

Simplifying Core Banking with API Orchestration



Contents

Introduction

3

Mapping Core Banking APIs for Our FinTech Client

4

Delivering a Seamless Conversational AI Experience

6

Validating Data Mapping

7



Introduction

Our client is a FinTech company that provides organizations in multiple industries, including banking, with fully automated AI assistance. Using conversational AI such as messenger bots and automated phone answering systems, our client enables these companies to offer an efficient and seamless customer service experience. Many of our client's partners are financial institutions that use core banking so their customers can make transactions at any of the bank's branches or online. To provide a consistent service that works over multiple banks, our client needed to reliably and predictably perform data integrations with a variety of core banking APIs. In addition to data extraction and data integration, they needed data orchestration to enable their conversational AI assistant to operate across partners consistently.

Lineate was hired to help our client build a number of these core banking integrations, including with third-party APIs Jack Henry Sycitar and Fiserv Banking Hub. We were chosen because of our extensive experience integrating a set of diverse third-party APIs under a single data orchestration layer for various industries. For example, we have extensive experience integrating and orchestrating third-party data platforms to create holistic customer views not just for financial data but also for marketing and engagement platforms.

The key challenge in controlling multiple third party services using a single technology is not simply performing the data integration, but providing a platform with a unified way to take action across the universe of services regardless of how they structure their data. We refer to this ability to activate differently structured data in the same way as data orchestration. This process doesn't just connect to and aggregate data from disparate sources; it also creates a master set of concepts and maps them across many providers that may represent each of these concepts in fundamentally different ways.

Mapping Core Banking APIs for Our FinTech Client

At Lineate, we typically start such data integration and unification projects by generalizing various object taxonomies into a single data orchestration model. For example, two providers might categorize users by age range but use different age groupings (say 18–29 versus 18–34). An orchestrated system would need abstract away such differences but remain able to take appropriate actions based on the overall concept of age.

Our FinTech client had already created a unified API as well as a tool to structure and simplify the development of mappings between data representations. But they needed to add new core banking APIs, which required writing mapping code into the system as well as building out the third-party data integrations. Core banking APIs are not simple, and any data integration requires encapsulating dozens of entities. At a high level, we have entities such as Account, Transactions, and Statements. But at the API level, these entities are often separated into entirely separate endpoints with no direct relation to each other (think Loans vs. Checking vs. Savings). These data taxonomies can be enormously complex. For example, the Fiserv DNA database has over 1500 tables.

The complexity of data integration for a diverse set of data taxonomies in the core banking domain is illustrated in our integration of Jack Henry Symitar. Symitar's API exposes internal database structures normalization to the API level, making any data integration highly dependent on the particular internal data implementation used by the API.

This dependence on a particular implementation made loading even relatively simple objects fairly complicated. For example, to load a flat object with Bank Account information we needed to fetch data from the Account, Card, Loans, and Shares objects in Symitar. And to update a single field for Account we needed to perform additional calls in order to load all the fields required by an update call. First, we had to understand whether its type was Share or Loan. This determined where we would look for details such as opening date and balance. For both Shares and Loans, we also needed to pull the Card object to read its limits and card number. Then, to load Transfers we had to go to either Shares or Loans for details.

We've shared only a couple examples of the sophisticated mapping we had to work through to translate Symitar's domain model understanding into that of our client's. The most complex parts of this data orchestration were analyzing the requirements of and mapping between APIs, including filling gaps in the functionality. We had to read thousands of lines of WSDL and other documentation in order to produce two lines of high-level mapping. In the end, we mapped 19 endpoints used by our client's downstream assistant systems into 5 services covering 30 functional areas of Symitar's SOAP API.



Validating Data Mapping

In the FinTech and banking industry, especially when it comes to core banking, the cost of mistakes is high. So we enforced the accuracy of the data mapping at every level. Our Solution Architects and engineers worked closely with the API vendors to ensure the correct use of endpoints to get specific data. Code paths were extensively covered with unit tests. Validation processes were planned from the start of the project. Automated verification was built in a clean room so that the logic of the verification was developed independently of the transformations and mappings of the integration code itself.

Delivering a Seamless Conversational AI Experience

Many industries, especially FinTech, often need data orchestration, which consists of a single set of unified tools running on top of diversified third-party APIs. These tools present data with different structures as a unified system. More importantly, data orchestration allows users to interact with a platform that covers a set of different third-party APIs without knowing what's under the hood. So, in the case of core banking, the stakes are higher because the cost of a mismatch or mistake is so high.

One of our client's top priorities is to create a positive experience for customers interacting with conversational AI. People using automated banking assistants do not want to be led through a complex phone menu or answer dozens of questions asked by a silly messenger bot. So the more we can utilize data orchestration to unify all those underlying differences and complexities, the simpler the automated experience will be for the end user.

At the end of the day, Lineate resolved our client's difficulties with data integration across multiple APIs using core banking data orchestration. Our solution was focused around two key points: carefully planning and managing the mappings between concepts between implementations, and employing incredibly rigorous testing and independent automated verification. With this combination we delivered a simple, seamless, and most importantly, reliable conversational AI experience for both our client and their banking customers.

Thank you.

Can we help you with your
ambitious goals?

Talk to us today at
lineate.com/contact

 Lineate