AI agents inside the bank

What they can reach, what they may do, and who decides

IN BRIEF

01 Banks have put AI assistants in front of millions of customers and most of their staff. Agents that act on a bank's systems are fewer and narrower, and they wait for a person at the regulated decision.

02 The hard part is not the model. It is reach and control: getting an agent to the bank's own systems, and deciding what it may do when it gets there.

03 Public catalogues of ready-made connections do not contain a bank's systems of record. Those are reached through the APIs the bank already runs.

04 A bank needs one place where every tool an agent can call is listed, every call is checked against policy, a change waits for a named person, and the record can be put in front of an auditor.

05 All of this can run on infrastructure the bank controls. Most packaged banking agents cannot.

IN THIS PAPER

01 Where banks are today 2
02 The systems of record are in nobody's catalogue 3
03 Two doors, one gate 3
04 Six ways into an older system 4
05 What an agent may do 5
06 Seven questions before an agent touches a system of record 6
07 What regulators already ask 6
08 Where to start 7

PART 01

Where banks are today

3 billion+
Bank of America assistant interactions since 2018

250,000
JPMorgan Chase staff with access to its model portal

61%
Valley National sanctions-hit reviews done by an agent

62%
Rabobank chats settled without a person

The scale is in assistants. Bank of America reports more than three billion interactions with its customer assistant since 2018, and says more than nine in ten of its employees use the internal version. JPMorgan Chase has given about 250,000 staff access to its portal of language models, and describes its work on agents that carry out tasks of several steps as early.

Agents that act are narrower, and already useful. Valley National Bank in the United States says an agent now carries out 61% of its sanctions-hit reviews, on about 14,000 alerts a month. Rabobank's agents take 20,000 calls and 7,000 chats a day and settle 62% of the chats without a person, including blocking a card from start to finish.

The line is drawn at the regulated decision. Across the announcements we read, agents gather evidence, summarise, draft and clear the obvious. A person decides the suspicious-activity filing, the escalated case, the credit exception and the movement of a customer's money. Ryt Bank in Malaysia shows the pattern in a consumer product: its assistant prepares a transfer from a typed sentence, and the customer taps Confirm before anything moves.

The control is moving into the platform. Commonwealth Bank of Australia says it registers every agent and every use, evaluates agents before they go live and while they run, and feeds those checks to a central control plane. Its chief executive put the principle in a phrase: an agent acting for someone should not operate on an "unconstrained basis".

Almost all of it runs in someone else's cloud. Of the packaged banking agents we read about, one vendor states that its agent can run on a bank's own premises. The rest run in the vendor's cloud or do not say. Banks are already pushing back at the edges. Rabobank runs its models inside its own cloud subscription for compliance, and turned down a search service because it stores data outside the bank's region. We found no public account of a bank running agents entirely on its own infrastructure.

That last point matters more as agents move from answering to acting. An assistant that answers questions needs documents. An agent that reviews a sanctions hit needs the screening system, the customer record and the payment. Those sit inside the bank.

PART 02

The systems of record are in nobody's catalogue

The Model Context Protocol, MCP, has become the common way to give an agent a tool. A system is wrapped once as an MCP server, and any agent can use it.

The largest public catalogue of these servers is Docker's. We read every entry. On 19 November 2026 it held 327 servers. They cover databases, document stores, ticketing, monitoring, identity providers, cloud services and web search.

Not one is a banking system. There is no core banking product, no card processor, no payment switch, no loan origination system, no treasury system, no sanctions screening service and no credit bureau. The entries filed under finance are payment services for merchants, market data, and tools for retail trading.

The banking vendors' own efforts do not fill the gap yet. We found MCP servers published by payment companies and market data companies. Among core banking and lending vendors we found two. The two card networks publish servers that serve documentation, not transactions.

There is a second fact underneath this one. **Most core banking vendors document their APIs for customers only.** The description of the API, the file that says what can be called and how, is something the bank holds. Nobody else can connect an agent to it in advance.

So the route to a bank's systems of record does not run through a catalogue. It runs through the bank's own APIs, described by the bank's own files.

PART 03

Two doors, one gate

A bank's estate has an edge and a centre, and an agent platform needs a door to each.

The first door is the catalogue, for the edge. Katonic's catalogue of MCP servers is drawn from Docker's and carries Katonic's own additions. A server is deployed as a container inside the bank's own cluster, from the bank's own image registry if it keeps one. This is how an agent reaches an Oracle database, a document store, a monitoring system or a web search.

One caution applies to any catalogue. A container that runs inside the bank may still call a service outside it. Only a server whose target the bank hosts keeps everything inside. Many entries are written by the community, not by the vendor named on them, and a bank will want to read the source and pin the version.

The second door is the bank's own APIs, for the centre. Most systems a bank runs can describe their API in a standard file, an OpenAPI or Swagger description. Katonic imports that file, from an address, pasted text or an upload, and each operation in it becomes a tool an agent can call.

- The file is checked before anything is created. The bank sees how many tools it will produce and any warnings.
- Each tool carries the operation's own inputs, with their types and which are required, so the model is told what the API expects and does not guess.
- Operations the agent should not have can be switched off one by one.
- A tool can be tried with real values, or as a dry run that shows the exact request and sends nothing.
- Each connection has its own sign-in, a limit on calls per minute, and a time limit.

Both doors lead to one gate. A tool from a catalogue server and a tool from an imported API sit in the same list, under the same policies, with the same record. An agent builder sees tools. A risk officer sees one inventory.

PART 04

Six ways into an older system

Not every system has a modern API. The work of giving it one is done in products the bank already owns, and it follows a small number of patterns.

	THE SITUATION	WHAT THE BANK DOES	WHAT THE AGENT PLATFORM DOES
01	**The system already has an API description**	Hands over the file and a credential, and names the operations agents may use	**Imports the file**
02	**The bank's API gateway already fronts the system**	Registers the agent platform as one more consumer	**Imports the gateway's file, or connects to the gateway's own MCP endpoint where it has one**
03	**The system speaks SOAP**	Puts a REST face over it and reviews the generated description by hand	**Imports the reviewed file**
04	**A mainframe transaction**	Exposes it through the mainframe's API product, mapping the fields and the caller's identity	**Imports the file that product generates**
05	**A system that works by message queue or file**	Builds a small adapter: submit a request, then ask for its status	**Imports the adapter's file**
06	**A system with only a screen**	Records the steps in a screen-automation product	**Imports that product's file, for reading only**

Two things follow. Nothing is replaced: the agent comes in through the same door as every other consumer of the bank's APIs, and is subject to the same limits. And the first two patterns cover a great deal.

A bank that has spent a decade putting an API gateway in front of its systems has already done most of the work.

PART 05

What an agent may do

Reaching a system is half of it. The other half is deciding what an agent may do there, and being able to show the decision afterwards.

Every call is checked before it runs. On Katonic the order is fixed: is this user allowed at all, do the inputs carry anything that looks like an injection attack, and what does the bank's policy say about this tool.

A policy gives one of four answers. Allow it. Allow it and record that it was watched. Hold it until a person approves. Block it. A policy can cover one tool, every tool on a server, or every tool, and it can apply to the whole bank or to one team. Where two policies meet, the stricter one wins.

A tool that changes something waits. Each imported operation is rated by what it does: reading is low risk, changing is medium, deleting is high. When an agent is published, any tool that changes something and has no policy of its own is set to wait for approval. A bank can relax that for a tool. It has to decide to.

If the policy cannot be read, the call is refused. A fault in the control must not become permission.

The agent acts as the person using it. Where a bank's API checks who is calling, Katonic passes the signed-in user's identity with each call. The value comes from the platform, not from the model, so an agent cannot be talked into acting as someone else.

Every call leaves a record: who asked, which agent, which tool, what the policy decided and who approved.

A POLICY GIVES ONE OF FOUR ANSWERS

01	02	03	04
Allow it.	**Allow it and record that it was watched.**	**Hold it until a person approves.**	**Block it.**

What one tender already asks

In 2026 a pension fund trustee tested agent platforms before it ordered one. It wrote the tests itself and scored them on invented data. What it asked for reads like the last three sections of this paper.

- The agent had to reach the trustee's member system through tools made from that system's own API description.
- The caller's identity had to travel with every call. One member seeing another's records failed the whole test.
- A number the documents did not contain failed the whole test. So did figures from two reporting periods mixed in one answer.
- A batch of forms had to stop for a person's approval and carry on from where it stopped.
- One row of the test data carried a hidden instruction to approve everything. The agent had to flag it, not obey it.
- A form that could not be read had to go to a person without holding up the rest.

Katonic was tested against it, and the trustee has since ordered the platform.

PART 06

Seven questions before an agent touches a system of record

These are worth asking of any agent platform, ours included.

01	**Is there one list of every agent in the bank, and of every tool each one can call?**
02	**Which of those tools change something, and who decided each one may run?**
03	**When a change needs approval, who approves it, and what do they see?**
04	**Whose identity does the agent act under when it calls a system?**
05	**What happens to a call when the policy cannot be checked?**
06	**Where does the model run, and what leaves the bank's network on each request?**
07	**Can every call be shown to an auditor, with the decision that allowed it?**

PART 07

What regulators already ask

Few rules are written for AI agents yet. Singapore's central bank issued guidelines on AI risk management on 7 October 2026 that bring agents into scope, and they take effect from October 2027. Elsewhere agents fall under rules a bank already lives with: technology risk, outsourcing and data protection.

Across the regulators we read, the same expectations recur. The board stays accountable, including for AI bought from a vendor. People oversee a system in proportion to its risk. The bank keeps an inventory of its AI systems. A system is tested before production and monitored after, with a way to suspend it. Records are kept. Third parties are assessed.

Malaysia shows how this lands. Bank Negara has no rulebook for AI. Its policy on risk management in technology applies; it has been in effect since 28 November 2025 and was reissued on 25 September 2026. Bank Negara's guidance on that policy names agentic AI as an emerging technology. A bank that first puts an emerging technology into a critical system consults the Bank beforehand, with a risk assessment and an outside review, whether the system runs in a cloud or in the bank's own data centre. Public cloud is permitted, and it brings further steps: a cloud risk assessment, control of customer data and encryption keys, approval where the arrangement is a material outsourcing, and a transfer assessment where personal data leaves Malaysia.

So running agents on the bank's own infrastructure does not take away the duty to assess and to consult. It takes away the questions about where the infrastructure is, who else shares it, and which country's law reaches it. And the controls in section 5 are the evidence a risk assessment asks for: a list of agents and their tools, a named approver, a record of each decision, and a way to stop.

PART 08

Where to start

Pick work where reading is slow, the decision stays with a person, and the systems involved already have an API.

THREE PLACES THAT FIT

Sanctions alert review

The agent compares each hit with the customer record and recommends clear or escalate, with its reasons. A compliance officer decides.

Policy and procedures

Staff ask questions and get answers from the bank's own documents, with the source cited. Nothing is changed, so there is nothing to approve.

Loan application intake

The agent checks a pack for completeness, extracts the figures and lists what is missing. It approves and declines nothing.

THEN, IN ORDER

01 Name the systems the agent needs, and which of the six ways each is reached.

02 Decide the policy for every tool before the agent is built.

03 Build it with the people who do the work today.

04 Run it on test data, and then beside the manual process.

05 Measure it against that process: time, and how often a person overturned it.

06 Go live for one team.

A fuller list of 26 agents across nine areas is published with this paper.

About Katonic AI

Katonic AI is the enterprise platform for serving models and building, running and governing AI agents. It runs on your machines, with your models, under your administrators. Katonic is listed in the Red Hat Ecosystem Catalog and runs on Red Hat OpenShift. It is in production at a top-tier bank in Asia, air-gapped, inside the bank's own perimeter. It was founded in Sydney in 2020 by Prem Naraindas, with its global hub in Dubai and teams in Sydney, Mumbai and Dubai.

KATONIC.AI

SOURCES

All read on 20 November 2026.

01 Bank of America, "A decade of AI innovation: BofA's virtual assistant Erica surpasses 3 billion client interactions", as reported by Financial IT, 22 August 2025.

02 CNBC, "JPMorgan Chase wants to be the first fully AI-connected megabank", 30 September 2025.

03 Valley National Bank's sanctions compliance manager, quoted in a UiPath announcement, 25 March 2026. ir.uipath.com/news/detail/436

04 Microsoft, "Rabobank conversational banking" case study, 11 August 2025. learn.microsoft.com

05 Ryt Bank, "Ryt AI". rytbank.my/ryt-ai

06 Commonwealth Bank of Australia, "Accelerate AI: 10 takeaways", 19 November 2026. commbank.com.au

07 Temenos, announcement of its financial crime agent, 21 May 2025. temenos.com

08 Docker, public registry of MCP servers, as it stood on 19 November 2026. github.com/docker/mcp-registry

09 Bank Negara Malaysia, Policy Document on Open Finance, 17 November 2026. bnm.gov.my/-/pd-openfinance

10 Bank Negara Malaysia, Risk Management in Technology policy document, reissued 25 September 2026, and its frequently asked questions of 1 July 2026.

11 Monetary Authority of Singapore, Guidelines on Artificial Intelligence Risk Management, 7 October 2026.