Legacy modernization blueprint:

A global bank's 3 step strategy for unlocking legacy systems with Anypoint Platform

Introduction

Few industries feel more pressure to modernize legacy systems than banking.

Banks operate within an uncertain and regulated economic environment and, as a result, their IT teams have been pressured to increase operational efficiency. Modernizing legacy systems - which consume over 75%[1] of banking IT spend - has been identified as a means to do so by simultaneously reducing costs and increasing the speed of software delivery.

Yet, many organizations still struggle with how to modernize their legacy systems. One large global bank provides a compelling example of how financial institutions, as well as organizations across other industries, can do so using MuleSoft's Anypoint Platform.

---

1 https://www.temenos.com/en/blog/2015/july/the-writings-on-the-wall-for-the-bank-batch-run/

The need for legacy modernization at a major financial institution

As one of the top ten largest banks, this bank and its IT teams operate on a massive scale, supporting tens of millions of consumer and small business relationships through thousands of retail financial centers, ATMs, and digital banking accounts.

The bank's Central IT team serves as a custodian of core systems of record—from mainframes and midrange systems (e.g. IBM's AS/400) to databases. The team works to provide and govern access to data assets for business technology teams, which serve the bank's various business lines (e.g. mortgages, global wholesale, retail banking). In addition, they govern access to externally provided services from vendors that different business technology teams must interface with.

Prior to adopting MuleSoft, the bank met customers' digital needs through a complex web of tightly coupled systems of record. As the bank made acquisitions and developed new applications on top of pre-existing code, this complexity only increased. As a result, applications became even more tightly coupled, and internal dependencies across applications increased. This lengthened the amount of time it took the Central IT team to make changes to existing applications and develop new ones, making it even more challenging to serve the growing demands of the business.

The challenges Central IT faced in cost-effectively maintaining their own applications and enabling rapid business IT innovation were a contributing factor to the bank's suboptimal efficiency ratio (expenses over revenue), an industry benchmark that they lagged behind peer banks on.

In response to a CEO-led initiative to increase operational efficiency and improve their efficiency ratio, the Central IT team embarked upon a strategy to address these challenges by modernizing their legacy systems with MuleSoft's Anypoint Platform.

# The bank's legacy modernization strategy

The bank's modernization strategy was designed to solve a number of legacy system challenges. To improve efficiency, Central IT aspired to control maintenance costs and increase agility by reducing dependence on system experts. Whereas some organizations pursue a path of connecting legacy systems to modern endpoints through point-to-point custom code, the bank concluded that this strategy would lead to even tighter coupling across their application stack, and make new projects even more time-consuming to deliver.

"We want to avoid point-to-point connectivity," explained the Senior Vice President of IT and leader of the initiative. "We didn't want to do an implementation for each service we're developing with its own rules."

Instead, Central IT aspired to surface legacy system functionality to line-of-business teams through decoupled, reusable services that could be easily discovered.

To do so, the team envisioned a three-tiered, API-led architecture, implemented with MuleSoft.
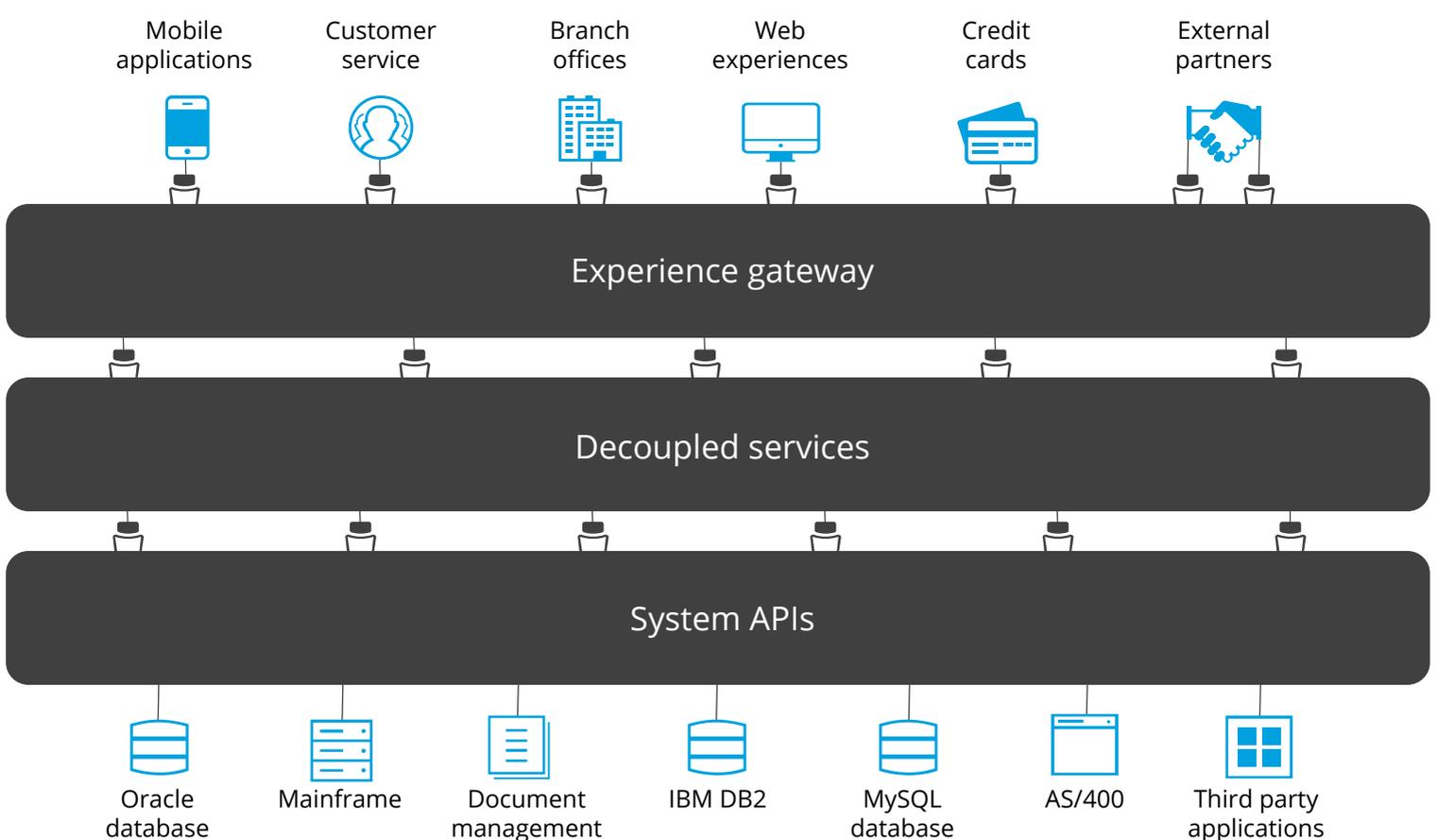
In this API-led architecture, **system APIs** provide governed access to their respective downstream systems. They also serve as wrappers that insulate these systems from excess traffic received by upstream applications. **Decoupled services** orchestrate data and functionality from these system APIs, abstracting away system complexity from business technology consumers. An **experience gateway** encapsulates business logic managing the authentication, authorization, and auditing of traffic that consume these APIs across different channels.

To implement this architectural approach, the bank identified a three-part strategy:

1. Developing system APIs to surface and govern data from source systems to upstream business processes.

2. Decomposing monolithic web services exposed by key systems of record into discrete microservices that can be easily used by business IT teams.

3. Re-architecting legacy process-level services and code to decouple them from source systems of record—making them more adaptable to evolving business requirements.

Below, we will detail how the bank used MuleSoft's Anypoint Platform to deliver on this strategy, and the benefits they realized in doing so. To do so, we'll take a closer look at how the successful implementation of these integration use cases improved how the bank serviced mortgages for its customers.

Redefining mortgage servicing by modernizing legacy systems with Anypoint Platform

Mortgages represent a microcosm of the legacy system complexity underpinning key banking business processes at the bank.
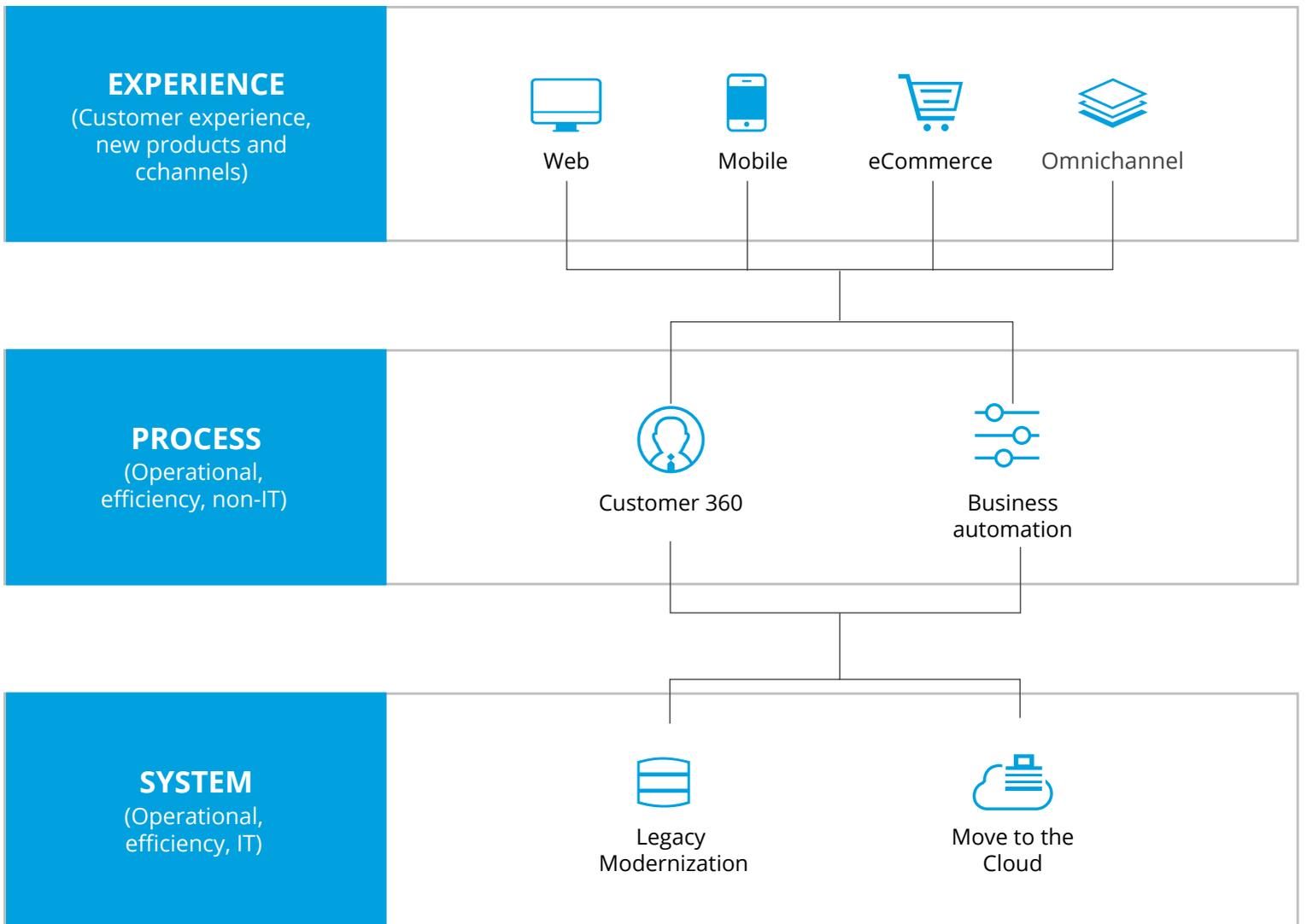
While many banks aspire to move toward a "mortgage-as-a-service" operating model – where mortgage-related data is accessible to any internal bank consumer that needs to make a credit eligibility decision – this vision has been complicated by the fact that loan information exists across a variety of legacy systems and databases. At many banks, collecting this data from different systems has been a largely manual process, leading to higher costs for loan servicing and a lower efficiency ratio. This is why the Central IT team made it a priority to modernize the systems containing loan data so that credit eligibility determination could be automated across business units.

Previously, the shared services supporting mortgages powered over 400 different internal and consumer applications, and each of these services was tightly coupled to the systems of record they consumed data from.

Furthermore, the bank was migrating over 300 applications supporting mortgage services to an external vendor. This vendor would be responsible for providing loan servicing and customer support for all mortgages offered through the bank, demanding seamless interoperability between the external vendor and any bank applications requiring mortgage data.

Last but not least, the underlying code for core mortgage-related services, written in .NET, had evolved into a "ball of mud" that was time-intensive to adapt whenever business requirements changed.

The bank's modernization strategy aimed to address each of these challenges in order to accelerate project delivery for applications requiring mortgage data and to increase the speed at which services could be adapted to meet evolving business needs.

EXPERIENCE (Customer experience, new products and cchannels)	Web	Mobile	eCommerce	Omnichannel
PROCESS (Operational, efficiency, non-IT)		Customer 360	Business automation	
SYSTEM (Operational, efficiency, IT)		Legacy Modernization	Move to the Cloud	

Legacy modernization serves as a foundational initiative to drive the development of modern customer experiences

Unlocking access to systems of record with APIs

In order to realize their vision of a decoupled architecture, the bank needed to abstract legacy system of record complexity from the services that consumed their data. To do so, they decided to represent the functionality from each web service operation through a microservice that would be governed through an API contract.

"With our approach, any connectivity to any service layer should be a separate component. If we have a service that gets data from one system of record, it should be represented as a single entity," explained the SVP of IT.

With MuleSoft's API designer, the bank was able to implement a design-first approach to their System APIs. "With our previous approach, we'd start writing code and design would come after the fact. By leveraging API Designer to build RAML API specs prior to implementation, we were able to more easily align with the clients consuming the data," said the SVP of IT.

With MuleSoft's unified platform for API lifecycle management and connectivity, the bank was able to not only design APIs to access system-level data, but also develop implementations within Anypoint Studio to expose this data in XML or JSON format. By leveraging MuleSoft's robust ecosystem of connectors, as well as Anypoint DevKit to develop custom connectors when needed, the bank rapidly developed reusable interfaces for each core system of record holding mortgage data.

Anypoint Platform also enabled the bank to develop and test these interfaces more quickly. According to the application development lead on the project, by leveraging DataWeave, MuleSoft's drag-and-drop graphical data mapping capability, developers were able to easily visualize the complex legacy data structures that their systems of record exposed, and map different fields to an XML or JSON payload. This enabled them to realize up to 5X increases in integration development speed. And with MUnit, a drag-and-drop unit-testing capability native to Anypoint Studio, developers were able to apply unit and functional tests on these integrations 6X faster than when they had to write their own testing scripts.

The end result was a foundation of data and services from the bank's systems of record, with the complexity of these systems abstracted away from consuming process level services.
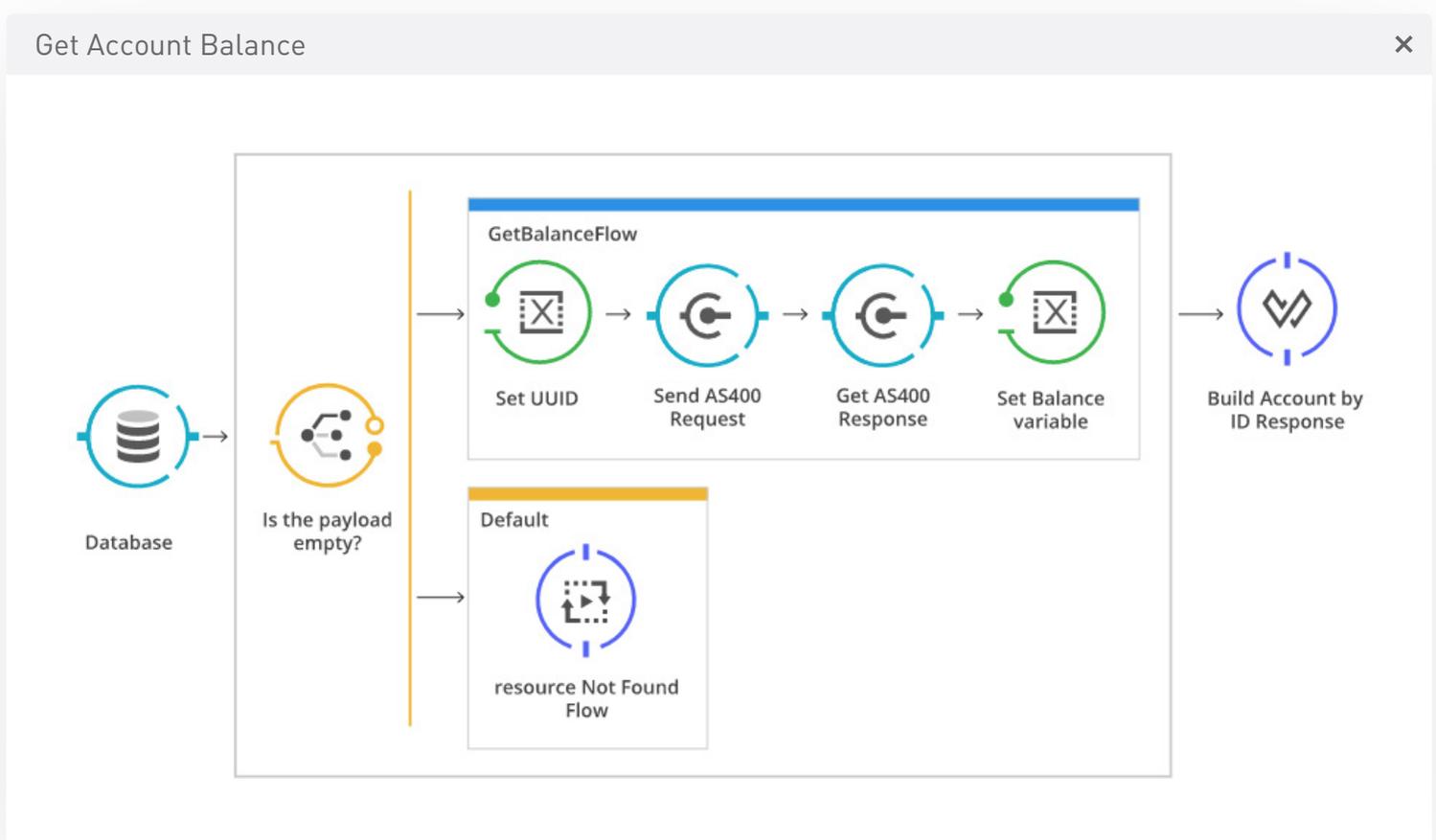
Drag and drop integration development helps power 5X faster legacy system connectivity.

Exposing web service functionality through fine-grained RESTful APIs and SOAP services

In addition to unlocking self-serve access to their own internal legacy systems, the bank also needed to enable the same functionality from the external mortgage vendor, which provided the underlying technology and personnel support for supporting loan servicing.

Supporting common business processes required providing applications with real-time, granular, bi-directional access to the external mortgage vendor.

None of these capabilities were natively supported by the external mortgage vendor. While it made "copies" of its data available to the bank's source systems, these were only delivered via batch file on a nightly basis, so this record could not be relied on for business processes demanding real-time data, such as determining eligibility for credit card applications.

And even though the external mortgage vendor surfaced a set of web services that could provide real-time data, these web services were not suitable for the bank's needs, since each featured tightly coupled operations that could not be exposed as-is to line-of-business IT consumers. For example, since each of the SOAP services included full create, read, update, and delete functionality for the data objects they represented, Central IT could not allow teams who were only authorized "read" access to use these services.

These tightly coupled operations also stymied Central IT's ability to create composite services using data from both the mortgage service vendor and internal systems. For example, the team wanted to create a composite capability — loan comments — that provided every note and comment made on a given customer's loan history. Without decoupling the "read" operations from these web services, the team would have to take a more time-consuming, heavyweight approach.

To address these challenges associated with tight coupling, the team aligned around an approach of decomposing each of the 70 WSDLs provided by the external mortgage vendor into discrete microservices, with each operation in the WSDL being represented by its own microservice, invokable via API contract.

Rather than manually developing hundreds of microservices to represent the mortgage vendor's functionality, the team built a "WSDL decomposer"— a component that dynamically ingests a monolithic WSDL, and exposes each operation's functionality through a distinct SOAP and REST service.

By leveraging the APIs exposed by Anypoint Platform, the bank was able to develop both the WSDL decomposer, as well as the supporting architecture for the automation pipeline, in less than 2 months. Once they had done so, they were able to use it to automatically generate over 600 microservices, each with their own RESTful and SOAP version.

In addition to leveraging Anypoint Platform to automate the decomposition of the vendor's web services, the bank also used the platform's APIs to automate uploading the microservices to Anypoint Exchange for line-of-business consumption.

"Before MuleSoft, the bank didn't have a way to publish and make APIs available. The only approach was an HTTP server — a tedious process to make these assets available," explained the SVP of IT. "With Exchange, you can go and find and search APIs that we've automatically uploaded to the platform."

To enable this capability, the bank configured the WSDL decomposer to take all of the constituent services, create an entry in Exchange for each with the Anypoint Exchange APIs, and upload the decomposed SOAP services. To surface the same functionality through RESTful APIs, the bank automated the process of mapping web service CRUD operations to their HTTP method equivalents (e.g. "create" to "post," "read" to "get") to create a RAML file. For each RAML file, the bank automated the process of uploading the RAML file to Exchange, creating the API, automatically generating examples, and publishing the API.

In total, automating both the decomposition of monolithic web services into discrete microservices, as well as the publishing of these assets into Exchange, drove an over 6X increase in productivity for the team.

The below architecture provides an illustrative example of how such an approach served different line-of-business teams.
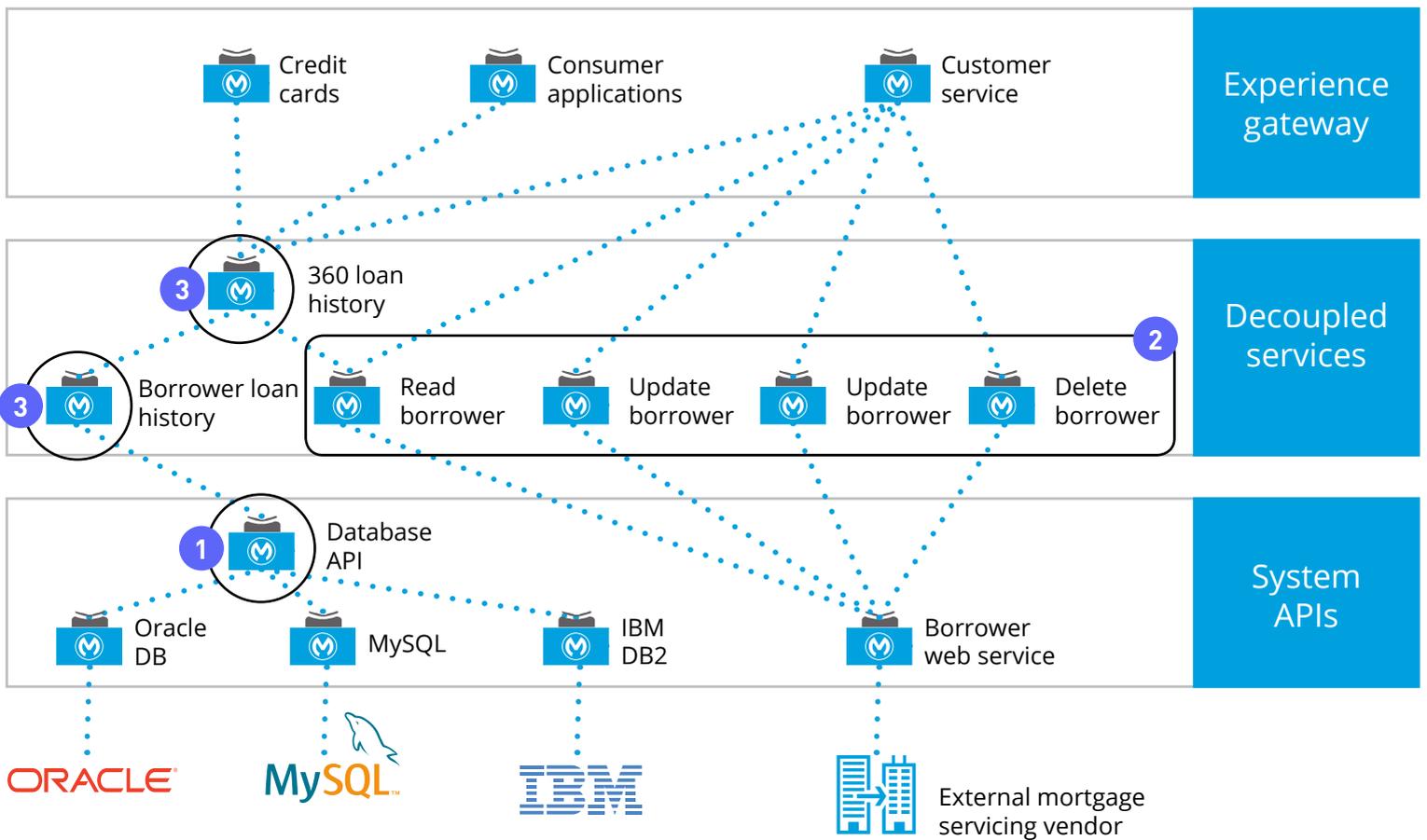

The bank's digital transformation reference architecture, supported by 3 distinct legacy modernization strategies

1 Strategy 1

Develop system APIs to surface and govern data from internal systems

2 Strategy 2

Decompose monolithic web services into discrete microservices

3 Strategy 3

Re-architect .NET process level services and decouple them from source systems

In this architecture, different line-of-business consumers responsible for developing applications requiring mortgage date receive selective access to distinct CRUD operations based on their role within the bank. For example, while customer service receives full access to the external mortgage servicing vendor's functionality (including deleting and updating records), consumer applications are limited to reading this data.

By enabling both self-serve access to legacy system data, and governed access to these legacy system assets, Central IT has increased the speed at which line-of-business IT teams can update over 400 different applications to meet evolving customer requirements. And with the automation pipeline they've built with Anypoint Platform's APIs, the central IT team can easily take the web services exposed from any future platform requiring modernization and decompose it in minutes instead of months.
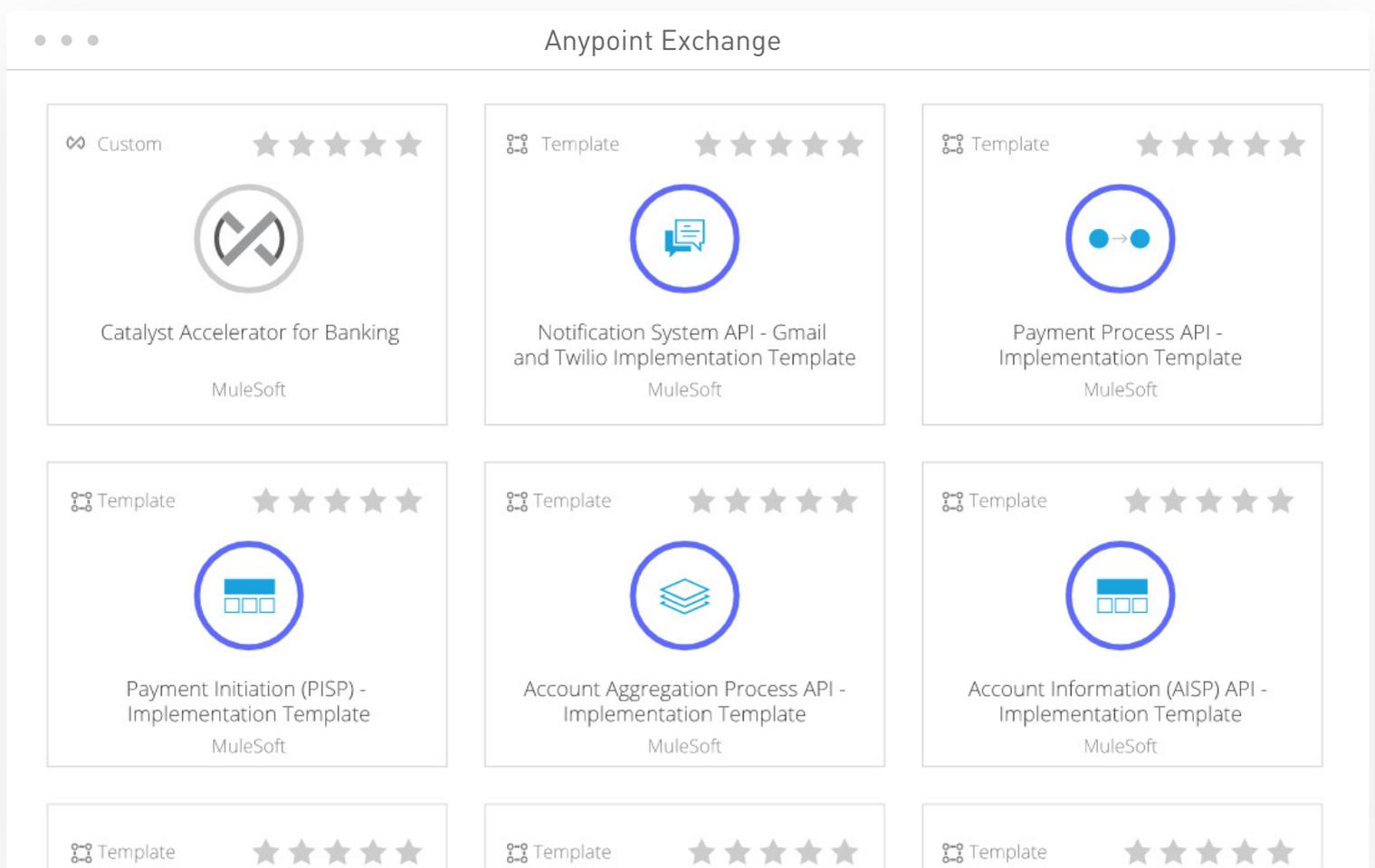
Anypoint Exchange enables line-of-business IT teams to easily self-serve from legacy system assets.

Re-architecting legacy .NET code to increase IT agility

As part of the process of modernizing core systems of record, the bank identified the need to re-architect the legacy .NET code underpinning application services.

Over the years, this code had been gradually reiterated on until it resembled a "ball of mud" that required specialized expertise and significant time investment to maintain and adapt. To make the code more easily understandable and adaptable to changing business requirements, the bank decided to decouple the code from the source systems it consumed data from, and re-architect the code into Mule integration flows (built through drag and drop components to transform and route data across different systems) that made it easy to visually understand a given service's operations and dependencies.

This proffered a number of benefits for the bank. First, it made these services easier to maintain and adapt in response to modern business needs.

"Working with .NET was enormously complex, so representing what was going on with a flow was monumental," explained the SVP. "Previously, making a change to the code required us to rewrite the code itself. With flow-based design, it becomes easier to not only see the code, but change it when needed."

Re-architecting the code also made it easier to implement business process automation within the Central IT function.

"We are doing a lot of business automation processing with orchestration and aggregation of data which has transformations, logic, and different paths we can take," shared the SVP of IT. "It's beneficial to implement something like that with a MuleSoft platform where we can visualize and construct the flows in a way where it's easy to see where the data is flowing."

MuleSoft's out-of-the-box connector for .NET, which allows IT teams to execute native .NET code in a Mule application, enabled the teams to make changes to the code and connect to new systems of record more quickly. "If you were to connect to a SQL or Oracle server, in .NET we'd have to write helper classes. In MuleSoft, it's a connector. You drag and drop, and there you go," said the SVP of IT.

Re-architecting the services with MuleSoft has also enabled Central IT to easily expose services as RESTful APIs returning JSON and XML payloads that can be centrally managed and easily consumed by business IT teams. The end result? The team leading the initiative forecasts that once these refactored services are put in production, they will be able to adapt services to meet new business functionality 30-40% faster than before.
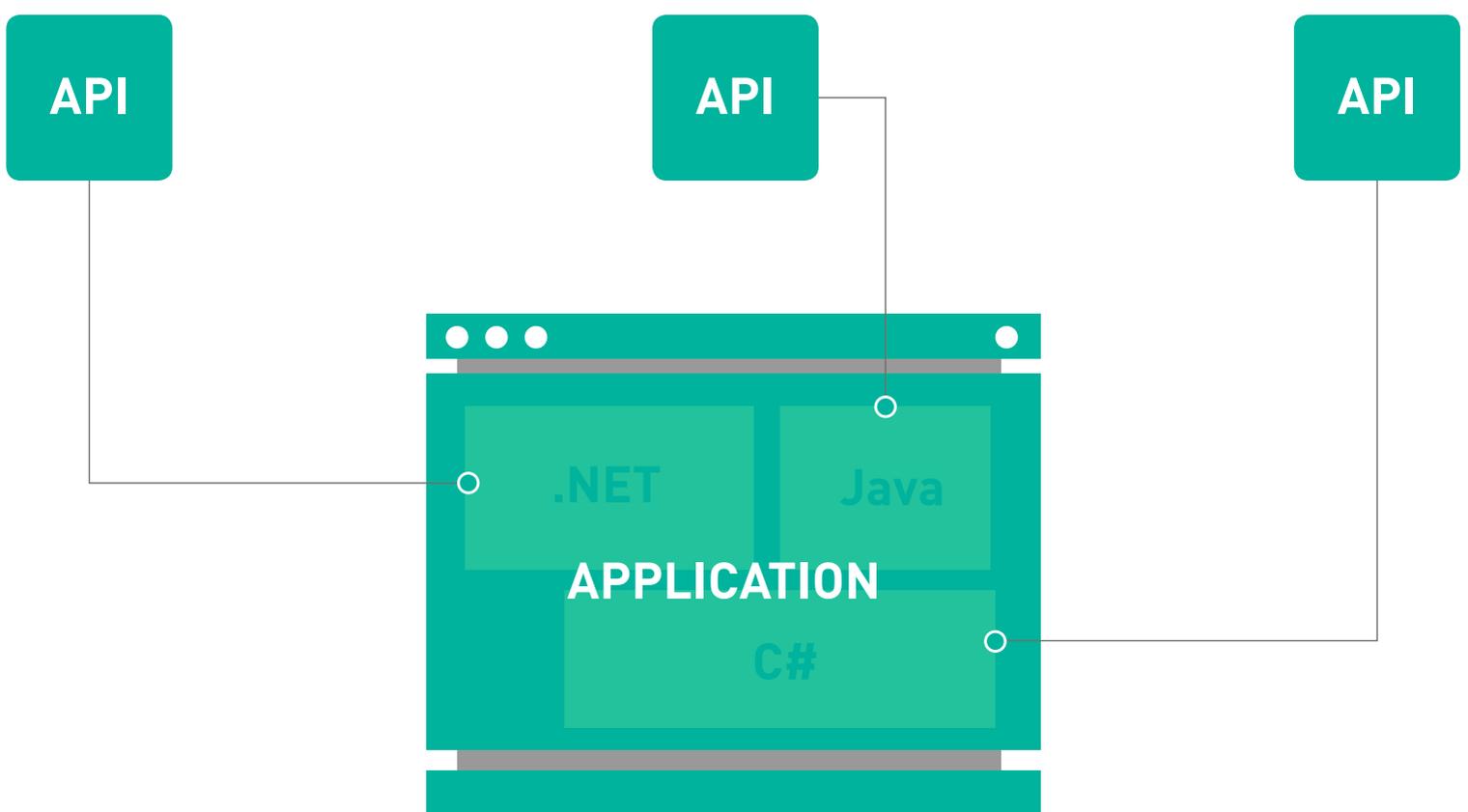
Re-architecting existing legacy applications and exposing functionality through APIs can improve IT agility by up to 40%.

The team driving the initiative identified three key reasons why MuleSoft was selected by the bank to support their legacy modernization initiative.

1. A platform that unifies integration and full API lifecycle management. "Integration is vital to getting data out of our systems of record," shared the SVP of IT. "But to enable the business to self-serve, we saw a real need to expose this data through APIs, instead of just relying on the legacy services we had available."

2. The ability to share legacy system data and services with line-of-business consumers through Anypoint Exchange. "We have over 1000 APIs on our platform, and as we grow, we're going to have a couple thousand more," said the SVP of IT. "It was critical for us to have a way that we could make those APIs easily accessible for our clients."

3. The ease of use of MuleSoft's platform, combined with the customer success model put in place to guarantee the bank's success. "With all the other technologies we were evaluating, we tried a PoC, we had to download libraries for each feature and function, and implementing even simple use cases was very time intensive. With MuleSoft, everything was packaged in a way that made it very easy for our developers to get started," explained the SVP of IT.

To date, the work driven with MuleSoft is expected to drive massive impact at the bank. Just within mortgages, over 400 applications are dependent on the services the team has developed. With Exchange, these services will now be available to all lines of business within the bank, enabling accelerated innovation. Furthermore, the underlying automation pipelines for decomposing monolithic web services are exposed through APIs, allowing other teams to employ a similar microservices-based approach where warranted.

This allows the bank to not only improve its operational efficiency, but also quickly develop superior customer experiences and new product offerings by reusing the underlying assets that the Central IT team has unlocked for the business.

"The vision in moving to MuleSoft is that it will help us provide services five times faster than before, and in a cheaper manner," said the SVP of IT. "As we implement this API strategy, we will be able to change and provide additional capabilities much more easily."

[Take a look at more resources to help you modernize your legacy systems with Anypoint Platform.](#)

MuleSoft®

MuleSoft's mission is to help organizations change and innovate faster by making it easy to connect the world's applications, data and devices. With its API-led approach to connectivity, MuleSoft's market-leading Anypoint Platform™ is enabling over 1,000 organizations in more than 60 countries to build application networks. For more information, visit mulesoft.com.