Making BIAN Machine-Readable

How an integration team built a company around one problem: mapping a bank's systems to BIAN without months of manual work



The Challenge

R34DY's founders came from banking integration. From inside banks they saw the same pattern again and again: fragmented systems, decades of technical debt, and a clear need for a shared standard. BIAN was, in their assessment, the best framework available. The obstacle was not the standard itself but the first step toward it.

Before a bank can align anything with BIAN, it has to understand its own landscape and map existing systems, data and processes to service domains and service operations. That work was manual, slow and expensive. Teams struggled to find the connections between their current architecture and BIAN's service domains, even with large effort, and the bank's ecosystem rarely fitted the BIAN object model cleanly, which left details and domain boundaries open to interpretation. The practical result: BIAN adoption stayed with the largest institutions, the ones that could afford the discovery phase.

Why a Standard, Not a Proprietary Model

R34DY evaluated the standards it knew from integration work: ISO 20022, SWIFT message standards, Open Banking APIs and proprietary frameworks. BIAN was chosen as the default framework for its service-oriented architecture and its coverage of banking operations as a whole. ISO 20022 remains in the platform where it belongs, for messaging. The company did not migrate from another standard; BIAN was the starting point.

© 2026 BIAN e.V.

Inside ABLEMENTS

The platform, called ABLEMENTS, was built from 2022 and has been in production since February 2025. Its approach, which R34DY calls Context Engineering, builds a semantic picture of a bank's systems, data and processes using vector embeddings and domain filtering, and uses that picture to propose how existing operations map to BIAN service domains.

Five modules cover the ground: Systems, Data, Process, Architecture and Flow Operations. Each analyzes part of the existing landscape and suggests BIAN alignments, so a bank can start where the pressure is highest and widen the scope later. A Business Configuration Module carries pre-configured BIAN domain filters that classify discovered systems and processes to service domains. A Framework RAG API validates results against BIAN documentation as the standard evolves, and a digital-twin function lets a bank simulate a transformation before committing to it. R34DY positions the platform as a partner to consultants and implementation teams: the discovery phase is automated so that people can spend their time on the transformation itself.

What Proved Harder Than Expected

R34DY's submission describes the difficulties as engineering problems, and three of them are directly relevant to anyone adopting the standard:

Turning BIAN into logic a machine can apply

The service landscape is written for architects. To let software classify systems automatically, R34DY had to translate the service domain architecture into machine-readable rules and pre-configured domain filters.

Bank landscapes that do not fit the model cleanly

Business Capabilities do not decompose neatly into service domains. Where the BIAN object model left room for interpretation, the automation had to make that interpretation explicit rather than hide it.

A standard that keeps moving

BIAN publishes new releases. Mappings that were correct against one version drift against the next, which is why R34DY built real-time validation against the current documentation instead of a one-off import.

The common thread is that the standard's depth is also its cost of entry. R34DY's answer was to absorb that cost in software so that a bank does not have to pay it in people.

© 2026 BIAN e.V.

Impact & Results

2–4 weeks

for BIAN mapping, from 6–12 months of manual work, as stated by R34DY

95%+

accuracy of automated service domain mapping, as stated by R34DY

85%

less manual effort for bank and consultant teams, as stated by R34DY

Feb 2025

in production, implementation since 2022

The figures above are stated by R34DY and have not been independently measured; no customer project is named in the submission.

> *Our real goal: make BIAN accessible to everyone through easy implementation, democratizing the best banking standard across institutions of all sizes.*
>
> – R34DY, BIAN Transformation Awards 2025 submission

What's Next

R34DY plans to deepen service domain coverage across all five modules, keep the Framework RAG API current with new BIAN releases, and add collaborative features so that banks can share implementation patterns. The company also intends to feed insights from automated discovery across its client base back into the BIAN community.

The Takeaway

Most case studies in this series describe a bank adopting BIAN. This one describes a vendor betting on it: a company founded on the assumption that the standard is right and that the only thing in its way is the cost of the first mapping. The bet itself is already an argument for the standard.

Every case study in this series began as an award entry. The BIAN Transformation Awards 2026 are open.
Entries close on March 16, 2027 – find the submission area at bian-services.com

This case study is based on R34DY's submission to the BIAN Transformation Awards 2025, in the "Best Use of Innovative Technology" category. All figures are as stated by R34DY and have not been independently measured.

© 2026 BIAN e.V.