misuse post-deploy.
- Continuously re-verify application access, not just at login.

## Identity for workloads

- Require workload identity for service-to-service calls, not network location.
- Enforce least-privilege authorization between services per request.

# DATA-DRIVEN ZT POLICY



## Make data visible and named

- Discover and inventory data across cloud, SaaS, and on-prem locations.
- Classify and tag data so policy can follow it everywhere.

## Enforce protection by lifecycle

- Encrypt data at rest and in transit with managed keys.
- Apply rights controls and DLP using classification as the trigger.

## Design for audit and adaptation

- Log data access decisions to prove who accessed what, and why.
- Treat data as the asset; network controls are supporting layers.

# SEGMENTED NETWORK DESIGN


## Move away from flat networks

- Replace broad internal reach with **microsegmentation** around critical environments.
- Use software-defined access to expose only what each request needs.

## Control east-west movement

- Treat internal traffic as hostile; enforce explicit paths and approvals.
- Encrypt in-transit, including east-west flows, to reduce lateral leverage.

## Design for breach containment

- Assume compromise; segmentation becomes the primary **blast radius** limiter.
- Expect multi-year rollout; poor segmentation causes outages and rework.

# AUTOMATION AS ARCHITECTURE

## Design the control plane

- Define clear **policy decision** and **policy enforcement** points per pillar.
- Orchestrate identity, endpoint, network, and data controls as one system.

## Treat policy like software

- Manage policy-as-code with versioning, tests, reviews, and audit trails.
- Standardize reusable policy patterns to reduce one-off exceptions.

## Make operations scalable

- Automate routine approvals and responses to keep pace with change.
- Replace ticket-driven access changes with repeatable, automated workflows.

# ANALYTICS ENABLES TRUST

### Unify telemetry across pillars

- Aggregate identity, endpoint, application, data, and network signals.
- Make context available to every policy decision point.

### Turn data into decisions

- Baseline normal behavior to surface anomalies faster.
- Capture decision telemetry: allow or deny, and the rationale.

### Architecture takeaway

- Without strong analytics, **continuous verification** becomes performative.

# INTEGRATION AND IMPLEMENTATION PHASES

# INTEGRATED PILLAR FABRIC

## Signals drive access decisions

- Identity and device posture are required inputs for each request.
- Those signals govern application use, data access, and network paths.

## Policy executes near resources

- Decisions must be enforced where workloads and data actually run.
- Network controls provide containment when identity is compromised.

## Operate as one control fabric

- Automation coordinates policy decision and enforcement points across pillars.
- Analytics supplies shared telemetry to tune policy and prove outcomes.

# DESIGN ACROSS PHASES

### Discovery: establish ground truth

- Build authoritative inventories and maps across users, devices, apps, data, and network flows.
- Stand up baseline telemetry so later controls make correct decisions.

### Phase One: turn on enforcement

- Make access identity-driven with MFA, conditional checks, and privileged controls.
- Contain breaches using **microsegmentation** and pervasive encryption in transit.

### Phase Two: become adaptive

- Evolve to context-aware decisions and automated responses at machine speed.
- Use advanced analytics to refine policy continuously and prove maturity.

# ARCHITECTURE PRIORITIES NOW

## Establish ground truth first

- Create authoritative inventories for identities, devices, and applications.
- Baseline normal access paths and exposure before tightening controls.

## Sequence enforcement by risk

- Start with **privileged access** to reduce impact of account compromise.
- Protect high-value data and critical environments with early segmentation.

## Plan as a multi-year architecture

- Deliver measurable capabilities in phases, not a single deployment wave.
- Engineer for policy, telemetry, and enforcement to evolve over time.

# 3 SLIDES FOR THE EXECUTIVE HUDDLE


# ARCHITECTURE TAKEAWAY

## Treat Zero Trust as an architecture program

- Use pillars as a coordinated control system, not separate tool buys.
- Plan for phased capability maturity, with measurable operational outcomes.

## Put decisions close to what you protect

- Design policy enforcement at the resource, not the network perimeter.
- Make telemetry and decision logs first-class architecture requirements.

## Future-proof the technology estate

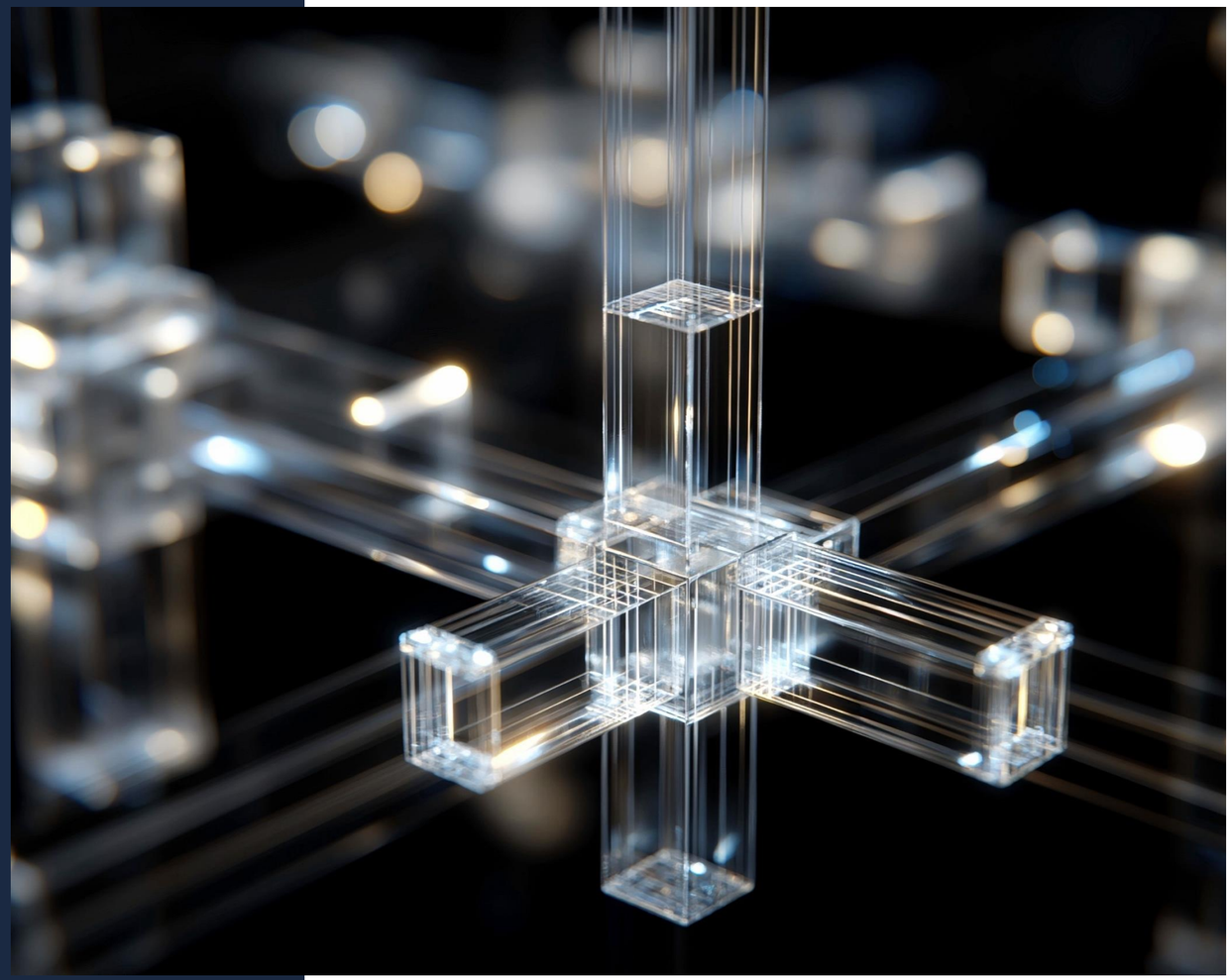
- Adopt **policy-as-code** to reduce lock-in and speed safe change.
- Architect now to avoid higher re-platforming cost after incidents.





Connect with Us

Read our Insights



# 3 STRATEGY QUESTIONS

## 1) Where are we really – past the project-status veneer?

- Which pillars are we calling “in flight” that haven’t cleared Discovery?
- If a regulator asked us to demonstrate continuous verification today, what would we show them?
- Which of our current security investments would we make again knowing what ZIG names as foundational?

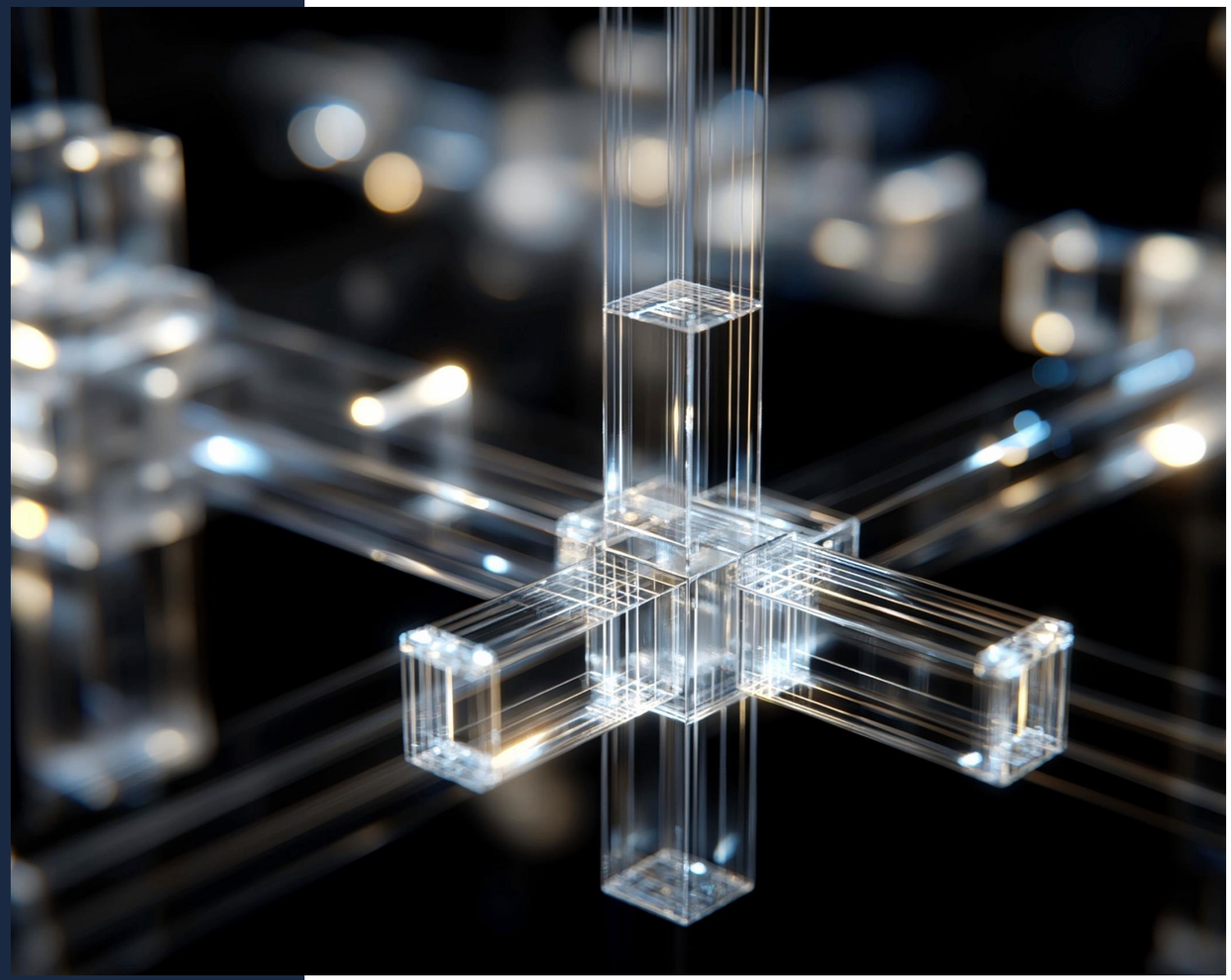
## 2) Where does the real disagreement live?

- Platform consolidation or best-of-breed integration — and who owns the integration fabric if we choose the latter?
- Policy as code or policy in vendor consoles — what are we committing the next decade to?
- Where are we paying ZT-tax for legacy architecture decisions, and what is the cost of leaving them in place?





[CONTINUED ON NEXT SLIDE]



# 3 STRATEGY QUESTIONS

**CONTINUED…..**

**3) Where does the ownership lie, beyond the security organization?**

- Identity, network, and security operations all touch policy. Who arbitrates when they disagree?
- Where does Zero Trust accountability sit on the org chart — and is that the right place?
- What does the board need to see quarterly to know this program is real?





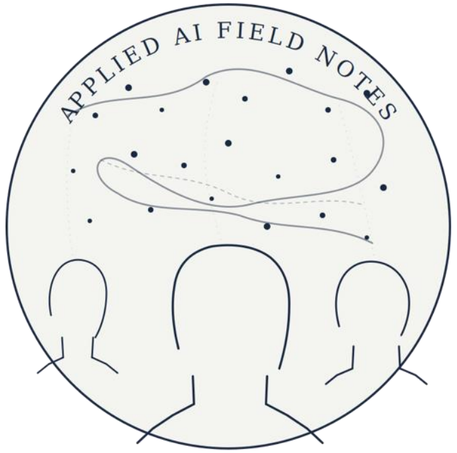
Connect with Us

Read our Insights

# REFERENCES

*Mapping the empirical evidence. Discovering the patterns that matter for Agentic AI in production.*

Read the detailed insights on the Luminity Digital Blog https://www.luminitydigital.com/insights




Subscribe to our Weekly Newsletter | **Follow us on LinkedIn** - https://www.linkedin.com/company/luminity-digital

**Primary Source**
NSA ZIG | ZIG Primer (PDF)

**Phase Resources**
Discovery | Phase One | Phase 2

**Supporting NSA Sources**
Cybersecurity Information Sheets (CSIs) | Technology Mapping

## The Seven Pillars

User Pillar | Device Pillar | Application and Workload Pillar | Data Pillar | Network and Environment Pillar
Automation and Orchestration Pillar | Visibility and Analytics Pillar

## Foundational Standards and Policy

NIST SP 800-207 — Zero Trust Architecture | NIST SP 1800-35 — Implementing a Zero Trust Architecture
Executive Order 14028 — Improving the Nation's Cybersecurity | National Security Memorandum 8 (NSM-8)
DoD Zero Trust Reference Architecture and Strategy | CISA Zero Trust Maturity Model

***Source Recency:***
*The NSA ZIG portal and Primer were released in January 2026. Supporting standards (NIST SP 800-207, EO 14028, NSM-8) predate the release and remain the foundational references.*

***Reading Order Recommendation***
*For architecture teams: NIST SP 800-207 first, then the ZIG Primer, then the relevant pillar pages and CSIs.*
*For program leadership: this brief, then the ZIG Primer overview, then CISA's maturity model for board-readable framing.*