

WHITE PAPER

Architectural Velocity vs. Agentic Threats in Banking

Redefining Application Landscapes Under Automated Vulnerability Discovery

Contents

03
Introduction

05
How Frontier AI Has Changed Vulnerability Discovery

08
Tactical Integration: The AI-Native Software Development Lifecycle

10
Strategic Architecture Management: The Application Landscape Matrix

16
Managing AI Complexity Across the Landscape

17
The BYOLLM Principle

18
From Assessment to Action

19
About the Authors

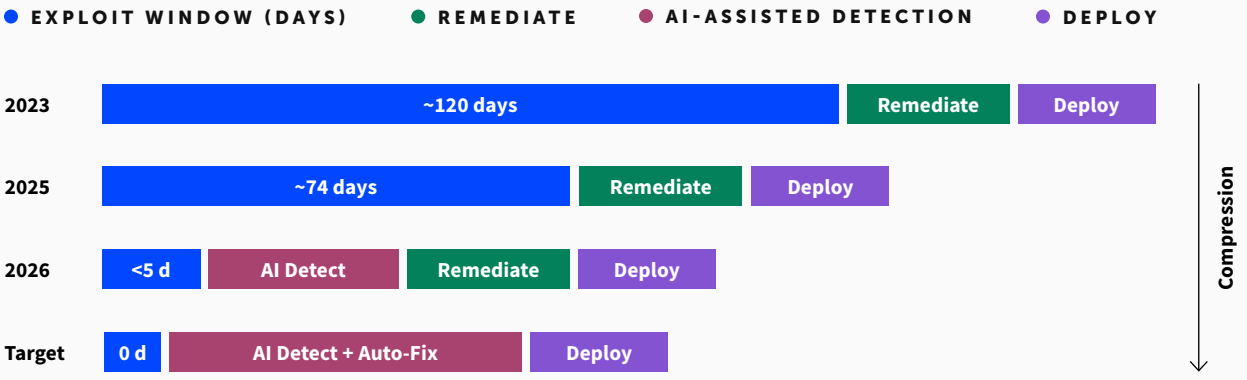
20
About EPAM

Introduction

LARS FETZER
Director, Advisory Financial Services, EPAM
Connect

DENNIS JOOSTEN
Head of Banking Practice, EPAM
Connect

The window between vulnerability discovery and exploitation has collapsed. As recently as 2018, defenders could count on roughly two months between public disclosure and active exploitation. By 2025, Mandiant's mean time-to-exploit estimate had turned negative: exploitation now routinely precedes patch availability. The average time to remediate a critical application vulnerability, meanwhile, remains around 74 days, **according to Edgescan's 2025 report**. No amount of process optimization can bridge this gap. The constraint isn't operational; it's structural. Architecture correction is not a discretionary improvement. It is the precondition for survival-grade deployment speed.



Frontier AI models have accelerated this compression in ways that demand a strategic response. Current-generation systems autonomously discover zero-day vulnerabilities in major operating systems and browsers, reverse-engineer closed-source software and generate working exploits at a cost of tens of dollars per finding. These capabilities aren't confined to a single provider. Rather, they're an emergent property of advancing code reasoning and will proliferate across the financial services industry within 12 to 18 months.

For European banks, the convergence of AI-accelerated threats with the Digital Operational Resilience Act (DORA) creates a response imperative on three fronts:

01

OPERATIONAL

Remediating existing vulnerabilities within regulatory timelines

02

TACTICAL

Embedding AI-assisted security into the software development lifecycle

03

STRATEGIC

Reassessing the application architecture itself

In addition, a fourth front is rapidly emerging concerning the governance of embedded AI in commercial software, where the stakes are rising fast and governance frameworks haven't kept pace.

	→ Operational	→ Tactical	→ Strategic
Objective	Remediate known vulnerabilities	Embed AI security into SDLC	Restructure the application landscape
Time horizon	Weeks to months	Quarters	1–3 years
Stakeholders	CISO, Security Ops	CTO, Engineering Lead	CTO, COO, Board
Velocity impact	Reactive; one-time fix	Continuous; integrated into flow	Structural; removes friction at source
Risk of inaction	Regulatory non-compliance	Accumulating exposure	Systemic architectural failure

How Frontier AI Has Changed Vulnerability Discovery

A Qualitative Shift in Capability

The trajectory of large language model (LLM) capabilities in cybersecurity has been steep. In 2023, frontier models could barely complete beginner-level capture-the-flag challenges. By early 2026, a qualitative leap had occurred: **Models began discovering and exploiting vulnerabilities** previously unknown to human researchers.

Anthropic's Claude Mythos Preview illustrates the scale of this shift. The **model identified a 27-year-old vulnerability** in OpenBSD and wrote a remote code execution exploit for FreeBSD's NFS server that chains six separate remote procedure call (RPC) requests to achieve root access without authentication. In evaluations by the U.K. AI Security Institute,

Mythos Preview **succeeded on expert-level cybersecurity tasks** 73% of the time. These are tasks no model could complete before April 2025.

This represents a categorical shift from pattern-matching tools that scan against known vulnerability signatures. Mythos Preview reasons semantically about code, understands business logic and data flows, hypothesizes where vulnerabilities may exist, and **writes proof-of-concept exploits to confirm**. Crucially, Anthropic's red team demonstrated the model could reconstruct plausible source code for closed-source software. In other words, proprietary code is no longer protected by obscurity.

The Economics of AI-Powered Attacks

Anthropic ran approximately 1,000 automated scanning sessions across various software targets at a **total cost of under $20,000**, yielding several dozen confirmed high-severity findings. The specific run that surfaced the OpenBSD bug cost under $50. The economics work like a search process: Most runs return nothing, but the marginal cost per attempt is negligible. What previously required an elite human researcher working for weeks can now be replicated by an automated agent in hours.

A **RAND Corporation analysis** concludes that attackers currently hold the advantage and AI favors defenders only if deliberately acted upon. In other words, passive defense postures that rely on perimeter security and periodic patching are structurally outmatched.

The Geopolitical Context

In June 2026, the U.S. administration issued an executive order requesting that AI developers voluntarily share advanced models with the government up to 30 days before public release. An earlier draft had proposed a 90-day review window. The reduction to 30 days reflects the political tension between security oversight and innovation speed. The framework is voluntary, and the 30-day access window precedes a model's release to the developer's wider set of trusted partners.

The consequence is nonetheless clear: under this arrangement, the U.S. government may have as little as a month to assess a frontier model for dangerous capabilities before it reaches the broader market. In a parallel development a month earlier, in May 2026, the U.S. Department of War announced agreements with eight leading AI companies to deploy AI on classified military networks, signaling that states are preparing to operationalize the same capabilities this paper describes.

The Threat Clock Is Outpacing the Patch Cycle

Mandiant's time-to-exploit data captures the trend in stark terms: an average of **five days in 2023**, and an estimated mean of roughly **negative seven days for 2025**. **Edgescan's 2025 report** measures mean time to remediate at 74.3 days, with 45.4% of enterprise vulnerabilities remaining unpatched after 12 months.

Exploitation now precedes the patch; remediation still takes weeks. This is not a metrics problem – it is a structural mismatch between threat velocity and institutional response capacity.

The cost of failing to close that gap isn't theoretical. In November 2023, ICBC's U.S. broker-dealer arm was **hit by LockBit ransomware** that entered through an unpatched Citrix NetScaler appliance carrying a known, already-disclosed vulnerability. The intrusion paralyzed ICBC's ability to clear U.S. Treasury trades, with roughly **$62 billion in trades** failing to settle in a single day. The vulnerability had a patch available; the binding constraint was the time taken to apply it.

Under DORA, a documented vulnerability left unremedied isn't merely an operational risk. It's a standalone compliance failure.

Operational Response: The Regulatory Imperative

DORA leaves no room for ambiguity. Once a vulnerability is identified and documented, the institution must remediate it within defined timelines. All documented vulnerabilities must be captured within the Information and Communication Technology (ICT) risk management framework. For major ICT-related incidents, initial notification is due within four hours of classifying the incident as major and no later than 24 hours after detection. Competent authorities can demand specific remediation actions, with penalties including fines, capital add-ons and public censure of executives.

The regulatory clock starts ticking when a vulnerability enters the risk register.

The question is not whether to act, but whether the organization can act fast enough.

Enforcement precedents from analogous regimes signal where DORA supervision is headed. In December 2022, the U.K.'s Financial Conduct Authority and Prudential Regulation Authority **fined TSB Bank £48.65 million** for operational risk-management failures arising from its 2018 IT migration. In Singapore, following repeated digital-banking outages, the **Monetary Authority of Singapore** raised DBS Bank's operational-risk capital multiplier to 1.8 times risk-weighted assets, resulting in roughly S$1.6 billion in additional regulatory capital and a six-month freeze on non-essential IT changes. Neither action was taken under DORA, but both signal the supervisory direction DORA now codifies for EU institutions.

The Path From Manual Patching to Automated Remediation

The vision of fully automated vulnerability remediation, where agentic AI systems autonomously detect, triage, fix, test, and deploy patches without human intervention, is technically within reach. Proof-of-concept implementations exist in which autonomous agent swarms orchestrate the entire cycle.

However, most institutions today operate with human-in-the-loop models in which AI assists detection and proposes fixes, but human engineers validate and deploy. The reasons are structural: regulatory requirements for auditability, legacy codebases that resist automated intervention and the nascent maturity of agentic AI in production banking environments.

Tools like **GitHub's Copilot Autofix** illustrate the current state of the art. This particular tool pairs static analysis with an LLM; when a vulnerability is detected, it generates a candidate fix and opens a pull request with a natural-language explanation for an engineer to review and merge.

GitHub reports remediation of over two-thirds of findings with little or no editing and a median time-to-commit of roughly 28 minutes. The engineer remains accountable for the merge, which is precisely the auditable posture DORA and banking risk functions require today.

The Run-the-Bank / Change-the-Bank Squeeze

As AI-discovered vulnerabilities increase in volume, the share of engineering capacity consumed by remediation rises correspondingly, squeezing the budget available for strategic Change-the-Bank initiatives. Teams that should be building new capabilities get locked into patching cycles. Without embedding security into the development process itself, the bank remains permanently reactive.

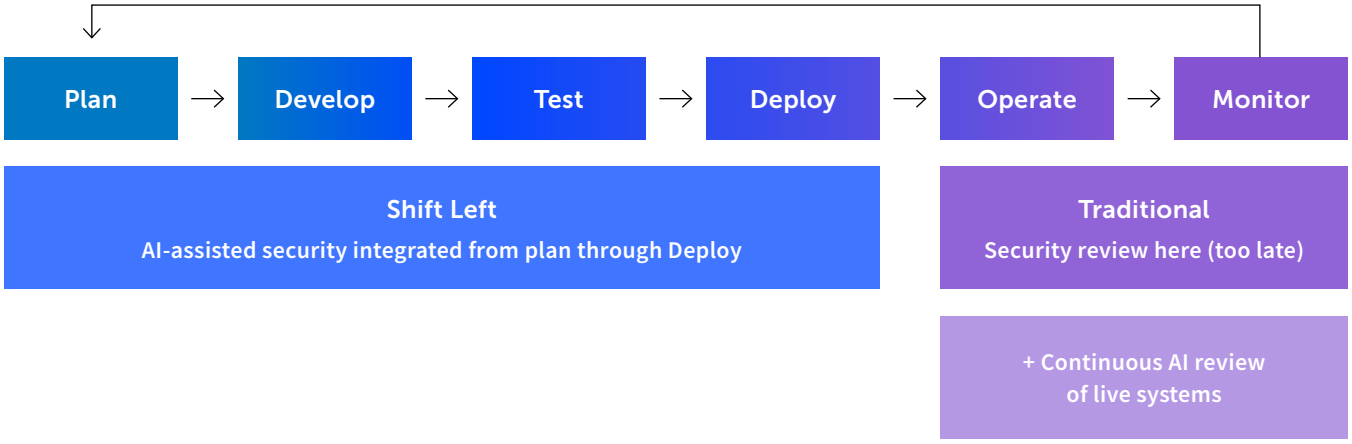
This does not diminish the urgency. It sharpens it.

Banks cannot and should not wait for fully autonomous remediation to mature before addressing the vulnerability backlog. They must act now with the tools available while simultaneously building the operational foundations for the next generation of automated defense.



Tactical Integration: The AI-Native Software Development Lifecycle

The second horizon shifts from remediation to prevention. The objective is to adapt the software development lifecycle (SDLC) so that security reviews keep pace with the threat environment, compressing the time from identifying a critical vulnerability to remediating it from weeks to under 24 hours.



Shift Left & Continuous Review

The concept of shifting security earlier in the development lifecycle, into design, coding and testing rather than post-deployment audit, is codified in the NIST Secure Software Development Framework (SP 800–218). What has changed is the urgency.

When a codebase that passed review six months ago may contain vulnerabilities only newer models can detect, the review cadence must be driven by the evolving capability frontier of analysis tools, not by the release calendar.

Shift left means AI-assisted code review early in the development process, automated vulnerability scanning during continuous integration and architecture-level threat modelling during design. Under DORA, these are risk management controls, not optional best practices.

In addition, existing codebases must be re-scanned regularly with current-generation models. Assurance must be continuous rather than periodic.

A vulnerability that no tool could find in January may be trivially discoverable by March. The cadence of assurance has to match the cadence of change.

Agentic AI as a Velocity Enabler

Without automation, continuous security review across a large application estate is prohibitive. Agentic AI makes these practices sustainable at scale through:

01

PRE-DEPLOYMENT GATES

Agents that analyze every code change for vulnerability patterns before it enters the release pipeline, flagging issues that require human review and auto-resolving trivial findings.

02

EXPLOITABILITY TRIAGE

Agents that assess each finding against the bank's actual configuration and reachability, determining whether a given vulnerability (a Common Vulnerabilities and Exposures, or CVE, entry) is genuinely exploitable rather than merely present. This turns raw detection signals into prioritized action.

03

POST-DEPLOYMENT MONITORING

Agents that continuously scan production against the latest model-derived vulnerability insights, triggering alerts when new exposures are detected.

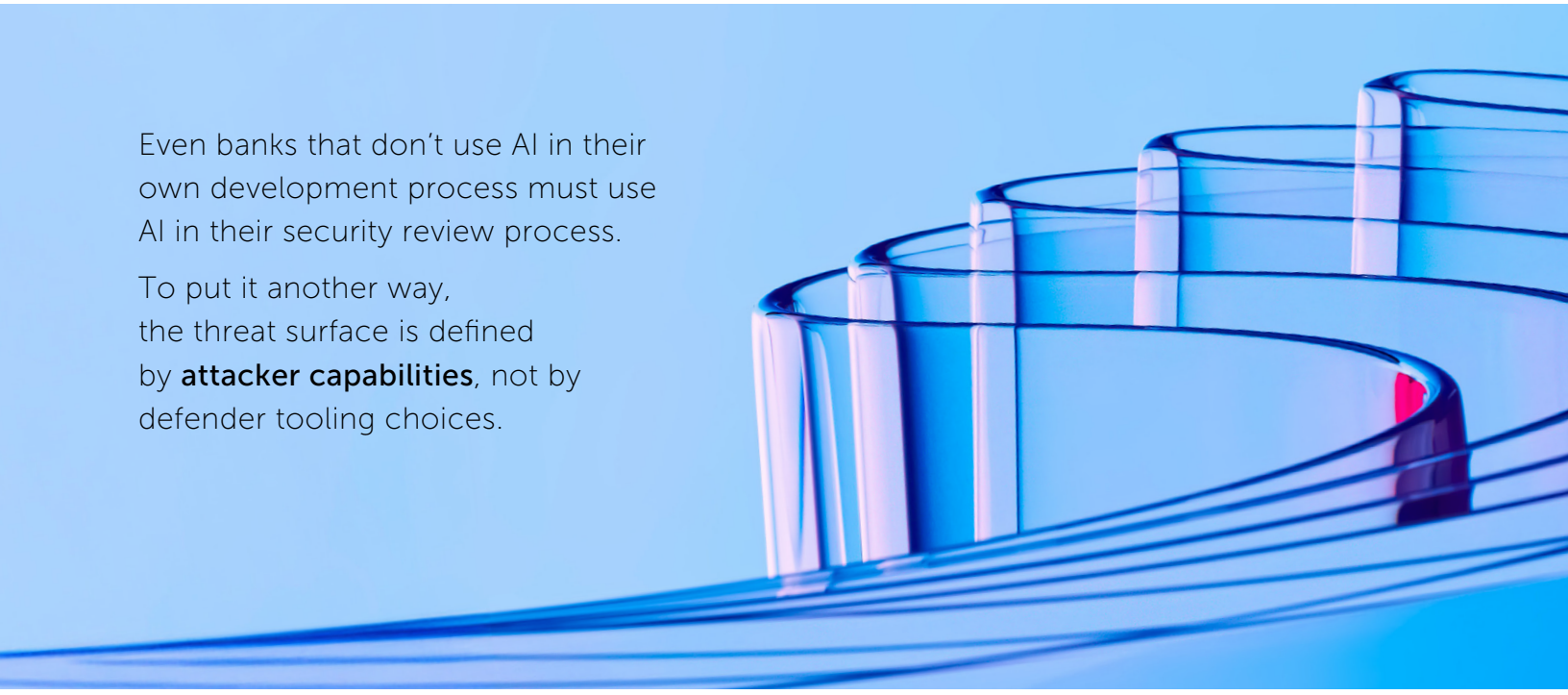
04

REMEDIATION ASSISTANCE

Agents that identify a vulnerability and propose a specific fix with test cases as a reviewable pull request, reducing the time from detection to resolution.

Even banks that don't use AI in their own development process must use AI in their security review process.

To put it another way, the threat surface is defined by **attacker capabilities**, not by defender tooling choices.



Strategic Architecture Management: The Application Landscape Matrix

Tactical and operational responses address vulnerabilities within the existing architecture. The strategic horizon asks a different question: Is the architecture itself capable of sustaining the deployment velocity the threat environment demands? To reach that determination, EPAM's banking practice employs a proprietary matrix framework designed to generate a score, referred to as ecosystem distance, that serves as a prioritization signal for the real-world fragility of the bank's architecture. Here's how it's calculated.

The Matrix Framework

EPAM's Application Landscape Matrix classifies every application along two dimensions on a nine-by-nine grid.

The X axis measures technology maturity:

Legacy: Vendor investment has stopped, developer talent is scarce and patches are slow to arrive. Every fix takes longer and costs more.

Contemporary: Well-supported, widely used technology, but still built as large, tightly bundled systems that are hard to change incrementally. Changes are safe but slow.

Modern: Cloud-native, actively evolving technology with fast, frequent patch cycles. Fixes can ship in days, not months.

The Y axis measures development paradigm:

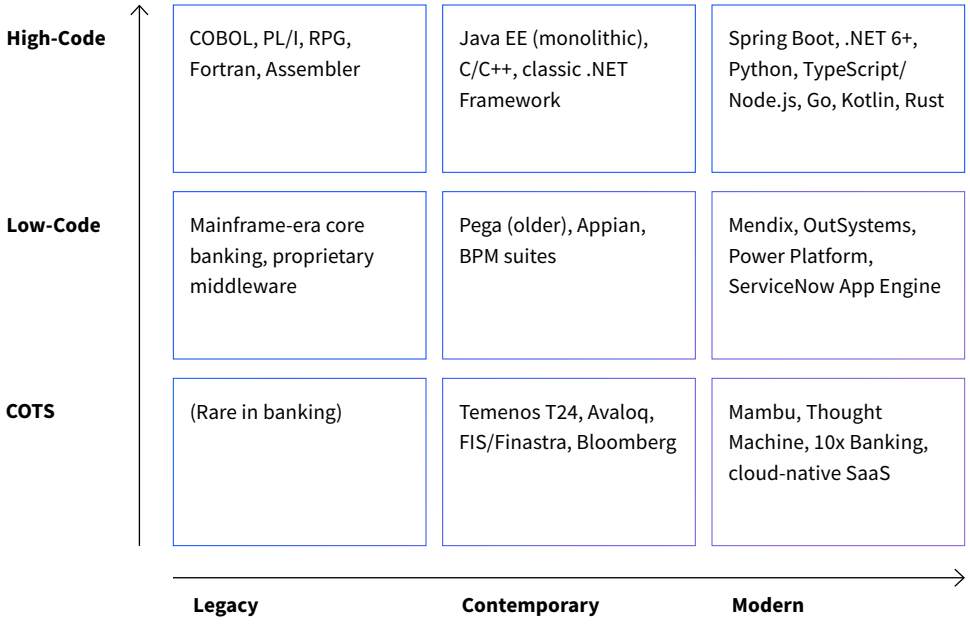
Commercial Off-the-Shelf (COTS): The bank buys and configures it but relies entirely on the vendor to patch it. The bank's speed is the vendor's speed.

Low-Code: Built quickly on a platform but limited by what that platform allows. Faster to build, but only as flexible as the vendor's roadmap.

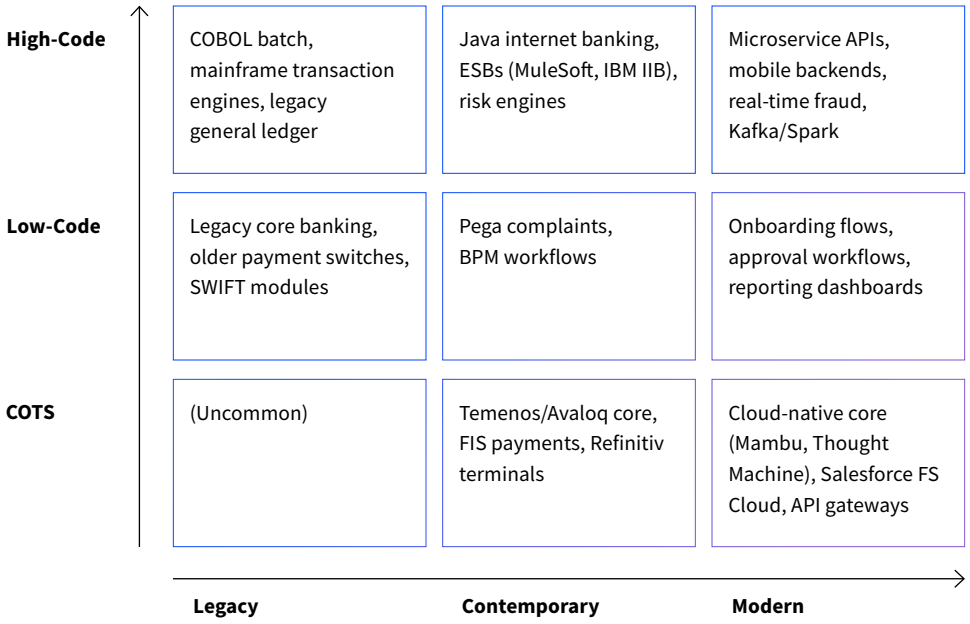
High-Code (Custom Development): Built in-house, with full flexibility and full responsibility for keeping it secure and maintained.

These two axes aren't arbitrary. Technology maturity moves from legacy to modern; development paradigm moves from vendor-controlled to bank-controlled. An institution can have two custom-built systems at opposite ends of the maturity axis, or a modern off-the-shelf product sitting next to a modern custom-built one on maturity but far apart on control. Every application gets a position on each axis independently.

EXEMPLARY PROGRAMMING LANGUAGES BY QUADRANT



EXEMPLARY BANKING APPLICATIONS BY QUADRANT



Measuring Ecosystem Distance

The strategic value of the matrix emerges when applications are evaluated not in isolation but as ecosystems: groups of applications that work together to deliver a business function.

Step 01

CLASSIFY

Assign each application a coordinate pair (x, y) on the nine-by-nine grid based on technology maturity and development paradigm.

Step 02

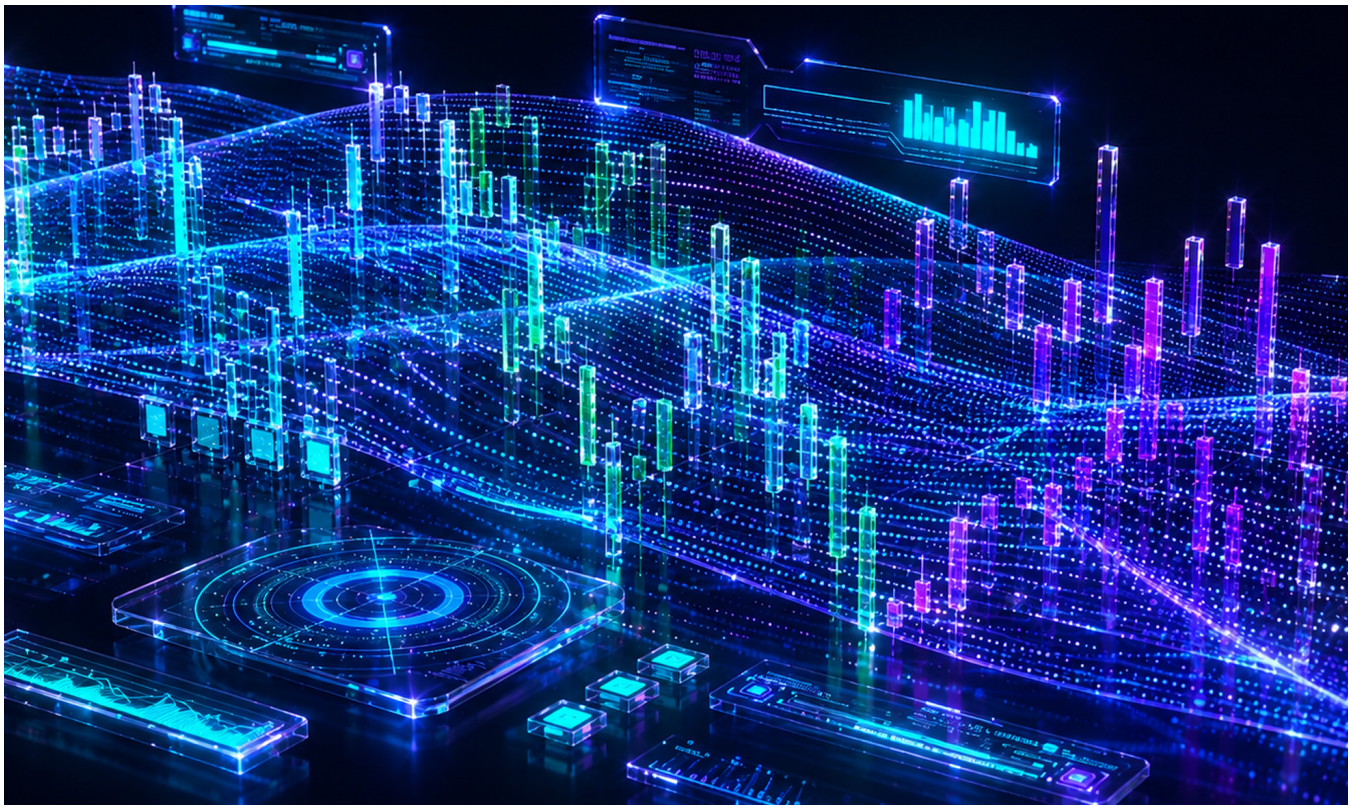
MAP ECOSYSTEMS

Identify functional ecosystems through dependency analysis: shared data stores, API calls, event subscriptions, batch interfaces and organizational ownership. An isolated legacy application with no coupling to other systems warrants monitoring but doesn't create inter-component friction.

Step 03

CALCULATE MAXIMUM PAIRWISE DISTANCE

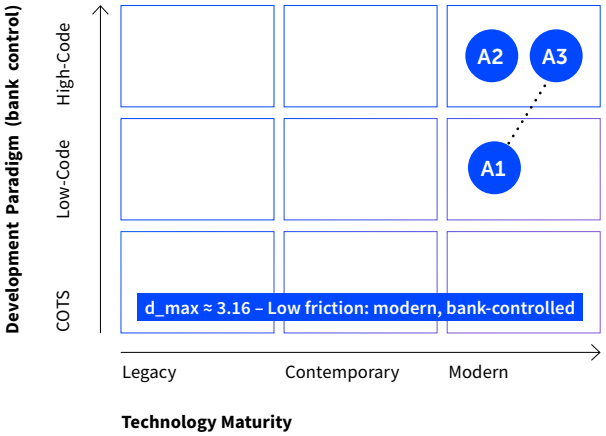
For each ecosystem, compute the Euclidean distance between every pair of components and select the maximum: d-max(E) = max sqrt[(xi-xj)squared + (yi-yj)squared]. This approach is preferred over centroid or variance methods because, in the era of AI-driven exploits, security and velocity are bound by the weakest component.



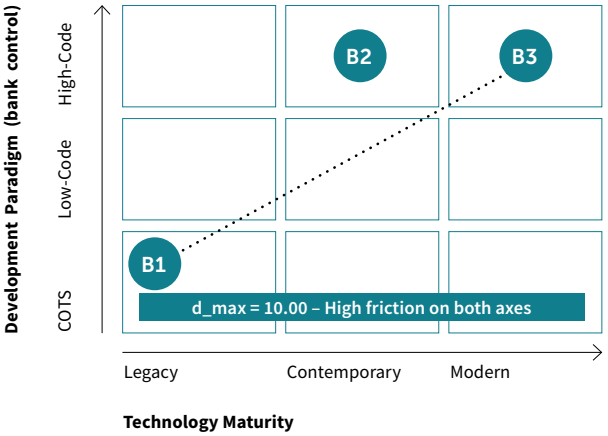
ARCHETYPICAL DISTANCE COMPARISON[1]

Archetypical distance comparison for digital challenger vs. traditional banks

Archetype A: Digital Challenger



Archetype B: Traditional Universal Bank



The Matrix in Motion

Archetype A

DIGITAL CHALLENGER

Microservices backend, API layer and internal workflows all sit close together on the matrix. Modern technology, largely built and controlled by the bank. The result is a homogeneous ecosystem with low friction; patches move fast because there's no weak link dragging on the rest.

Archetype B

TRADITIONAL UNIVERSAL BANK

The mobile app is modern and bank-controlled, but it ultimately depends on integration middleware and, beneath that, a legacy core banking system the bank doesn't fully control. That legacy core sits at the opposite end of the matrix from everything else. This gap between the newest and oldest components drives friction up on both axes simultaneously.

[1]Coordinates mapped to a 9x9 grid; axes ordered independently. On this grid, one step between adjacent cells corresponds to a distance of 1; the maximum possible distance across the full grid is √128 ≈ 11.3. Bubble positions follow the 9×9 coordinates; the cells shown are aggregated.

High ecosystem distance translates directly into real-world fragility. The Knight Capital collapse of August 2012 offers a stark illustration. Knight deployed new order-routing code to its automated trading platform, but the update reached only seven of its eight servers.

The eighth still carried a dormant test routine from 2003 that a repurposed configuration flag silently reactivated. For roughly 45 minutes, that single divergent component **fired millions of erroneous orders** into the market, executing over four million trades across 154 stocks and generating initially reported losses of approximately $440 million. The firm was acquired in a fire sale within days. In a tightly coupled ecosystem operating at machine speed, the most divergent component determined the fate of the whole. This is the Archetype B pattern in practice: the weakest, most divergent component determines the resilience of the entire ecosystem.

The Impact of High Ecosystem Distance

Where the distance metric identifies high friction, the consequences are concrete, not theoretical. A wide gap between a portfolio's oldest and newest components plays out in four ways: who is left to maintain it, how fast it can move, how exposed it is to AI-driven attack, and how much a single slow component ends up constraining everything connected to it:



TALENT

The average COBOL developer is over 55; approximately 10% retire annually. A vendor-commissioned survey by Micro Focus/OpenText found that 60% of organizations using COBOL identify finding qualified developers as their biggest challenge. Systems that cannot attract maintainers accumulate unresolved vulnerabilities.



VELOCITY

A modern microservice with automated CI/CD ships a patch within hours. A legacy mainframe with quarterly releases and manual regression testing takes months. When both sit in the same ecosystem, the slower component dictates the pace.



AI VULNERABILITY SURFACE

Legacy code in languages with small communities, limited documentation, and infrequent patching presents a disproportionately large attack surface. It's simultaneously harder to defend and easier for AI models to reverse-engineer.



COUPLING

The tighter the coupling between heterogeneous components, the more the slowest component constrains the ecosystem. An isolated legacy satellite with no dependencies may persist safely. A legacy backend embedded in a critical payment chain cannot.

Tightly coupled ecosystems should converge toward the same quadrant. Loosely coupled satellites may remain in legacy quadrants if operationally isolated. But any component deeply embedded in a critical business process and sitting in a high-vulnerability, low-velocity quadrant represents a compounding risk.

From Distance to First Move

A high ecosystem distance is a trigger, not a verdict. Before any migration decision is taken, a low-regret sequence applies:

01

DECOUPLE BEFORE MIGRATING

Insert an API or event facade so the legacy component's release cadence no longer gates the modern ones.

02

RAISE THE LEGACY PATCH SLA

Match the patch standard of the ecosystem's most exposed path before any rebuild begins.

03

WRAP, DON'T REWRITE

Place modern, observable, automatable controls (scanning, gating) around the legacy component.

04

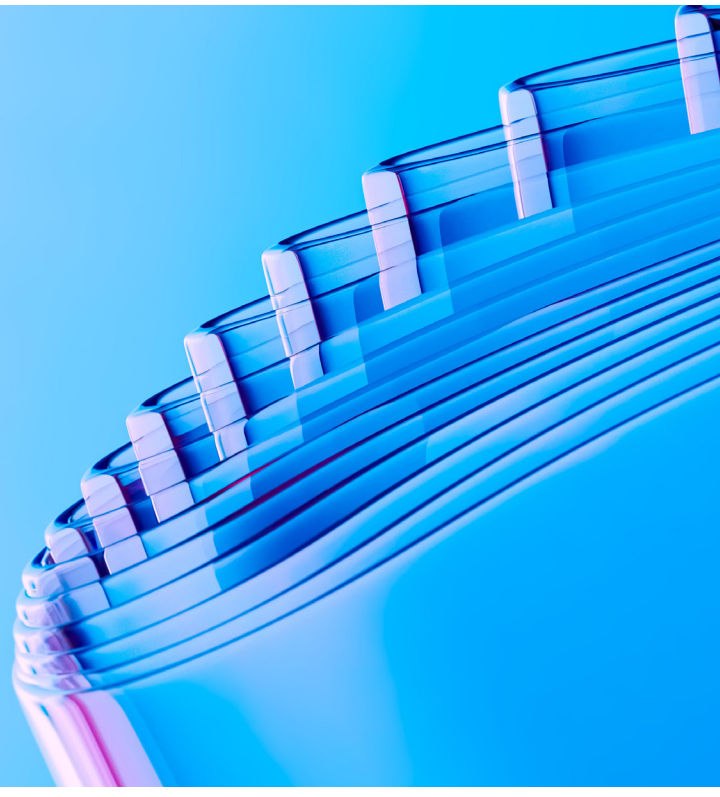
BUILD THE MIGRATION BUSINESS CASE

only where decoupling can't contain the risk.

Typically take weeks to months

Multi-year decision the metric is designed to inform, not replace

Key takeaway – use agentic AI to reduce complexity, not to tolerate it.



A CRITICAL CAUTION

Agentic AI deployed to absorb architectural complexity rather than reduce it becomes life-support for the very ecosystem distance identified as the binding constraint. The complexity isn't removed; it's hidden, and it compounds. This is symptom management, and it returns as a boomerang: the more agents are relied on to hold a fragmented estate together, the deeper the fragmentation entrenches, until the accumulated latent risk snaps back at machine speed.

The strategic use of agentic AI is the inverse: deploying it to decouple, consolidate and migrate components so that ecosystem distance actually shrinks over time. The goal should be to deploy agents to simplify the landscape, not to make an unsimplified one more tolerable.

Managing AI Complexity Across the Landscape

Embedded AI in Enterprise Software

Enterprise software vendors are integrating AI capabilities at an unprecedented pace. Miro now offers **multi-LLM integration** with autonomous AI features and model selection. Figma uses **Claude Sonnet 4.5** as its default LLM for code generation. ServiceNow has expanded into an AI-native platform with an **Autonomous Workforce of AI specialists**. Salesforce has embedded Einstein GPT across its CRM suite. Databricks has integrated an AI assistant across its data platform.

Each tool that was previously a predictable, well-understood COTS component has become a conduit for LLM interactions, often processing sensitive corporate data. This isn't inherently problematic; AI features can deliver significant productivity gains. But it introduces architectural complexity that must be actively governed.



Shadow AI, Model Volatility & Velocity Risk

Shadow AI describes a condition in which AI models operate within the bank's technology estate without explicit evaluation, approval or governance. It arises when a COTS vendor embeds an LLM into its product, routing corporate data through a model the bank hasn't assessed for security, data protection or regulatory compliance.

Under DORA, financial entities must **maintain a Register of Information for ICT third-party arrangements** and manage third-party risk through documented contractual arrangements, including sub-outsourcing chains. When a vendor embeds an AI model hosted by a further third party, this creates a dependency chain that must be captured in the Register of Information. The **EU AI Act** reinforces this from the model side: Providers of general-purpose AI models must supply downstream integrators with technical documentation on the model's capabilities and limitations. For a bank consuming an embedded model through a COTS vendor, this is a concrete legal basis on which to demand model documentation.

Beyond governance, there's an operational stability risk that directly affects architectural velocity. AI models are deprecated and replaced at a pace incompatible with enterprise banking cycles. In January 2024, **OpenAI retired 33 models simultaneously**, requiring all dependent applications to migrate. **GPT-4o was retired** from the ChatGPT interface on Feb. 13, 2026. Each deprecation event is a velocity shock: Established workflows break, trained agents require revalidation and the bank must re-test integrations on compressed timelines.

The BYOLLM Principle

The practical response has two components. First, establish a Bring Your Own Large Language Models (BYOLLM) principle, not only as a procurement criterion but as an architectural guardrail.

Products that allow integration with the bank's approved model stack should be preferred; those that hard-code non-configurable AI components require additional risk assessment and, potentially, feature deactivation. Second, for stable, well-defined tasks, consider small language models that can be hosted internally and aren't subject to external deprecation cycles.

Procurement decisions in financial services are increasingly architecture decisions. Every COTS product that introduces an unmanaged AI component extends the attack surface and introduces unplanned velocity constraints. The response isn't to reject AI in COTS; it's to govern it.

The Landscape-Wide Imperative: Reduce Complexity, Not Absorb It

Architectural distance and AI complexity are two faces of one problem. The first is the complexity a bank has accumulated in its own estate; the second is the complexity it now imports through agentic adoption and embedded-AI procurement.

Treated separately, each looks like a governance task. Treated together, they point to a single strategic objective: net reduction of complexity across the landscape. Reduction does not happen by default; and without an explicit mandate it does not happen at all.

The strategic posture is the same for both dimensions: use AI to simplify, not to sustain. Measure every architectural decision by whether it shrinks ecosystem distance, and every AI procurement by whether it can be governed within the bank's approved model stack and retired without a velocity shock. Complexity that can't be reduced must at least be contained and made visible.



From Assessment to Action

The response framework outlined in this paper requires technical expertise, strategic advisory capability and execution capacity.

This requires coordinated workstreams across four key fronts:

01

OPERATIONAL

AI-Assisted Vulnerability Assessment

Comprehensive scans of the technology estate using current-generation AI models, producing structured findings with severity classifications, affected components and concrete remediation steps.

02

TACTICAL

SDLC Modernization & Agentic Integration

Redesign the SDLC for an agentic threat landscape; shift security left, add agentic pre-deployment gates and build continuous, automated code review into the pipeline.

03

STRATEGIC

Application Landscape Assessment

Map the entire application estate using the matrix framework, apply the two assessment dimensions, calculate ecosystem distances and prioritize an architecture roadmap accordingly.

04

GOVERNANCE

COTS AI Risk Evaluation

Assess the commercial software portfolio for embedded AI: vendor transparency, configurability and BYOLLM readiness.

The window between vulnerability discovery and exploitation is narrowing. Institutions that act now, not merely to fix today's vulnerabilities but to build architectures that sustain the deployment velocity the threat environment demands, will be the ones that maintain both regulatory compliance and competitive position.

Architecture is no longer a back-office concern. It's the primary determinant of whether a bank can defend itself at the speed the threat landscape requires.

About the Authors

Lars Fetzer

Lars Fetzer is a Director at EPAM Advisory and Co-Lead of the Financial Services Industry Practice. His 20-plus-year banking career spans central strategy, business development and digital execution roles for private and business banking clients across Asia, Europe and Germany at a global universal bank, from shaping growth strategy to leading large-scale, multi-country digital delivery organizations. This blend of strategy, delivery and leadership underpins his work at EPAM Advisory, where he advises senior banking leadership on digital transformation, AI-driven operating models, governance and regulatory alignment. He is co-author of the EPAM Consumer Banking Report 2026.

Connect

Dennis Joosten

Dennis Joosten is Senior Director and Client Partner in EPAM's Banking & Financial Services practice, where he leads Europe's Banking Practice. His 25-plus-year career spans CTO and Technology Advisory roles for Europe's leading banks, from core banking transformation and Fintech integration to cloud, composable-architecture and generative-AI adoption. This underpins his work at EPAM, where he drives architectural optimization for global banks - platforming, sovereignty and AI-driven modernization - keeping agentic and core-banking transformations secure and compliant with evolving regulation. He is a regular speaker and publisher on agentic AI in banking, including as co-author of EPAM's Consumer Banking Report.

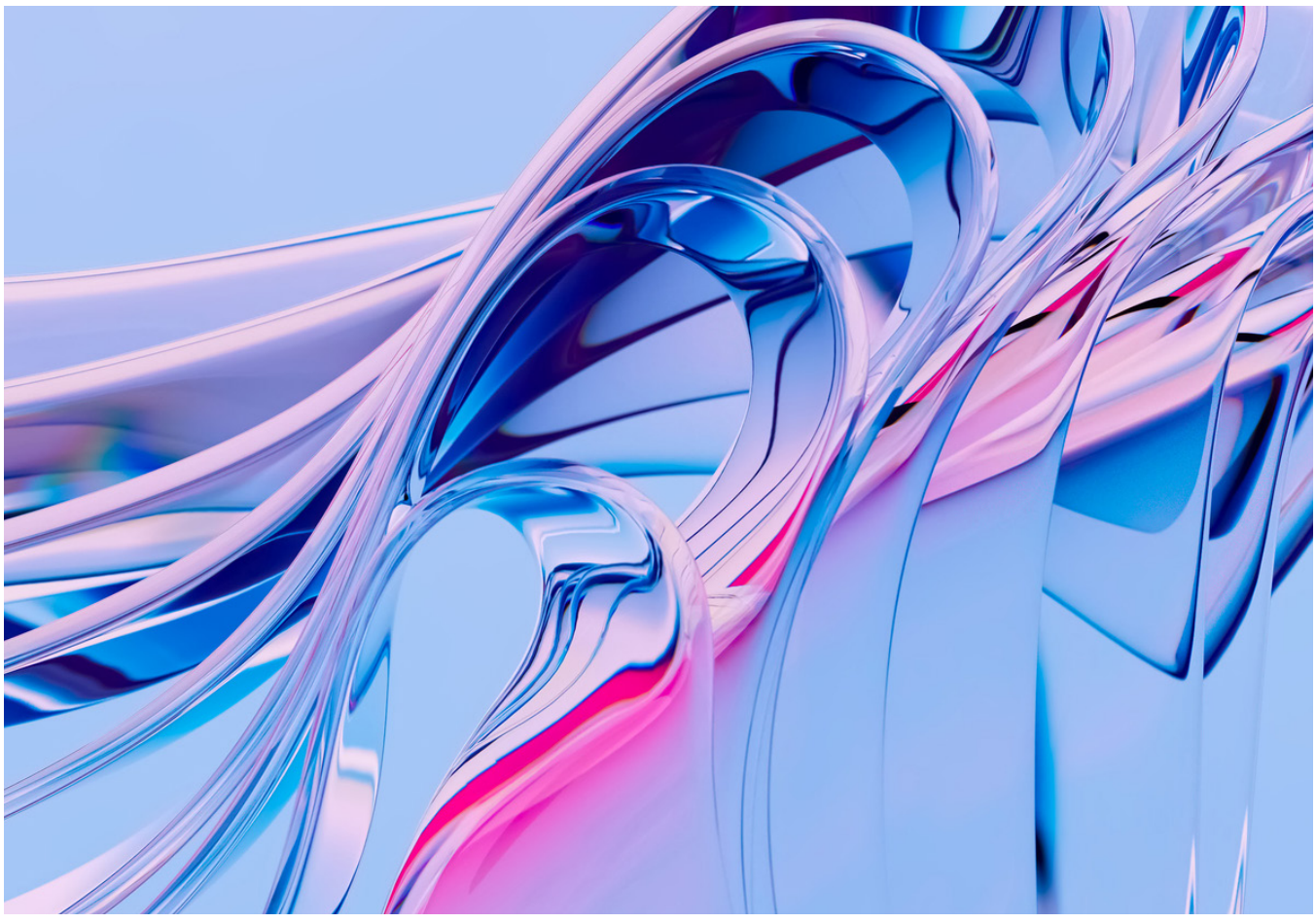
Connect

About EPAM

EPAM (NYSE:EPAM) is a global leader in AI transformation engineering and integrated consulting, serving Forbes Global 2000 companies and ambitious startups. With over thirty years of expertise in custom software, product and platform engineering, EPAM empowers organizations to become AI-native enterprises, driving measurable value from innovation and digital investments. Recognized by industry benchmarks and leading analysts as a leader in AI, EPAM delivers globally while engaging locally, making the future real for clients, partners and employees.

We are proud to be recognized by Forbes, Glassdoor, Newsweek, Time Magazine, Great Place to Work and kununu as a Most Loved Workplace around the world.

Learn more at www.epam.com and follow us on LinkedIn.





Global

41 University Drive, Suite 202
Newtown, PA 18940, USA

P: +1-267-759-9000
F: +1-267-759-8989

© 2026 EPAM. All Rights Reserved.