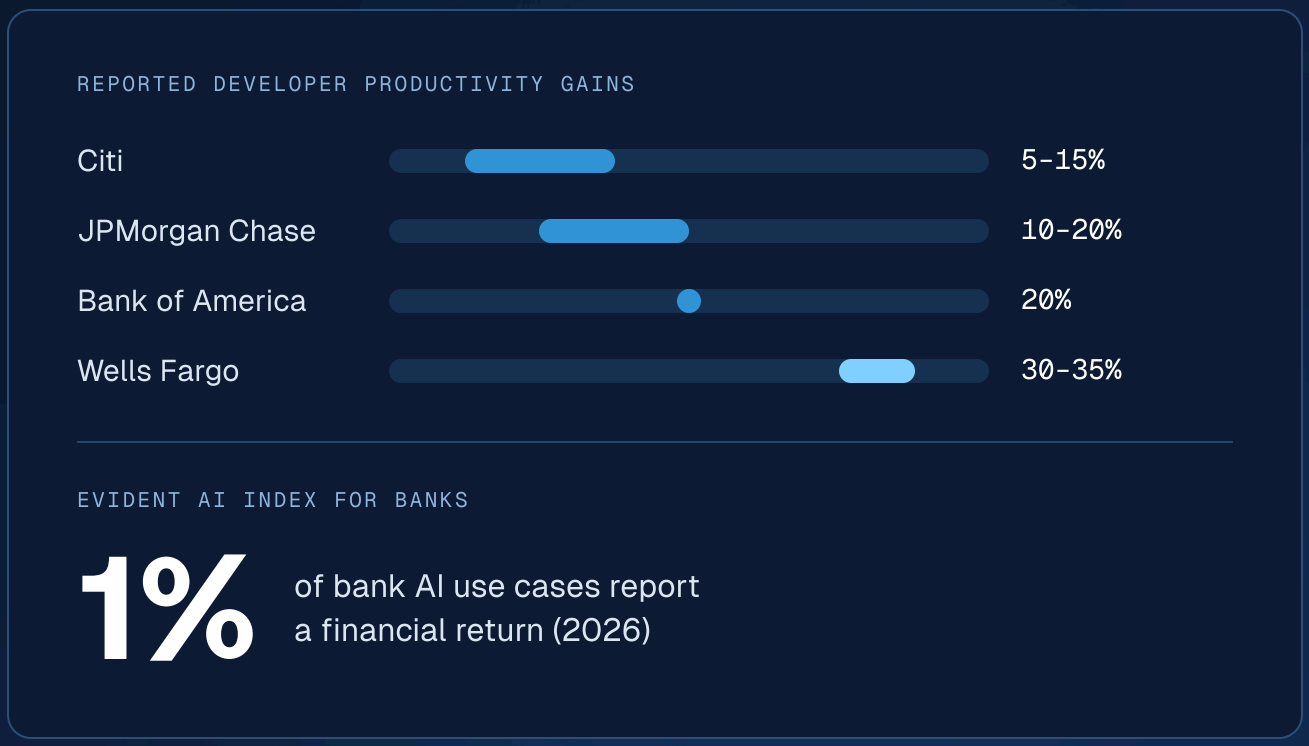


GUIDE · AI IN BANKING

Coding Faster Won't Save the Bank

What AI Agents Mean for How Banks Build Software



Zach Gardner
Chief Architect, Keyhole Software

KEYHOLE SOFTWARE

GUIDE · AI IN BANKING

Coding Faster Won’t Save the Bank

What AI Agents Mean for How Banks Build Software

By Zach Gardner, Chief Architect, Keyhole Software

Introduction

If you want to start an argument at a bank technology conference, ask someone how they measure developer productivity.

The largest US banks have all put a number on what AI is doing for their engineers, and the numbers are real enough. JPMorgan says tens of thousands of its engineers are 10% to 20% more productive with an in-house coding assistant.[1] Wells Fargo's CEO says his engineers are 30% to 35% more efficient writing code.[6] Bank of America has 18,000 developers using coding agents and reports 20% gains.[3] (Citi reports a more modest 5% to 15% from Copilot, which, to be fair, is the one I'd be most inclined to believe.)[8]

Profit is harder to find. Evident's 2026 AI Index for Banks found that only 12% of the use cases it tracks report impact against operational KPIs, and barely 1% disclose concrete financial returns.[14]

5–35%	**12%**	**1%**	**49%**
developer productivity gains reported by large US banks[1, 6, 8]	of bank AI use cases report impact on operational KPIs[14]	of bank AI use cases report a concrete financial return[14]	of banks have deployed generative AI[17]

Meanwhile, AI agents are starting to act on behalf of customers, and that changes what a bank has to build. Software acting for a customer needs its own identity and limits, instant payments need fraud decisions made before the money leaves, and agents need APIs they're allowed to use. A bank that only uses AI to write code faster could collect the savings and still lose the customer relationship to whichever agent ends up in front of that customer.

WHO THIS IS FOR

This paper is written for technology leaders at US banks, particularly regional and community banks, who are trying to figure out where AI belongs on the roadmap and what it's going to cost them if they get it wrong. It covers what the productivity numbers actually show, where coding agents and AI-assisted modernization fit, what it takes to build for customer agents, and how to govern all of it while the rules are still being written.

What Bank AI Productivity Numbers Actually Say

Here's what the big banks have said publicly about AI and their engineering teams. Most of it comes from executive remarks, not audited disclosures, so it's worth reading the source column.

Bank	What they've disclosed	Source and notes
JPMorgan Chase	Tens of thousands of engineers 10% to 20% more productive with an in-house coding assistant. 2024 technology budget of $17 billion.	Global CIO Lori Beer to Reuters, March 2025.[1] The bank estimated AI benefits at about $2 billion a year in 2025,[2] and it ranked first on the Evident AI Index again in 2026.[14]
Bank of America	18,000 developers using coding agents, with 20% productivity gains "coming out of those parts of the lifecycle."	CTIO Hari Gopalkrishnan at the November 2025 investor day, per CIO Dive.[3] The bank spends about $13 billion a year on technology, $4 billion of it on new initiatives,[4] and has said it expects to double its AI budget in 2027.[5]
Wells Fargo	Engineers 30% to 35% more efficient writing code.	CEO Charlie Scharf, December 2025: "We've not reduced the number of people we have coding today, but we're getting a lot more done."[6] Headcount was 197,466 at June 30, 2026.[7]
Citi	GitHub Copilot available to 40,000 developers, with 5% to 15% gains for regular users. Cognition's Devin agents since July 2025, limited to internal repositories.	CTO David Griffiths, American Banker, July 2025.[8] Agent changes go through testing and human review: "We're certainly not allowing any kind of deployment of code by the agents."
Goldman Sachs	Devin piloted alongside its roughly 12,000 developers, starting with "hundreds of Devins" that "might go into the thousands."	CIO Marco Argenti to CNBC, July 2025.[10] The "3 to 4x productivity" figure you'll see online comes from vendor and secondary blogs, not a measured result.
Morgan Stanley	DevGen.AI, an in-house tool that turns legacy code into plain-English specifications, saved about 280,000 developer hours in 2025.	Business Insider, 2025.[11] Morgan Stanley now says the tool supports more than 600 production use cases.

Depending on which bank you ask, AI makes developers somewhere between 5% and 35% more productive. Most of that spread comes from measuring different things. One bank counts time saved per task, another counts throughput, another counts the share of suggested code that developers accept. Evident's 2026 report notes that "results from developer tools are converging across banks,"[14] which makes sense when everybody bought roughly the same tools.

The independent research is murkier. METR's 2025 randomized trial found that experienced open-source developers took 19% longer on tasks when using early-2025 AI tools, while believing afterward that AI had sped them up by about 20%.[18] METR's follow-up estimated an 18% speedup for returning developers, but in February 2026 it called its own data an unreliable signal, mostly because developers now refuse to work without AI, which makes a control group hard to come by.[19] And none of these studies looked at bank developers working in thirty-year-old codebases.

My read is that AI coding tools reliably speed up boilerplate, test generation, documentation and making sense of legacy code. The gains on complex, novel work are smaller and much harder to measure.

Adoption outside the biggest banks is further along than you might think, too. Cornerstone Advisors' 2026 survey of bank and credit union executives found 49% of banks and 59% of credit unions had deployed generative AI.[17] The same survey found that "planned system replacements continue to fall well short of actual deployments," which anybody who has sat through a core conversion already suspected.

From Coding Assistants to Coding Agents

Copilot-class assistants are table stakes at large banks now. Citi, Bank of America, Wells Fargo and JPMorgan all run them at scale, and I'd be surprised to find a regional bank that isn't at least piloting one.

Coding agents are the next step. Instead of suggesting code, tools like Cognition's Devin take a ticket and produce a proposed change. Citi and Goldman Sachs have both deployed Devin for patches, upgrades, vulnerability fixes and language rewrites.[8, 10] Cognition says Brazil's Itaú uses it to automatically fix 70% of its security vulnerabilities (earlier reporting says 70% of static-analysis alerts, and either way it's the vendor's number).[13]

What I like about Citi's approach is how boring it is. Agents can only retrieve from internal repositories, every change goes through automated testing and a human reviewer, and agents don't deploy anything.[8] That's roughly the posture I'd recommend for any bank starting out.

KEYHOLE IN PRACTICE

We've seen what happens without that discipline. A founder with no formal software background brought us a stock alerting platform they'd built over the better part of a year, mostly with AI coding tools. It had a working interface, but large parts of the source code had never made it into the repository (the app had been running from cached files on the server, which were later deleted), and the alert-processing jobs kept timing out.

Our team worked backward from what was left, rebuilt the alerting engine with proper dependency management and CI/CD, and reworked the data layer with async I/O, batched queries and caching (using Claude and Cursor ourselves, for what it's worth). Core functionality was back within two weeks, as described in our case study on rebuilding the platform.[58] It's a small, non-bank example, but the failure mode is one every bank should recognize: code nobody fully understood, with no record of how it got there.

For a bank, I suspect the highest-value use is reading old code. Morgan Stanley built DevGen.AI specifically to read legacy code and produce plain-English specifications that a developer can work from, and that's how it saved those 280,000 hours.[11] Plenty of banks have business rules that live only in a COBOL program written before half the team was born, and a tool that can explain those rules is a big deal.

Legacy Modernization: Cheaper to Understand, Just as Hard to Cut Over

On February 23, 2026, Anthropic published a post on using Claude Code to break the cost barrier to COBOL modernization, covering code analysis, dependency mapping and documentation. IBM's stock fell 13.2% that day, its biggest drop since October 2000.[30] IBM's Rob Thomas answered the same day with a blog post whose argument boils down to one line: "Translating code is one thing. Modernizing a platform is something else entirely."[31]

He has a point, even if it's a self-interested one. The expensive parts of a mainframe exit are data architecture, replacing the runtime, preserving transaction integrity and all the non-functional requirements nobody wrote down because the old system has met them without complaint for forty years.

Gartner landed in roughly the same place. Its June 2026 prediction is that more than 70% of mainframe exit projects initiated in 2026 will fail to produce their intended benefits, because buyers overestimate what generative AI can do. It also expects 75% of vendors in the mainframe exit market to pivot or shut down by 2030. Alessandro Galimberti, a VP Analyst at Gartner, put it plainly: "There

is a widening gap between the marketing promise of GenAI and its real-world ability to transform and migrate complex legacy code."[29]

I'm cautiously optimistic anyway, for a narrower reason. AI makes modernization economically viable in ways it wasn't before, because it compresses the discovery work. Documenting what the legacy system actually does, mapping its dependencies and generating regression tests used to eat a big chunk of the budget before anyone moved a single line. That work is now much cheaper.

KEYHOLE IN PRACTICE

We've run a smaller version of this experiment ourselves. In a proof of concept for a food wholesaler (not a bank, but COBOL batch jobs look much the same everywhere), our team used EnterpriseGPT to explain what a mainframe batch job did, document its data structures, convert COBOL working storage into Java objects and draft the Spring Batch skeleton. That cut manual effort by 20% to 30%, and the conversion took about 16 hours.[57]

Developers still refined the business logic by hand, and deep Java and Spring Batch knowledge remained essential. The AI-accelerated COBOL modernization case study has the details, and that 20% to 30% lands right in the range the big banks report for their developers.

Cutover is where I'd stay conservative. The programs I'd bet on define what "equivalent" means before they start, move one capability at a time, run old and new in parallel, and switch only when the outputs reconcile. If a vendor is promising an AI-driven big-bang rewrite of your core, I'd politely show them the Gartner numbers and ask what makes them part of the other 30%.

KEYHOLE IN PRACTICE

That's also how our longest-running financial services modernization has worked. Since 2011, we've helped an investment and asset management firm move its COBOL batch processing, including fee calculations, vendor file processing, fund reconciliation and end-of-day reporting, onto Spring Batch and Spring Boot. It happened in phases over more than a decade, with more than 300 batch jobs enhanced between 2016 and 2025, and the firm eventually eliminated its COBOL dependency.[56] The batch processing case study walks through it.

Your Core Provider Is Now an AI Vendor

For most community banks and credit unions, AI strategy runs through the core provider whether they planned it that way or not. In 2026, all three of the big ones announced agentic platforms, each with a different foundation model partner.

Provider	What they announced	Model partners
Fiserv	agentOS (May 2026), a governed marketplace that launched with four Fiserv-built agents and nine third-party agents across areas like risk, regulatory reporting, deposit operations and reconciliation. Six institutions co-developed it, and it was expected to be widely available by August 2026.[21]	OpenAI, with AWS
FIS	A Financial Crimes AI Agent (May 2026) that assembles AML evidence across systems, being developed with BMO and Amalgamated Bank.[22] Separately, an agentic commerce offering with Visa and Mastercard (January 2026) that gives issuers "know your agent" (KYA) data.[23]	Anthropic
Jack Henry	An AI-driven security platform built with Google Cloud (June 2026),[24] plus about 25 discrete core services through the Jack Henry Platform that banks can adopt "incrementally and at their own pace, avoiding 'big-bang' core replacements."[25]	Google

The pitch is compelling, especially if you don't have a data science team. As Jim Reuter, CEO of First Interstate BancSystem, put it when agentOS launched: "Delivery of an agentic operating system through our core allows us to simply integrate agentic tools into our daily workflows."[21] ICBA has made a similar argument, that partnering with a core provider "is the most practical, and often the only, path forward for community banks seeking to responsibly leverage AI."[27]

I don't disagree, but I'd go in with eyes open about concentration. The OCC noted in a November 2025 request for information that the three largest core providers served more than 70% of depository institutions in 2022, and the largest alone served 42%.[26] Those three providers now run on three different foundation models. That means your bank's AI roadmap is partly your core vendor's roadmap, and your fourth-party exposure now includes OpenAI, Anthropic or Google whether or not you ever signed anything with them.

I'd still use vendor AI, and I'd spend real time on the contract. ICBA's 2026 AI security readiness guide suggests asking vendors whether AI features are already active in your environment and whether you'll be notified before more are switched on.[28] I'd also want to know:

- Which model sits behind each feature, and whether you'll hear about it before that changes
- What happens to your data once it reaches the model, including whether it can be used for training
- Whether you can switch individual features off and audit how they behave
- What you get to take with you if you leave

Building for AI Agents in Banking

After Money20/20 USA in October, Chris Nichols, President of Institutional Banking at SouthState Bank, wrote up ten lessons from the conference for community and regional bankers. His central worry is economic: "The real threat isn't that technology will replace banks. It's that technology may increasingly own the customer relationship while banks retain the balance sheet, compliance burden, and operational risk."[20]

The piece is written for bankers (and SouthState sells correspondent services and tools to the same banks it's writing for, so keep that in mind). Read it as an engineer anyway and you'll find a fairly concrete build list, one that lines up with what the card networks and core vendors are describing.

Agent Identity and Delegated Authority

When software moves money on a customer's behalf, the bank has to know which agent it's dealing with, what that agent is allowed to do and whether the action matches what the customer actually wanted. Nichols makes a point here that I think every payments architect should tape to their monitor: "A payment might be cryptographically valid and still be fraudulent or unauthorized."[20]

In practice, that means extending identity and access management to non-human identities. You need a registry of agents, scoped credentials per agent (instead of an agent logging in with the customer's password), transaction limits, a list of permitted counterparties, a way to revoke access and an auditable record of what the customer consented to. Can a customer's accounts-payable agent send $75,000? Can it add a new beneficiary? Those rules have to be enforced in the payment path, at the moment of the transaction. It's a multi-year platform build.

The networks are already building their half of this. Visa introduced its Trusted Agent Protocol in October 2025 as part of Visa Intelligent Commerce.[32] Mastercard Agent Pay binds agentic tokens to a specific agent, and Mastercard launched Agent Pay for Machines in June 2026.[33] On its October 2026 earnings call, Mastercard CEO Michael Miebach said one-time virtual card numbers are "a bridge" and "not a long-term solution that we will support," which tells you where the network expects tokenization to go.[34] Google's Agent Payments Protocol (AP2) moved to FIDO Alliance stewardship in April 2026.[35] And at Money20/20 itself, Proof and Circle showed an agent proving whom it acts for and then paying in USDC.[36]

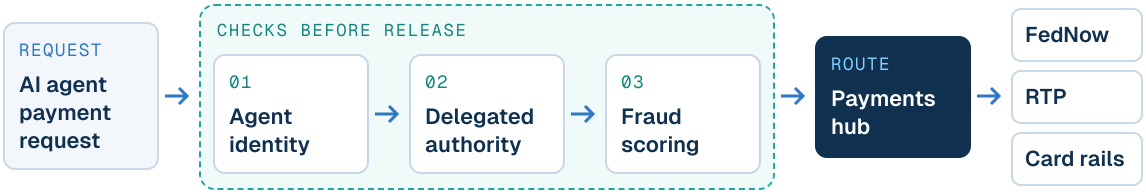
On the issuer side, that adds up to agent token provisioning and lifecycle management, customer controls for agent limits and merchant categories, storage of intent records for disputes, and fraud models that actually use agent identity signals.

Payment Hubs and Fraud Decisions Before Release

Nichols observes that the banks succeeding with instant payments have a payments hub and price instant payments as the default choice for customers. The national numbers back him up. FedNow passed 1,800 participating banks and credit unions in 2026 and settled about 5 million payments in Q2 2026, up 83% from the prior quarter, but the dollar value barely moved.[38] A Richmond Fed brief found that 71% of RTP member banks also use FedNow, while only 46% of FedNow banks use RTP.[39]

What holds most banks back is internal architecture. A modern hub (ISO 20022-native, multi-rail, with routing by speed, cost and risk, plus real exception management and reconciliation) may honestly be the highest-return modernization project available to a lot of regional and community banks, and it's a lot cheaper than replacing a core.

Fraud has to move with it. Post-settlement controls don't help much on a rail where the money is gone in seconds, and Nichols puts it well: "Faster payments coupled with AI without faster risk decisions simply accelerate the consequences of fraud."[20] That pushes fraud decisioning into a low-latency, in-line service that sits next to the payment hub and scores behavior, counterparty, context and delegated authority before release. If your fraud stack today is a handful of point tools and a manual review queue, that's an architecture problem, and AI-enabled fraud is making it more urgent (more on that in the governance section).



Agent payment path: identity, authority and fraud checks ahead of the hub.

In the path above, an agent's request has to clear identity, delegated authority and fraud scoring before the hub routes it to a rail.

Block, Build or Both

The hottest debate at the conference, according to Nichols, was what to do about third-party agents. Some banks are blocking them: about a dozen products profile click speed, scrolling, mouse movement and typing, and when the score says "automated," the bank can kill the session and force the customer to log back in without their agent.[20] Others are building their own. Huntington launched The 1866 Studio in October 2026, an AI-powered small-business operating system built with Google Cloud, and JPMorgan Chase has gone the same route.[37]

Most banks will probably end up doing both. Behavioral detection becomes a tunable product feature, with thresholds and experiments and a policy layer that decides which agents get blocked and which get admitted through sanctioned APIs. Those APIs (secure, consented, token-based, with

machine-readable product information) matter whatever happens with the open banking rule, because an agent can't recommend your deposit product if it can't read it. And the bank still needs its own customer-facing intelligence, so it stays in the customer's decision loop instead of becoming what Nichols calls "an interchangeable financial utility."[20]

The One Lesson I'd Push Back On

Nichols also argues banks should drop the human-in-the-loop construct in favor of "human-on-the-loop" monitoring, with other agents doing quality control, because designing without the human as a default forces better process design. I see the logic, and for low-risk, high-volume processes I think he's right.

It's also well ahead of where regulators, vendors and the most advanced banks are today. FIS says every decision on its financial crimes agent stays with the investigator.[22] Citi keeps a human on every agent-written code change.[8] For credit decisions, money movement and production deployments, I'd treat human-on-the-loop as something to design toward over time. Where it does make sense now, plan on investing heavily in evaluation harnesses, test coverage, observability and kill switches to replace the review queue.

Open Banking, Stablecoins and Tokenized Deposits

Section 1033 Is Stalled, and the API Work Shouldn't Be

The CFPB's Personal Financial Data Rights rule would require banks to give consumers, and the third parties they authorize, access to their account data through a developer interface. That's a big part of why it matters for agents, since an AI agent acting for a customer could be one of those third parties. The rule was finalized in October 2024, with the largest institutions originally due to comply in April 2026.

Those dates are on hold. In *Forcht Bank, N.A. v. CFPB*, a federal court in the Eastern District of Kentucky issued a preliminary injunction on October 29, 2025, barring the CFPB from enforcing the rule until it finishes reconsidering it.[44] The Bureau issued an advance notice in August 2025 and sent a revised proposal to OIRA for review on August 4, 2026.[45] As of this writing, I haven't seen it published.

I wouldn't build to the original deadline, and I also wouldn't pause API work. Data sharing continues through bilateral agreements and aggregators in the meantime, FDX remains the recognized standard, and agents make secure, consented, token-based APIs more important whatever the final rule ends up saying.

Stablecoins, Tokenized Deposits and the GENIUS Act

The GENIUS Act was signed in July 2025, and the OCC released its first final implementing rule on October 21, 2026, effective January 18, 2027, after roughly 328 comments on the proposal.[46] A few points matter for technology planning:

- Insured banks issue payment stablecoins through a subsidiary, not directly.
- The definition of a payment stablecoin excludes deposits, including tokenized deposits.
- The OCC eased several proposed requirements, including cutting the daily liquidity floor from the proposed 10% to 5%.[46]
- The FDIC approved its own proposal in April 2026, which would require a standard two-business-day redemption period and wouldn't extend pass-through insurance to stablecoin holders on reserve deposits.[47]

Because tokenized deposits are excluded from the stablecoin definition, they remain deposits under the existing framework, with the insurance and supervision that come with that. That strengthens the case Nichols makes for banks to explore them. His example is a $10 million commercial operating balance that moves to a stablecoin platform and takes the fees and the data visibility along with it.[20]

Whichever way a bank goes, the engineering list looks similar: wallet and custody capabilities, a ledger that runs 24/7/365, integration with tokenized networks and reconciliation between on-chain records and the core. Most cores weren't designed for any of that, so middleware and event-driven architecture come first.

Before spending on it, though, I'd borrow another of Nichols' ideas and decide "what needs to be true" first. Would 10% deposit growth justify building tokenized deposits? Or does the bank have to lose 20% before it acts?

Build, Buy or Partner

One number from McKinsey's 2026 State of AI survey has stuck with me. Of the 1,719 people surveyed, 32% said their organization had decided against buying at least one software product or feature because it could build it in-house with agentic coding tools.[16] That's a leading indicator, and it suggests the build-versus-buy math every bank did five years ago is worth redoing.

The large banks are already acting on it. Capital One built Chat Concierge, its first proprietary multi-agent assistant, on Meta's open-source Llama.[12] JPMorgan built its own coding assistant. Morgan Stanley built DevGen.AI.

Here's the rough decision rule I'd start with:

- **Build** what shapes the customer relationship and uses your data advantage: agent-facing APIs, customer intelligence, pricing.
- **Buy** regulated commodity processing, like the core ledger, rail connectivity and KYC utilities.
- **Partner** through the core vendor where scale, model access and examiner familiarity matter most. For most community banks, that covers most of AI.

Community Banks Are Playing a Different Game

JPMorgan's 2024 technology budget of $17 billion is larger than the total assets of most community banks. Nobody should pretend a $2 billion bank is going to out-build that, and it doesn't need to. Evident found that 80% of the leading banks in its index have adopted sophisticated AI control mechanisms, compared with 40% across the rest of the industry,[14] which makes me think governance capacity divides banks more than access to models does.

A community bank's realistic advantage is speed of adoption through its vendors, plus the relationships its bankers have with customers. Unfortunately, the relationship is exactly what third-party agents are positioned to take, which is why the agent posture and API decisions in this paper matter more to a community bank than to JPMorgan.

What Happens to the Developers

Headcount at the largest banks is falling, mostly through attrition. Wells Fargo had 197,466 employees at the end of June 2026,[7] and Citi was at about 219,000 under a plan to cut 20,000 roles by the end of 2026.[9] Executives increasingly connect AI coding gains to future headcount, though Scharf was careful to say Wells Fargo hadn't reduced its coders yet.

I'm more worried about the junior pipeline. Boilerplate, test writing and documentation are exactly the tasks agents now handle well, and they're also how early-career engineers have always learned a codebase. If a bank stops hiring juniors because agents do that work, it's going to have a hard time finding senior engineers in five years.

Meanwhile, new roles are showing up: AI platform engineers, evaluation and red-team engineers, people who own agent identity and security, and AI governance staff. Banking Dive reported AI governance talent across the top 50 banks grew 33% in the past year.[15] The core developer skill is moving from writing code toward specifying it, reviewing it, testing it and being accountable for what AI wrote.

AI Governance in Banking Is a Patchwork

If you were hoping regulators would hand you a rulebook for generative and agentic AI, I have some bad news.

Model Risk Guidance Left AI Out on Purpose

On April 17, 2026, the Federal Reserve (SR 26-2),[40] OCC (Bulletin 2026-13)[41] and FDIC (FIL-15-2026)[42] issued revised model risk management guidance that replaces SR 11-7, the document every bank model validator has had memorized since 2011. It's explicitly risk-based, and its expectations are aimed mainly at banks with more than $30 billion in assets. It also says, in so many words, that generative and agentic AI models "are not within the scope of this guidance" because they're novel and evolving quickly. The agencies "plan to issue in the near future a request for information" on AI.[40]

Please don't read that as permission. Examiners will still apply safety-and-soundness principles, the 2023 interagency third-party risk management guidance, fair lending law and UDAP/UDAAP, and without a federal AI framework to point to, the documentation burden lands on the bank itself.

Treasury's Framework Is the Closest Thing to a Playbook

In February 2026, Treasury released the Financial Services AI Risk Management Framework (FS AI RMF), built with more than 100 financial institutions and the Cyber Risk Institute. It adapts the NIST AI RMF to financial services and includes a risk and control matrix of 230 control objectives organized by adoption stage, along with an AI lexicon and a self-assessment.[43] It's voluntary. It's also, in my opinion, the most practical examination-ready control set available for AI-built and AI-powered software right now, and I'd map every generative and agentic use case to it.

AI-Written Code Needs Scanning

Veracode's October 2026 GenAI code security update, which tested 18 models, found the average security pass rate for AI-generated code rose from 56% to 62%, and the share of tasks introducing a vulnerability fell from 44% to 38%.[51] That's real progress, and it still means more than a third of AI coding tasks introduced a known flaw. Java, which runs a lot of bank back ends, scored just 31%. (Veracode sells application security testing, so factor that in, though its methodology has been consistent across more than 100 models.) Its separate 2026 State of Software Security report found 82% of organizations carry security debt, meaning known flaws left unresolved for more than a year.

Static and dynamic scanning of AI-generated code should be mandatory, with extra attention on cross-site scripting, injection and Java services. Agents with tool access and payment authority widen the attack surface further through prompt injection and agent hijacking, which is one more reason the scoped identities from earlier in this paper matter. For the fundamentals that apply to

any Java service in a regulated environment, AI-written or not, Kent Lautenschlager's post on secure Java development in regulated industries is a good place to start.[59]

Fraud Is Getting Cheaper to Commit

FinCEN's November 2024 alert on deepfake fraud (FIN-2024-Alert004) describes rising suspicious activity reports involving AI-falsified identity documents used to open accounts and executive impersonation used to authorize wires. It asks banks to use the key term "FIN-2024-DEEPFAKEFRAUD" in SARs.[53]

The FBI's 2025 Internet Crime Report, released in April 2026, included its first AI section, with $893 million in reported losses across 22,364 AI-related complaints. Total reported losses hit $20.9 billion, up 26% from the prior year.[54] Deloitte projected back in 2024 that generative AI could push US fraud losses to $40 billion by 2027, from $12.3 billion in 2023.[55] (That's a projection, and the FBI numbers only count what gets reported, so I wouldn't compare them directly.) In KPMG's 2026 banking technology survey, 76% of banking executives said cyberattacks had increased over the past year.[52]

Fair Lending and State AI Law

AI-assisted credit agents inherit every ECOA and Regulation B adverse-action obligation the bank already has. Colorado repealed and replaced its AI Act with SB 26-189, signed May 14, 2026 and effective January 1, 2027, which requires notice, an explanation of adverse decisions and human review on request for automated decisions in consequential areas like lending.[48]

The state picture is messy. Executive Order 14365 (December 2025) directed the Attorney General to set up an AI Litigation Task Force to challenge state AI laws,[50] xAI sued Colorado in April 2026, and the Justice Department intervened two weeks later.[49] Texas's TRAIGA and California's SB 53 both took effect January 1, 2026. An executive order can't preempt state law on its own, and Congress hasn't passed anything that does, so banks operating across states should comply with the state laws that apply to them while keeping an eye on the litigation.

I'd build to the strictest credible standard now instead of waiting for things to settle down. Retrofitting controls onto agents that are already moving money is going to cost a lot more than designing them in.

Where I'd Start: An AI Roadmap for Regional and Community Banks

If I were running technology at a regional or community bank, this is roughly the order I'd work in.

01 **Keep rolling out coding assistants and scoped agents, with guardrails.** Require static and dynamic scanning on AI-generated code, human approval on production merges and retrieval limited to internal repositories (the Citi pattern). Tag AI-written code so you can find it later during an audit or an incident.

02 **Measure business outcomes alongside tool usage.** Nichols suggests customer acquisition cost, operating expense per account, credit cycle time and incremental revenue.[20] Pair those with DORA metrics on the engineering side, and stop reporting "hours saved" as if it were a return.

03 **Make agent identity and delegated authority a top-three platform investment.** That means a registry, scoped credentials, limits, permitted counterparties, revocation and an auditable consent record, integrated with the Visa and Mastercard agent token programs.

04 **Fund a payments hub with in-line fraud decisioning before you fund a core replacement.** For most regionals and community banks I think it's the better return, and it's the thing standing between broad FedNow and RTP participation and actual usage.

05 **Use AI for modernization discovery, and stay conservative on cutover.** Let it document legacy logic, map dependencies and generate regression suites. Then migrate one capability at a time with parallel runs and reconciliation.

06 **Decide your agent posture on purpose.** Block, build or (most likely) both, but write the policy down instead of letting a fraud tool's default threshold decide it for you.

07 **Adopt the Treasury FS AI RMF as your AI control baseline**, and plan to comment on the interagency AI request for information when it comes out.

08 **Renegotiate core and AI vendor contracts.** Ask for notice before AI features are activated, notice of model changes, limits on data use, audit rights, exit provisions and visibility into fourth parties. ICBA's resources are a good starting point for community banks.

09 **Keep building APIs and run the numbers on tokenized deposits.** Continue FDX-aligned API work regardless of 1033's timing, and decide what needs to be true before you invest in digital money.

10 **Protect the junior pipeline.** Keep hiring early-career engineers onto AI-augmented teams, add evaluation, red-team and agent-security skills, and retrain developers toward specification and review.

Items 1, 3 and 4 are where I'd want a funded plan by the end of 2027. The rest can follow.

Your Coding Agent Controls Are a First Draft of Agent Identity Policy

One connection I keep coming back to is that the two halves of this paper are closer than they look. The controls Citi uses for Devin (access limited to internal repositories, a human reviewer on every change, nothing deployed by the agent, a record of what it did) are close to the controls a bank needs for a customer's agent that wants to move money. Scoped access, a named owner, a limit on what it can do without a person, an audit trail.

So the engineering teams working out how to govern coding agents are, without really meaning to, writing the first draft of the bank's agent identity policy. I wouldn't run those as two separate AI programs. Use the coding agents, pay close attention to what it took to trust them, and get those people in the room when the payments team starts designing for customer agents. I suspect that experience will be useful sooner than most roadmaps assume.

At Keyhole, we've spent nearly two decades working with banks, credit unions and investment firms,[61] on everything from COBOL batch modernization to a payment management system for a nationally recognized bank,[60] and we're using AI tools on more of that work every year. If you're trying to figure out where your bank's roadmap should go next, we'd be happy to compare notes.

How We Researched This

We reviewed bank disclosures and executive remarks, analyst and industry research, core provider announcements, regulatory texts and security research, current as of October 30, 2026, and checked each figure against a primary source or multiple outlets. A few limits apply.

- Most bank productivity figures are self-reported by executives, use different definitions and aren't audited. Treat them as directional.
- Vendor-produced research (Veracode, Cognition, the core providers) and SouthState's article, which promotes SouthState's own tools, carry commercial interest. So do the Keyhole case studies we cite, which describe our own work.
- Gartner's mainframe prediction, Deloitte's fraud projection and the timing of a revised 1033 rule are forecasts, not measurements.
- We left out figures we couldn't confirm, including a widely repeated claim that Bank of America streamlined some software tests by 90%, a 17-million-line figure for Morgan Stanley's DevGen.AI and the "3 to 4x" productivity number attributed to Goldman Sachs.
- Regulatory status (Section 1033, the GENIUS Act rules, state AI laws and the coming interagency AI request for information) was changing quickly at the time of writing. Re-check anything date-sensitive before relying on it.

References

BANK DISCLOSURES

1. Reuters via Yahoo Finance, "JPMorgan engineers' efficiency jumps as much as 20% from using coding assistant," March 13, 2025.
2. Entrepreneur, "JPMorgan Chase Is Saving $2 Billion a Year Using AI," 2025.
3. CIO Dive, "Bank of America runs 270 AI models across operations," November 10, 2025.
4. American Banker via Yahoo Finance, "'Revenue opportunities everywhere': BofA's tech chief on AI," November 17, 2025.
5. Let's Money, coverage of Bank of America's plan to double its AI budget in 2027, September 24, 2026.
6. diginomica, Wells Fargo and Goldman Sachs remarks at the Goldman Sachs US Financial Services Conference, December 10, 2025.
7. Wells Fargo, Q2 2026 Earnings Supplement.
8. American Banker, "Citi is rolling out agentic AI to its 40,000 developers," July 14, 2025.
9. Citigroup, Q2 2026 Financial Supplement.
10. CNBC via NBC Chicago, "Goldman Sachs is piloting its first autonomous coder in major AI milestone for Wall Street," July 11, 2025.
11. Business Insider via AOL, "A team of engineers saved Morgan Stanley more than 280,000 hours this year," 2025.
12. Capital One Tech, "Driving the future of car buying with agentic AI," March 5, 2025.
13. Pulse 2.0, Cognition funding coverage including the Itaú figure, May 31, 2026.

INDUSTRY RESEARCH

14. Evident, "2026 Evident AI Index," October 6, 2026.
15. Banking Dive, AI governance talent at the top 50 banks, October 6, 2026.
16. McKinsey, "The state of AI in 2026: On the road to ROI," August 2026.
17. Cornerstone Advisors, "What's Going On in Banking 2026."
18. METR, "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity," July 2025.
19. METR, "We are Changing our Developer Productivity Experiment Design," February 24, 2026.
20. Chris Nichols, SouthState, "Money 20/20 USA 2026: Ten Lessons to Shape Bank Strategy," October 26, 2026.

CORE PROVIDERS AND MODERNIZATION

21. Fiserv, "Fiserv Launches agentOS, The Operating System for Agentic AI in Banking," May 14, 2026.
22. FIS, Anthropic partnership and Financial Crimes AI Agent, May 4, 2026.
23. FIS, agentic commerce offering with Visa and Mastercard, January 12, 2026.
24. Google Cloud, Jack Henry AI-driven security collaboration, June 25, 2026.
25. Jack Henry, FY2026 Annual Report.
26. OCC, request for information on core service providers, 90 FR 54882, November 28, 2025.
27. American Banker, "It's time for community banks to break free of 'core' limitations," July 30, 2026.
28. Independent Banker, ICBA AI security readiness guide, June 2026.
29. Gartner, mainframe exit prediction, June 18, 2026.
30. Malay Mail (Bloomberg), "IBM shares hit 25-year low as AI promises to update ageing mainframe code," February 24, 2026.
31. IBM, "Lost in translation: what the AI code debate keeps getting wrong," February 23, 2026.

AGENTIC COMMERCE AND PAYMENTS

32. Visa, Trusted Agent Protocol announcement, October 14, 2025.
33. Mastercard, "Mastercard launches Agent Pay for Machines," June 2026.
34. American Banker, "Mastercard touts the importance of agentic tokens," October 2026.
35. FIDO Alliance, "Google donates Agent Payments Protocol to FIDO Alliance," April 29, 2026.
36. Investing News, "Proof and Circle Show How AI Agents Can Prove Who They Act For and Pay in USDC," October 19, 2026.
37. Google Cloud, Huntington introduces The 1866 Studio, October 19, 2026.
38. Federal Reserve Financial Services, FedNow quarterly statistics, Q2 2026.
39. Federal Reserve Bank of Richmond, "FedNow and the Development of U.S. Fast Payments," Economic Brief 26-28, August 2026.

REGULATION

40. Federal Reserve, SR 26-2, April 17, 2026.
41. OCC, Bulletin 2026-13, "Model Risk Management: Revised Guidance."
42. FDIC, "Agencies revise interagency model risk management guidance" (FIL-15-2026).
43. ExecutiveGov, Treasury AI Lexicon and FS AI RMF release, February 2026.
44. Moore & Van Allen, "CFPB enjoined from enforcing Personal Financial Data Rights Rule."
45. Consumer Finance Monitor, "CFPB sends new Section 1033 open banking proposal to OIRA for review," August 6, 2026.
46. WilmerHale, "OCC finalizes GENIUS Act rule for permitted payment stablecoin issuers," October 27, 2026.
47. Mayer Brown, "FDIC proposes GENIUS Act rules," April 2026.
48. Ropes & Gray, "Colorado scales back AI law," May 2026.
49. US Department of Justice, intervention in xAI lawsuit challenging Colorado's law, April 2026.
50. Executive Order 14365, "Ensuring a National Policy Framework for Artificial Intelligence," December 11, 2025.

SECURITY AND FRAUD

51. Cybersecurity Media, coverage of Veracode's October 2026 GenAI code security findings.
52. KPMG, 2026 Banking Technology Survey, June 17, 2026.
53. FinCEN, Alert FIN-2024-Alert004 on deepfake media fraud, November 13, 2024.
54. Help Net Security, coverage of the FBI IC3 2025 Internet Crime Report, April 7, 2026.
55. Deloitte, "Deepfake banking fraud risk on the rise," May 29, 2024.

KEYHOLE SOFTWARE

56. Keyhole Software, "Case Study: Modernizing Enterprise Batch Processing for a Financial Institution."
57. Keyhole Software, "AI-Accelerated COBOL Modernization to Spring Batch," December 2025.
58. Keyhole Software, "Rebuilding and Scaling an AI-Generated Stock Alerting Platform."
59. Kent Lautenschlager, "Best Practices for Secure Java Development in Regulated Industries," Keyhole Software, July 15, 2025.
60. Keyhole Software, "Custom Payment Management System with .NET and React."
61. Keyhole Software, "Banking & Financial Services Software Development."

ABOUT THE AUTHOR



Zach Gardner

Chief Architect, Keyhole Software

Zach Gardner is Keyhole Software's Chief Architect with more than a decade of development experience. Zach has led numerous initiatives to modernize technologies and processes with real-world experience as an educator, architect, technical lead, developer, and mentor. He primarily lives in Microsoft Azure, leveraging modern architectural approaches to enable hybrid app modernization that meets business needs at an exceptional cadence. He also mentors new developers and provides architectural input to complement the existing talent of clients.

WORK WITH KEYHOLE

Figure out where your bank's roadmap goes next.

At Keyhole, we've spent nearly two decades working with banks, credit unions and investment firms, on everything from COBOL batch modernization to a payment management system for a nationally recognized bank, and we're using AI tools on more of that work every year. We'd be happy to compare notes.

Explore our financial services work

keyholesoftware.com

KEYHOLE SOFTWARE