BANKING INTEROPERABILITY ACCELERATOR GUIDE

A practical guide to modern banking integration

Build faster today without adding complexity tomorrow

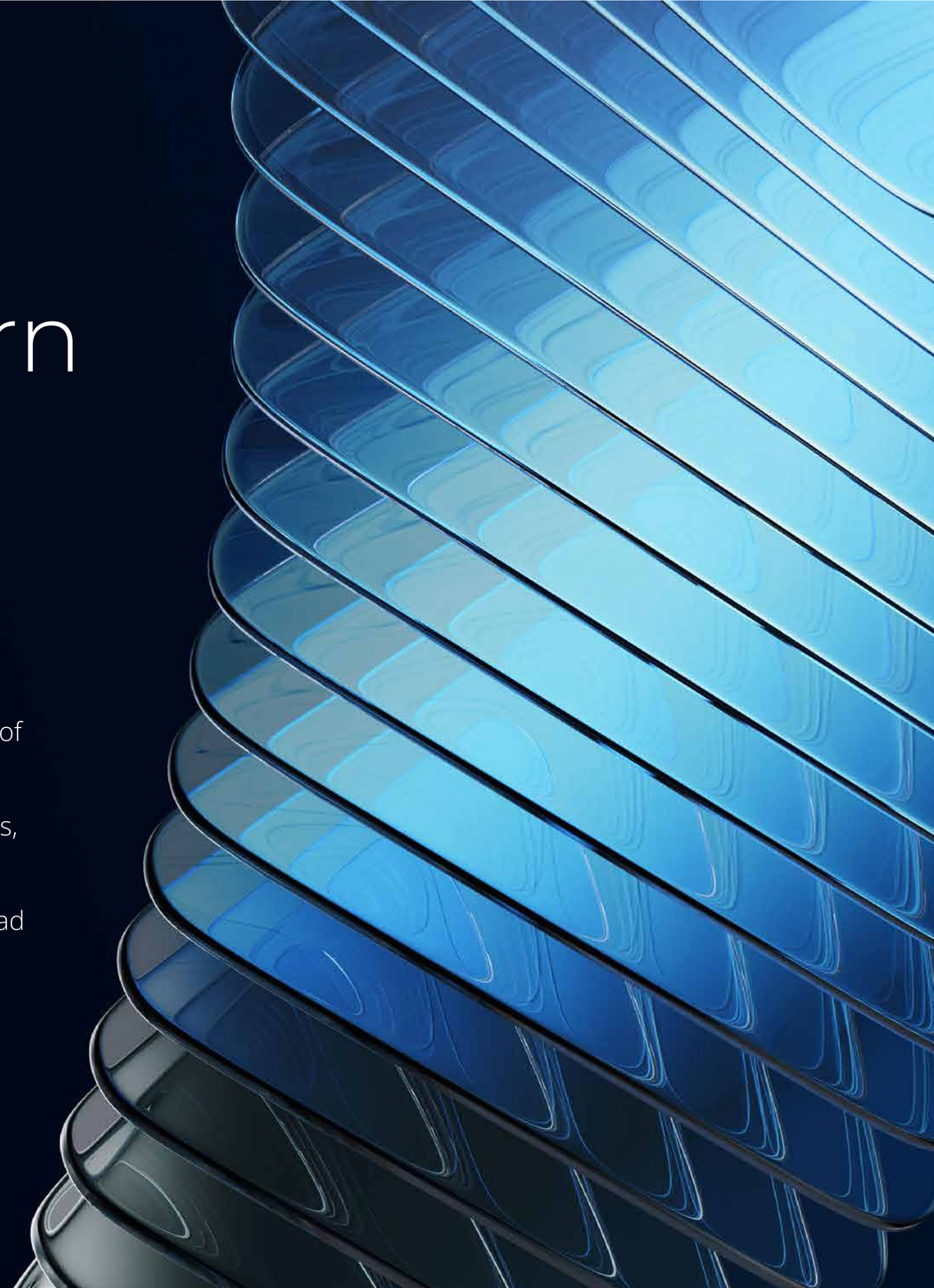
Banks don’t set out to create complex integration environments. They accumulate them.

New platforms are introduced. Banks acquire businesses. Payment rails evolve. Products change. Legacy systems remain because replacing them is difficult, risky or simply unnecessary. Over time, different parts of the organisation begin defining and exchanging the same banking information in different ways.

The result is an integration landscape in which every new programme can trigger another round of analysis, modelling, standards interpretation, mapping and bespoke translation.

That work may not appear as a line item called integration complexity, but its effects are familiar: longer lead times, repeated effort, growing dependencies and increasing reliance on people with scarce banking and standards expertise.

bbd
software development



Contents

Banking integration carries a hidden tax 03.

Is this relevant to your bank? 04.

Why the problem compounds 06.

Rosetta: BBD’s Banking Interoperability Accelerator 09.

Where Rosetta creates value 12.

Modernisation does not need another big-bang programme 14.

Proven in live banking environments 15.

One shared language, applied to your banking environment 17.

Frequently asked questions 21.

01 Banking integration carries a hidden tax

Modern banking depends on constant change. New payment capabilities must be introduced, legacy platforms need to coexist with modern services, customer and party information has to move between systems, ISO 20022 requirements continue to shape payment environments, and new products need access to data and capabilities that may sit inside platforms built decades apart. Yet the architecture underneath that change can make every programme harder than it needs to be.



The hidden tax is not integration itself.
It is the amount of integration effort spent repeatedly rebuilding the same foundations.



THE SAME MODELLING WORK GETS REPEATED

Architects and analysts repeatedly reinterpret standards and remodel payment workflows that have already been solved elsewhere.



POINT-TO-POINT RELATIONSHIPS MULTIPLY

Every bespoke system-to-system connection becomes another dependency that needs to be understood when either side changes.



SPECIALIST EXPERTISE BECOMES A BOTTLENECK

ISO 20022, BIAN and deep banking-domain knowledge are valuable precisely because they are scarce. When every integration requires that expertise from first principles, specialist capacity is spent rebuilding foundations rather than solving what is genuinely unique to the programme.



MODERNISATION BECOMES PROGRESSIVELY HARDER

The more tightly systems are woven together through bespoke relationships, the more difficult it becomes to change one component without understanding the potential for technical debt.



THE SAME BANKING CONCEPTS ARE DEFINED DIFFERENTLY

A customer, party, account or payment can be represented differently across platforms. Before systems can exchange information reliably, teams first need to establish what those concepts mean in each environment.

02 Is this relevant to your bank?

Rosetta, BBD's Banking Interoperability Accelerator, is most valuable where integration complexity has become a recurring delivery problem rather than an isolated technical task. If several of the following feel familiar, the issue may be the absence of a reusable integration foundation.

INTEGRATION EFFORT IS CONSUMING TOO MUCH OF THE PROGRAMME RUNWAY

Too much early-stage effort goes into understanding existing systems and dependencies before teams can focus on the change the programme is actually meant to deliver.

THE SAME CUSTOMER, PARTY OR PAYMENT MEANS DIFFERENT THINGS IN DIFFERENT SYSTEMS

Data can only move reliably once those differences have been interpreted and reconciled.

ADDING A NEW SERVICE MEANS CONNECTING IT SEPARATELY TO SEVERAL EXISTING PLATFORMS

Each new relationship adds another dependency to design, test, maintain and change.

A SMALL NUMBER OF SPECIALISTS CARRY DISPROPORTIONATE INTEGRATION KNOWLEDGE

Progress depends heavily on people who understand both the banking domain and standards such as ISO 20022 or BIAN.



LEGACY PLATFORMS STILL WORK, BUT INTEGRATING AROUND THEM KEEPS GETTING HARDER

The challenge is not necessarily replacing those systems, but containing their complexity so newer capabilities can evolve around them.

YOUR STANDARDS STRATEGY IS CLEARER THAN YOUR IMPLEMENTATION PATH

ISO 20022 or BIAN may be part of the architectural direction, but translating that intent into practical delivery still requires significant modelling and integration work.



If the same integration questions keep resurfacing from programme to programme, the opportunity is not simply to make the next integration faster. It is to stop starting over.

03 Why the problem compounds

Traditional point-to-point integration works well enough when only a few systems need to communicate. The problem appears as the environment grows.

If System A needs to understand System B, a mapping is created. If System C then needs the same information, another mapping may follow. Add another platform, another payment flow or another region and the number of relationships continues to grow.

Each connection solves the immediate requirement, but collectively they create a web of translations and dependencies.

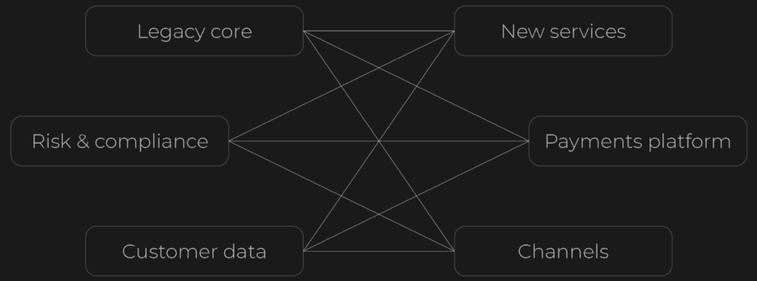
A canonical approach changes that pattern. Instead of requiring every system to understand every other system’s internal language, each participating system integrates around a shared representation of common banking information.



One language in the middle. Different systems at the edges

As systems, payment flows and regions grow, point-to-point mappings multiply. A shared banking language changes the pattern.

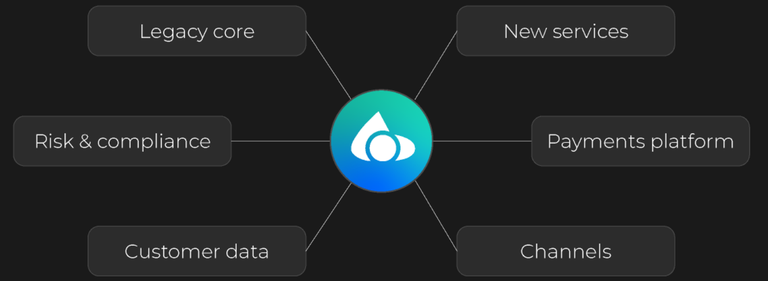
Without Rosetta



Every new connection creates another mapping and dependency.

VS

With Rosetta



Each system connects through one shared canonical foundation.

The business implication: The value is not merely architectural neatness. It is that the work completed for one integration can begin contributing to the next.

What changes when systems share one banking language?

The difference is easiest to see in the work teams no longer need to repeat.

WITHOUT A SHARED FOUNDATION	WITH A SHARED BANKING LANGUAGE
Remodel familiar banking domains	Start from established banking models
Create bespoke mappings between systems	Integrate around a common representation
Reinterpret standards programme by programme	Reuse standards-derived definitions and documentation
Depend heavily on scarce specialists for foundational work	Focus specialist effort on what is unique to the bank
Add another point-to-point dependency	Create a reusable integration boundary
Standardise everything before value can be realised	Apply interoperability incrementally where it matters

The shift is subtle but important. Teams stop treating each programme as an entirely new modelling exercise and start from a reusable foundation that can be validated against the specific environment. That is the role Rosetta plays.

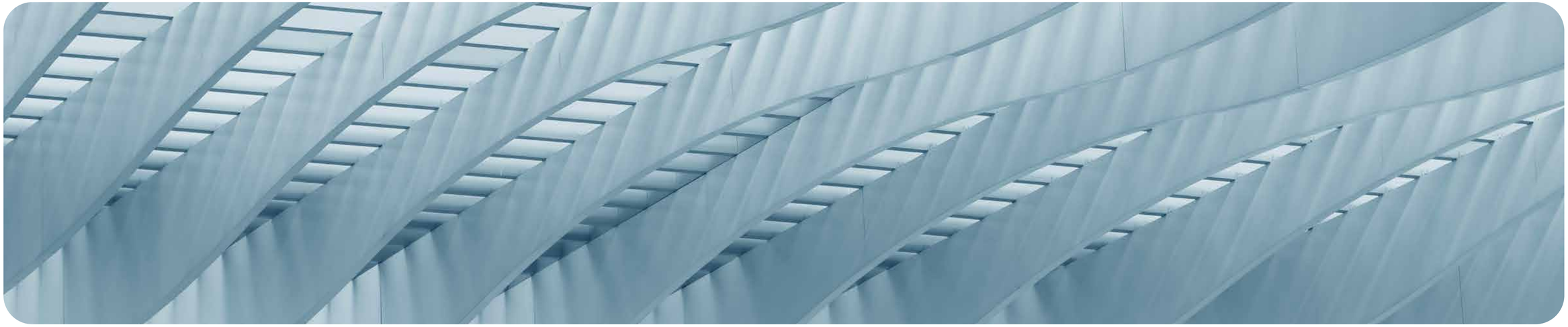
04 Rosetta: BBD’s Banking Interoperability Accelerator

Rosetta gives BBD a reusable ISO 20022 and BIAN-derived canonical foundation for banking integration and modernisation work.

It provides a shared model for common banking concepts, together with standards-based definitions, mappings, translation capability and field-level documentation.

Rather than beginning with a blank page, BBD starts from that foundation and adapts it around the bank’s systems, data structures, regulatory context and programme requirements.

The outcome is one shared language across the banking systems involved, without asking the organisation to replace every platform or impose a single enterprise model before work can begin. Rosetta is already running in production as the foundation of a large bank’s payments modernisation platform.



Rosetta is not another banking platform

It does not replace the core. It does not require the bank to abandon its existing systems. And it is not a standalone software product that teams are expected to implement themselves.

BBD applies Rosetta as part of integration and modernisation delivery, using the accelerator to reduce repeated foundational work while engineering around the realities of the bank's environment.

Start further ahead

One of the least visible costs of banking integration happens before much software is written.

Teams need to establish what information a service requires, how banking concepts relate to one another, how those concepts are represented in the existing environment and how they should map to external standards. That work is necessary. Repeating it from scratch is not.

Rosetta allows BBD teams to begin with an existing, documented standards-based model and ask: *How does this fit your bank?* rather than: *What should a payment, party or customer look like?*



A checklist instead of a blank page

MOVE INTO DELIVERY SOONER	Start with a defined model to validate and adapt rather than building familiar banking domains from first principles.
USE SCARCE EXPERTISE WHERE IT MATTERS	Architects and analysts can spend more time on what is specific to the bank, its systems, regulatory requirements and business model.
BUILD GREATER CONSISTENCY	Shared definitions and structures reduce the likelihood that different programmes independently solve the same problem in different ways.
CARRY WORK FORWARD	Models, mappings, documentation and integration patterns established during one initiative can become part of the foundation for future work.

05 Where Rosetta creates value

1 PAYMENTS MODERNISATION

Payment environments often sit at the intersection of external standards, legacy infrastructure and new digital capabilities.

Rosetta provides a common foundation for payment information and the systems that produce, process or consume it, reducing the need to recreate translation logic across each new flow.

When a second payment rail was added in that live programme for example, the team reused the canonical model, translation service and adapter patterns built for the first, so the new work was the rail itself.

BUSINESS VALUE

Modernise payment capabilities without adding another generation of bespoke integration.

2 LEGACY INTEGRATION

Legacy systems often remain business-critical long after the technologies around them have changed. Rather than forcing those systems to become something they were never designed to be, BBD can isolate their proprietary structures behind adapters that translate information into the shared canonical model.

BUSINESS VALUE

Preserve valuable legacy capability while reducing how much of the organisation needs to understand its complexity.

3 PAYMENTS MODERNISATION

Modernisation rarely happens in one clean switch. Rosetta can provide a stable integration boundary while services and underlying systems evolve progressively.

BUSINESS VALUE

Change the platform without requiring every connected system to change at the same time.



4 NEW BANKING CAPABILITIES

A new product or service frequently needs access to existing customer, party, account or payment information. Without a common model, every new capability risks creating another bespoke set of mappings.

BUSINESS VALUE

Connect new capabilities without continuously increasing point-to-point complexity.

5 ISO AND BIAN LED PROGRAMMES

Standards provide the architecture. They do not automatically provide implementation. Rosetta helps BBD move from standards-led intent towards an executable integration foundation, adapted to the bank's environment.

BUSINESS VALUE

Reduce the gap between architectural ambition and delivery.



06 Modernisation does not need another big-bang programme

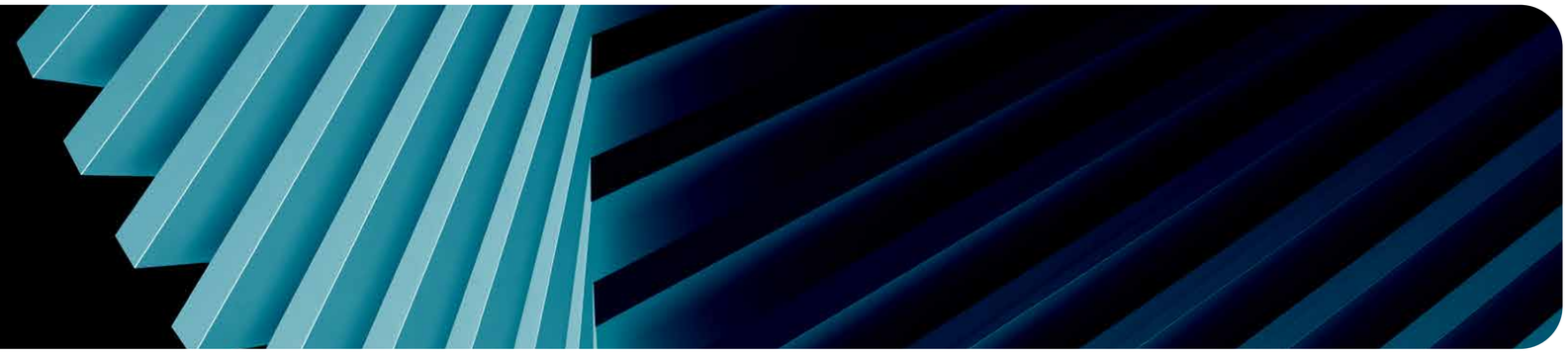
Standards initiatives can make business leaders nervous for good reason. They can sound like years of architecture governance, organisation-wide alignment and foundational work before tangible value appears.

Rosetta is not positioned that way.

BBD can apply the approach within a bounded initiative: a payment flow, a platform, a group of services or a modernisation programme. As further work is introduced, the shared foundation can be extended.

That means a bank can improve interoperability system by system and programme by programme, without waiting for every part of the organisation to adopt a common model simultaneously.

Start where the value is.
Expand as the bank evolves.



07 Proven in live banking environments

Rosetta is not an architecture diagram waiting for its first implementation. Its underlying model and translation capabilities are already running in production within a live payments-modernisation engagement.

Creating a consistent view of customer and party data across systems

One practical application involves party reference data. The bank holds customer and party information across different underlying systems and regional environments. Those source systems use different technologies, structures and interfaces, meaning the same customer information can be represented differently depending on where it originates.

Using Rosetta, BBD created a common representation for that information. Requests can retrieve party data from the relevant source systems, translate it into the shared canonical model and return it to downstream services in a consistent format.

The consuming service does not need to know whether that information originated in one legacy platform or another. It works with one representation, while the differences between underlying systems remain behind the integration boundary.



Extending the foundation to accounts and products

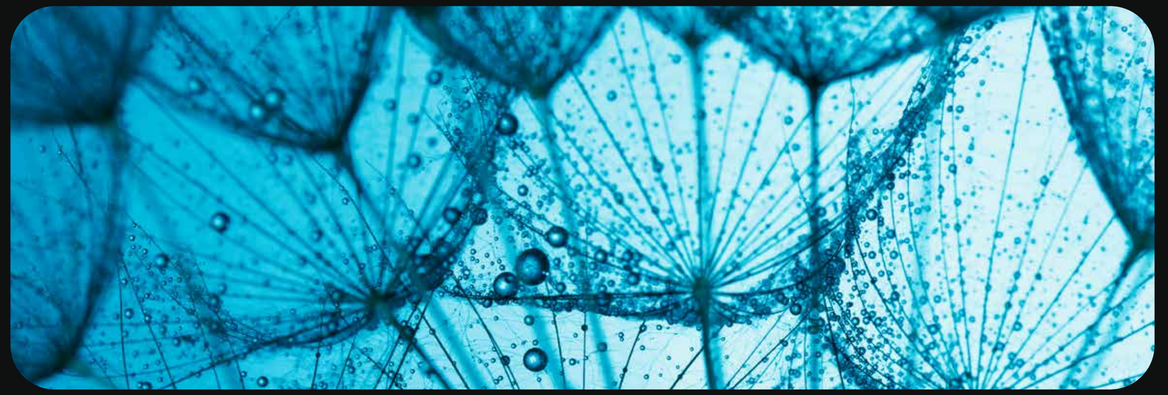
The same approach is now being extended to account and product information. A payment service asks one question, which product agreement an account belongs to, and receives one answer in the shared model. Behind that boundary, the request is routed to whichever system holds the account: a cache kept current by events from the party data service, the core banking platform for the bank's home market, a different core platform for accounts in other countries, or the legacy platform as a final fallback.

Each new domain is added to Rosetta first, and the service is then built on the generated model. The party data service already publishes the account events that the new service consumes, so each delivery builds directly on the last.

THE BROADER OUTCOME

The programme now has a reusable integration foundation that allows modern services to evolve around the bank's existing technology estate without requiring every underlying system to change at once.

That is precisely the kind of modernisation Rosetta is intended to enable.



08 One shared language, applied to your banking environment

Rosetta provides the reusable foundation. BBD does the work required to make that foundation useful within the real environment.

Rosetta provides:

- ISO 20022 and BIAN-derived banking models
- Reusable mappings and translation capability
- Standards-based definitions and code sets
- Field-level documentation and implementation patterns

BBD applies:

- Fit-gap analysis and client-specific extensions
- Mapping to existing systems and data structures
- Legacy and platform adapters
- Integration architecture, regulatory considerations and engineering

Your bank gets:

- ✓ A standards-aligned foundation adapted to the requirements of the banking environment
- ✓ Less need to build and maintain bespoke translation between systems
- ✓ Greater consistency across integrations and programmes
- ✓ A foundation that can be reused and extended as the environment evolves

This is deliberately different from imposing a rigid canonical model on the bank. The BIAN-derived portions can be adapted where client-specific requirements demand it, while ISO 20022 provides the standards-based foundation for relevant payment messaging.

From banking standards to client-ready integration foundation

How BBD applies a reusable foundation to real banking environments.

01 / INPUT

Standard sources

- ISO 20022
- BIAN

02 / PROCESS

BBD generation + curation

03 / FOUNDATION

Rosetta Canonical Foundation

- ✓ Shared banking models
- ✓ Standards-based definitions
- ✓ Reusable mappings
- ✓ Field-level documentation

04 / APPLICATION

Applied to your banking environment

- ✓ Payment platforms
- ✓ Legacy core
- ✓ Channels
- ✓ New services

Standardise the common language. Adapt at the edges.



What sits underneath the simplicity?

Rosetta is built around a canonical model generated from the official ISO 20022 eRepository, with BIAN-aligned service-domain models layered on top.

Its translation service, Rosetta Babel, converts in both directions between that canonical representation and ISO 20022 XML or JSON across the pain (001–014) and pacs (002, 003, 004, 007, 008, 009) message families, including multiple versions of each. Translation is driven by more than 3,000 generated mapping descriptors rather than hand-written code, so supporting another message version is largely a data change rather than a development project.

The architecture can also use adapter boundaries around proprietary or legacy platforms, allowing their internal formats to remain contained while modern services work with the shared representation.

The foundation is accompanied by field-level documentation that gives architects, analysts and engineers clearer context about the models being used. That documentation is generated from the same mapping descriptors that drive translation, so it always describes what the system actually does and cannot drift out of date.



ONE MODEL IN EVERY LANGUAGE

The same model is published for Java, Go, JavaScript and Python, so services written in different languages share one definition of each banking concept instead of maintaining their own copies.

DATA PROTECTION BY DESIGN

Raw payment messages can be parsed once at the edge and held in a separate claim-check store, while downstream services store and process only the canonical form. Personal information lives in fewer places, which supports data-protection obligations such as POPIA.

STANDARDS UPGRADES BECOME A REGENERATION

Because the ISO 20022 layer is generated rather than hand-written, moving to a new ISO 20022 release regenerates the model instead of remodelling it, and a correction made once in the generator reaches every service.



09 Frequently asked questions

CAN WE BUY OR LICENCE ROSETTA?

Rosetta is not sold as a standalone software licence. BBD applies it within banking integration and modernisation engagements, with the work scoped around the outcome you need.

That means you benefit from Rosetta's reusable models, mappings, translation capability and integration patterns without having to procure, implement or maintain another platform yourself.

BBD adapts the foundation to your systems, workflows, data structures, regulatory requirements and programme priorities, whether that is supporting payments modernisation, integrating legacy and modern platforms or establishing a more reusable integration foundation.

The commercial model is therefore engagement-led rather than licence-led: you buy the outcome and delivery capability, not the accelerator itself.

DO WE NEED TO STANDARDISE THE WHOLE BANK BEFORE ROSETTA CREATES VALUE?

No. Rosetta is designed for incremental adoption. BBD can apply it within a bounded initiative, such as a payment flow, platform or group of services, and extend the approach as the environment evolves.

DOES ROSETTA REPLACE OUR CORE BANKING OR LEGACY PLATFORMS?

No. Rosetta is not a core replacement. It creates a shared integration foundation around existing and modern systems, with adapters used where necessary to isolate proprietary or legacy formats.

IS ROSETTA A SOFTWARE PRODUCT OUR TEAMS BUY AND IMPLEMENT THEMSELVES?

No. Rosetta is a BBD accelerator used within our banking integration and modernisation delivery. The client buys the outcome, while BBD applies and adapts the reusable foundation to the environment.



WHAT DOES THE BANK KEEP WHEN THE ENGAGEMENT ENDS?

The models, mappings, translation services and integration code delivered as part of the engagement are handed over into the bank's own repositories and build pipelines.

The bank receives the relevant source code, tooling and technical documentation needed to build, deploy, maintain and extend those components. For the components delivered as part of the solution, BBD grants the client a free, perpetual, irrevocable and royalty-free licence to use and modify them for the client's business, at no additional cost.

The bank can therefore continue to run and extend what has been delivered using its own teams or an appointed third party, with BBD available to support further changes where required.

WHAT IF OUR SYSTEMS DO NOT CONFORM NEATLY TO ISO 20022 OR BIAN?

They don't need to. BBD uses fit-gap analysis, client-specific extensions, mappings and adapters to connect the standards-based foundation to the systems and data structures already in place. BIAN-aligned areas can be adapted where client requirements demand it. In the live programme, Babel normalises a counterparty's non-standard payment status report into the standard form, so downstream services never see the deviation.

IS ROSETTA ONLY RELEVANT TO PAYMENTS?

Payments are a strong and proven application because ISO 20022 is central to modern payment environments. The same approach can also support party data, legacy integration, platform modernisation, new banking capabilities and BIAN-led banking domains where a shared model reduces repeated integration work.

HOW DOES ROSETTA HELP TEAMS MOVE FASTER?

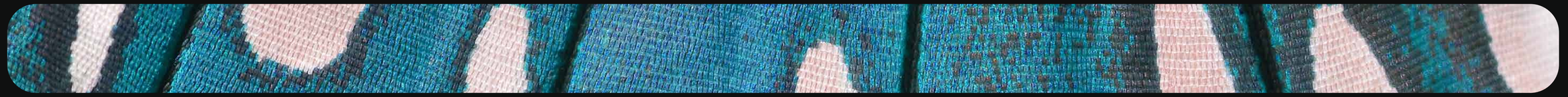
It changes the starting point. Rather than asking architects and analysts to model familiar banking concepts from a blank page, BBD begins with documented standards-based models, mappings and definitions that can be validated and adapted. This reduces repeated foundational work, but any time or cost improvement should be measured within the specific engagement rather than assumed.

Make every new integration build on work already done.



DOES STANDARDS-BASED MEAN MAINTENANCE-FREE?

No. Rosetta's ISO 20022 layer is generated from the official ISO 20022 eRepository and its BIAN-aligned models are curated alongside it, so upstream changes still need curation and governance. What changes is where that effort lands. Moving to a new ISO 20022 release is a regeneration rather than a remodelling project, and a fix made once in the generator reaches every service. In a live programme, a single generator fix corrected 5,830 inconsistent definitions across the whole model in one regeneration.



WHERE SHOULD A BANK START?

Start with a bounded area where integration friction is already visible and where reuse will matter: a priority payment flow, a platform modernisation initiative, a legacy integration challenge or a new capability that needs to connect to several established systems. BBD can then assess fit, adapt the model and extend the foundation as value is proven.

WHAT THIS MEANS FOR YOUR NEXT BANKING PROGRAMME

Every integration project leaves something behind. The question is whether it leaves another bespoke dependency or a foundation the organisation can reuse.

Banks can continue treating every new system, platform and payment flow as a fresh integration problem. Or they can start creating an environment in which common banking concepts are defined once, translation logic becomes reusable and each programme makes the next one easier to deliver.

Rosetta gives BBD a practical way to help make that shift. Not through wholesale replacement, an enterprise-wide standards mandate or by introducing another platform that the bank has to own.

But by bringing a reusable, standards-based banking integration foundation into the work already being done.



Talk to our banking experts

Speak to your BBD contact and our Rosetta expert about how we can help reduce integration complexity and build a more reusable foundation for banking modernisation.

CLICK HERE FOR MORE INFO →

CLICK HERE TO CONTACT US →



JOHNATHAN BOTHA

Payments Expert

johnathan@bbd.co.za

+27 72 293 5749

bbd
software development