



Where work actually happens

## The Middle Office Is the Layer Most Banks Never Quite Finished

Banks invest in customer-facing application layer (front office) and stabilise core systems (back office). The operational layer in between (middle office) is usually built in pieces over time – and the resultant gaps are where unrealised value sits.

Domagoj Markovina

CS Computer Systems

Walk through almost any mid-sized bank and you'll find two things that work well and one that's harder to pin down.

The customer-facing layer works. The mobile app and online banking is well thought through and responsive, the contact centre has a modern desktop. A decade of digital investment has gone into making this layer dependable, and in most banks it is.

The core systems work too, in their own terms. Core banking is rarely loved by the people who run it – the platforms are often old and the change windows are short – but it does what it's designed to do.

Between those two layers is where work actually happens. And in most banks, this is the layer that was never quite finished.

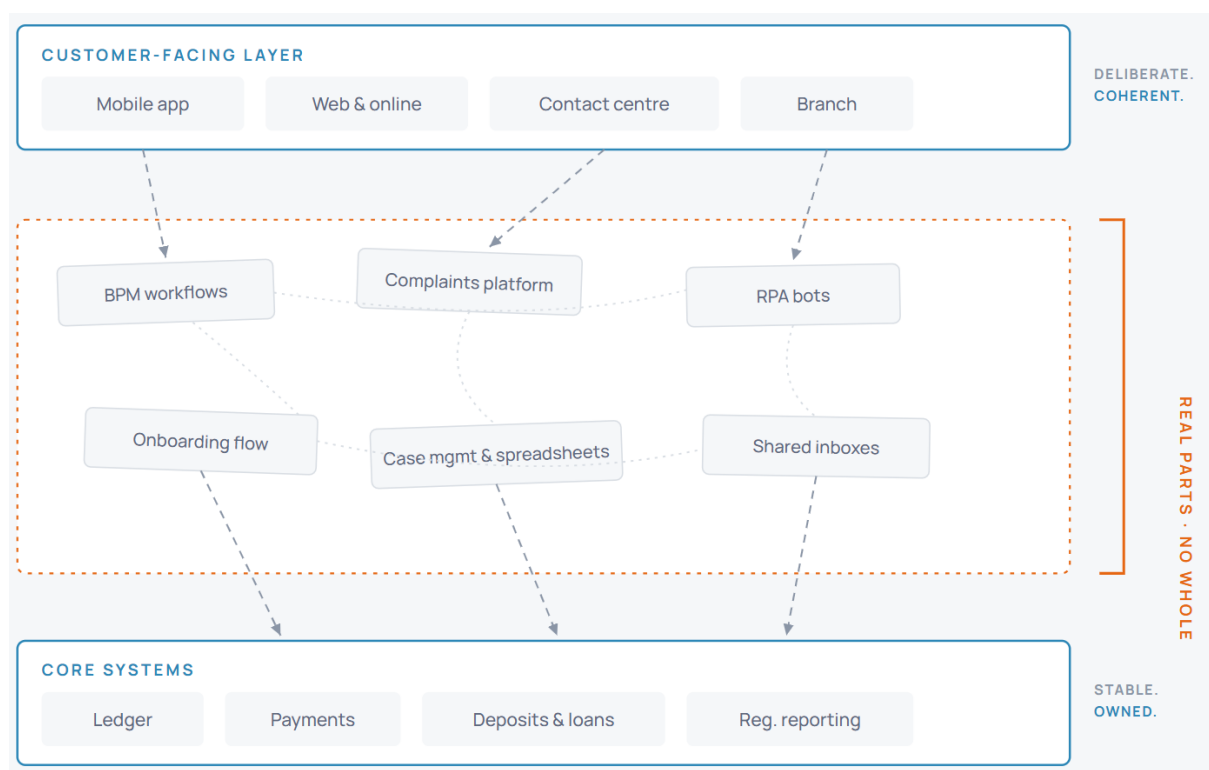


Figure 1 - Genuine investments that don't yet connect into a single layer. Front and back office layers are often well built – what is missing is coherence across them.

### The layer where customer outcomes are produced

The operational layer sits between the channels customers touch and the core systems that hold the money. It's where an application becomes an account, a complaint becomes a resolution, an onboarding request becomes a customer, a corporate query becomes a deal.

In most banks it isn't one coherent layer. It's a collection of things assembled over time: a BPM workflow here, a complaints platform there, an onboarding process, some robotic automation holding hand-offs together, a few case-management tools, and a layer of spreadsheets and shared inboxes underneath it all. Much of it is genuinely well-built. The complaints system does its job. The onboarding flow works. But the pieces were built at different times, for different reasons, and they don't connect into a single, owned layer.

This is what people often mean when they talk about "legacy in operations". The legacy usually isn't the individual systems – most of them are fine per se. The legacy is the absence of a deliberate architecture above them.

It matters because this layer is where customer experience is actually produced. A current account isn't a row in a deposit system. It's the experience of opening one, using one, having a problem with one, and getting that problem resolved. The app is the window. The core is the ledger. **The middle office is the bank, delivered.**

### Why this layer is so often fragmented?

It's worth being clear about something, because it's easy to read all this as criticism of the people who built these systems. It isn't. Where the operational layer is fragmented, it's rarely because anyone failed to try. It's usually the predictable result of constraints the technology function didn't control.

**Budgets tend to follow visibility.** Channel programmes are visible to the board and get funded. The operational layer behind them is harder to show and harder to fund, so it gets improved in increments rather than as a whole.

**The core often couldn't be touched.** Short change windows and real risk on core systems pushed operational logic outward, into the tools that were safe to change. Fragmentation was frequently the responsible choice at the time.

**Business units usually bought point tools to solve specific pain points** – each solving a genuine local problem, without trying to fulfil a single architecture mandate. Sensible decisions individually; a patchwork collectively.

**And mergers bolted stacks together.** Integrations leave banks running several overlapping operational systems at once. Consolidation is a multi-year programme, rarely fully finished before the next change lands.

None of this is a verdict on anyone. It's the context the work has been done in. The useful question isn't "why wasn't this built properly in the first place". It's "given how it had to be built, what would it take to make it coherent now?"

## Where the gap shows up?

Because no one budgets for “the middle office” as a single line item, the cost of the pieces not connecting tends to surface in other numbers.

It shows up in onboarding. A significant share of digital applications stall somewhere between submission and an active account – a document that didn’t reach the right team, a check that stalled, a step that needed a manual hand-off that didn’t happen. Each failure has its own explanation. Collectively, they describe a layer that in practice doesn’t run as a layer.

It shows up in contact-centre volume that won’t fall despite years of digital investment. Customers call because something in the operational layer didn’t complete, or completed wrong. The call is the symptom; the cause is upstream.

It shows up in complaints, where repeat rates stay stubborn even after a new complaints system, because resolution still depends on processes in other systems that don’t connect to it. The complaint gets recorded and routed, but the underlying issue is often resolved at the symptom level rather than the root cause.

It shows up in regulatory load. Modern regulatory regimes assume operational processes can be evidenced and audited consistently. A layer spread across many tools can meet that bar – usually with far more manual effort than it should take.

And it shows up most expensively in transformation programmes that don’t move the metric they were sold against. A polished new app sitting on a fragmented operational layer can become a faster route to the same delays and hand-offs. The board sees the channel improvement. The customer sees the unchanged wait. Closing that gap is almost always a middle-office problem, not a channel one.

## The same complaint, in two operating models

It helps to make this concrete. Consider what happens to a single complaint in two different banks – and most banks sit somewhere between the two.

In the first, the complaint is logged in one system, emailed to a team to action, and its resolution depends on data held in other systems the handler has to chase. It’s often fixed at the symptom level, and the regulatory reporting is reconstructed after the fact. It’s managed, in the end, by the diligence of the people involved rather than by the system.

In the second, the complaint goes through a defined intake and is categorised against a current taxonomy. It’s routed to the right team with the regulatory deadline visible. The handler sees the customer’s full history. Required communications are generated against

the deadlines. If it stalls, it escalates automatically. Root-cause data is captured in a structured way, and the regulatory reporting view updates in real time.

Same event. Different operating model. The goal isn't to leap from the first to the second overnight. It's simply to move forward.

### What a coherent layer actually does?

Rather than a checklist of things a bank is failing at, this is more useful as a quiet diagnostic. How many of these does your current setup do, and how consistently across different processes? Most banks do some of them well in some places. The pattern of where they hold and where they break is usually the map of what to fix first.

A coherent operational layer does five things:

**1. It orchestrates the work** – it knows what has to happen, in what order, by whom, against what deadlines, and with which hand-offs to which systems. The question to ask is whether your processes run end to end, or stop at the edges of each tool.

**2. It holds a process record alongside the customer record** – a record of the work itself: what stage it's at, what's blocking it, what was decided, by whom, against which policy version. The test is whether you could reconstruct a case without asking the people who ran it.

**3. It surfaces the right context to the right person** – the agent, the compliance officer and the operations manager each see the same underlying record through a view relevant to them, rather than working from separate, disconnected screens.

**4. It handles exceptions explicitly.** Most processes break not in the normal path but in the unusual 5–15% of cases. A layer that only runs the happy path forces every exception into ad-hoc human work, which is where most cost and most failure live.

**5. And it changes at the speed of the business, not the speed of the core.** Policy, regulatory and product changes land in the operational layer first. It needs to be configurable enough to absorb them quickly. The honest question here is how long it takes you to add a step to a live workflow today – and whether that timescale sits with the business or with a long release cycle.

### Making the layer deliberate

A deliberate middle office is a strategic asset. An accidental one is a quiet, recurring tax on every digital programme and every regulatory cycle.

The good news is that making it deliberate is usually less dramatic than it sounds. For many banks it isn't a rip-and-replace exercise. It's consolidation – bringing the operational work onto one configurable layer that orchestrates between the channels and the core, with the customer record emerging as a by-product of running the work well rather than as the thing everything else is bolted onto.

The category of platform that does this is sometimes called intelligent process automation, sometimes low-code business platforms. We'd describe it more plainly as a middle-office operating layer. The stronger options are unified by design rather than assembled by acquisition: they treat process and customer record as equal concerns, and they stay configurable enough that the people who run the processes can change them. The point is to hold a deliberate position on the layer – which sometimes means a new platform, and often means using what you already have, more coherently.

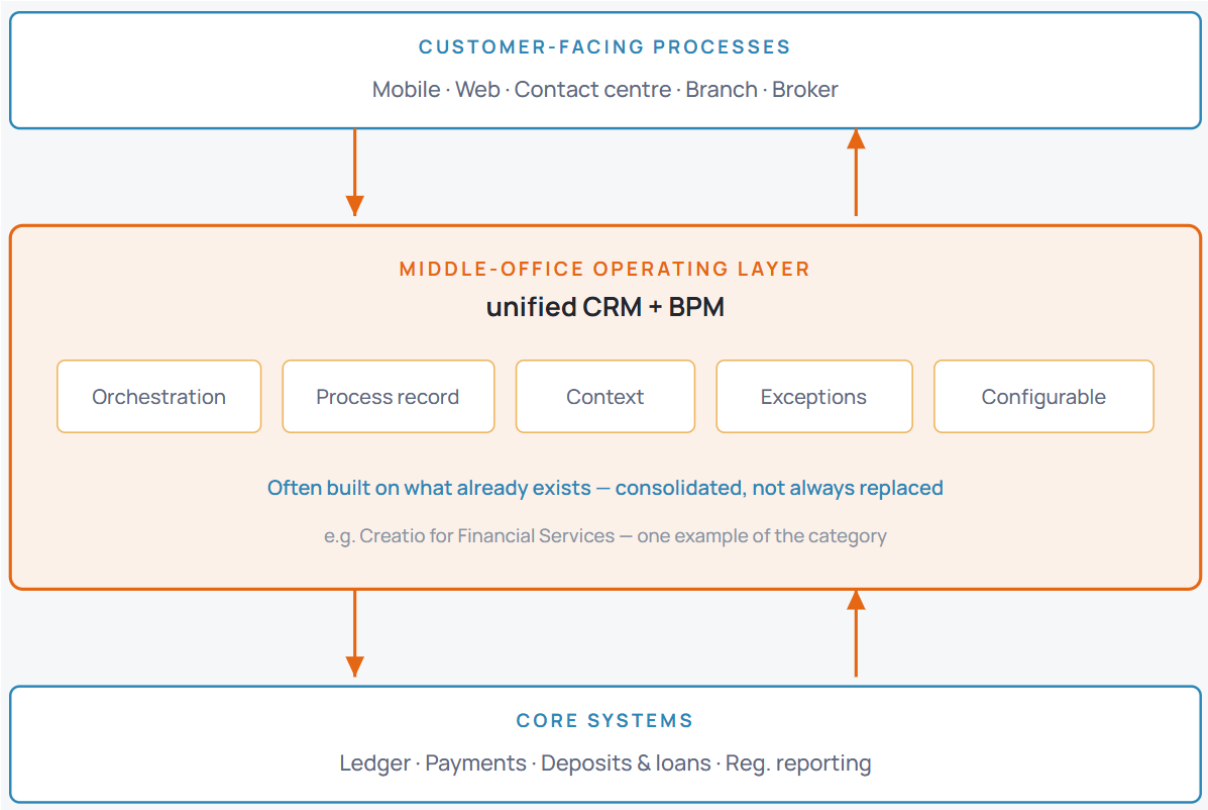


Figure 2 - The same bank, made coherent. One configurable layer orchestrates the work between channels and core. For many banks this is built by consolidating what already exists, not by replacing it.

### Where does a useful conversation start?

There's no single right answer, because it depends on the bank. Some need to consolidate point tools onto one platform. Some need to use more of what they already have. Some are best starting with a single high-value process – complaints, onboarding, claims – and building the layer outward from there.

What every bank benefits from is asking the question before scoping the solution.

1. Where does our operational layer actually sit today, and which parts already work well?
2. Are those parts connected into one layer, or running side by side?
3. What would it take to make the whole thing coherent, and how much of that is consolidation rather than replacement?
4. And what would change for the customer if we did?

If those questions are useful, that's where the conversations worth having begin. The platform question comes later – and sometimes the honest answer is to make more of the platforms you already run.

If you'd like to talk through where your operational layer sits today and what it would take to make it coherent – we work with clients on exactly this kind of problem.