THE FUTURE IS HEADLESS

Why the next generation of sanctions screening will be run by agents, not screens

> *"The best interface is no interface."*
>
> *– Golden Krishna, designer and author*

For twenty years, almost every improvement in sanctions screening has, in the end, been an improvement to a screen: a faster alert queue, a richer case file, a smarter dashboard. The assumption underneath never changed: a human being would sit in front of the system and look at the result.

That assumption is about to break.

The financial industry is going real-time, and it is taking itself apart in the process: the monolithic applications of the past are being replaced by composable ecosystems of components wired together through APIs. At the same time, AI agents have become capable enough to read a screening alert, work out whether it is real, act on it, and write down why, all in seconds. Put those two trends side by side and the conclusion is increasingly hard to avoid: in the near future, most screening alerts will be handled by software, not by people.

This paper is about what that means and about why the systems built for the screen era will struggle to make the journey, while those built API-first, "headless", are already there.

From monoliths to ecosystems

Start with the big picture, because screening does not live in a vacuum. Three forces have been quietly rewiring financial technology:

- **Everything is going real-time.** Instant-payment schemes (FedNow, SEPA Instant, UPI, PIX…) have collapsed the settlement window from days to seconds. A control that used to run comfortably overnight now has milliseconds to do its job. With more than sixty live instant-payment markets clearing billions of transactions a year, the direction of travel is not in doubt.
- **Monoliths are giving way to components.** Banks and fintechs increasingly assemble their platforms from specialized parts connected through APIs, rather than buying one giant suite that tries to do everything. The unit of architecture is no longer the application; it is the component.[1]
- **Finance has become an ecosystem.** Open banking, embedded finance and Banking-as-a-Service all rest on the same idea: a capability can be exposed through an API and consumed by someone else's system.

The common thread is simple. Function is being separated from interface. And anything that can only be reached by a human clicking through a screen is starting to look like a relic.

An ocean of false positives

Now zoom back into screening, where this matters more than almost anywhere else.

In the domain of screening, the overwhelming majority of the alerts it produces are false alerts. This is due to the obligation of result the banks are under, as well as their (understandingly) very low risk appetite on sanctions compliance. Industry benchmarks routinely put the false positive rate above 99%. For every hundred alerts an analyst opens, more than ninety-nine are a waste of their time: a near-namesake, a common name in the wrong country, a counterparty they already cleared last week.

Staff augmentation is not a strategy, it is a holding pattern.

The volumes of transactions make this brutal. A mid-sized international bank can generate hundreds of thousands of alerts a month; a global one, millions. And the math only gets worse as instant payments push ever more traffic through the screening engine in real-time.

[1] For a more detailed view, you can read our white paper on that subject:
Revolutionising Compliance - The shift from monolithic systems to composable ecosystems

How has the industry coped so far? Mostly by throwing people at the problem: bigger teams, often offshored, sometimes helped by tooling that shaves a few seconds off each review. As everyone in the industry noticed by now, this does not scale. Volumes grow geometrically, headcount grows linearly and expensively. Good investigators are scarce, turnover is high, and quality drifts from one shift to the next.

The first wave of automation fell short

So, over the last five years or so, the industry tried something smarter: automation.

On the surface, the idea looked sensible. Take all the historical alerts and the decisions analysts made on them, train a machine-learning model on that data, and let the model auto-dispose the new alerts that look just like the old ones. It shipped under many names: auto-disposition, alert suppression, intelligent triage… but the recipe was always the same: learn from the past, repeat it in the future.

It helped. On the easy, repetitive alerts that look exactly like a thousand cleared before them, these models genuinely take work off the table. But the approach has three structural limits that no amount of tuning removes:

- ***It only knows what it has already seen.*** A classical model recognizes patterns from its training data and is lost outside them. Show it a brand-new sanctions entry, an unfamiliar counterparty type, or a transaction shaped unlike anything in its history, and it cannot reason its way through, as it can only pattern-match. As sanctions regimes have turned more volatile, that blind spot has grown.
- ***It cannot explain itself.*** A model like this returns a score, not a reason. To keep auditors comfortable, institutions bolt a templated "rationale" onto the output: "auto-closed: score below threshold, customer on cleared list", but that is a label, not an explanation of why the model decided what it did.
- ***So everyone stays cautious.*** Because of the two limits above, these models are deployed timidly, on the safest sliver of alerts, while the bulk of the work stays in human queues. The gain is real, but capped.

Notice the pattern: the first wave automated the cases we had already solved. It could not handle the ones we hadn't.

Enter the agents

This is exactly where LLM-based agents change the game and why the change is qualitative, not incremental. An agent does not need to have seen a case before to deal with it. Given the alert, the transaction context, the customer record, the matched list entry and the institution's policy, it can reason, like an analyst, about whether this particular hit makes sense. A name that matches a sanctioned individual but sits in the wrong decade, the wrong country, with the wrong identifiers? An agent can work that out, even on a combination it has never encountered and, crucially, it can write down why. Not a simple template, but an actual explanation: what it looked at, what it weighed, and why it came to a conclusion. In many cases that rationale is richer and more consistent than what a rushed human captures from the same data.

Automation could only repeat past decisions. An agent can reach a decision it has never made before and still explain it.

But an agent is only ever as good as what it is given to reason about, and this puts a new and underappreciated demand on the screening engine itself: it must emit as much high-quality metadata as it possibly can. A human analyst can squint at a sparse alert and fill the gaps from experience; an agent thrives on structure. Every extra, machine-readable detail the engine can attach to a hit becomes raw material the agent can weigh. Where the screen era treated such metadata as optional information buried beneath the UI, the agentic era treats it as the primary product. The richer and more structured the engine's output, the sharper and more defensible the agent's decision.

So does the analyst disappear? Certainly not.

The realistic future is not an empty room. It is the same analysts we have today, each now supervising a large team of agents rather than working a personal queue. The analyst's job shifts from alert disposition to setting policy, calibrating how the agents behave, reviewing samples of their work, handling the genuinely hard cases, and signing off on the contested ones. This is not only a more scalable model, it is almost certainly the one audit and supervisory authorities will insist on, because accountability has to rest with a named human, not with a model.

The pyramid doesn't disappear, it inverts. Agents handle the many, humans handle the few and own the outcome.

We have seen this kind of adoption curve before in compliance, and agents will likely climb it in three steps:

- ***Phase 1: Assist.*** The agent enriches and triages: it prioritizes the queue, drafts a rationale, and points the analyst at what matters. The human still decides.

- ***Phase 2: Recommend.*** The agent proposes a disposition; the analyst validates it or overturns it. Every correction makes the next recommendation better.
- ***Phase 3: Supervised autonomy.*** The agent disposes of the clear-cut alerts on its own, at a quality that can exceed the tired human average, while analysts supervise the fleet and audit oversees the whole, exactly as it would any other risk function.

Why the old guard can't follow

If the destination is so clear, why can't the legacy systems simply drive there? Because of how they were built.

Most established screening platforms are end-to-end suites, fifteen or twenty years in the making. They are far more than a matching engine: around it sit alert workflow, case management, list management, tuning, reporting, configuration and audit… a whole constellation of modules. And almost every one of those modules was designed around a screen, for a specific human role: the analyst, the supervisor, the list manager, the auditor.

Where these systems have APIs, the APIs were usually added later, as integration bolt-ons, and tend to expose only the most obvious function: send a name and get back the hits. Everything an autonomous agent would actually need to work end-to-end (pull the full alert context, fetch the list entry, record a disposition with its rationale, check prior decisions on the same entity, nudge a threshold, produce an audit-ready trail) often lives only behind the GUI.

You can bolt an API onto a screening system. You cannot bolt a new mindset onto 20 years of code.

So "making it agent-ready" is not a matter of pointing an agent at the existing API: it means either re-engineering large parts of the platform to expose those functions, or building fragile robots to click through screens meant for people. Both are slow, costly and brittle.

The real problem is one of intent. A system built on the assumption "a person will look at this screen" is a fundamentally different animal from one built on "another system will call this function", and you cannot easily retrofit the second into the first.

Headless by design

The alternative has a name borrowed from modern web architecture: headless. A headless system has no privileged front-end. Every capability is a function that can be called programmatically, and the screen, if there is one, is just another client of the same API that everything else uses.

For a screening solution, being headless means all of it is callable: not only "screen this name", but list management, policy management, rationale capture, threshold and settings configuration, statistics, metrics and audit access. None of it is trapped behind a UI. That single design choice changes everything for the agentic future:

- ***Agents use the same door as everyone else.*** There is no special "agent mode" to build. Whatever a developer can call, an agent can call, under the same credentials, permissions and audit trail the institution already controls.
- ***The operations are the right size.*** An agent that judges an alert a false positive can record the disposition, attach its reasoning and update the statistics through clean, discrete calls, no pretending to be a human clicking buttons.
- ***A new discipline appears: Agent UX.*** For twenty years we obsessed over user experience for humans. The next twenty will be about user experience for agents: how an API answers should be phrased so a model reads it unambiguously, how errors should surface, how capabilities should be discoverable. Expect this to extend the interface itself: richer APIs, command-line access built for automation, and emerging standards that let agents discover and call tools in a common language.

The hard questions: accountability, explainability and model risk

None of this is free, and it would be dishonest to pretend the road is smooth. Four challenges deserve a clear answer rather than a hand-wave:

- ***Accountability.*** When an agent closes an alert, who is responsible? Not the agent. Institutions will need governance that ties every decision back to a named human owner: the compliance officer who owns the program, supported by the model owners who govern the agents, with explicit boundaries and escalation rules.
- ***Explainability.*** A regulator or auditor must be able to ask "why was this discarded?" and get a real answer. Headless systems help, because logging a structured decision record is natural when every action is already an API call, but the explanation still has to be written for humans to read, not just machines.
- ***Model risk management.*** Agents are models, and supervisors are starting to treat them as such: they will need to be inventoried, validated, monitored for drift and re-tested. Existing frameworks are the starting point : SR 11-7 in the United States and, more

recently, the interagency SR 26-2 issued in April 2026 by the Federal Reserve, OCC and FDIC (which replaces both SR 11-7 and SR 21-8). Interestingly, SR 26-2 puts generative and agentic AI explicitly outside of its current scope while reminding institutions to govern them under sound risk-management principles anyway (a strong hint about where the supervisory spotlight is turning). In Europe, the EBA's guidelines play a similar role.

- ***Audit.*** Auditors will want to replay a sample of decisions: the inputs available at the time, the model version, the prompt, the output, etc… and confirm the behavior matched policy. A headless system, where every interaction is already an explicit, loggable call, is built for exactly this.

None of these are reasons to wait. They are reasons to design carefully. And it is worth remembering that the status quo i.e. armies of people manually reviewing the same false positives forever has its own audit and consistency issues that we have simply learned to live with.

The future is headless

The shift to agents is not a far-off scenario or a vendor's daydream. It is the logical result of three things already happening: the economics of manual review breaking under real-time volumes, AI agents becoming genuinely capable of advanced reasoning, and the whole financial industry moving to composable, API-first architecture.

In that world, the most valuable properties of a screening solution are no longer how good its screens and dashboards look. They are whether the entire system can be cleanly driven by another piece of software, how richly it can describe what it found, and whether it can keep up with the load. Platforms designed around screens, with an API stapled on afterwards, will need expensive surgery to keep up. Platforms that were API-native from day one and headless by design are already standing where the industry is heading.

The most future-proof screening engine is no longer the one with the best screen. It is the one that needs none.

Being callable is necessary, but not sufficient. An agent reasons on what the engine hands it, so the quality and richness of the metadata the engine produces becomes a first-class concern rather than an afterthought. The best engines for the agentic era will be the ones that say the most about every hit (the scores, the matched fields, the list data, the rationale behind every selection) in clean, structured, machine-readable form. In the screen era this detail was hidden behind the interface; in the agentic era it is the product.

And none of this matters if the engine can't keep up. The same real-time, instant-payment world that makes agents necessary also pushes ever-growing volumes through the

screening engine, at latencies measured in milliseconds and on a 24/7 basis. An engine that produces beautiful metadata but buckles at peak load, or adds seconds to a payment that is supposed to settle in less than one, has solved the wrong problem. Scale and low latency are therefore not separate, nice-to-have performance metrics; they are part of what makes an agentic operating model viable at all. The engine has to do more (richer output, more data points analysed, more context attached) for more transactions, faster, than ever before.

For a financial institution, the choice is being made right now, with or without a conscious decision being taken. The platform chosen this year will likely still be running at the end of the decade. So the question to ask is not only "does it screen well today?" It is: "will it still make sense when most of its alerts are read by an agent, and a person only looks at the few that truly matter?".

HOW TO CONTACT US

If you want to have more information on our solutions and services, please contact us at the following address:

info@neterium.io

Information about our solutions and future events are available on our website

www.neterium.io

Dive into our latest insights : explore our white papers, webinars, and podcasts here:

www.neterium.io/insights

To learn more about the Neterium API, including the full developer documentation, tools and other resources, please register on our User Portal at

portal.neterium.io

Follow us on LinkedIn and Twitter/X:


linkedin.com/company/Neterium


@neterium

NETERIUM

the next generation of screening infrastructure