

# Digital Transformation in Banking:

How APIs are Fueling the Next  
Wave of Financial Services

# Table of Contents

<b>1. Introduction</b>	<b>3</b>
The Proliferation of APIs in Banking	3
Historic Perspective	4
Example – Credit Card Network	5
Example – the Swift Payment Network	6
The Movement to Digital – Core and Spokes Architecture	7
Bottom Line	9
<b>2. Factors Accelerating API Creation in Banks</b>	<b>10</b>
Fintech Applications	10
Global Regulation	10
API Aggregators	12
<b>3. How APIs are used in Retail Banks</b>	<b>13</b>
Account Information	13
Account Creation	14
Product Information	14
Payment Initiation	15
ATM Bank Locations	15
New Products	15
<b>4. How to Do APIs Right</b>	<b>16</b>
Work Inside Out	16
Create an API Platform – Enterprise API Hub	17
API Deployment Flexibility	19
Developer Experience Matters	20
Make Room for New Products	20
<b>5. APIs are Imperative for Growth</b>	<b>21</b>

# 1. Introduction

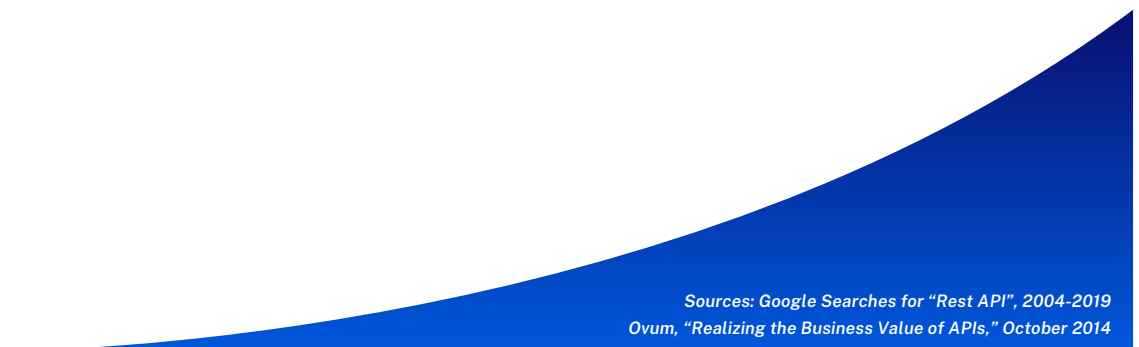
Application Programming Interfaces (APIs) have emerged as a key component of enterprise modernization efforts, particularly in industries such as banking and insurance where the availability of API-based resources provides leading institutions with new channels for reaching customers.

This ebook explores the driving forces behind this rise in adoption for APIs in financial services and banking sectors, examines the type of APIs being created and exposed by banks around the world, and presents a set of recommendations for doing APIs smartly based on data from more than a dozen banks.

This book is recommended for any executive, manager, or technological influencer affecting digital strategy at a bank or financial services company. It can serve to provide some context and examples around the adoption of APIs in banking, as well as a practical guide for those embarking on an API journey.

## The Proliferation of APIs in Banking

Much has been written about the overall growth of new startups and businesses with API-centric models. Many of these businesses have already surpassed billion-dollar valuation, and continue to rise in importance and size.



Source: IBM, "Building for the Open API Economy", 2016



Source: Public Market Information, 2019

When one thinks about the banking sector, the image that tends to come to mind is of heavy regulation, risk averse executives, and slow development. So how is it that an industry so infamous for moving slow is at the forefront of the global API journey? The answer seems to be a combination of historic predisposition to relying on APIs, combined with other factors that are accelerating the adoption of APIs at a breakneck pace.

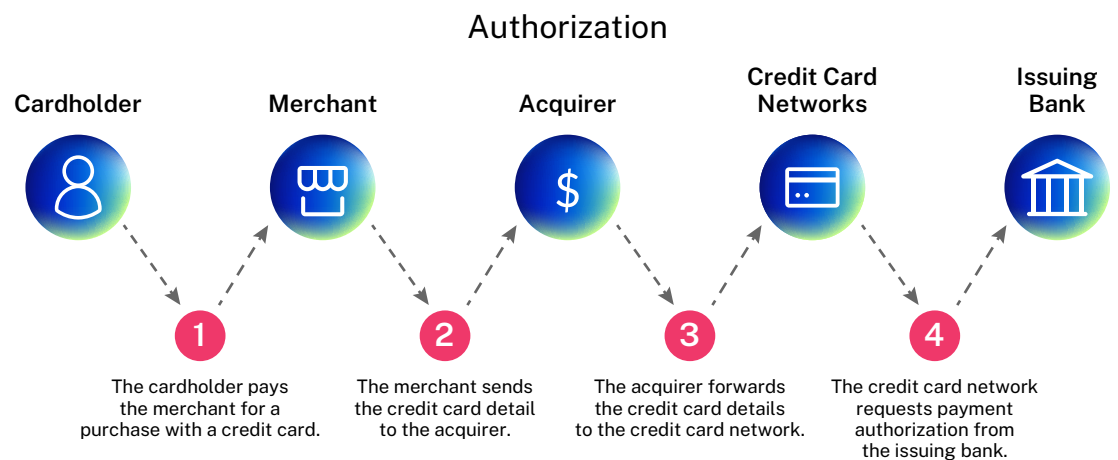
## Historic Perspective

When you come to think about it, APIs and integrations in general have been core to the finance industry since its very beginning. This is due to the number of parties and elements involved in making this huge industry work, and their need to communicate with one another. The financial industry is centered around storing, providing, and moving capital -and all of those require the cooperation of multiple parties, interfacing through APIs.

## Example – Credit Card Network

Though it may not appear as such to the average consumer, every credit card swipe involves a complex dance of multiple integrated systems across many institutions, and thus exemplifies the deeply interconnected and API driven nature of the financial system.

As an example, let's take a simple scenario – let's say you are at the closest coffee shop, you order a cappuccino and you are asked to insert your card into a card reader to pay. You instantly get a confirmation of payment and a receipt. You may be unaware, but to make this seemingly simple transaction work, the terminal, a credit card network and at least two banks are involved, all through a complex network of API calls.



This complex song and dance between the merchant, acquirer bank, credit card network, and issuing bank was originally done 'offline'. The merchant would copy the credit card details using a specialized slip, and physically send it to the acquirer (or call them using any regular phone).

In 1970, IBM introduced the magnetic strip<sup>1</sup> in a pilot with American Express and American Airlines, and paved the way to digitizing the payment process and making it real time. Initially relying on the telephone network (before widespread internet deployment), the merchant's terminal would "call" the acquirer bank's computer and they would confirm the transaction in real time. This can be seen as a very early example of an API Call ('call' being used both figuratively and literally in this instance). As the internet became more prevalent, the whole process moved to relying on TCP/IP and those calls were replaced by what we would view as an API today.

In a way, the credit card network can be seen as one of the earliest examples of a complex process involving multiple parties happening in real time, and it was enabled by APIs.

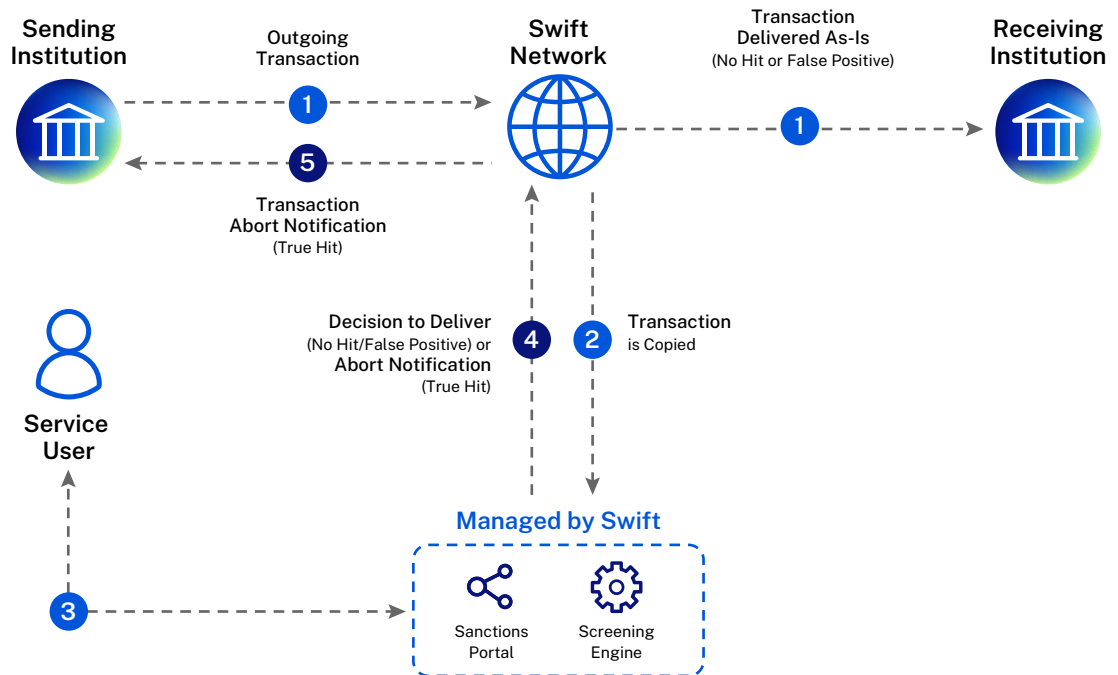
<sup>1</sup> <https://www.ibm.com/ibm/history/ibm100/us/en/icons/magnetic/>

## Example – the Swift Payment Network

Swift – or the Society for Worldwide Interbank Financial Telecommunication – was formed in 1973 with the goal of letting customers and corporates globally send money from one bank to another – even when those banks are in different countries.

It did that by creating a central server that handled the transaction-making, and an interface that banks can connect to to initiate and receive transactions through the network. That interface is – in a way – an API specification, describing how different banks can make calls to interact with this global payment network. In fact, if you look at the full name of the SWIFT network – Society for Worldwide Interbank Financial Telecommunication, you might notice that “Interbank Telecommunications” sounds awfully similar to Interapplication Communications – which is exactly what APIs are.

One part of the SWIFT network is the ISO 9362 standard, which creates a standard for bank-codes. (if you have ever sent money internationally you may have seen codes like BOFAUS3N -Bank of America. These are SWIFT bank codes). These are a standard parameter in the system, much like an API would introduce.



The SWIFT system today provides the backbone of the global financial system with over 11,000 member banks<sup>2</sup> and over half of cross border payments remitted through the SWIFT network<sup>3</sup>. It is hard to imagine today’s global economy functioning without this critical piece of infrastructure.

<sup>2</sup> <https://www.swift.com/about-us/discover-swift/technology-operations>

<sup>3</sup> <https://www.ft.com/content/631af8cc-47cc-11e8-8c77-ff51caedcde6>

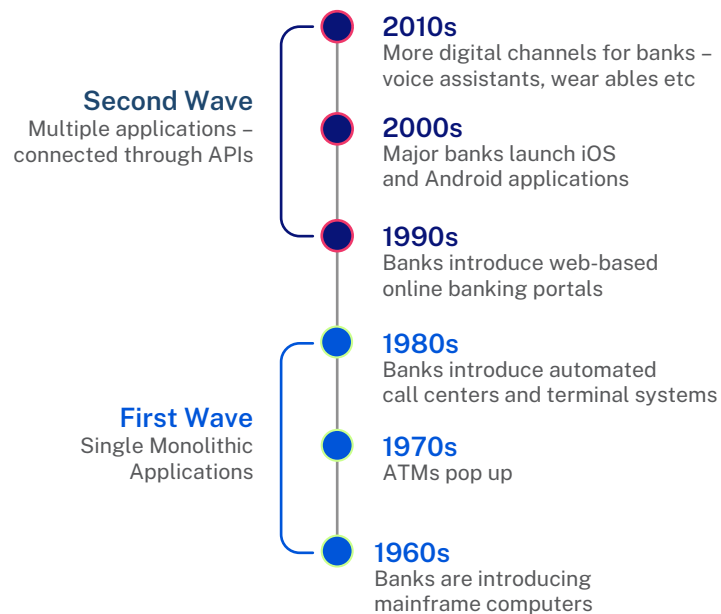
## The Movement to Digital – Core and Spokes Architecture

While the networked driven nature of the banking space led to the creation of APIs for integrating different banks together, the internal development of digital software and services within banks also drove the creation and adoption of APIs and a Service Oriented Architecture (SOA) in a significant way.

The first wave of computing and digitalization in banking happened during the 1960s when mainframes were first introduced into major banks. As many of the banks' tasks are oriented around managing, moving and manipulating big sets of data (account balances, transaction records, loan documents etc.), computers were a great fit and helped automate many of the daily tasks in the bank. Many banks still use similar mainframes to manage accounts and transactions – their “Core Computing.”

The second wave of computing for banks started in the 1990s, as banks have introduced digital experience to give their customers direct access to banking services and products outside of the branch. In 1994 Stanford Federal Credit Union was the first bank in the US to offer an online banking portal. By late 2001, eight major US banks had over 1 million online users each. As the 21st century progressed, banks have started offering more digital channels – mobile apps, Alexa skills, wearable apps, and so forth, and today the majority of banking happens outside the branch. Consumer behavior has changed – banking customers want to bank on the go or from their homes, not in a branch. This means that every bank is now also a software company, developing applications for multiple channels.

### 50 Years of Digital Banking



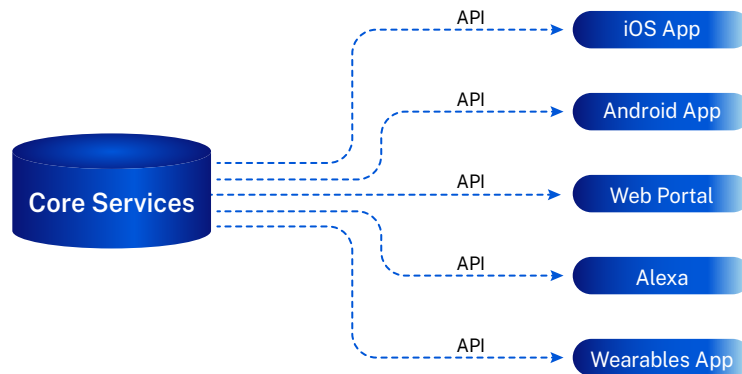
The combination of this trend has yielded a need for a new architecture. There are now two types of software developed in the bank:

- **Core Services**—providing the core account and transaction keeping services.
- **Consumer Applications**—delivering banking to consumers on a myriad of digital channels.

Each one of these types of software has very different requirements. Their development operates in different paces, and they rely on different technologies.

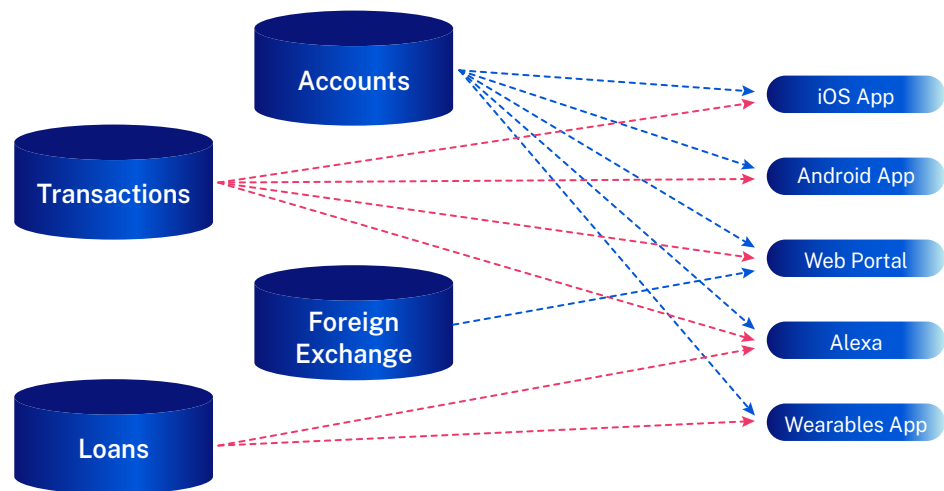
	Core Services	Consumer Applications
<b>Platforms</b>	Single	Multiple
<b>Technologies</b>	Only proven technologies	New technologies on new platforms
<b>Pace</b>	Slow & Safe	Fast & Agile
<b>Tolerance for Errors</b>	Low	Medium

Thus, banks must separate their development efforts between different teams. Some would work on core services and some would build applications used by consumers, interfacing with those core services. The connection between the teams would rely on APIs — and thus APIs have been introduced into the internal development practices of the banks.





This image gets further complicated when considering that “Core Services” includes more than just one service. It must handle many types of financial products and instruments, from loans and deposits to inter-account transactions and foreign exchange. It is a collection of services, all with their own APIs — making the architecture more of a “Many-to-Many” relationship:



## Bottom Line

The banking space is a global, interconnected, and partnership-driven industry. As such, it is not surprising that as banking moved toward digital integrations and APIs were paramount for digitization. From the credit card network to the SWIFT and ACH global payment networks, banks are reliant on many external APIs. In tandem, the development of digital channels and experiences on top of the vast set of core services existing in the banks has made internal APIs a core part of the software architectures.

Combined, these trends mean every bank is dealing with API integrations and management on an unprecedented scale, and must make these APIs a top part of its software strategy. Looking at larger companies as a class, the average organization has over 300 different APIs under its wing. In the banking experience this number is likely higher, and will keep rising. There are however several factors that will increase the rate at which APIs get created and their importance in the banking space.

## 2. Factors Accelerating API Creation in Banks

While historically APIs have been a core component of most software architecture in many financial institutions over the past decades, the discussion about them has noticeably picked up in recent years. While to some extent it can be attributed to the general market shift toward APIs and Microservice architectures, the banking sector — due to a combination of factors — has seen an amplification of this overall switch to APIs.



### Fintech Applications

Tech companies and startups launching financial services



### Global Regulation

Global regulators requiring banks to open APIs



### API Aggregators

API aggregator firms creating private APIs for banks

### Fintech Applications

Most estimates put the Fintech application market to be over \$300B by 2024, fueled by the demand across all industries to incorporate financial services into almost every offering. For example, companies like Apple, Google Pay, or Facebook have integrated payment systems with their applications to provide customers with a more seamless experience. Similarly Fintech startups such as Venmo, Transferwise, and Robinhood want to offer consumers finance solutions but like the technology companies, they do not want to become a bank.

These companies partner with traditional banking institutions such as Goldman Sachs, Wells Fargo, CITI, etc. to become the “back-end” for their offering. Fintech companies rely on banks to expose their APIs so that they can offer customers real-time access to financial services while relying on the bank to provide security and regulatory compliance for every transaction.

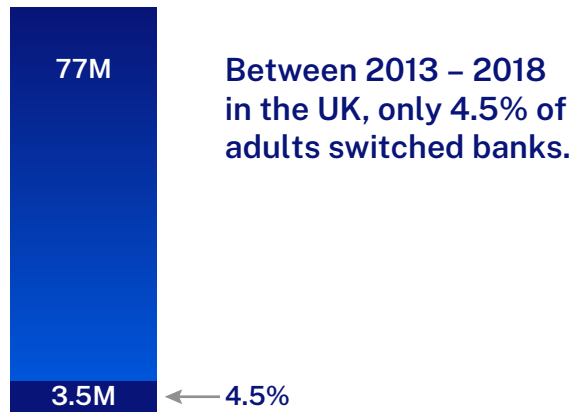
### Global Regulation

The financial crisis in 2008 has led to more than a decade of financial and regulatory reform involving consumer protection, capital requirements, Federal Reserve authority, and more. The government added regulation to prevent this type of crisis from occurring again. It also bailed out many of the largest banks financially as these traditional institutions were “too big to fail.” The theory was that the collapse of some of the world’s largest banks would be disastrous to the greater economic system.

Despite what has occurred, the general population continues to place their trust in a few major financial institutions. Most people open a bank account when they are 18 years old and never change banks throughout their lives. In fact, less than five percent have ever switched their bank to another institution. Additionally, most of these banks collect data about their customers and use it to offer customized services such as loans and credit card offers, which makes switching banks even harder. As a result, a few banks control the market.

This lack of competition leads to:

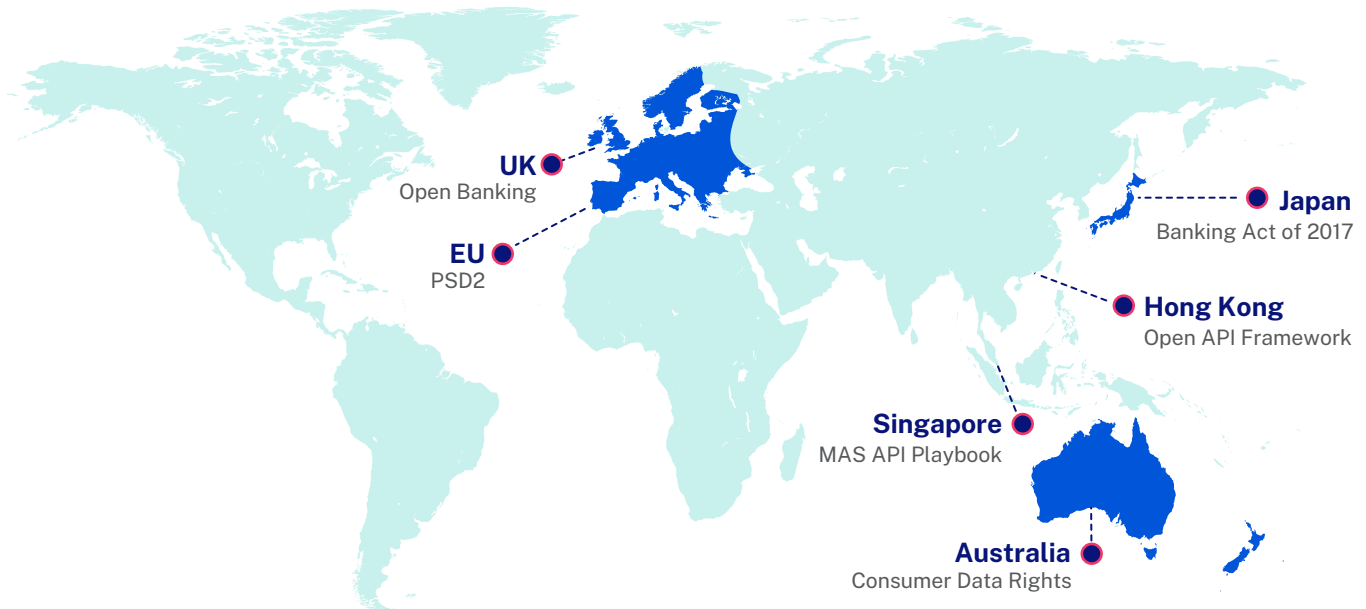
- Stagnation in innovation and services
- Overpayment of banking fees
- General market risk



Source: Wired, 2018

In 2016, the UK began to require open banking (or PSD2), a system that required banks to open up their Application Programming Interfaces (APIs) to all third-parties to access the financial information needed to develop new applications and services. This opened up competition and brought more integrated offerings to consumers.

This trend is not unique to the UK – as many countries around the world embarked on a journey of open banking, powered by APIs. By 2020, countries in almost every continent have launched an open banking API initiative and many will follow suit in coming years. This regulation acts as the “stick” that gets banks to invest in exposing APIs.



## API Aggregators

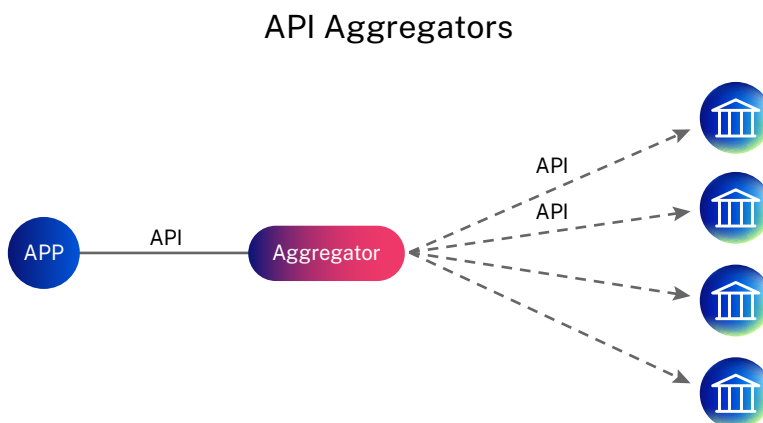
As many companies move to launching financial services and apps that require access to their customers' financial data, there is a need to integrate with the customers' banks.

There are two key challenges with connecting to these banks though:

1. There are over 2,000 banks in the US alone. Connecting to all of these banks is a daunting task.
2. Many of these banks do not have an API, requiring more creative data extraction methods.

This need led to the creation of a new type of service provider — the API aggregator. Under this concept, a customer's personal and financial data is available in a single location making it easier to incorporate into services and transactions. Since most banks do not provide APIs, aggregators solve this problem by providing a single interface to all banks that accesses personal and financial data. When there is not a direct API interface, many rely on "screen scraping" to obtain the information.

The aggregation of financial data has become a lucrative business for companies in the Fintech market.



This causes banks to become concerned about security. In addition, they are experiencing high traffic to their sites, but it isn't actual customers. However, most banks understand that they need to provide access to this data to leverage the potential of fintech applications and remain competitive in the market. This is another factor pushing banks to expose more APIs.

### 3. How APIs are used in Retail Banks

There are six types of core APIs in retail banking:

- Account Information
- Account Creation
- Product Information
- Payment Initiation
- ATM Bank Locations
- New Products



Account Information



Account Creation



Product Information



Payment Initiation



ATM Bank Locations



New Products



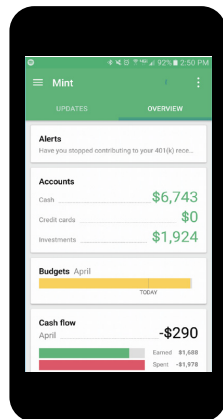
#### Account Information

An API that accesses a user’s account information is the core API for most banking initiatives. These APIs provide access basic account data including the ability to:

- Log-in to bank account
- Validate user Accounts
- Confirm user balance and funds
- Get user transactions

An example of this is the Wells Fargo Account and Transaction Information API that provides customers with the ability to access their UK account balance and transaction information. Or the RBS’ or Barclays Account & Transaction APIs that provides access to account information, balances, etc. for their respective customers. This API is often used in online banking when customers are interested in getting account information for use in other applications.

#### Account Information – PFMs



Get Customer’s Account

Get Account Balance

Get Account Transaction History

Get Product Offers – Cards





## Account Creation

This set of APIs enable users to create or apply for bank accounts programmatically. These APIs:

- Check user eligibility
- Start new application
- Submit details
- Check application / account status

Some of these APIs may help with the application submission process or enable an organization's customers to open an account and transfer funds. Banks like Standard Chartered, Capital One, and Chase offer these types of APIs.



## Product Information

These APIs access product information for banks, obtain details for each product. In short these types of APIs enable the application to access the

- Product catalog
- Product details
- Current rates / interest
- Eligibility for the user
- Customized product offer

**Payment Calculator** **REDFIN**

**\$4,849 per month**

30 Year Fixed, 3.672% Interest [Customize calculations](#)

Principal and Interest \$3,024 Property Taxes \$838

HOA Dues \$836 Homeowners' Insurance \$151

Home Price \$824,000 Down Payment \$164,800 20%

Today's rates for this home 30 Year Fixed ^

Loan Type 30 Year Fixed

<b>3.625%</b> Interest Rate Sebonic Financial NMLS#66247	<b>3.632%</b> APR as of 4/9	<b>3.625%</b> Interest Rate eClick Lending NMLS#227765	<b>3.641%</b> APR as of 4/9
---	--------------------------------	---	--------------------------------

[View all rates](#) Ads



Get Custom Mortgage Rate



## Payment Initiation

These APIs enable applications to create and track payments. Some common APIs for this include:

- Validate IBAN / SWIFT code
- Create a payment
- Approve a payment
- Track payment status



## ATM Bank Locations

These types of APIs reveal the locations for ATMs and branches and enable applications to locate:

- ATMs in radius
- Branches in radius
- ATM information
- Branch hours
- Available service at each branch



## New Products

The biggest area for innovation involves creating new products and services leveraging existing banking data and infrastructure in a unique way. For example, CapitalOne has created services using APIs, where they can verify customer identity using shared identity attributes. Wells Fargo enables users to search and retrieve critical tax documents. Deutsche Bank has APIs that enable businesses to verify someone is over 18 years and confirm their identity using APIs that connect to banking data. Through APIs, organizations can offer extended services.

## 4. How to Do APIs Right

Although many banks and financial services firms have already been participating in the API economy, the torrid pace in which new APIs are created and the scope of where those APIs are being used — both within the bank and by third parties — is leading many to start thinking about the second generation of their API strategy.

Naturally, many organizations gravitate towards thinking of APIs in different “buckets”: internal APIs, external APIs, developer APIs etc. This classification can be counterproductive, and tends to create more fracture and discord within the organization as opposed to a clear and unified strategy.

As APIs go through a lifecycle, beginning as internal technology and then becoming exposed to certain partners, so must the API strategy. As such, the following recommendations refers to managing and structuring both internal and external APIs.

First, it is important to start with internal APIs so that you build up your API Muscle in your firm. The second principle calls for utilizing an API platform, so that you have the right tooling and infrastructure in place. The third principle involves creating flexibility in the API release or API creation process. Teams should be made aware of APIs and should be thinking of them as soon as possible in the software development lifecycle. The fourth principle is about building your solution with the developer in mind. As the developer is the heart of your software supply chain, if you do not create an experience that is easy and intuitive, they will not use it. And finally, make sure you are leveraging your APIs to continue to innovate and offer new products and services for partners and customers.



**Work  
Inside Out**



**Create an  
API Platform**



**Flexibility in  
Deployments**



**Developer  
Experience  
Matters**



**Make Room for  
New Products**



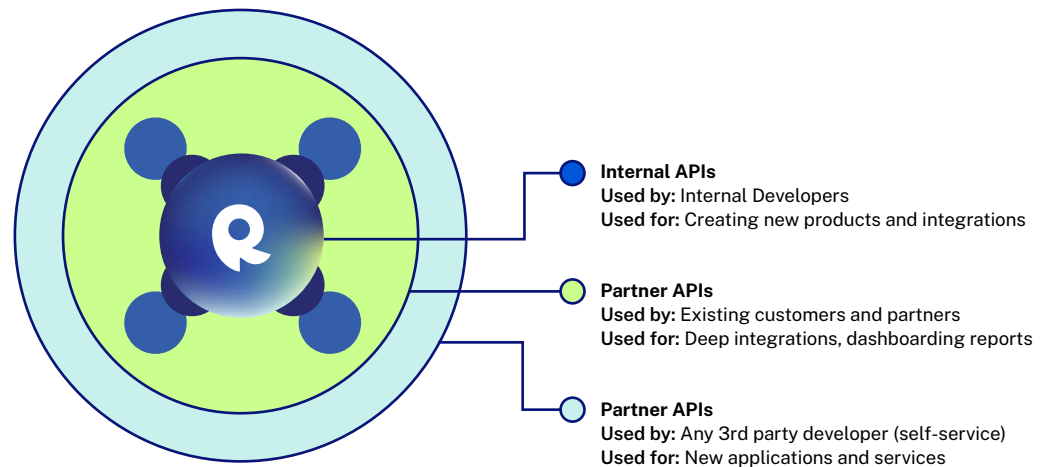
### Work Inside Out

The reality is that the strategy should begin with a focus on internal APIs. A recent survey indicates that almost 70 percent of those surveyed are considering or are using internal APIs/microservices in their infrastructure. Most companies have more than 300 APIs in their infrastructure, with a plan to increase this number significantly over the next few years.

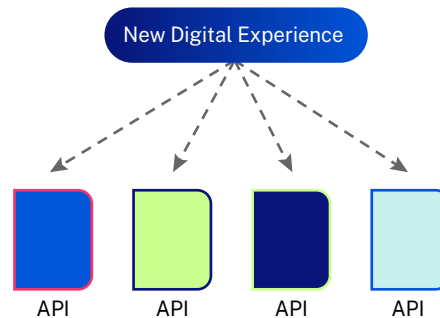
When creating API platforming or tooling, start by looking at the internal APIs that internal developers are using to create new products and integrations. Use them as a testing ground for new tooling and as an opportunity to build your understanding and skill set with APIs. Once you have a platform, tooling, design processes, and everything around the internal APIs, start leveraging these for partner APIs. These APIs can be exposed to partners and customers and then after that, the APIs can be made public.



## APIs – Start Inside Out



If you were to begin with public APIs, it is most likely that you do not have the underlying systems and services required to support the third-party developers using your APIs. Additionally, you are trying to create API skills within the most complex use case – which is building a public API developer community before you've actually practiced any of these skills internally. The core part is to start with the internal services and move your way outwards.

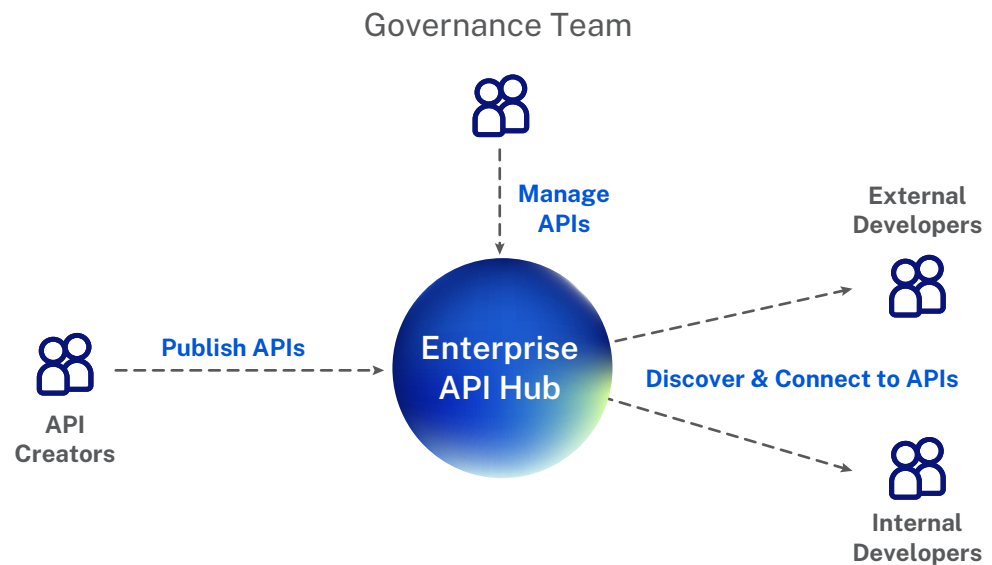


Building an internal API strategy has several benefits. Once you start organizing these internal APIs, you can start thinking about internal initiatives such as hackathons, product development programs etc. that can leverage some of the internal data sources and functionality that was previously siloed behind APIs that weren't fully managed anywhere. Supporting internal APIs internally is critical before expanding and enabling external developers to leverage these APIs.



## Create an API Platform – Enterprise API Hub

When thinking about building that API tooling/mindset for the second generation of APIs, it is essential to establish an Enterprise API Hub. This is a single place for managing all of the APIs used by all the stakeholders in the organization. For the API creators, this is where they can publish and share their APIs. For both internal and external developers, they can discover and connect to APIs and for the Governance team, the Enterprise API Hub enables them to monitor and manage how APIs are used across your organization. For each one of these stakeholders, there is a different set of features and capabilities that enables them to effectively publish, find, share, and manage APIs.



#### API Creators can:

- **Publish API documentation:** The Enterprise API Hub provides a centralized location where it can be accessed by all.
- **Integrate with CI/CD and API Gateways:** Most likely there is already a pipeline in place to manage the software development workflow. As a result, the platform must integrate with the existing infrastructure and processes.
- **Access and Visibility Control:** This enables you to give workflow approvals for how APIs are used internally and fine-grained access control over how partners and customers can access them.
- **Support All API Types:** The platform must support REST, SOAP, and some of the newer APIs that companies are investing in such as GraphQL and Kafka.
- **Create a Monetization Strategy:** Support is needed so you can monetize your APIs and leverage the platform to attribute cost.

#### Internal and External Developers can:

- **Access API Search:** This capability enables a full text search across all APIs.
- **Use Tagging and Filtering:** The flexibility to create a simplified curation when performing an advanced search of hundreds/thousands of APIs.
- **Conduct Interactive API Testing:** The faster the developer can access the documentation, locate the API, and obtain a response code, the more likely they are to adopt it.
- **Select Code Snippets for Integration:** It should be easy to access and integrate APIs using preferred code language.
- **Monitor and Analyze:** Developers can immediately get a bigger picture of how the APIs are impacting the applications.

**For Governance Team, the can:**

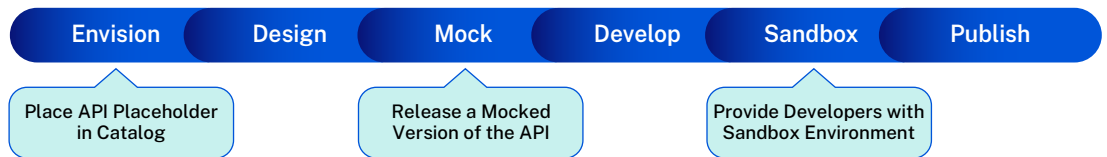
- **Control API Visibility:** Use a whitelist or blacklist model that indicates who can consume/ access data.
- **Define API Readiness:** Establish and enforce the standards that API must meet before they can be leveraged by other developers.
- **Ensure Compliance:** Gather analytics that provide insights into every API to ensure compliance.
- **Track API Usage:** Manage usage for SLAs, internal costs, and to mitigate breaches.

When we are looking at hundreds or thousands of APIs, an API Hub is a critical piece of the toolchain.



## API Deployment Flexibility

It is important to think about APIs and their deployment in a more flexible way — specifically giving access to developers as much as possible as soon as possible. Often the sharing and publishing of APIs is the last step of the process. But the problem is that the API can take four to five months to publish. This leads to several issues. First, if the developer does not know the API is in development, they develop something on their own. The other issue is that the team developing the API is missing out on key feedback throughout the creation process.



A better solution is to involve the developer for every phase of development. In the envision phase, place an API placeholder in your hub/catalog with a basic description so developers know it is not ready but they can see the API is being created and have the ability to comment on it. Once it has gone through the design phase, mock the API and release a mocked API version to the marketplace.

Developers can easily see the API, make requests, and understand what the APIs is going to do once fully developed. They can even create applications that are dependent on the API prior to having the API. Then, create a sandbox for that API so developers can use it and test with it even if there isn't a final version in production. The guiding principle is give access to it as much as possible as soon as possible so developers can work more productively.



## Developer Experience Matters

When you are creating your APIs, it is critical to remember that developer experience matters. Take for example, Stripe, a Fintech company that has an API that allows individuals and businesses to make and receive payments over the Internet. There are many payment systems out there -- many that were in place before Stripe. However, companies/users are choosing to integrate Stripe's payment API because it is easy-to-use, has a nice JS option, great documentation and includes many examples.



*We are making a payment platform that doesn't suck.*

Patrick Collison, CEO, Stripe



It is important to remember that developers are becoming more influential so when creating your APIs, build it with developer experience in mind as it is critical in garnering adoption. In short, you should make sure to:

- Build well structured APIs
- Invest in good developer portals / marketplaces
- Polish your appearance
- Implement self-service when possible
- Deprecate APIs carefully



## Make Room for New Products

As banks go through the journey of discovering which APIs they can expose, traditionally they start by augmenting internal services and traditional banking products with an API layer. While this can provide significant value to both customers and partners, an API delivery model can unlock new product types and opportunities, enabling the bank to leverage its existing data and infrastructure to provide new services with a low marginal cost.

The position of the bank is in many ways contradictory to that of most Internet software companies. Where most apps have user anonymity, banks must have hard authentication. Where most software companies have only online presence, banks have branches spread throughout. Whereas websites are unable to validate anything about their users' life (address, marital status, age, etc.), banks must possess such information.

A smart bank can use those data points and infrastructure and offer — as APIs — this data to other websites and applications that need such services. These can include anyone who cares about address confirmations, KYC (Know Your Customer), age validation, and more.

When building an API strategy, the banks should ardently promote the creation of such APIs to creatively generate new and exciting revenue streams.

## 5. APIs are Imperative for Growth

What is well understood in the banking industry is that customer experience is key to success. As all services become digital, easy-to-use banking and financial applications that can be accessed anywhere at any time are in high demand. APIs enable banks to provide customized and relevant services to their customers by providing the critical connection between customer data and applications/programs. The proliferation of APIs has been accelerated by the emergence of Fintech companies offering new technology, new regulation/open standards, and the emergence of API aggregators.

APIs help customers in a number of ways, providing a simplified and integrated experience when making purchases and completing transactions. For banks, APIs provide a key to new revenue streams, offer access to additional data, encourage new product development, and open up new channels.

To capitalize on this next wave of API-lead microservices, financial institutions need to build an API strategy that works from the inside out using internal APIs, leverages an enterprise-class API platform, provides a flexible deployment model, offers developers a friendly user experience and becomes the cornerstone for future innovation. To learn more how RapidAPI can help you on your API journey, visit us at [www.rapidapi.com/enterprise/](https://www.rapidapi.com/enterprise/)