ACROSS THE LINE

A vendor-neutral pattern for running one application across a classification boundary — without asking the boundary to do the work.

June 2026

We create intelligent customers.

TABLE OF CONTENTS

The pattern in brief:.....3
The problem it answers:.....4
The pattern:.....5
1. Data-centric release.....5
2. Ephemeral cryptography.....5
3. Confidential computing.....6
4. Federated zero trust.....6
5. Crypto-agility.....6
How it is built:.....7
Assurance, in the auditor’s terms:.....8
When to use it — and when not to:.....9
Author:.....10
Prior art and sources:.....11

THE PATTERN IN BRIEF:

This is a context-blind pattern. It is the architecturally sound answer to a recurring problem — how to run a single application across a classification boundary, using untrusted infrastructure on the far side, without a perimeter guard becoming the bottleneck — stated without sight of any particular system. The integrator brings the mission; the pattern brings the shape of the answer. It is vendor-neutral by construction: every element below is a published standard or a primitive, not a product.

1. Stop asking the boundary to do the work. Legacy cross-domain solutions inspect at the line, which is where latency, state, and cost accumulate. Protect the data before it reaches the line, and the line becomes a dumb pipe for pre-validated, encrypted packets.
2. Make security a property of the data, not the place. Validate and label at the source; bind the label cryptographically to the payload; encrypt per transaction. The packet then defends itself wherever it travels, including across infrastructure you do not control.
3. Treat the far side as hostile, and compute there anyway. Confidential computing lets the untrusted host process ciphertext it cannot read; the host is a managed risk, not a trusted party. This is what lets you reach commercial AI without surrendering the mission data to it.
4. Federate trust by cryptography, not by directory. In a coalition there is no shared identity provider and partners come and go. Establish trust through attestation and trust anchors, fail closed when verification fails, and the pattern survives a partner leaving the room.
5. Build for the crypto transition now. Decouple the cryptographic service from transport and application so the move to hybrid post-quantum key establishment is a swap, not a rebuild — answering "harvest now, decrypt later" before it answers you.

THE PROBLEM IT ANSWERS:

Doctrine and architecture are pulling in opposite directions. Multi-domain operations and modern data strategy demand that information move and be acted upon across classification levels at the speed of the decision; the architecture that protects those levels is built to keep them apart. The result is an interoperability paradox: the faster the mission needs to be, the more the separation costs it.

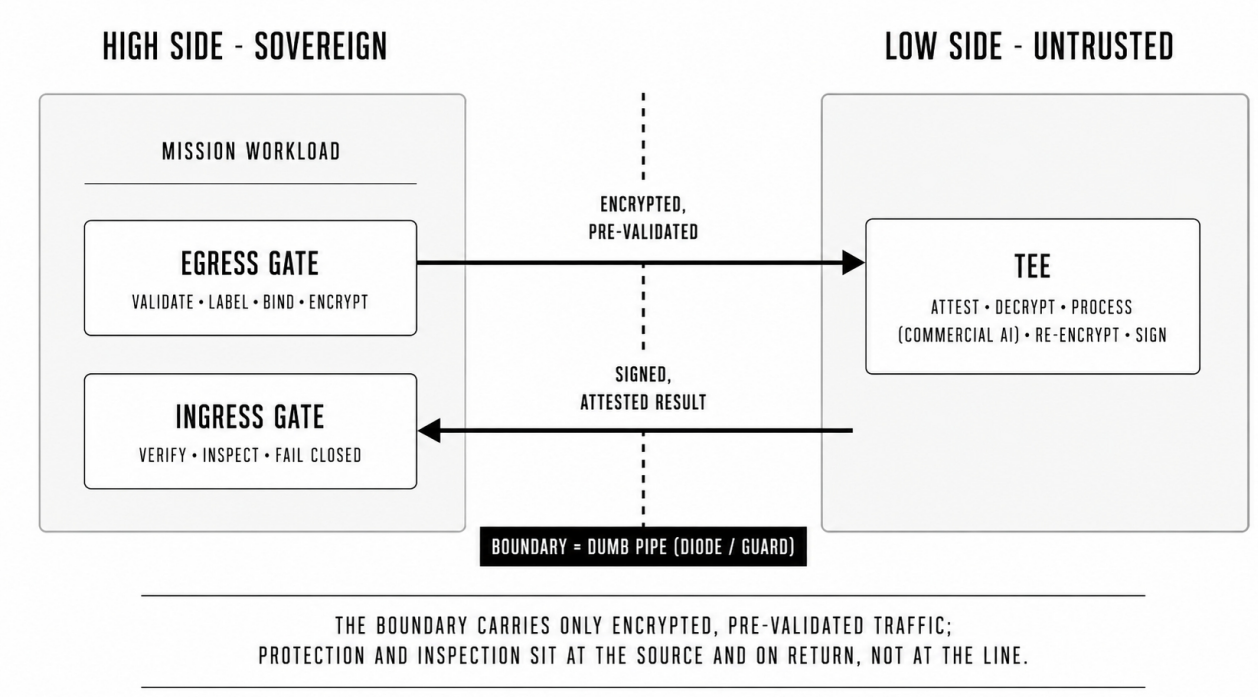
Five forces make the conventional answer — a high-assurance guard performing deep inspection at every boundary — increasingly unequal to the task:

- **Latency.** Deep inspection, transformation, and release at the boundary add a meaningful fraction of a second per crossing. For telemetry, full-motion video, or an AI inference loop that must close in near-real time, a per-hop penalty of that order fragments the decision cycle across the very boundaries the mission needs to cross.
- **An untrusted substrate, not just an untrusted wire.** The low side is no longer the public internet; it is a mission-partner environment or a commercial cloud segment hosting useful capability. The threat is therefore not only interception in transit but the compute platform itself — the host, the hypervisor, the administrator.
- **State and scale.** Stateless guards struggle with the multi-turn interactions modern APIs, replication, and model inference require, and hardware guards do not scale elastically when mission data surges in high- intensity conflict.
- **No shared centre.** In coalition operations there is no common identity provider and no single trust authority; partners may join, drop out, or lose connectivity to a hub at any moment.
- **Data in use, and the quantum clock.** Encryption at rest and in transit leaves the dangerous third state — data in use, exposed in host memory during processing — unaddressed. And traffic recorded today is plaintext tomorrow once a cryptographically relevant quantum computer exists.

The pattern does not abolish the guard. It moves the centre of gravity away from it, so the boundary stops being the place where speed goes to die.

THE PATTERN:

Move security from the perimeter to the payload. Five coordinated moves, each resting on a published standard.



1. DATA-CENTRIC RELEASE

Before anything reaches the boundary, validate the payload against need-to-share policy using attribute-based access control, attach a machine-readable confidentiality label, and bind that label cryptographically to the data so the two cannot be separated downstream. Release is decided at the source, on content, not at the line on packets. The label-and-bind step follows the NATO confidentiality-labelling conventions (STANAG 4774/4778), implemented by carrying the label as authenticated associated data.

2. EPHEMERAL CRYPTOGRAPHY

Use a fresh ephemeral key per transaction via elliptic-curve Diffie–Hellman, authenticated encryption (AES-256-GCM) for confidentiality and integrity together, and signatures for provenance and non-repudiation. The property that matters operationally is perfect forward secrecy: a node captured at the tactical edge yields nothing about the traffic before or after it, because the keys that protected that traffic no longer exist.

3. CONFIDENTIAL COMPUTING

Run the far-side processing inside a trusted execution environment so the untrusted host operates on ciphertext it cannot read. This closes the data-in-use gap and is what makes commercial capability — a large language model, a vision model, burst compute — usable on sensitive data. Treat the enclave as a managed- risk component, not a sealed box: require remote attestation before any data is released to it, keep its code paths minimal, and key every transaction separately so plaintext never lingers.

4. FEDERATED ZERO TRUST

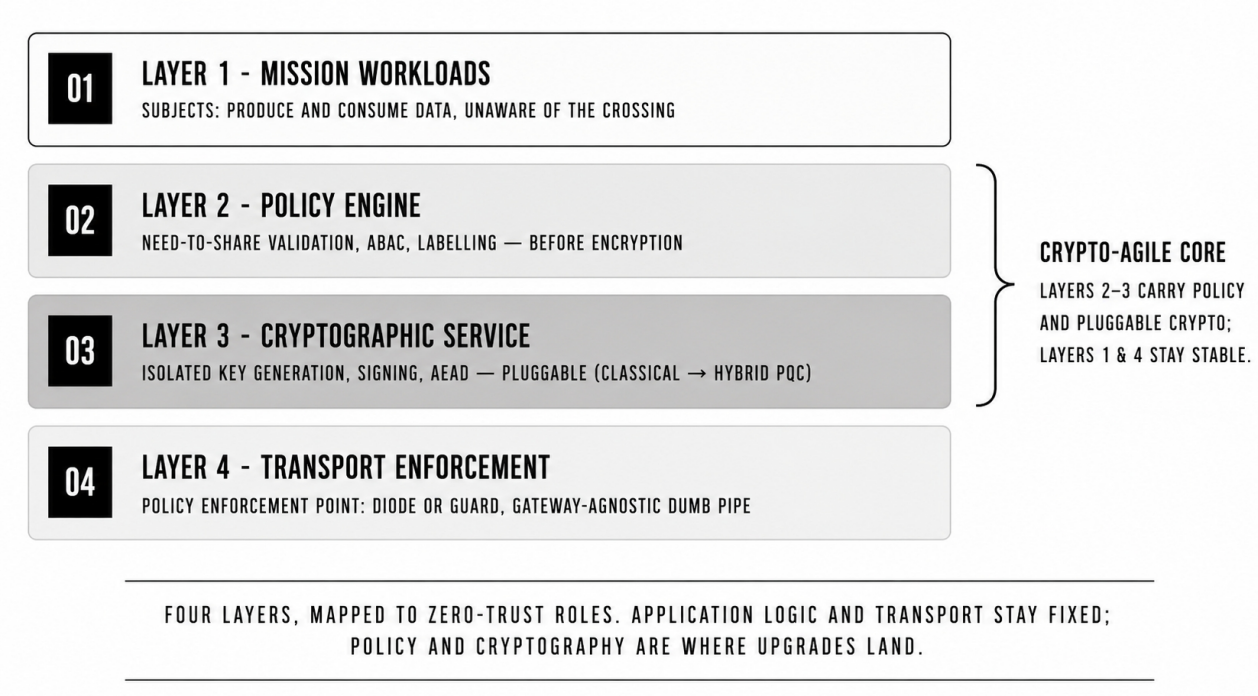
Apply "never trust, always verify" without the usual machinery of a central policy point and a shared directory, because in a coalition neither exists. The high side extends trust to the low side only so far as it can verify a cryptographic signature and a valid attestation against an approved trust anchor and measurement allowlist. Verification failure fails closed. The model degrades gracefully when a partner disconnects, which a directory-dependent model does not.

5. CRYPTO-AGILITY

Keep the cryptographic service architecturally separate from the transport and the application, so migrating to hybrid post-quantum key establishment — classical ECDH combined with a lattice-based mechanism (ML-KEM, FIPS 203), signatures moving to ML-DSA (FIPS 204) where bandwidth allows — is a change in one layer, not a re-engineering of the system. This is the standing answer to "harvest now, decrypt later."

HOW IT IS BUILT:

The pattern separates four responsibilities, mapped to the zero-trust roles in NIST SP 800-207. The separation is the point: it is what makes the design stable where it must be stable and upgradeable where the threat moves.



Defence in depth still applies. Cryptography protects the pipe; inspection protects the domain. The pattern keeps an egress gate that examines plaintext against release policy before encryption, and an ingress gate that treats everything returning from the low side as hostile — checking for malformed structure, unexpected size, or executable content before it touches the high side. The pattern complements accredited cross-domain solutions; it does not replace inspection, release policy, governance, or the security case that high-classification deployment requires.

ASSURANCE, IN THE AUDITOR'S TERMS:

A pattern that cannot be mapped to the assurance framework an accreditor already uses is a pattern that will not be deployed. The design satisfies the UK NCSC cross-domain security principles by cryptographic enforcement rather than perimeter inspection alone. A condensed mapping:

PRINCIPLE	HOW THE PATTERN MEETS IT
Protocol & content attack protection	Protocol-break transport pipe; egress and ingress validation with schema and content checks.
Prevent unauthorised export	ABAC release gates at source; fail-closed defaults on any verification failure.
Session isolation	Per-transaction cryptographic context; no shared session state to subvert.
Persistent-compromise protection	Attestation plus key rotation; revoke a compromised enclave measurement from the allowlist.
Authentication	Mutual authentication; service identity backed by attestation rather than a shared directory.
Data in transit / at rest	Authenticated encryption with per-transaction keys; minimise plaintext persistence.
Audit & accounting	Signed results and policy logs; a protected, non-repudiable audit trail.
Component integrity & patching	Signed updates and measurement allowlists; trusted-base version enforced by attestation policy.

The mapping is the basis for an accreditation conversation, not a substitute for one. It shows the principles are addressed; the accreditor decides whether they are satisfied for a given mission and classification.

WHEN TO USE IT — AND WHEN NOT TO:

A pattern earns its place by being honest about its edges. This one is the right answer in a defined envelope and the wrong answer outside it.

Use it when a low-latency, data-intensive, or AI-driven workflow genuinely needs to cross a classification boundary in near-real time; when the far side offers capability worth reaching but cannot be trusted; and when the operating context is federated — coalition partners, intermittent links, no shared centre — so a directory-dependent model would be brittle.

Do not use it as a substitute for physical separation where the threat model demands a hardware air gap and the latency cost of one is acceptable; where the inspection and release policy cannot be authored correctly, since the system is only ever as sound as that policy; or where the governance to bootstrap and maintain trust between partners — trust anchors, key rotation, revocation, incident response — cannot be sustained.

Three residual risks travel with the pattern and should be carried openly into any security case. The trusted execution environment reduces exposure to a compromised host but does not eliminate it; sophisticated physical attacks remain in scope, so the enclave is a managed risk, not a guarantee. Enclave memory limits may force model quantisation for large models. And the root of trust between partners is a governance problem the architecture enables but does not solve.

What the pattern buys, in one line: cryptographic overhead on the order of a fraction of a millisecond — negligible against tactical links measured in tens to hundreds of milliseconds — stateless operation that survives disrupted and intermittent connectivity without reconnection penalties, commercial capability reached without surrendering the data, and a migration path through the quantum transition. The published prior art cited below demonstrates these properties empirically for one instantiation of the pattern.

AUTHOR:

James Patrick, CITP MBCS

Founder of Nuraghe. Chartered IT Professional, TechUK National Security Committee, former intelligence officer, peer-reviewed NATO author. He assures secure systems for the most demanding classified environments in allied and UK governments, and authored AI 101, now taught across defence from Major to four-star General.

PRIOR ART AND SOURCES:

This pattern is assembled from public standards and established primitives, stated context-blind and vendor- neutral. It is offered as the architecturally sound shape of the answer, not as a product and not as an accreditation argument.

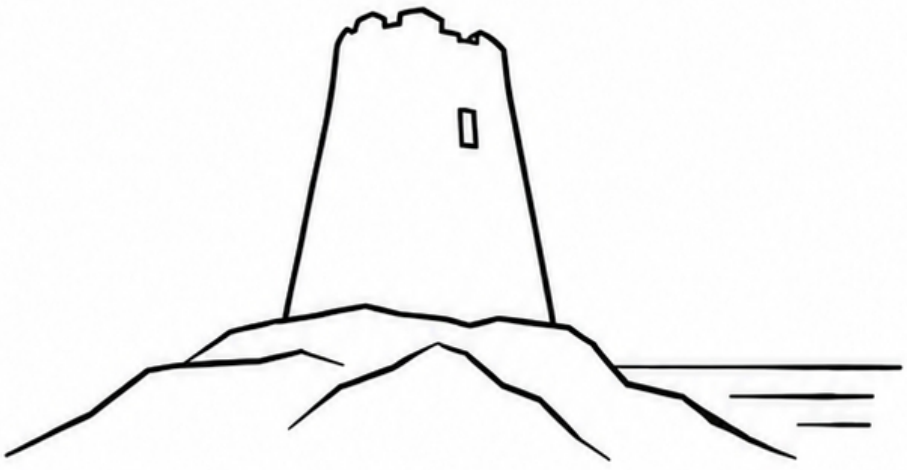
1. Prior art. J. Patrick and M. Galkovsky, "Cross-Domain Hybrid Computing Architecture (CDHCA) for Interoperable Multi-Domain Operations," International Conference on Military Communications and Information Systems (ICMCIS), Bath, 12–13 May 2026. A published treatment of this problem space, including a vendor-specific instantiation and an empirical evaluation of the cryptographic path. Cited here as prior art establishing feasibility; the pattern in this note is stated independently and vendor-neutrally.
2. UK National Cyber Security Centre, Cross Domain Solutions: Security Principles. The thirteen principles to which the assurance mapping refers.
   https://www.ncsc.gov.uk/collection/cross-domain-solutions
3. NIST, Zero Trust Architecture, Special Publication 800-207 (2020). The subject/policy-engine/policy-administrator/policy-enforcement-point roles the four-layer model is mapped to.
   https://csrc.nist.gov/pubs/sp/800/207/final
4. NIST, FIPS 203 (ML-KEM) and FIPS 204 (ML-DSA), 2024 — the post-quantum key-establishment and signature standards behind the crypto-agility move.
   https://csrc.nist.gov/pubs/fips/203/final
   https://csrc.nist.gov/pubs/fips/204/final
5. NIST, SP 800-38D (GCM authenticated encryption) and SP 800-186 (elliptic-curve parameters) — the symmetric and asymmetric primitives.
   https://csrc.nist.gov/pubs/sp/800/38/d/final
6. NATO, STANAG 4774 / 4778 — confidentiality metadata labelling and the binding of labels to data, behind the data-centric release move. And NATO, Data Strategy for the Alliance (2025), for the doctrine of data protected and governed at the source.

Vendor-neutral by construction. The pattern names no provider, no product, and no programme; instantiation choices belong to the integrator and the accreditor.

NURAGHE

(nʊˈrɑːgeɪ)

Sovereign technology strategists
for government and defence.



We create intelligent customers.