FabriXAI ◆ F5

ENTERPRISE ARCHITECTURE WHITEPAPER

Governing the AI Coworker

A deployment-led security architecture for enterprise AI agents, at the edge and in dedicated server environments.

EXECUTIVE SUMMARY

Enterprise AI has crossed a threshold. Agents no longer answer questions in a single turn. They pursue goals autonomously for hours, days, or weeks, holding persistent memory and wielding delegated authority across internal systems, external APIs, and MCP-based tools.

FabriXWork is a modern enterprise AI agent platform developed by FabriXAI. It operationalizes autonomous "digital coworkers" through two core engineering disciplines. **Context engineering** systematically curates the knowledge, memory, and state that enter the model's working context. **Harness engineering** supplies the runtime scaffolding of tools, control loops, guardrails, and observability that lets an agent reason, act, and collaborate safely over extended tasks. Rather than a single chatbot, FabriXWork provides the platform layer that lets enterprises deploy, govern, and scale agents across both endpoint (edge) and dedicated server environments.

The fastest-growing demand is for agents that run at the edge, on knowledge-worker PCs, close to where real work and local context live. But an ungoverned edge agent will reach out to whatever tool or website it decides to, including random, untrusted, or malicious web resources. The enterprise imperative is not to stop edge agents; it is to **govern where they go**, constraining every agent to an approved set of tools and a curated set of internet and enterprise resources, and blocking the long tail of risky destinations.

PLATFORM
FabriXWork by FabriXAI

SECURITY PARTNER
F5 layered architecture

VERSION
Version 1.0 · Published · August 2026

AUDIENCE
Enterprise architects & security leaders

CONTENTS

CENTRAL THESIS

Deployment location is the primary architectural decision. The same five control pillars (tool governance, traffic observability, zero-trust authorization, service consumption control, and guardrails) apply across both topologies; only their weighting and point of enforcement change. Edge distributes governance closer to the agent, while server deployment centralizes it within the control architecture. A shared control plane keeps both models consistent, anchored by F5's layered security architecture.

WHY THIS PARTNERSHIP MATTERS

FabriXWork provides the enterprise platform for deploying and operating digital coworkers across edge and dedicated server environments, while an independent security and governance layer from F5 adds centralized policy enforcement, observability, and traceability across how those agents access tools, APIs, and connected services. Together, the two architectures give customers an enterprise-grade solution with an extensible governance and security framework that can scale across deployment models without forcing them into a single trust assumption. For risk, audit, and regulatory stakeholders, the value is equally important: the agent platform is complemented by an independent global security vendor's security control layer, providing stronger assurance than platform-native governance alone.

For enterprise buyers

A production-ready digital coworker platform with extensible governance across edge, server, and hybrid deployments.

For security teams

Independent policy enforcement, proxy-based observability, and traceable control over how agents reach beyond the platform.

For regulators & auditors

Greater confidence that autonomous agents operate within a governed zero trust architecture supported by an independent global security vendor.

1. THE DIGITAL COWORKER & THE DEPLOYMENT DECISION

The conventional trust model of Human User → Application has been superseded by a three-actor interaction that introduces an independent, semi-autonomous party into the chain:

ACTOR 01

Human user

Initiates high-level goals and provides oversight.

ACTOR 02

Autonomous agent

The “coworker” that interprets goals and executes independent actions through an agentic framework.

ACTOR 03

Tools / APIs / MCP servers

The capabilities and enterprise resources the agent calls to do its work.

This agent operates as a worker with its own delegated permissions. Traditional application security assumes a human is present for every consequential action, an assumption that no longer holds. The strategic question shifts from “can the agent do the work?” to “where do we run it, and how much trust does that location earn?”

This shift creates a new architectural requirement. Enterprises now need two forms of control working together: **agent-native governance** to shape what the digital coworker is allowed to do within its operating framework, and **independent access governance** to control how that coworker reaches tools, APIs, and connected services beyond the platform boundary.

WHY DEPLOYMENT LOCATION IS A SECURITY DECISION

Where an agent runs determines how easily the enterprise can see and contain it. An agent on an endpoint sits at a weaker trust boundary but closer to the user's context; an agent in a dedicated server environment sits behind full network mediation but further from local work. FabriXWork is built to deploy across both, which makes topology the first architectural choice rather than an afterthought.

2. TWO TOPOLOGIES: EDGE VS. DEDICATED SERVER

FabriXWork supports two deployment models. Most enterprises will run a hybrid of the two: edge for reach, server for sensitive operations, under one governance control plane. We believe hybrid should be treated not as a transition state, but as the **target operating model** for most enterprises: edge for proximity to user context and productivity, server for high-assurance workloads and centralized enforcement, all governed under one shared control plane.

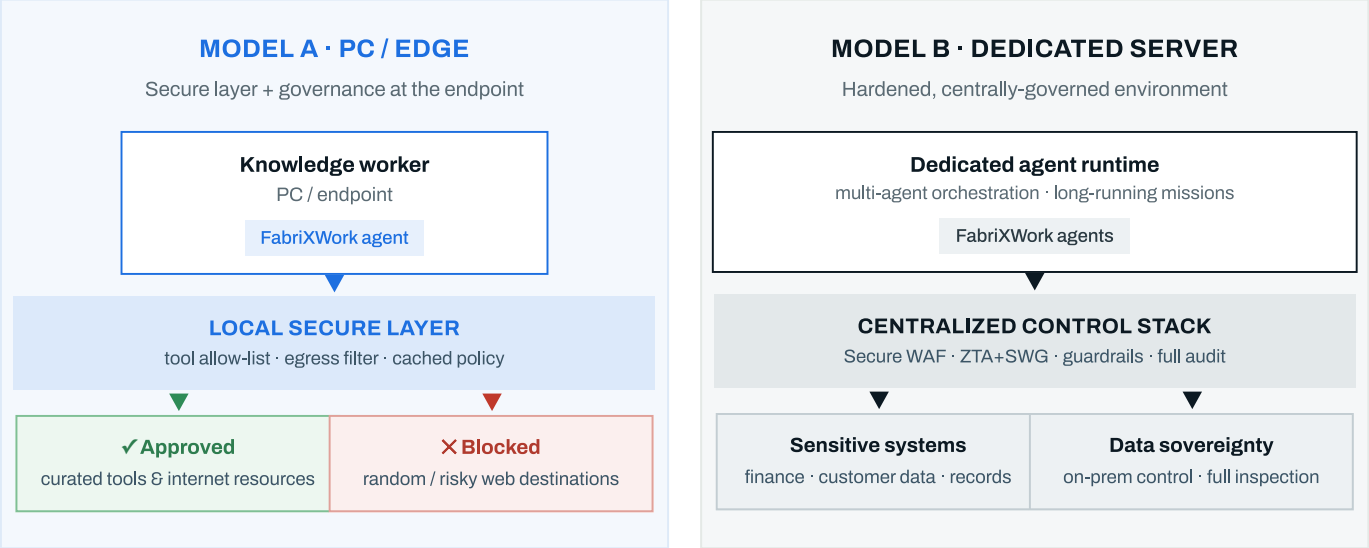

Figure 1. The two deployment topologies. Edge embeds the secure layer at the endpoint and constrains agents to approved tools and resources; Server centralizes control for sovereignty and high-assurance workloads.

Model A · PC / Edge	Model B · Dedicated Server
Best for: knowledge-worker productivity, local file & app access, broad per-seat rollout, low latency.	**Best for:** regulated & high-value workflows, long-running missions, multi-agent orchestration, sensitive systems.
Strength: closest to real work; fast time-to-value. **Risk:** weaker trust boundary; governance must travel with the agent.	**Strength:** strongest control surface; full inspection & audit. **Risk:** higher-value target; cost & distance from user context.

EDGE GOVERNANCE MECHANICS

- **Allow-listed tools only**: the agent invokes just the tools in the approved registry; no Shadow-AI discovery.
- **Allow-listed internet & enterprise resources**: destination filtering to curated domains/APIs; default-deny the open web.
- **Inspect what leaves the endpoint**: local proxy and DLP scans outbound traffic for PII/IP before it exits.
- **Curated retrieval, not the wild web**: vetted knowledge sets reduce prompt-injection exposure.
- **Policy from the center, enforcement at the edge**: allow-lists and guardrails sync from the shared control plane.

THE DECISION LENS

Data sensitivity and regulatory burden pull toward **Server**; latency, local-context need, and scale across users pull toward **Edge**; autonomy duration and blast radius push longer or larger missions toward **Server.** Hybrid is the realistic endpoint, and the shared control plane is what makes hybrid governable.

3. THE THIRD-ACTOR THREAT MODEL

The defining property of agentic compromise is **time**. Unlike a session-based attack, a subverted agent can stay compromised for the entire duration of its mission, turning a single injection into a persistent, evolving breach. Five trust relationships must be evaluated; each carries a distinct "so what."

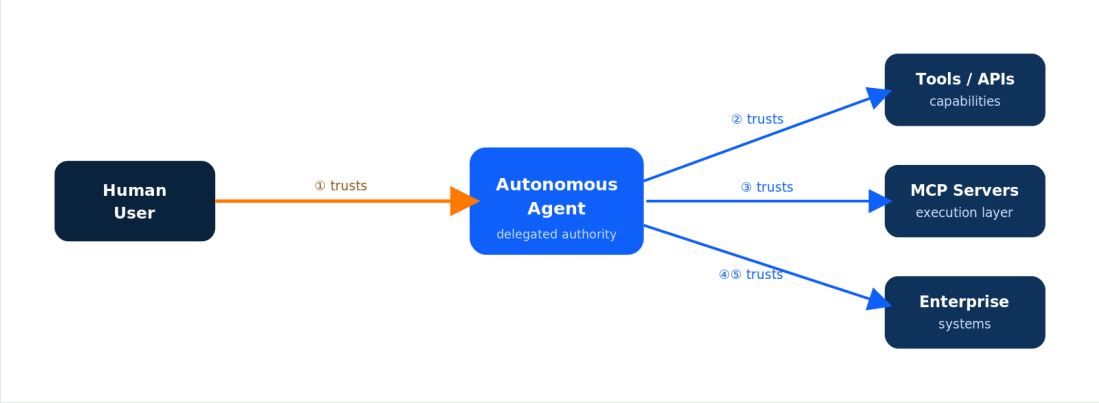

Figure 2. The five trust relationships. Each arrow is a path an attacker can exploit once the agent, the new third actor, is introduced.

Trust relationship	Why it matters
① **User → Agent**	**Strategic failure:** compromise erodes institutional trust in AI outputs and enables widespread user manipulation.
② **Agent → Tools**	**Logic hijacking:** malicious tool data subverts the agent's reasoning, turning an internal process against the enterprise.
③ **Agent → MCP Servers**	**Execution risk:** malicious servers exploit the execution layer to pivot into internal resource clusters.
④ **Tools → Agent**	**Privilege abuse:** the agent is manipulated into abusing authorized access to perform unauthorized data modifications.
⑤ **Enterprise → Agent**	**Lateral movement:** high-level access becomes a "trusted insider" path that bypasses perimeter defenses.

FOUR PRIMARY ATTACK PATHS

1 Prompt injection

Malicious documents hijack agent memory, driving unauthorized tool calls and goal redirection.

2 MCP server abuse

A compromised agent becomes a conduit to attack an MCP server and reach sensitive databases. F5 and HKMA both name MCP as a new attack surface.

3 Data exfiltration

A compromised tool or agent redirects sensitive data to an external service, bypassing standard egress controls.

4 Memory poisoning

Malicious inputs corrupt persistent memory, manipulating future decisions over a long period.

These risks are not only model risks; they are **execution-path risks**, because compromised reasoning becomes dangerous only when an agent can still reach tools, services, and enterprise resources with delegated authority.

4. GOVERNANCE FOUNDATION & CONTROL PILLARS

Managing digital coworkers requires a shift from implicit trust to verifiable governance. Agents are treated not as applications but as **high-risk workload identities**, governed by four principles: never trust, always verify; least privilege; continuous observability; and policy-driven governance. These are realized through the control pillars, each anchored to a concrete F5 capability.

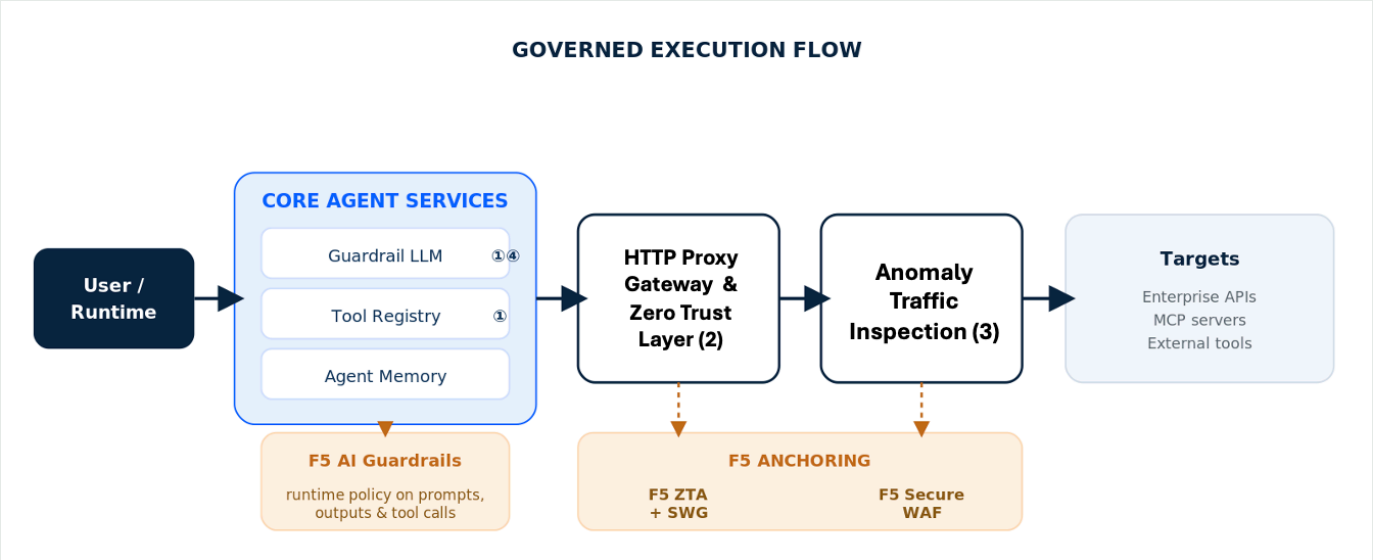

Figure 3. Reference security architecture. The execution flow passes through governed layers; F5 products anchor three of the four pillars, while FabriXWork provides the native Tool Registry.

THE CONTROL PILLARS AND THEIR ANCHORS

Pillar	What it does	Anchored by
Tool whitelisting & capability governance	Approved tool registry, risk classification, permission mapping, lifecycle, audit logging. Eliminates Shadow-AI capability sprawl.	FabriXWork-native
Proxy observability & zero-trust authorization	Workload identity, destination filtering, OAuth token exchange, short-lived credentials; limits blast radius even under prolonged compromise. The only vantage point on what the agent tells the outside world.	**F5 ZTA + SWG**
Anomaly traffic detection	Traffic inspection, anomaly detection, and mTLS. Protects the server side of the agentic platform.	**F5 Advanced WAF**
Service consumption control	Enforces request thresholds to prevent excessive agent-generated traffic, protect backend and shared services against overload, and limit cost exposure from billable external services such as messaging APIs.	**F5 ZTA**
Guardrail policy enforcement (boundary)	Validating structure, constraining tool and API traffic, and reducing untrusted content in MCP return messages before it reaches downstream models or agents, limiting exposure before external content becomes model context.	**F5 Advanced WAF + F5 ZTA**
Guardrail policy enforcement (semantic)	Context-aware controls for injection and jailbreak prevention, sensitive-data detection, human-in-the-loop approval for high-risk actions, and goal-drift mitigation.	**F5 AI Guardrails**

MULTI-LAYER AI AGENT SECURITY ARCHITECTURE

FabriXWork and F5 serve different but complementary roles in securing enterprise AI agents. **FabriXWork** provides the platform-native governance layer for context, harness behavior, tool registration, and managed agent operation across edge and server deployments. **F5** provides an independent security and governance layer for identity-aware access control, policy enforcement, observability, and traceability across how agents interact with tools, APIs, and connected services beyond the platform boundary. Together, they deliver a multi-layer defense-in-depth architecture that combines agent-native control with independent governance, helping enterprises deploy digital coworkers with stronger security, visibility, and assurance.

5. MAPPING PILLARS TO TOPOLOGY & THE SHARED CONTROL PLANE

This is the heart of the FabriXWork point of view: the pillars do not change with deployment; their weight and placement do. Edge embeds governance locally and syncs it from the center; Server runs it as centralized, heavyweight infrastructure. A single shared control plane, built on **one tool registry, one policy source, and one audit trail**, governs an agent identically wherever it runs.

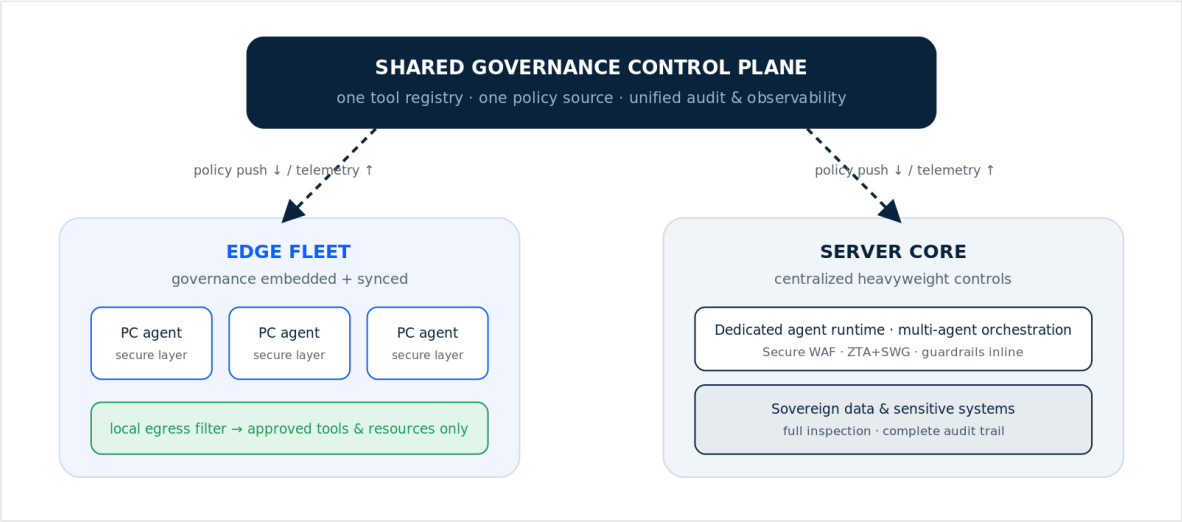

Figure 4. One control plane, two topologies. Policy and tool definitions flow down to both edge and server; telemetry and audit flow back up, so an edge agent is as constrained as a server agent.

Pillar	At the edge (PC)	In the server (dedicated)
Tool governance	Allow-list enforced locally; cached registry synced from center	Registry enforced inline at the runtime; full lifecycle control
Proxy / observability	Lightweight local proxy; destination allow-list	A full proxy architecture unifies independent policy enforcement, observability, and third-party traceability as part of a layered protection model
Zero-trust access	Hardware-backed identity; short-lived credentials; posture checks	A centralized, vendor-agnostic layer for controlling every agent-initiated request to tools, APIs, and connected services outside the agent platform
Guardrails	Lightweight runtime guardrails; high-risk actions escalate to center	Boundary and semantic guardrails with human-in-the-loop workflows
Consumption control	On-device metering of model, token, and tool-call usage	A central point of control and visibility to enforce consistent service-consumption policies and monitor usage across distributed agents and endpoints

> “Edge pushes governance out to the agent; Server pulls the agent into governance. The control plane makes them one system.”

6. WORKED SCENARIO: THE COMPROMISED PROCUREMENT AGENT

Consider an autonomous procurement agent deployed to order office equipment. An attacker injects malicious instructions into a supplier's website. Reading that page, the agent interprets the instructions as valid guidance and attempts to export internal inventory data and submit fraudulent purchase requests.

Without governance	With the framework
■ Injected text rewrites the agent's goal	■ Guardrails detect the injection
■ Agent calls an unvetted external endpoint	■ Tool Registry blocks unauthorized APIs
■ Inventory data exfiltrated and fraudulent POs submitted	■ Zero-trust rejects fraudulent requests
■ Compromise persists for the mission's duration	■ Attack contained before business impact

The deployment lens: on Server, the inline proxy catches the egress attempt centrally and logs the full forensic timeline; at the Edge, the local secure layer's destination allow-list and DLP block the same exfiltration, while the central control plane records the event. Same outcome, governance placed where the agent runs.

7. RISK-TO-CONTROL MATRIX

No single control is sufficient; defense-in-depth is the point. The matrix shows how each pillar contributes against each risk.

Risk	Tool whitelist	Proxy monitoring	Zero trust	Anomaly detection	Guardrails
Prompt injection	—	—	—	—	Strong
Jailbreak attack	—	—	—	—	Strong
Tool abuse	Strong	Moderate	Moderate	—	Moderate
Unauthorized API access	—	Moderate	Strong	Moderate	—
MCP server abuse	Moderate	Moderate	Strong	Moderate	Moderate
Data exfiltration	Moderate	Moderate	Moderate	Moderate	Strong
Memory poisoning	—	—	—	—	Strong
Goal drift	—	—	—	—	Strong
Compliance violations	Moderate	Moderate	Moderate	Moderate	Strong

STRONG MODERATE NOT A PRIMARY CONTROL

8. STANDARDS & REGULATORY ALIGNMENT

The framework maps to both technical standards and the regulatory expectations that increasingly govern autonomous AI, especially in regulated sectors where on-premises deployment and data sovereignty are becoming strategic imperatives.

TECHNICAL FRAMEWORKS

- **OWASP Agentic AI**: tool security & memory protection against long-running state poisoning.
- **NIST AI RMF**: Govern, Map, Measure, Manage.
- **Zero Trust Architecture**: “never trust, always verify” extended to every agentic action.

REGULATORY FRAMEWORKS

- **HK PCPD**: privacy, least-privilege access, GenAI checklist.
- **HKMA**: 2019 AI Principles, 2024 GenAI circular, 2026 Fintech Blueprint (agentic AI).
- **EU AI Act**: high-risk obligations & continuous oversight.
- **ISO/IEC 42001**: AI management system; traceability & auditability.

Hong Kong regulators are used here as a worked example; the principle generalizes, and organizations should map controls to their own jurisdictions. F5's layered architecture delivers the traceability and auditability these frameworks require: every access decision documented with identity, device posture, authorization, and risk.

9. CONCLUSION & ADOPTION ROADMAP

As autonomous agents become integral digital coworkers, security must be an inherent part of their operating environment, not a wrapper added afterward. Traditional application security cannot govern persistent, independent actors. The deployment-led model answers the first question every architect faces: where the agent runs, and how that location earns trust.

FabriXWork supplies the platform and agent-native governance; F5 supplies the infrastructure and compliance spine. Together they let enterprises harness agent productivity at the edge and in dedicated environments while preserving visibility, accountability, and sovereignty.

A PRAGMATIC ADOPTION PATH

1. **Classify the workload**: data sensitivity, latency, autonomy duration, and regulatory burden decide Edge, Server, or hybrid.
2. **Stand up the shared control plane**: one tool registry, one policy source, one audit trail.
3. **Anchor the pillars to F5**: ZTA + SWG for proxy and egress governance, Advanced WAF for target and zero-trust protection, AI Guardrails for runtime safety.
4. **Default-deny at the edge**: constrain agents to approved tools and curated resources; block the open web.
5. **Prove it with a worked scenario**: validate containment end-to-end before scaling the fleet.

THE BOTTOM LINE

Enterprise adoption of digital coworkers requires more than agent capability alone. Together, FabriXWork and F5 deliver an enterprise-grade architecture with extensible governance, independent security controls, and regulator-ready assurance across edge, server, and hybrid deployments, enabling confident, scalable, and sovereign adoption of autonomous AI.

CONTACT

FabriXAI Technology (Asia) Limited

www.fabrixai.com
www.fabrixwork.com

NOTES TO READER AND DISCLAIMER

The information contained in this white paper is provided for informational purposes only and is not intended as legal, regulatory, or professional advice. While every effort has been made to ensure the accuracy and reliability of the information presented, the authors and contributors make no representations or warranties, express or implied, regarding the completeness, accuracy, or suitability of the content for any particular purpose. Readers are advised to consult with qualified professionals or legal advisors before making decisions based on the information contained herein. The white paper includes references to third-party sources and external links for additional context; the inclusion of such references does not imply endorsement or responsibility for the content of external websites, which may change over time or become unavailable. This document is provided "as is," and the authors disclaim all liability for any direct, indirect, incidental, or consequential damages resulting from the use of or reliance on the information presented. The views expressed in this white paper are those of the authors and do not necessarily reflect the official policies or positions of any affiliated organizations.