API STRATEGY FOR OPEN BANKING

BY
THE NORDIC APIS WRITING TEAM

WITH A FOREWORD BY
EYAL SIVAN

NORDIC APIS

API Strategy for Open Banking

Insights and case studies from leading open banking experts and API strategists.

Nordic APIs

© 2018 - 2020 Nordic APIs

Contents

Foreword: Embracing Open Banking . . . . . **i**

Preface: APIs Support the Open Banking Movement . . . **iv**

The Premise of PSD2 And Open Banking **1**
Open Banking: The Premise and Promise 2
What PSD2 Means For Banks 3

6 Reasons to Embrace an API Strategy for Open Banking **4**
1. Compliance 5
2. Improved Digital Agility 5
3. Premium API Products 6
4. Increased Customer Satisfaction 7
5. Potential for Collaboration 7
6. Wider Client Base 8
Conclusion 8

Bring on the Players: Who Wins in Open Banking? **9**
What Open Banking *Really* Means 10
Comply-first Providers 11
Protectionists 12
Open-first Providers 13
Plotting the Opportunity 14
Final Thoughts 16

Case Study: Nordea's Journey to PSD2 Compliance, 300 Signups in 72 Hours **17**

A World Beyond PSD2 Compliance 20
Final Thoughts 20

FinTech and APIs: Making the Bank Programmable . . . 22
What is FinTech? 23
Advantages of Exposing a Bank with an API 24
Banks and FinTech Can Play Nice 24
Use of APIs: In-Account App Marketplace Concept . . . 25
Data Transparency and the Rise of Open Banking . . . 26
New Platforms Lead to Unexpected Innovation 27
More Advances in the Financial Sector 28
Conclusion 29

How Can Consumers Relate To Open Banking? 30
Building Context for Consumers 31
Open Banking Must Foster Trust With End Users 33
Control Matters 34
The Open Banking Marketplace 35
Final Thoughts: How to Establish Consumer Faith in Open Banking 37

How Banks Are Becoming Uberized 38
APIs are Nothing New 39
Smartphones: Kindling a Change 40
Time to API Up 41
Building with Purpose 41

How Does Open Banking Apply to US Banks? 45
Regulation in Europe 46
Regulation in the US 47
The Role of the Market 49
Final Thoughts 51

Case Study: From API Doing to API Thinking at ING Bank 52
APIs versus Web Services: What's the Difference? 53
API Doing vs API Thinking 54

APIs and Customer Journeys 54
Why API Doing is Equally Important 55
From API Doing to API Thinking 57

Open Banking Amplifies the Need For Definition Driven APIs 58
Adjusting Practices With The Shifting API Landscape . . 60
How OpenAPI Specification (OAS) Accelerates API Development 61
Supporting OAS Throughout the API Lifecycle 62
Final Thought: Drive Open Banking API Strategies with OAS 64

High-Grade API Security For Banks 65
Regulatory Compliance Considerations 66
Identifying Vital Data 67
Potential Vulnerabilities 69
API Security Methodologies 70
Security is The API Provider's Responsibility 74
Recent Exploits and Breaches 74
Conclusion 75

Is OAuth Enough for Financial-Grade API Security? . . . 76
Can OAuth Make The Grade? 77
Some Tokens Are Unbearer-able 78
Away With The PKCEs 79
Signed, Sealed, Delivered 80
What's Next For Financial Grade API Security? 81

OpenID Connect: Overview of Financial-grade API (FAPI) Profile 82
What is FAPI? 83
Adding Resilience: The Read-Only Profile 85
Bullet-Proofing: The Read-Write Profile 86
Improving OAuth 2.0: JWT-Secured Authorization Codes 88

Decoupling Authentication: Client-Initiated Backchannel Authentication 88
Final Thoughts 91

Case Study: Growing Internal API Consumption in Danske Bank 92
The Path Towards APIs 93
Set-and-Forget Performance 94
Identifying Setbacks… and Addressing Them! 94
The Results 96
Summary 97

It Started With PSD2 and Personal Data 98
The Status Quo 99
Regulatory Impact 101
The Open Banking (and Data) Landscape 102
Final Thoughts 103

Nordic APIs Resources 105

Foreword: Embracing Open Banking

by **Eyal Sivan**

It is with great pride that I write this foreword, for it is no exaggeration to say that Nordic APIs played a pivotal role in my journey, which led me to dedicate my life to **open banking**.

The first time I ever heard of open banking was while presenting at the Nordic APIs Summit event in Stockholm, where I met Gunnar Berger from Nordea, featured here in this book. After listening to him speak, describing large banks as battleships and fintechs as speedboats, I discovered he had a curious title that I had never heard of before: Head of Open Banking.

After speaking to Gunnar, doing some reading, and listening to European politicians argue about PSD2 on YouTube, I became immediately fascinated with open banking and its seeming contradictions. Here was a situation where governments were the ones driving innovation while market forces were slowing it down, a situation where regulation was trying to reduce barriers to entry instead of erecting them. In open banking, there seems to be an attempt to create a financial system that is both more competitive and healthier at the same time. In short, a better way to handle money.

Upon further research, it became clear open banking activity was taking place everywhere, not just in Europe. Moreover, regardless of whether the approach being taken was driven by regulation or by the market, the aim seemed to be the same: to develop a **common standard**, enabling all the players in the

financial ecosystem to communicate with each other seamlessly and universally, acting as a foundation not only for great innovation but also for the establishment of modern data rights.

So, what exactly does this common standard called open banking look like? If you are reading this book, you probably already know that open banking standards are implemented as **a set of open APIs**. Today, in most open banking implementations, there are two main types of APIs: one to get information about you or your banking activity and another to move money around using payments. But that is just the beginning. Already, based on those two basic functions, we are seeing moves into premium APIs, data-driven aggregation, and AI-powered smart money. Mature markets like the UK are seeing exponential growth in usage. Despite some bumps in the road, open banking has already proven that a strong standard acting as a base can lead to explosive innovation, increased competition, and a modernized financial market. Make no mistake: open banking is here now, and it is here to stay.

To those building the APIs that power open banking, understand that you are building the future. The APIs you build will form the digital economy's roads and bridges, the bedrock of the 21st century. Your APIs will control how all of us trade, save, and invest to build ourselves better lives. In this book, you will find articles by some of the smartest open banking pioneers out there, like Chris Wood, who was one of my early teachers. But rest assured, much of this story has yet to be written, and much of open banking has yet to be built. Perhaps you will be the next one to add a piece of the puzzle.

Open banking appears to be a rare opportunity to raise all boats if done right - a true nonzero-sum gain. Creating a common standard for the financial services ecosystem means that: Consumers win through better, cheaper, more innovative financial products; governments win through effective regulation that defends both competition and consumer data rights; technology

companies, large and small, win because they can create and build on open standards; and banks win (although many don't know it yet) because there is more banking activity overall. A **win-win-win-win**.

Today, I am fortunate enough to hold the same title as my friend Gunnar, Head of Open Banking, and I cannot help but smile at the thought of it. Not merely because I have a good job at a great company, but because it feels like I am helping to build something bigger.

It is my distinguished pleasure to assist Nordic APIs in spreading knowledge of open banking. I sincerely hope that you come away from this book as excited as I am about the possibilities of open banking and the open data future it makes possible.

– Eyal Sivan

Head of Open Banking at Axway

Mr. Open Banking

mropenbanking.com

Preface: APIs Support the Open Banking Movement

by **Bill Doerrfeld**

Since Nordic APIs ran our first story on open banking and FinTech in 2016, our writers have been carefully attuned to the world of banking software architecture.

We've traced the EU's Payment Services Directive 2.0 (PSD2) regulation, and the rise of Application Programming Interfaces (APIs) to help externalize personal data for end consumers. We've also followed the modularization of banking components, as valuable infrastructure becomes reusable through modern microservices designs.

Not only are financial institutions opening up with APIs to meet global regulations, but innovative banks are finding success in fully-fledged API-driven banking products, supplementing core business models. Though there are many new opportunities, banks must apply a keen strategy to accel in this new paradigm. New entrants must be made aware of open banking pitfalls and marketing gaps.

At Nordic APIs, we live for new strategies and technologies and strive to introduce the benefits of API-first platforms to technical and non-technical audiences. In our volume, *API Strategy for Open Banking*, we compile our top articles on open banking from the Nordic APIs writing team.

Within each chapter of *API Strategy for Open Banking*, a Nordic APIs blogger digs into a niche aspect of open banking, covering PSD2, open banking benefits, developer experience tips,

frameworks for high-grade security and access management, and more. We've featured best practices as well as case studies from our conference speakers, who represent some of the largest open banking initiatives in the world.

So, please enjoy *API Strategy for Open Banking*, and let us know how we can improve. If you haven't yet, consider following Nordic APIs and signing up to our newsletter for bi-monthly blog updates and event announcements. We also accept blog contributions from the community - if interested, please visit our Create With Us page to submit an article.

Thank you for reading!

– Bill Doerrfeld, Editor in Chief, Nordic APIs

Connect with Nordic APIs:

Facebook | Twitter | Linkedin | YouTube

Blog | Home | Newsletter | Contact

The Premise of PSD2 And Open Banking

by **Thomas Bush**

Open banking brings agility to modernize the financial sector

There's a particularly apparent trend in today's economy: we're moving away from large, centralized systems towards collaborative, access-based ones. It's changed the way we shop with websites like eBay, the way we travel with apps like Uber, and the way we get a good night's rest with portals like Airbnb.

One industry that's been slow to change in this global movement is banking, where it's all the more critical. We still use massive institutions, like JPMorgan Chase, Citi, and HSBC, to manage our finances — and what's more, we use them to manage all aspects of our finances, from investment to insurance to mortgages.

However, it's somewhat of a double-edged sword with banking: we want the trustworthiness and reliability of a big bank, but with the agility and innovativeness of a hundred startups.

The solution? — **open banking**.

Open Banking: The Premise and Promise

The premise of open banking is to put the power of financial institutions in the hands of others. On the one hand, this means increased transparency. It also means the development of APIs to allow third-party developers to build their own financial services and platforms using all the data and functionality that banks usually keep to themselves.

While open banking may not sound so attractive for big banks or the major payment service providers like Mastercard and Visa, it's mostly good news for everyone else. Consumers get a better choice of financial services without sacrificing on security, FinTech startups get to explore otherwise impossible ideas, and governments get to tackle the issue of having large, hegemonic financial organizations.

In fact, most governments in Europe — and even the European Union itself — are already pushing for open banking. The UK has its own Open Banking momentum while the EU has passed *PSD2*, the second edition of the Payment Service Directive, which will encourage APIs that "unlock customer data" and "enable consumer choice."

What PSD2 Means For Banks

The original Payment Service Directive was enacted in 2007 to regulate payment services across the European Union and encourage competition across borders. The revised edition — PSD2 — was passed in 2015 and shifts the directive's focus from broad safety regulation of the financial world to the development of more **innovative** and transparent financial services.

So what exactly does PSD2 mean for European banks? To be compliant with these new rules, they'll have to implement a significant amount of specific, robust functionalities, which will ultimately require an overhaul of much of their technical infrastructure, including security concerns associated with doing so.

Fintech group Difitek compiled this PSD2 compliance checklist for banks, which includes, among other things:

- An **API function** allowing users to grant others access to their data
- An **API authentication** process which verifies both the user and application
- API **documentation**, developer SDKs, code samples, and tutorials

6 Reasons to Embrace an API Strategy for Open Banking

by **Thomas Bush**

Outside of compliance, there are many business reasons to adopt an API-first open banking.

Open banking is an initiative that allows third-party financial services companies to access users' banking data. The primary goal of open banking is to put power back into the hands of customers, enabling them to securely use third-party financial products and services that rely on banking data or functionality. Web APIs are the technology underpinning much of this new ecosystem.

With the introduction of new regulations like the European Union's Second Directive on Payment Services (PSD2), many banks have no choice but to open up, giving others access to their users' data for not much in return… or so it may seem! In reality, there are plenty of strong reasons for banks to embrace open banking, with concrete financial incentives. Let's look at six of them.

1. Compliance

Of course, the main reason banks are implementing open banking practices is compliance — or at least preparation for compliance. While the European Union's PSD2 is the best example of a sweeping regulation that requires banks to share customer data with third parties (this is known as X2SA — *Access to Account*), it's not the only one. For example, Hong Kong has its *Open API Framework*, while Australia has the *Consumer Data Right* (CDR) act. Other major jurisdictions are moving in the same direction: the US Treasury has recommended the introduction of financial data sharing regulations, despite the country's hitherto market-driven approach.

How it affects the bottom line: Of course, compliance isn't about driving additional revenue: it's about staying in business. Compliance improves profitability by avoiding unnecessary fines and fees.

2. Improved Digital Agility

A major challenge of open banking is being able to share data securely, quickly, and efficiently. As a result, many banks are having to redesign their entire data architectures, often employing an API-based microservices approach to make data more

accessible. Greater digital agility, then, is both a necessity and benefit of open banking.

In turn, improved digital agility has its benefits. Not only does open banking improve security and transparency, but it also makes it easier for banks to leverage their own data internally — e.g., for service personalization or to create frontend applications — where it may have been impractical, or even impossible, to do so previously.

How it affects the bottom line: An improved digital infrastructure enables data to be better used internally to improve the customer experience, thereby increasing customer lifetime value.

3. Premium API Products

One particularly exciting benefit of open banking is the potential to create new, revenue-generating API products with relative ease. For an example, look no further than Nordea's developer program. Nordea used the open banking shift as a springboard from which to create paid banking APIs for its corporate customers. Offering API-driven payments, instant reporting, and various foreign exchange tools, these "Premium" APIs go well beyond compliance, building on the hard work that was involved in opening up their systems for open banking.

How it affects the bottom line: By developing and selling access to new API products, banks are able to create additional direct revenue streams. These premium APIs can also be used as up-sells or cross-sells for other banking products (such as certain corporate accounts).

4. Increased Customer Satisfaction

Open banking gives customers huge amounts of freedom as to the number and scope of financial services available to them. On the one hand, this appears to be a clear negative for banks, as it allows third-party organizations to capitalize on user data, where previously only they could. However, a greater selection of financial service integrations — whether or not they are the bank's own — ultimately improves the customer's banking experience, making them less likely to seek alternatives. As they say, a rising tide lifts all boats!

How it affects the bottom line: Since the customer is more satisfied with their banking experience, they are less likely to look for alternatives. This increases customer lifetime value, improving long-term profitability in a predictable way.

5. Potential for Collaboration

As mentioned earlier, open banking is designed to make it possible for third-party financial services companies to gain access to customer data. If banks are willing to take this a step further, they can actively assist these third-party companies in doing so for a whole host of benefits. For example, banks can offer additional functionality, dedicated support, or even developmental collaboration to chosen third parties. In exchange, these third parties can return the favor with various non-monetary offerings, such as additional product functionality for the bank in question or cross-branding.

How it affects the bottom line: By building collaborative relationships with third-party financial services companies, banks are able to create unique value propositions and employ creative marketing strategies, thereby winning new customers.

6. Wider Client Base

Until now, we've focused on the benefits of *sharing data* with others. However, it's important to recognize that open banking is a two-way street: in other words, it will allow banks to gain access to user data from other participating financial institutions (especially other banks). This creates a massive opportunity for banks to create their own integration-based financial products and services.

> "Banks able to move fast to develop a modular business and technical architecture can leverage their brands to dominate parts of the value chain - whether front, middle or back, while dynamically integrating offerings and data from other players." - PWC

How it affects the bottom line: Whereas banks could previously only offer additional financial products and services contingent on banking data to their own customers, they can now serve customers of other banks, with the potential for significantly more revenue.

Conclusion

It may be compliance that has pushed banks to invest in open banking, but there's no doubt that the movement has numerous other benefits with tangible financial impacts. Many of these benefits, such as direct improvements to customer satisfaction and digital agility, result in increased customer lifetime value. However, open banking also improves security, opens new doors for collaboration, and allows banks to make bigger plays with additional financial products and services of their own.

Bring on the Players: Who Wins in Open Banking?

by **Chris Wood**

Simply being an API provider doesn't guarantee an open banking advantage.

It's fair to say that the current noise around open banking is almost deafening. We constantly hear about the potential of this new market — driven by APIs — to change financial services by increasing consumer choice, crushing incumbent banks, or making a barrel of cash for a cool startup.

There is, however, more to open banking than hyperbole. The opportunities and challenges – for the incumbent banks and

"Fintechs" alike – offer a means to reshape an operating model that has typified banking for many years. Equally, there are other reasons for entering the market, such as legal or regulatory coercion that will result in naturally cautious financially-regulated entities being forced into becoming API providers with "open" interfaces that need to be managed, maintained, and protected.

Standing on the cusp of this revolution, we, therefore, have an evolving marketplace being shaped by both regulation and market forces. The reaction to both of these is defining how the protagonists are forging an open banking identity and the behaviors that go with it.

In this post, we characterize the players in the Open Banking market with three levels of action: **Comply-first**, **Protectionist**, and **Open-first**. By understanding why providers are in the market and what they expect to get out of it, we'll get an idea of how it might evolve in the future. We'll find that merely being an API provider in this market won't be enough to guarantee a competitive advantage.

What Open Banking *Really* Means

Open banking is first-and-foremost a means for increasing consumer choice in financial services through regulation. You may have thought that the phrase "open banking" means "open to everyone" — a field day for the geeks who have longed to have an API to populate their Warhammer budgeting spreadsheet without all the nasty downloading of CSV files from online banking. Joking aside, what it actually means in its current form is pretty far from that ideal. "Open" in this context is "open (via an open API) to regulated entities" as defined by legislation like PSD2 or the CMA Order. These regulations decree new roles in financial services, which can be summarized in the term third-party provider (TPP).

Given this context, Open Banking implies API usage. However, it supplies banking customers more rather than complete choice. More choice means we, as customers, can work with a variety of companies to help fulfill our banking needs. In the case of our aforementioned geek, they'll need to work with a TPP to help them aggregate the data via APIs for their spreadsheet. Some of these companies will be the organizations we already know as our account providers. Others will be new companies that provide products to make our lives easier. These organizations can be characterized in one of three different ways:

- Comply-first Providers
- Protectionists
- Open-first Providers

Comply-first Providers

While regulatory deadlines like PSD2 have been approaching for many years, many large banking organizations are still struggling to meet their obligations. Comply-first providers are the financial institutions who are being compelled to deliver on these regulations to open up customer accounts to TPPs.

Banking as an industry has historically had limited forays into the API economy, with a hesitation driven by their risk-averse and project-driven corporate culture that typifies so many large banks. Their API-related initiatives have largely been piecemeal, with little cohesive drive towards API delivery and limited buy-in or sponsorship from senior leadership to make being an API provider a success.

This culture, which tends to result in delivering the bare minimum through open APIs, means that these organizations are already playing "catch-up" both in their API delivery mechanisms

and in the API economy at large. The first and most important goal for comply-first providers is to meet regulatory deadlines and avoid fines, penalties, or bad press.

An interesting side note that lends weight to this argument is a report conducted by Paysafe. In Open Banking and PSD2: A confused roadmap to innovation, they cite current impasses in the landscape and the fact that banks "have had a mixed approach to PSD2. The majority have treated it as an exercise in minimum compliance rather than looking for customer-led outcomes."

Examples of such organizations are not hard to find. Some qualities of comply-first providers include:

- Do not market their API.
- Have a boilerplate or no developer portal.
- In Europe, they only offer APIs that facilitate access to the core PSD2 roles (Account Information Services Provider or Payment Initiation Services Provider).

Despite their false start as an API provider, many of these organizations are also uniquely positioned in that they are directly responsible or have a significant influence in shaping the market. For example, the CMA Order in the UK made the nine largest banking institutions directly responsible for the writing of open banking standards. The largest incumbents, therefore, have a huge amount of influence on the direction of the market.

Protectionists

Protectionists take a step up from Comply-first providers in attempting to seize a slice of the pie as they meet their regulatory commitments. Their motivations, however, do not generally lean towards openness per se: Their goal is to protect what

they already have by wooing their existing customer base. In this guise, Protectionists often take up the TPP role, effectively dogfooding their own APIs while taking advantage of their competitors.

Examples of Protectionists in the market are found in the UK. Barclays released an aggregation feature in their mobile app while HSBC introduced Connected Money in May 2018. The premise of both is simple — to offer existing customers account aggregation features via their existing banking provider — and in doing so, taking some of the heat out of customer demands for "shiny new toys."

The Protectionist provider, therefore, delivers just enough of open banking, safely dipping a toe in the water in a risk-averse manner while delivering on their regulatory commitments. They can also influence their customers' perceptions of open banking by associating their first experience with the bank's tooling. This perception in a nascent market might prove highly beneficial to Protectionists.

Open-first Providers

There are, of course, the players for whom open banking is the reason they get up in the morning. Open-first providers are the API consumers and providers who are at the bleeding edge of open banking, looking to take advantage of or exploit opportunities in the new market. Most of the open-first protagonists carry the "FinTech" label, but that's not to say they have to. There are the new kids on the block — the likes of the Starling Bank — who are opening developer portals to be at the forefront of Open Banking as a banking services provider. Established banks like Nordea are also taking a plunge into this market, with or without the major standards wrappers like UK Open Banking or the Berlin Group Standards.

Of course, the consumers of the APIs, the Account Information and Payment Initiation Service Providers (AISP/PISP) enshrined in the PSD2 regulation, are the big news as they directly interact with real human beings. Aggregators like Yolt, who was the first AISP to connect to CMA APIs in the UK, are making waves in the marketplace. Likewise, Zopa is bolstering its offering by using open banking data to help consumers prove their income when applying for a loan.

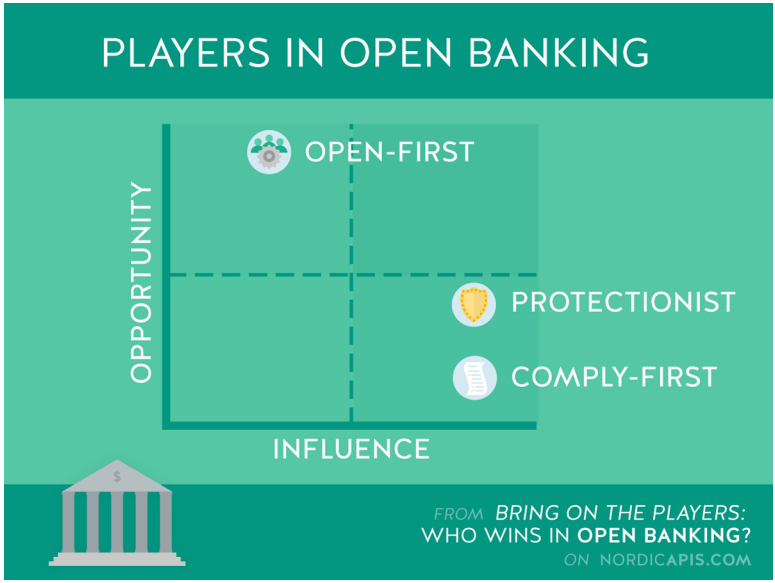
Whether an API consumer or provider, Open-first providers have some tough challenges in the market:

- As consumers of open banking APIs they need to deliver products that are so good they become a byword for innovation in the banking space and offer a credible, convenient alternative for customers to their existing bank. They are also — to a certain extent — at the mercy of the Comply-first providers, as they have a dependency on the availability of their APIs.
- API providers need to deliver APIs that allow them to differentiate their product set and make them attractive to consumers. It's more than just providing the regulatory minimum; it adds value by offering features that drive consumers.

Given the size of the challenge, it's no surprise that Open-first protagonists are coalescing into organizations like FDATA, to give them greater clout against the incumbents.

Plotting the Opportunity

Looking across the players, the diagram below shows where they sit in the current state of the market in terms of both their influence and the opportunity available to them:



The ideal in this diagram (without trying to be too much like Gartner) is to be in the top-right quadrant i.e., influential and with a great deal of opportunity. In this context:

- **The Comply-first** providers will need to extend their initial offerings to bring new, API-powered open banking products and services to the market to add competitiveness to their offerings. In doing so, they should look to leverage the qualities of their organizations that make them successful – for example, the enormous amount of collateral they hold or their ability to acquire or clear a variety of currencies.
- **The Protectionists** will need to leverage their position as a both a TPP and API provider to differentiate their services by offering them outside their existing customer base – holding a current account should not be the only criteria for entry. These organizations are also uniquely positioned given their duel role of TPP and API provider.

- **The Open-first** providers need to carry on doing what they do best: innovating, moving quickly, and delivering great products that delight their customers and enhance their reputation. By producing killer apps and experiences, they become synonymous with what's innovative in open banking.

Final Thoughts

Open Banking still has some way to go before the market reaches anything near a state of maturity. In its current guise — with multiple technical standards being developed and APIs coming into the market all the time — it's hard to see exactly how it will develop. The key activity for many has been all about regulatory compliance. So, when all the banks become API providers, what happens next?

The most likely scenario is the Comply-first, and Protectionist players will realize how big the opportunity really is. By providing APIs, they have already begun the transition to becoming banking service providers. If they accept that as their raison d'etre and accept that APIs can be used as a means to deliver on their strengths, the developments in the next phase of open banking could be positively seismic.

Case Study: Nordea's Journey to PSD2 Compliance, 300 Signups in 72 Hours

by **Thomas Bush**

The financial industry has opened up, to much Fin-Tech excitement.

For most, Nordea needs no introduction. As one of the biggest banks in Europe and the biggest bank in the Nordic region, Nordea serves over **10 million** private customers. Of course, like all other European banks, Nordea is subject to PSD2 regulations.

Gunnar Berger, Head of Open Banking at Nordea, told the story of their journey to PSD2 compliance at our 2017 Platform Summit in Stockholm, Sweden.

Nordea was one of few European banks that chose to take a proactive approach to compliance. PSD2 was coming whether they liked it or not, and as Gunnar was well aware, Nordea is "*sort of a supertanker when it comes to changing course; — it takes a while*," so when it came to building an open banking platform, they were going to do it well and in good time. Development on their PSD2 open banking platform began in Autumn 2016.

As with any behemoth project, Nordea was bound to run into some problems while building this platform. In this case, the issue was trying to balance the very real requirement of basic legal compliance with the desire to build new opportunities. Since developers would pick innovation over compliance any day of the week, the open banking team slowly started falling behind on the main aim of their project: to make Nordea PSD2 compliant.

No Presents This Year

Come Christmas, Gunnar realized it was only a year until the requirements for PSD2 had to be fulfilled — or so he thought. With that in mind, he asked his development team to focus just on the compliance aspect of their new open banking infrastructure, setting aside the more creative work.

While this wasn't much fun for the developers, it got the project back on track and meant that Nordea could start looking for beta testers from February of 2017. Without giving it too much thought, they published a signup page where external developers could apply for early access, expecting to get only a few signups. Who would want to test out the boring new legal stuff, anyway?

Within 72 hours, the page had garnered **300 signups**, with developers both big and small leaving enthusiastic comments about what they wanted to build with the platform. Supposedly, Nordea was the only bank communicating with the Nordic FinTech world at the time, although by this point, the "communication" was limited to just a signup form.

The Results Are In…

By the time signup closed, **700 applications** were tallied. Clearly, FinTechs were much more interested in the possibilities of open banking than had been anticipated — and this was evidenced in just one of the hundreds of banks trying to meet compliance.

> "Now we had created an expectation in the market — not that we will deliver some really crappy compliance APIs by the end of this year — an expectation of something much more."

It was only right that Nordea matched the enthusiasm of the applicants, and so they created blogs and newsletters, met up with developers, and decided to let everyone access the platform. Needless to say, this stirred up even more excitement and helped Nordea gauge — with minimal investment — the feasibility of developing their open banking platform beyond basic compliance.

Thankfully, they had realized by now that they had until late 2019 to meet the technical requirements for PSD2, and so by now, Nordea could even begin collaborating with 3rd parties on more innovative aspects of the project.

A World Beyond PSD2 Compliance

With this huge influx of enthusiasm and now great expectations for the platform, Gunnar's vision changed completely. This was the perfect medium to connect an old, big bank with nimble and innovative FinTechs.

It was the open banking system that could leverage Nordea's huge client base and reliable infrastructure against the FinTechs' brains and fast footedness. It would allow FinTechs to put their products in the hands of ordinary people — not just the young or tech-savvy — without having to spend millions of euros on a sales team.

The vision for what was previously a PSD2 compliance platform is now to create an **open banking marketplace** — or perhaps an even more universal developer platform. Gunnar wants to work with third parties across the ecosystem, with partners not only using the new solutions but also creating them or even creating the APIs needed to create them!

If you've ever wondered how big banks will stay relevant into the future, now you know. Instead of doing the very minimum to meet legal requirements, Nordea opened up access to their platform early, collected feedback, and then decided to dive in with both feet when they saw an opportunity.

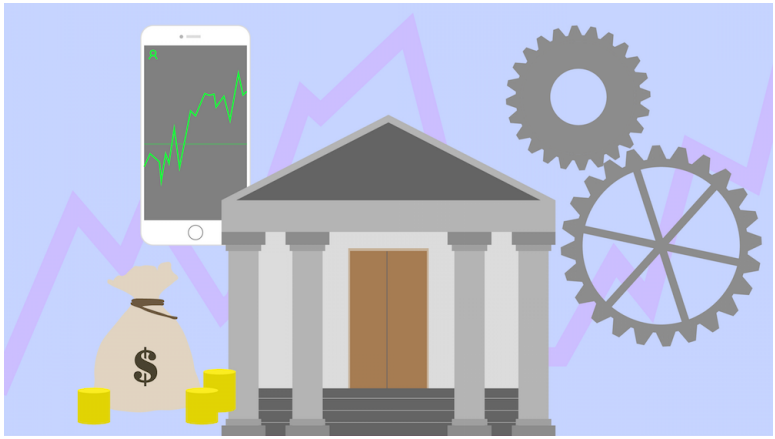
Final Thoughts

Open banking is allowing financial services to extend into a comprehensive network of services. The world of banking may no longer be closed behind locked doors, operating separate to all other technology — and is there anything really stopping banks like Nordea from expanding their open banking hub into a flexible, cross-industry development platform?

Even banks not legally affected by PSD2 might benefit from building similar development platforms, as they'll be able to stay relevant into the more dynamic, universal service markets that Gunnar predicts. In an age of endless consumer electronics and worldwide interconnectivity, it's no wonder that banks are starting to redefine themselves as *technology* companies.

FinTech and APIs: Making the Bank Programmable

by **Bill Doerrfeld**



Making the bank programmable is a win-win-win for banks, developers, and end consumers.

Banking institutions are usually portrayed as monoliths, slow to change even amidst obsolescence and new technological advances. This won't be for long, as the financial sector is becoming increasingly open and programmable.

Even large US banking institutions like Citigroup, BBVA Compass, Bank of America and Capital One acknowledge the benefits of opening internal systems for third-party developers to integrate into new apps. And, as alternative banking services like

Simple or Holvi emerge, we are seeing more of a diversity of services that embrace new consumer expectations brought on by the Internet and mobile devices.

In this article, we tune into FinTech experts to hear what organizations like Fidor, the Open Bank Project, and others did to lead the banking revolution. We'll find that we're moving toward a new financial market of customer choice, third party services, and open data — all powered by APIs. Essentially this means making the bank programmable.

What is FinTech?

FinTech stands for financial technology; a broad sector of innovative and emerging financial services. Examples include crowdfunding platforms like Kickstarter, new online-based currencies like Bitcoin, virtual wallets, micro stock investment apps like Robinhood, account aggregation and analysis services like Personal Capital, payment splitting services like Splitwise, mobile-optimized peer-peer payment transfer apps like Venmo — the list goes on.

Replacing paper checks, physical charities, paper money, and even investment firms, FinTech services are suitable to the digital times we are living in now, and are shaking existing banking infrastructure and international financial systems.

According to Stefan Weiss, head of APIs at Fidor:

> "The customer expects a different behavior from service companies, and banks, somehow, are stuck in the last century, and the movement of money is stuck in the last century."

Weiss sees the Fintech craze as not simply a short-lived phenomenon. The technologies being developed now are going

to have everlasting impacts on the future financial and banking industry, and institutions need to adopt change or face destruction. FinTech activity is international but very apparent in London, where FinTech venture capital investments are hitting record highs.

Advantages of Exposing a Bank with an API

APIs are yet again another vital cog driving disruption, now fueling FinTech startups and open banking data initiatives worldwide. According to Weiss, Fidor decided to build their banking platform with a community of third party developers in mind for the following reasons:

- A bank API allows the end-user to have a **quicker onboarding** experience.
- An API enables a bank to acquire **partners** that specialize in niche FinTech services with optimized front-end user interfaces.
- A bank API allows **seamless integration** with crowdfunding platforms, payment splitting apps, and more — great for startups with innovative financial-oriented products that may lack the budget and legal counsel to hold funds or establish their own bank.

Banks and FinTech Can Play Nice

Weiss stresses that banking licenses are tough to acquire. Even smaller legal licenses to transfer payments or handle credit, likely needed to start a FinTech, are difficult to mitigate for most

small developer teams unfamiliar with the many laws that regulate these handlings. Banks can provide this knowledge via APIs and partnerships to help FinTech companies excel, especially those who are creating APIs themselves.

Banks need to create well-designed, standardized APIs, along with self-serving adoption processes complete with documentation, sandboxes, simulated account structures, and more to get in the hands of developer users quickly. Onboard more partnerships and make it cheaper for FinTech startups to launch, and you've got a recipe for a banking API success.

Use of APIs: In-Account App Marketplace Concept

Weiss recognizes a long term goal for Fidor is leveraging their API to allow developers to create add-on services in a marketplace format, enabling the end-user to customize their banking experience. The idea is that a bank can provide an API and open app store (i.e. appstore.mybank.com) and pair apps with specific accounts. Win-win-win. Banks can acquire new partners, third party developers bring innovation, and customers are empowered with more choice and customization. It's true that for banks, building an application manager is no easy task, however, but API management solutions exist to aid this process.

> "Just because you *can* offer an API for something doesn't mean that you *have to* offer an API for something. APIs are products, have to be handled as products. You must have a customer, and a reason for the customer to buy the product"

Weiss admits that even though their full production API is making 30,000 requests daily, the team consistently strives to reit-

erate their product, tuning into the customer and considering "what is a beautiful API?"

Proof of Concepts

Open Banking Project is a separate organization striving to reinvent how banks handle their data. The Open Bank Stack, written in Scala and running on the JVM, is secured with OAuth and is a "semantic API vertical for the banking space". They've had numerous FinTech apps developed that tap into their API service. Examples include Savetastic, which pulls data from a bank to calculate potential savings, and Social Finance application, which enables account users to choose who they want to share account data with.

In a session with Nordic APIs, Simon Redfern, Founder and CEO of Open Bank Project, even demonstrated The Singing Bank, which produces varying tonalities associated with withdrawals and deposits. Combining FinTech with musical composition — now that's a tough act to follow.

Data Transparency and the Rise of Open Banking

From timestamps to transaction IDs, banks collect enormous amounts of data. In fact, it was this information that led the Fidor team to consider how it could be used to create a *better* bank.

To Redfern, "better" means more open, more transparent, and less corrupt. Apathy in companies perpetuates distrust, which is bad for society in general. With the philosophy that transparency can increase empowerment, Simon was inspired to reevaluate how banks handle and expose their data. Could company bank accounts be open for everyone to see? What if all personal bank

holdings were open for the public eye? How could this data be used programmatically to create new, ethical services?

New Platforms Lead to Unexpected Innovation

What's the similarity between a violin and a smartphone? In a session with Nordic APIs, Redfern, a composer at heart, argues that they're both platforms.

A violin and iPhone are both standardized interfaces. Yet, standardization doesn't inhibit musical creativity any more than the current smartphone design channels app developer curiosity and innovation. With platforms, weird things can happen, such as a rising class of Instagram entrepreneurs in Kuwait using the platform to sell their sheep. In FinTech, the idea is that an open standard API could revolutionize banking with unprecedented consequences.

Xignite CEO Stephane Dubois acknowledges that:

> "The role of technology in advancing the financial service industry is more critical than ever before. The use of APIs by today's banks is becoming increasingly common as they help to drive speed and cost-effectiveness compared to traditional legacy systems."

Just as Facebook and other social media giants have become platforms, Redfern believes that **banks will inevitably become platforms as well**. The idea is that banks can push innovation and allow developers the ability to create brand new products, and the Open Bank Project can allow banks to easily adopt an API. To Redfern, open banking is made up of four distinct facets:

- Open standards
- Open Source
- Open data options
- Open innovation

> "APIs are great for abstracting away aging IT systems which are barriers to innovation….We think banks need to open up their infrastructures, and they should be developing ecosystems so they can better respond to their customers and so they can compete better in the market"

It's been proven that customers are interested in using new entrant services rather than their own. According to the Millenilal Disruption Index, "71% would rather go to the dentist than listen to what banks are saying" — proof to some that the industry is ripe for "seismic" change.

More Advances in the Financial Sector

We find many accounts that the FinTech industry is growing at an enormous rate. Further indication that people desire alternatives to the politics of brick and mortar banks, Bitcoin has altered the state of the monetary conversation on a fundamental level. As Weiss puts it:

> "Bitcoin changed many things. It changed the conversation, it changed technology, because Bitcoin is more or less an internet payment system that is much more suitable to the times we are living in now, than banks, credit cards, interchange fees, etc. It really shook the foundations of international payment streams. It opened a discussion on security as well,

> and what internet protocols are capable of… Probably in 50 years from now, there will be no reason for a bank anymore."

Weiss acknowledges that what banks actually can do today is regulate and certify trust and knowledge about financial transactions. If banks wish to remain competitive, many argue that through opening up APIs to outsiders, they can form new ecosystems that encourage innovative product creations and new end-user experiences.

Conclusion

To put it simply, **making the bank programmable is a win-win-win**.

For **developers**, they get to mess around with the power and knowledge of banks to create innovative services and terminate compliance issues.

For **end-users**, they can experience a new breed of services that tie into their accounts. Politically, it could be argued that open banking transparency could decrease corruption.

For **banks**, they open new revenue streams by monetizing partner resources and experience increased customer satisfaction.

So, why not make the bank programmable?

How Can Consumers Relate To Open Banking?

by **Chris Wood**

92% of consumers haven't heard of open banking. So how do we make them care?

Open banking has garnered a lot of attention recently. With the introduction of UK open banking and Payment Services Directive 2 (PSD2) becoming law across the **European Union**, regulations are fundamentally altering the way banks handle personal data. The movement is also not restricted to Europe, with regulators in the USA having released their intentions in October 2017 on access to bank accounts.

Open banking promises a revolution in how we consume financial services products. At the heart of the paradigm shift is **APIs** exposed by financial institutions. These APIs allow us to share access to our accounts with apps and platforms of our choice. We might, for example, choose to securely share **financial data** with an app that gives us an aggregated visualization of our finances, helping us better plan our lives. Alternatively, we may choose to take advantage of payment initiation direct from a checking account and use this from an online accounting package. Opening banks with standardized APIs will alleviate a lot of friction between financial services.

However, most end consumers don't even know this great transformation is occurring. There is a sense in the media — in the UK, at least — that open banking is a "quiet revolution". A recent survey conducted by the consumer organization Which? found that **92% of consumers had not heard of open banking**. Of those surveyed, 51% were unlikely to share their account information with third parties. Mainstream media outlets, including The Guardian, also question the concept as a whole. They cite concerns over fraud and liability once data has been shared.

Given the huge investments made in open banking across the EU, how can technology providers and legislators bring open banking up the consumer agenda and make it more than just some clever enabling technology?

Building Context for Consumers

The greatest challenge the open banking initiative faces is **breaking down the technology barrier**. Open banking is usually linked with the technology that makes it possible, including the relative merits of APIs over the practice of screen scraping. This fact foists the enabling technology front-and-center and, in many cases, makes the protagonists focus on *it* rather than the **so-**

lutions open banking makes possible. Instead, we need simple use cases and explanations to help set a context for end users.

An example is the European Banking Authority's publication of a video guide on the disadvantages of screen scraping. While technology is an important part of the debate for the technologist, the regular consumer will simply want to know: *What's in it for me?*

Answering that question is not always at the forefront of the debate. Generally, those involved seek to profit, disrupt, or regulate financial services, and their message is tailored accordingly. Obviously, those who seek to profit — by providing the apps and services that leverage open banking — want to paint a **consumer-focused** picture. For others, it's not so straightforward.

For example, the European Union message on PSD2 focuses on delivering choice and flexibility in how customers access financial services and who they choose to access their account, regardless of who they bank with. However, there is a subtext that introducing payment initiation services direct from a consumer's account may break the hegemony of Visa and Mastercard in payments. Recent research tends to support a decline in card payments in a post-PSD2 world.

This introduces a significant issue, namely: How can PSD2 — or any open banking scheme — engineer a payments "network" in a few short years and foster the same **trust**, **reliability**, and **security** that Visa and Mastercard promise? The advantage the payment networks have at the moment is that they "just work". This is juxtaposed to the fledgling open banking solutions that are composed of many moving parts.

Open Banking Must Foster Trust With End Users

The possible journeys for customers to authenticate themselves to pay from their bank vary from one implementation to the next, with many banks forcing a user to use a physical passcode device. Such journeys are likely to involve redirection to online banking with multiple brands; for example, a merchant, the bank themselves, and possibly an aggregator providing PSD2-compliant payment services that a consumer may have never heard of.

By this rationale, what is open banking really giving consumers other than the perception of **greater risk** and a more fragmented customer experience? For it to resonate with consumers, **open banking needs to be marketed by its champions as more than just a technology solution**. It needs to be successfully **evangelized** with the message refocused in several ways:

- **Emphasis user control**: The technology slant needs to take a back seat, and the protagonists instead focus on how consumers are at the heart of open banking, **controlling access** to their accounts on their own terms.
- **Evangelize great use cases**: The champions of open banking need to evangelize how the technology allows consumers to unlock the potential of banking services from a myriad of different providers. Focus on how consumers can start to take advantage of the account data that — in the EU at least — *they legally own**.
- **Ease any doubts regarding security**: Finally, they need to imbibe consumers with trust in the approach and confidence of the solutions built on open banking.

Doing all of the above will have significant effects on the current

open banking approach. They are also likely to have long-term benefits that far outweigh the cost of addressing them.

Control Matters

Many of the concerns expressed by consumers focus on *who* can access their account. These include:

- How do I give access to my accounts?
- Who should I trust?
- How do I stop third parties accessing my account?
- Who is held responsible if there's a problem?

These concerns expose a structural issue with how open banking is manifested. It models itself on the infrastructure that underpins it — the internet and a collection of APIs. This decentralized and dispersed model means that consumers lack a cohesive view of what they have consented to. In most implementations that are either live or planned, this information is siloed across multiple repositories and is likely hosted in online banking platforms.

Without this centralized element of control, the user is forced to aggregate this information by hand, which is inconvenient, impractical, and makes understanding harder. To foster trust in open banking, the underwriters of the schemes across the world will need to address this by providing schema-sponsor facilities to allow citizens to see and manage what they have consented to clearly. Enabling this comes in two forms:

- **A digital identity scheme**: Already present and very successful in many countries, a standardized digital identity can be used by consumers to attest who they are when

consenting to account access, binding each consented activity with a common identifier. The Nordics provide great examples of the success of digital identity. For example, over 60% of the population in Sweden use BankID as a means of digitally proving who they are.

- **A means for consumers to collate and manage consent in a cohesive way**: This either means a centralized repository or — more practically — a network that allows consumers to manage consent from a repository they own. A personal data store would provide the perfect vehicle for this.

Lastly, high-grade API security is paramount for open banking to thrive. Delegated authority protocols like OpenID Connect and it's financial services extensions being built by the OpenID Foundation will help build faith for the technophiles in the solutions that underpin open banking. However, for the average consumer, security means control and the subject control needs to be addressed for open banking to be successful.

The Open Banking Marketplace

The flipside of control for consumers is building trust in the **third parties** that access their accounts. Understanding the role an app takes in your financial ecosystem should not be an act of blind faith.

Take the UK open banking initiative as an example. All integrators are businesses regulated by the Financial Conduct Authority (FCA). If a consumer wants to check if an organization is registered for open banking, they need to look them up on the FCA register — an impenetrable mechanism even for those that know what they're looking for. This offers little convenience

to the consumer. Compare that to a payment card in a bricks-and-mortar shop. The customer needs to do nothing to check the merchant's status other than put their faith in the Visa logo displayed in the store.

For open banking schemes to build a perception of trust for consumers, they could create some "signage" — markers that help consumers easily recognize the status of any organization professing to integrate with their account via open banking. Such markers might include:

- An open banking **kitemark** would be a quick indicator that a third party is legally registered for open banking. The implementation of this requires some thought, of course, given how easy it is to spoof websites, URLs and images on the internet. However, with the right construct, this could not only boost consumer confidence but also add an open banking brand that helps foster a scheme's identity in a given country.
- An easily accessible directory of open banking participants, designed with consumers in mind. This may be the financial services equivalent of the UK Checkatrade website intended for finding tradespeople.
- The development of an **open banking marketplace** that extends the directory idea and brings together all the solutions that use the open banking APIs in a searchable, consumable way. This has benefits for the **consumer**, who can easily search for and consume offerings from different providers in one place. It also would benefit **providers**, who get an accessible shop-window for their wares. This also benefits the **regulators** of any open banking scheme, as they can easily assess, monitor, and certify solutions entering the marketplace.

If the open banking schemes can co-create a semblance of front-tend signage, it could help foster the open banking mission.

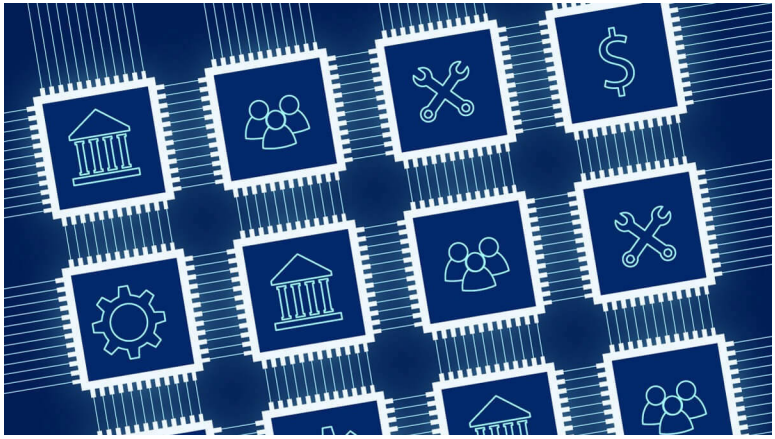
Final Thoughts: How to Establish Consumer Faith in Open Banking

Open banking represents a significant opportunity for many parties in financial services. The large incumbent banks can differentiate their offerings. Fintech will benefit massively from APIs created expressly to access accounts. Regulators across the world will shake up the hegemonistic practices of large incumbents in financial services.

However, this promise will only be realized if consumer concerns on **trust** and **security** are addressed. Open banking needs to be marketed by its champions so that consumers understand what the technology actually means for them. This is important for right now — for building trust in consumers as they start to use solutions that integrate with open banking. But it is doubly important for the future, as the services offered to expand and the opportunities grow. If this can be achieved, then open banking just might cause the financial services revolution it promises.

How Banks Are Becoming Uberized

by **Thomas Bush**



As banks treat their assets more like products, monolithic infrastructure is decomposing into an amalgamation of reusable components.

You've probably heard the whole "banks are becoming tech companies" sentiment. The "X are becoming tech companies" idea has been around for years, and used to depict the uberization of various industries. It captures evolution in the way we consume and use everything; from food to money. As consumers, we want options at our fingertips, and perhaps it's this forcing banks across the world to open up their services with

APIs and microservices, and ultimately turning them into tech companies.

In a previous role, Eyal Sivan was Senior Director of Enterprise Architecture at the Canadian Imperial Bank of Commerce (CIBC). At our 2018 Austin API Summit, he told the story of how and why one of Canada's Big Five adopted an API and microservice architecture.

APIs are Nothing New

Another statement you'll have certainly heard is that APIs are "nothing new," which is true. Eyal acknowledges this — APIs have been around since the '70s — but back then, enterprise architecture could be described by a single, dreaded word… **monolithic**.

Monoliths were big and bothersome. As Eyal says, "sometimes they were on mainframes that filled entire buildings, and you would speak to these things however you could speak to them." More an artifact of early digitization than anything else, monoliths had too many interfaces, systems kept breaking down, and were thus costly to maintain.

Enter the ERP, or Enterprise Resource Planning solutions. These were proprietary building blocks that you could use to cut up, package up, and clean up your "spaghetti" (as Eyal so affectionately remembers it). Needless to say, arbitrarily boxing up code didn't really add much to the systems architecture; ironically, it ended up costing more.

Next came the EAI, or Enterprise Application Integration. While this was an improvement over the ERPs thanks to a slightly more standardized framework, it still fell short in time — fixing none of the problems but being equally expensive and equally proprietary.

Finally, Service-Oriented Architecture (SOA) rolled around. The industry now had a standard for API contracts, and with the introduction of an enterprise service bus, each piece of the puzzle could be connected, albeit indirectly. However, as the service bus inevitably began to take on more logic, it grew big and messy in the same way an EAI would.

Part of the problem was everyone interacting with the enterprise service bus in their own way. So, the next step was to federate the different schools of thought and give each party their own domain service bus, which could contain whatever logic and follow whatever rules, while still connecting to the main service bus.

Finally, there was a system that worked well for everyone…

Smartphones: Kindling a Change

…that is, until smartphones were invented — completely changing the way we interact with IT services. Consumers now had a device that could perform any digital action from everywhere.

> "Everything changed. Suddenly people could order a taxi, a hotel room, a pizza, with a couple of taps on their phone. And consumers expected exactly the same kind of interaction from everyone they do business with, probably most of all the banks."

This changed the landscape of software development forever. Cost was no longer the deciding factor — *speed* was. Developers now wanted to create flexible software: easy to maintain and easy to upgrade.

To Eyal, the obvious solution was to break things apart, replacing every monolith with a whole suite of APIs and microservices which could be worked on individually.

Time to API Up

Once it was clear that an API and microservice architecture was the one to pursue, the bank pulled out their wallet and went to market. As Eyal explains, buy-over-build was the natural posture at a bank with plenty of money.

Analysts soon discovered that there were no leading solutions in the field, even after exploring the open-source projects, which are usually at the cutting edge of any new technologies. So CIBC took the bold decision to build.

They started with an open-source core — the Light framework — before employing the owner of that project around whom to build an entire team. With this, they created an API Foundation, and shared it amongst stakeholders to review. While analysts unanimously agreed that the documentation "sucked," the platform was something new — and it held up to tests extremely well.

Eyal justifies the decision to build their own solution with three reasons:

- It hedges against an incredibly volatile market.
- It allows you to steer the direction of your platform (in this case, it was CIBC's only solution for supporting GraphQL).
- It cultivates critical development skills (OAS, Swagger — now OpenAPI, DevOps, daily releases, automated testing/deployment, and so and so forth).

Building with Purpose

The decision had been made. CIBC was going to build their own microservice and API architecture, and now it was time to get

work. But before they could do so, they'd need to establish their guiding principles — among other things — Eyal explains.

Guiding Principles Come First

Every organization has its own priorities, so there's no one-size-fits-all approach to building any kind of network architecture — this is another advantage of developing your own solution.

With that in mind, you need to establish your guiding principles before you get going. The dichotomy Eyal focuses on is proliferation versus standardization and governance. That is to say, do you value a solution that's easy to build upon (with, say, adding new functionality every day), or a solution that's well-kept and easy to maintain?

For the record, the right principle is probably somewhere in between.

The Goal Product

Next up, it's important to appreciate the nature of the product goal. Eyal stresses that they did not build a "magic box" that takes care of everything, which would be a modern repackaging of SOA that eventually hits a wall.

Instead, they took a distributed gateway approach to embrace microservices — the "secret sauce." Let's note here that while you might be able to package all the same functionality into a monolithic mainframe and expose that with APIs (which would be significantly simpler), you'd then stuck with no space to grow.

CIBC then took the microservices and gave each one an embedded gateway. That way, the microservices can act and be managed independently from one another in every aspect, from security to analytics. There's no blanket ban on centralized gateways, but let them be used for the stuff that really is centralized.

According to Eyal, the rise in popularity of these architectures, or service meshes, could worsen volatility — yet another reason to cultivate development and management skills in-house.

CIBC's Service Mesh Plus

Aside from the embedded gateways, CIBC has purpose-built plenty more features into their service architecture to make things as smooth-sailing as possible — hence the label of Service Mesh Plus.

For one, they've chosen to move away from the typical homogenous mesh (where each user shares the same mesh), including a series of light proxies and routers to improve compatibility.

What's more, they're looking to implement federated authorization — a trusted network across OAuth servers. ##Acting Like a Tech Company While some of these things might seem foreign to you, Eyal notes that doing APIs and microservices the right way forces you to act like a tech company. That is "the critical skill here; treat your assets more like products."

The uberization of banks is coming, so acting like a tech company is "the shift for banks to survive," he goes as far as to say.

That's presumably why Eyal's not afraid to share how far CIBC has come in that regard:

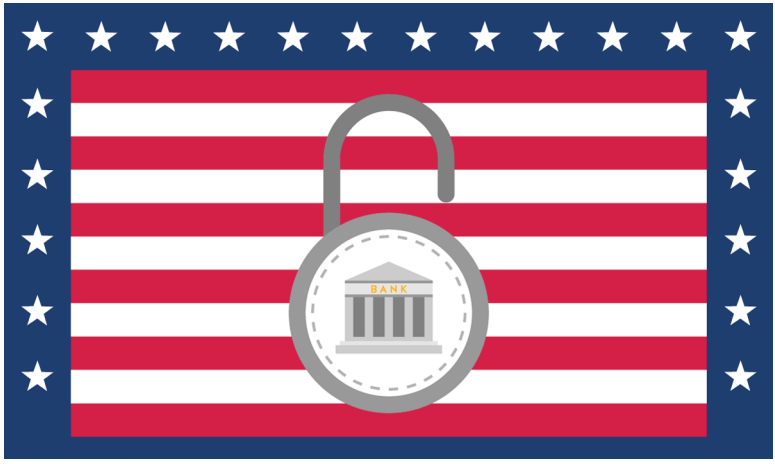
- The API Foundation is used across the bank, with a governance council mandating its use — unless there's a good reason not to.
- There's a managed container environment with a development network already in place.
- They've created an adjacent developer training program, with the goal to create an internal community around the platform.

- There's a pilot of the API marketplace available today, with a self-serve model allowing developers to use the API without a single meeting or call (allowing cost savings of 50-70% per integration). ##Eyal's Tips for Your API Platform Here are some of Eyal's tips for building your own high-yield API platform, with the fantastic metaphor of a brick-and-mortar building:
- **The four pillars**: Developing a powerful network is a balancing act. Grow your API, cloud, DevOps, and agile strategies together to keep things running smooth.
- **The foundation**: Your solution needs to be built onto a solid foundation, which runs from your technology and operations team, through the company's organizational structure, down to corporate values and culture. Can you convince bank officials that innovation is the right focus to have?

Eyal finishes with what he considers the single most important idea for success: **build for change**. IT systems are changing, and while we can't know where they're headed, we can build with that uncertainty in mind.

How Does Open Banking Apply to US Banks?

by **Chris Wood**



The future of open banking in the US market is more likely to be driven by demand than by regulation.

Open Banking has been on our radar for several years now, with Nordics APIs first publishing a post on PSD2 back in 2016. Since then, there's been continual coverage in the financial industry press on the promise of Open Banking, with predictions of a "revolution" as FinTechs unlock the banking market and provide the tools, apps, and banks do not offer functionality that bank customers want yet.

Banking APIs are hitting the European market in earnest in 2019 to meet the deadline for delivering PSD2. The potential for this revolution to become a reality has never been greater. It's no secret that Open Banking is most advanced in Europe because of PSD2, which has brought with it standardization through standard API specifications developed to deliver it. This has led banks down the path from private, closed APIs to public, open APIs, and allowed them to become functioning members of the API Economy.

Most commentators agree – including recent discussions in a Senate Committee — that it's high time that the US got in the Open Banking game. However, the context of Open Banking in Europe — with regulation currently driving the market — could be met with suspicion by many US citizens. Regulation from the center has overtones of socialism, a highly divisive word in US culture, and attempts at a top-down Open Banking implementation would leave many questioning its value. Banks and their customers alike, therefore, need to understand what's in it for them, relying on influences that resonate with the market.

Therefore, in this post, we take a look at the prospects for creating such a market. Without the regulatory drivers, unlocking the potential of Open Banking in the US may be more complicated than it first appears.

Regulation in Europe

Before discussing the US market, it's worth noting the scope of European regulations. The need for banks to provide open APIs in the European market has mostly been driven by the requirements of the Second Payment Services Directive (PSD2), which concerns itself with access to payment accounts. The context for change is centralized and driven by the European Union,

responsible for legislating in many areas across the 27 member states.

In this context, encouraging an Open Banking market through regulation is relatively "straightforward," especially in a market where openness is already written in law. For example, passporting of financial services – which allows an organization authorized by the competent local authority in one member state to operate in any across the EU – is one such a regulation that enables an open market to work across national boundaries. PSD2 introduced roles to enhance this open market, allowing third-party providers (TPPs) to become regulated as either an Account Information Service Provider (AISP) or Payment Initiation Service Provider (PISP).

PSD2 is also a maximum harmonization directive. This means national law may not exceed what the legislation intends to deliver. As each member state must provide an implementation that is equivalent to the law requirements, standards help massively as each member state essentially attempts to achieve the same thing. The market has been supported by standards bodies that have built API specifications for them to adopt – the Berlin Group, UK Open Banking and STET – easing the process of bringing APIs to market.

Regulation in the US

Where central legislative bodies take a less active role - or where there are several bodies with overlapping responsibilities - API-enabling banks to open the market is more complicated. US legislation is generally dealt with federally, with further laws set on a state-by-state basis. Top-down change is rare, with central government institutions often setting guidelines but less frequently enacting legislation aimed at wholesale, nationwide regulation.

The driver for Open Banking in this environment is Section 1033 of the Dodd-Frank Act which legislates that US citizens can permit access to their financial data. Open Banking is being "encouraged" in this context by a central body, the Consumer Financial Protection Bureau (CFPB). In October 2017 they outlined their principles for "Consumer-Authorized Financial Data Sharing and Aggregation" which, in the words of the document itself:

> … recognizes that many consumer protections apply to this market under existing statutes and regulations. These Principles are not intended to alter, interpret, or otherwise provide guidance on — although they may accord with — the scope of those existing protections.

The document goes on to lay out the principles by which consumer-authorized access should be achieved, including granting access itself, the scope of access, and how consent is obtained. In general, the principles are high-level and are in stark contrast to the exhaustive (although still subject to interpretation) detail of PSD2 or the Regulatory Technical Standards that govern customer authentication. The nature of the principles indicates that US government agencies are unlikely to seek centralized regulatory means to foster Open Banking.

In lieu of regulatory pressure, some bodies are seeking to set standards that will encourage open API adoption across the US market. For example, the National Automated Clearing House Association has released standards under the banner of Afinis with account validation and bank contact APIs. However, it is early days for these efforts given key features, such as payment submission (via ACH) and transaction status APIs, are still under development. The initiative also lacks the impetus that maximum harmonization offers, as the CFPB principles indicate that federal law will always take precedence. The same could also

be said of the Financial Data Exchange with the Durable Data API, a similar group primarily aimed at providing consolidated account access.

From an objective viewpoint, it is difficult to assess whether these initiatives serve the principles of the CFPB document, or whether they are aimed at the interests of the participants in the groups. The immaturity of the standards and weaker regulatory drivers also means there is less impetus for banks to participate in developing and adopting common standards. On this basis, it appears that an Open Banking revolution led from the center is unlikely. Consumers in the USA are therefore likely to be more reliant on market forces to bring them Open Banking and the benefits it delivers.

The Role of the Market

When Open Banking is discussed in the context of any market, there is a tendency to lightly gloss over what's gone before. Perhaps this is down to the fact that the "Open" in Open Banking is the same "Open" in open API. In the manner of the Old Testament, Open Banking is the Word, and the Word is API…

Regardless of whether open APIs are (obviously) a good thing and (of course) should be implemented, a marketplace for sharing account data with third parties already existed in Europe. Local initiatives such as HBCI/FinTS in Germany, direct collaboration with banks (API-driven or not), or screen-scraping made it possible. The examples of successful third parties in this space are too numerous to mention. The role of Open Banking in this market - and the legislation that drives it - is to standardize the interfaces and ensure their adoption, making access for third-parties - and by proxy, banking customers - ubiquitous.

Despite the obvious advantages of open APIs that the stan-

dards brought, many protagonists in the market argued against the proposed (now rejected) ban on screen scraping in Europe, which we covered in a previous post.

If the strong regulatory drivers are missing in the US market, then third parties with compelling products must fill the space in the same manner as the pre-PSD2 market in Europe. The development of Open Banking in the US will rely on the players to ubiquitously open up bank accounts for customers as it already does today. For example, Yodlee provides access to 99% of US banks, delivered to API consumers through a single API. Similarly, Plaid delivers a single banking API for third parties and connects to over 1700 banking institutions for both account access and authentication. Behind the scenes, such providers are responsible for connecting their API platform to the myriad of systems supported by the banks to deliver a seamless experience for customers. On the payments front, the success of players like Stripe is well documented. However, The Clearing House (TCH) is also standardizing how real-time payments are made in the US, which has historically been fragmented across multiple platforms. While this platform is currently not an open API - being message-driven in a closed network - the messaging interfaces are based on ISO 20022. They could provide the basis for open payments APIs.

Organizations like Plaid, Yodlee, and TCH provide a live proving ground for the value of Open Banking. By delivering compelling solutions in the marketplace - which many US consumers want and use - they offer a persuasive argument that providing access to an account can only be a "good thing". Such evidence mirrors that found in Europe, where providers like Figo were already delivering near-complete coverage of the German market before the rollout of PSD2-compliant APIs. Therefore, the success of such providers is critical to the success of Open Banking in the US. In the absence of regulatory-driven API standards, they will provide the *de facto* Open Banking APIs for early adopters.

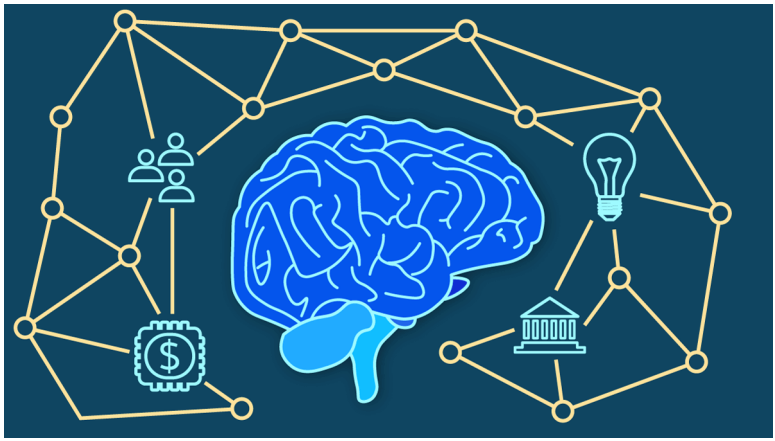
Final Thoughts

The future of Open Banking in the US market is more likely to be driven by the market than regulation. However, there is still an opportunity to introduce standardization through open APIs behind the scenes. As TPPs like Plaid and Yodlee continue to increase their footprint, a means for banks to easily offer an interface and a go-to API standard will help increase penetration. A partnership model could develop, with TPPs proactively working with banks and providing the interface on their behalf. In this context, it might be that TPPs can help bring standardization and open APIs to the banks, and organizations like FDATA would do well in establishing a US footprint.

However, there is also an equal opportunity for a **BigTech** to steal a march on the Open Banking market. While this is equally possible in Europe, the regulatory aspect to the Open Banking market means that exceeding the provisions of an AISP or PISP to introduce the "killer" app might be difficult. In the US, the constraints on *what* Open Banking can do are not defined by regulatory roles, and the authorities are far more accepting of big business influence. In the same way, as Apple influenced Visa and MasterCard to provide a solution to introduce Apple Pay, one of the FAANG groups might find a way to broadside the US Open Banking market. If this comes to fruition, the hype might become a reality, and Open Banking really will revolutionize financial services.

Case Study: From API Doing to API Thinking at ING Bank

by **Thomas Bush**



"API thinking" takes a holistic outside perspective that considers developer consumer needs.

In any sizable organization, it's hard to feel like you can make a difference with just your tech skills. There's no surprise that software engineer Flavia Sequeira felt precisely that when she joined ING, the multinational banking firm with over 50,000 employees, in 2013.

You see, building an impactful IT product can take a lot of time and effort. So, it's hard for management to get excited about that when it doesn't offer an immediate value proposition. Who

needs an API when we can be fixing server downtime, or getting the user interface all dressed up for next month's holiday?

That was the case with APIs when Flavia joined ING. The business folks were aware they wanted a more open approach to technology, one that could potentially change the face of 21st-century banking, but it wasn't obvious where to start. Surprisingly, building an API wasn't an obvious solution to that, especially since the IT team had already created a web service.

As Flavia described in her session at the Nordic APIs Platform Summit, it was quite the journey to educate her coworkers on the benefits of an API thinking, as opposed to just an API doing. And it all starts with a case of semantics…

APIs versus Web Services: What's the Difference?

Flavia tells of an early API community session at ING, where overwhelming negativity towards APIs filled the room. The experienced developers simply couldn't see what APIs had to offer that their web service didn't.

The common consensus is that web services offer a more limited range of functionality than APIs, delivering particular data in a standardized format. This is opposed to providing a comprehensive development suite for building onto an existing IT product. Flavia had (and still has) her own take on the matter:

"APIs are built outside-in", she announced in the meeting, followed by a long silence.

When asked to expand on what she meant, Flavia explained that APIs are built from the customer's perspective, to the **customer's needs**, while web services are built from the organization's perspective, offering what they perceive as important.

This means that, whereas web services might offer just a single service or set of services for customers to make use of, public-facing APIs allow for much more **interactivity** with your product, such as tracking customer journeys.

API Doing vs API Thinking

The above anecdote makes it clear that Flavia joined ING with a completely different mentality towards APIs. She wasn't as much interested in API *doing* as she was API *thinking*, and that was an entirely novel approach.

She didn't ask herself: * What data and functionality do we think is useful for users? * How can we provide them with that data and functionality? (In other words, how we do we build this API?)

Instead, she asked herself: * How do our users want to engage with our services and what do they want to build? * How can we make it as easy as possible for them to do so? * How will that better help us understand and serve the needs of our customers?

Of course, that first set of questions — the API doing questions — are just as important as the second set of questions. In fact, they're the more practical questions for a developer to ask.

However, it's the second set of questions — the API thinking questions — that really justify an organization to build an API. What's more, they solve that initial dilemma of convincing management what APIs have to offer.

APIs and Customer Journeys

Flavia's anecdote continues, in parallel with what we've just said above. She recalls how a senior employee then asked her to expand on her thoughts on customer journeys, presumably

wondering why a software engineer would be tracking a customer's experience with ING from start to finish.

The truth is that well-designed IT products are tightly linked to an awareness of customer behaviors, requirements, and desires. Not only does the latter help to build products that the user will love, but products that the user loves will themselves help to flesh out customer behaviors, requirements, and desires. This occurs by encouraging open thinking in developmental stages and providing empirical data in operational stages.

This is why customer journeys from the business department at ING help to build better APIs, and why better APIs help the business department to create fuller, more effective customer journeys.

APIs share that same symbiotic relationship with other business tools, like service blueprints, and business practices, like design thinking. These pave the way for a more functional, complete API, while the API helps to visualize and facilitate new opportunities.

What's more, APIs can be a solution to decoupling front and back ends — allowing the end product to link up closely with its inner workings, without making the two entirely reliant on each other.

This is API thinking. It's understanding what APIs can do for the organization, from a holistic perspective, and not just meaninglessly building APIs because they are trendy.

Why API Doing is Equally Important

When you're tracking this kind of high-end thinking, it's easy to forget about the true value that APIs can provide in modern-day banking — and that's very much in the doing.

See, unlike the business tools mentioned above, APIs offer just

as much to the user as they do to the bank itself. API usage in banking, which is part of a greater movement called open banking, allows FinTech services to draw from the huge information base and varied functionality that banks already have.

Think about it — there are thousands (if not millions) of third and first-party apps built into our mobile phones, social media platforms, and even houses. Yet there's hardly an equivalent for today's banking services.

Imagine having an app store for your bank account. You could choose from thousands of services, built with the collective thinking power of third-party developers — big and small — that would solve a multitude of problems and change banking for good.

The possibilities for this are endless: collect all your accounts in one place, send cash over social media, manage your spending patterns, verify past transactions, and so forth.

With the way things are today, it's not that it can't be done, but more that it's impractical. Of course, any financial services — banks included — need to be well-secured, with information locked away in internal systems. This is why the existing (and highly limited) model of giving FinTech services direct access to your account (known as screen scraping) simply can't go on. The finance management service Mint performs screen scraping regularly — gaining direct access to your bank accounts — but it feels safe enough knowing their parent company, Intuit, already holds millions of social security numbers.

However, requiring user login information is a significant barrier-to-entry for smaller FinTech developers. These applications may only need a limited scope of data to operate, such as your spending dates or locations your credit card has been used at.

APIs are an obvious solution to this, allowing third-party services to interact with banks, drawing only the necessary information,

and performing only the necessary services. With standardized authentication and delegation protocols widely established for APIs, not only would it be more secure — it would be more effective.

Changes in regulation, like with PSD2 in the European Union, are already pushing for and preparing for this change. Ultimately, though, it's up to banks to open themselves up to the world in a way that is usable.

From API Doing to API Thinking

When Flavia joined ING, they didn't just lack in API execution, or "API doing", but they also lacked in API understanding, or "API thinking." This was making it hard to execute a definite platform strategy.

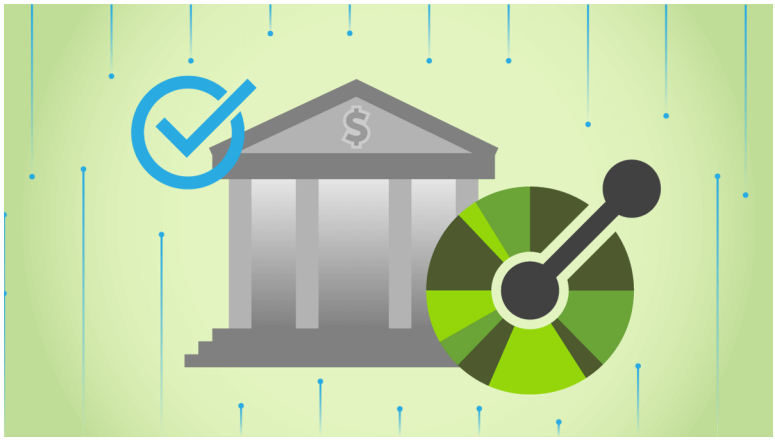
She's solved this by encouraging others to look at APIs holistically — to understand what they bring to an organization and its clients or partners, and to employ that same information to then build better IT products.

Along Flavia's journey, her career description has changed from "software engineer" to "API thinker." She doesn't just build IT products at ING; instead, she educates fellow developers on how those products, especially APIs, can be designed true to both ING's values and its customers' experiences. This, in turn, will allow them to create a better overall customer experience.

Ultimately, APIs aren't made from doing or thinking alone — it should be a mix of the two. Despite Flavia being an API thinker, there has to be a number of API doers at ING, too.

Open Banking Amplifies the Need For Definition Driven APIs

by **Saoirse Hinksmon**



Adopt an API specification like OpenAPI to organize and standardize API design practices across an org.

In today's global economy, it's no secret that consumer power is growing. Organizations all over the world are rethinking their digital strategies to maximize their customer focus and engagement. Customer engagement is the catalyst that jumpstarts digital transformation, mobile-first, and other strategic initiatives to bring legacy systems into the modern digital age. To accomplish

these initiatives, many companies look to APIs as the primary building blocks, rapidly increasing the number of internal, external, and partner APIs they rely on for critical business transactions.

A tangible representation of consumer power and API growth lies in the acceptance and promotion of the Open Banking Standard. In recent months, open banking has been the buzzword for FinTech organizations worldwide. The Open Banking Standard tackles concerns for more institutional transparency and more options for consumers when selecting banking services and providers. Under the Open Banking Standard, banking data is shared through secure open APIs to promote consumer control and lower entry barriers into a market that has historically has been difficult to break into.

With the promise of a cultural revolution, Open Banking Standard requires financial institutions to expose APIs for consumers and vendors alike. However, creating and exposing public-facing APIs to a variety of consumers would require standardization to ensure that people are able to understand what the API does and what information it is supposed to return.

The tricky thing about creating and deploying valuable APIs is that they need to serve both human and machine needs, both of which are inherently different. Humans prefer more instinctual, easily understandable instructions, while machines require more explicit, logic-driven instructions. Implementing an API standard that both humans and machines can understand, especially with the lens of open banking, is essential to finding long-term success with exposed and internal APIs.

Adjusting Practices With The Shifting API Landscape

As financial organizations increase their reliance on APIs, a variety of different challenges arise. We've mentioned the challenge of designing an API so that both humans and machines can understand their functionality, but the challenge does not end there. Some of the hindrances that can impact API development and growth include:

- Collaboration
- Quality
- Scalability
- Availability
- Performance
- Security
- Compliance

With the mounting challenges, organizations also have to keep in mind that practitioners sometimes have different needs from stakeholders, and striking the right balance between practitioner and stakeholder requirements is the only way to progress as an organization.

How can financial organizations manage all these factors, while maintaining the right balance for practitioner and stakeholder needs?

- **Use OpenAPI**: Take a definition driven approach to API development.
- **Share responsibilities**: Ensure quality is a shared responsibility for Dev, Test, and Ops teams.
- **API Virtualization**: Consider using service virtualization to accelerate development.

- **Keep iterating**: Monitor end-user experience and performance in pre and post-production.

How OpenAPI Specification (OAS) Accelerates API Development

In Definition Driven API Development, you design the API definition before any other lifecycle operation. The design of the API's interface, requests, and responses are finalized before other lifecycle functions, like building the API's business logic or testing the API for errors or defects. The definition-driven approach brings with it some great benefits, such as better developer experience, team collaboration with independence, and a faster go to market time. All these help create a consumer-centric approach to API development.

The OpenAPI Specification (OAS) is to REST what WSDL was to SOAP, providing a common framework that designers, developers, testers, and DevOps teams can use to build and maintain APIs. The specification offers a set of rules to develop and implement a REST API. It is language-agnostic and is both human *and* machine-readable. OAS offers a way for all stakeholders to discover and understand the capabilities of a service without requiring access to source code, additional documentation, or network traffic inspection.

For financial institutions, the OAS framework can help bridge the gap between consumers and institutions, without requiring much additional effort from either party. Adopting this framework can help re-frame negative consumer perception of exposed APIs. It can also help break down the technology barrier between consumers and banking providers to build trust with end-users. All of which will be increasingly vital for finding success with an open banking initiative.

Supporting OAS Throughout the API Lifecycle

Instilling quality in the Organization

Typically, software teams are segmented by responsibility, although now, there is a shift to make organizations more interoperable. Sharing duties with explicitly defined metrics empower teams to collaborate while maintaining responsibility and accountability. Sharing quality metrics can help promote rapid development cycles, reduce the number of bugs, and ensure that the APIs pushed to production retain their quality throughout the entire lifecycle.

Service Virtualization

Rather than mocking and sandboxing, which can limit how many scenarios can be tested at a time, service virtualization empowers teams to use virtual services instead of production services, enabling frequent and comprehensive testing even when key components are missing from the system architecture. It is especially useful in the development of complex cloud, API, and SOA-based systems, as well as at any point in a production cycle where important hardware and software components aren't readily available for testing purposes. For API design and development in the financial industry, service virtualization offers a variety of benefits:

- Rapid creation
- Virtualization can be pre-packaged, modified, and reused
- Is not heavily manual
- Import real data into tests for more realistic results

More and more companies are using service virtualization to improve productivity, reduce testing costs, and deploy higher-quality software in a shorter timeframe. In addition to emulating major software applications, third-party services, and even whole backend systems, the virtual assets can also be reliably shared and used by the entire production team, facilitating more efficient parallel development practices.

Monitoring End-User Experience in Pre and Post Production

For most companies, API monitoring is a relatively new concept. However, for companies relying on APIs for critical business transactions, understanding the availability, performance, and functional correctness of their APIs is paramount for their digital experience delivery.

Monitoring key API transactions, whether they are a simple endpoint call or APIs called in sequence, allows organizations to fully understand how their APIs are behaving. Identifying performance problems, remediating issues, and fixing functionality mishaps is easier than ever when you choose an API monitoring tool that can reuse functional test scripts and OAS files, and perhaps even natively add new chained API transaction monitors.

Another trending strategy is to monitor APIs in pre-production environments. These monitors do not replace testing efforts, but rather complement the testing process. Pre-production monitoring offers benefits such as:

- Ensure that any new updates or changes will not break the monitors in production
- Gain a view of performance from an on-going perspective
- Capture performance and functionality metrics outside of the specific test case

- Insight into the state of your test environments themselves, so you can maintain the environment stability

Final Thought: Drive Open Banking API Strategies with OAS

The implementation of open banking and PSD2 will only serve to heighten the pressure financial services feel to create and deploy new or additional public-facing APIs. Fostering trust and transparency with both practitioners and stakeholders is only the beginning of the challenges teams will face in crafting great APIs, but using a framework like OAS can help streamline consistency and collaboration. Thanks to a vibrant community, many tools are available to support OAS defined APIs from development to deployment, making the adoption decision clear and the transition easy.

High-Grade API Security For Banks

by **Kristopher Sandoval**

Consider these data regulations and API security open standards to address them.

Financial institutions occupy an exclusive zone for APIs primarily because of how stringent the regulatory compliance rulesets are. The data that financial institutions leverage is protected widely by a variety of regulatory ordinances. As such, this data has to be stringently controlled, secured and managed – hence why high-grade API security is such a serious concern.

In this chapter, we're going to discuss the regulatory severe compliance considerations inherent in financial data exchange

and management. We'll then look at some methods for securing such data when it comes to APIs (like API keys, OAuth and JWTs). We'll see what can be done (and should be done) to ensure consumer security and protection.

Regulatory Compliance Considerations

To contextualize our conversation, here are some of the most common regulatory guidelines financial sector providers may come across.

- Basel II is a set of international standards that requires financial organizations to evaluate and mitigate operational risk losses of financial data. It is concerned explicitly with insufficient security for data and system failures due to improper configuration or poor expectations of system demands. As such, it is a good starting point for many systems planning to utilize financial data.
- PSD2 - The second revision of the Payment Services Directive hails from the European Union, and was designed by the European Commission to regulate payment services throughout both the EU and the European Economic Area. The regulation is primarily concerned with consumer protection and establishing rights, obligations, and expectations for payment providers and financial institutions.
- The FFIEC Uniform Rating System for Information Technology (URSIT) is a US federal standard for examining a company for proper Auditing, Management, Development, Acquisition, Support, and Delivery processes. URSIT is a great tool for setting up a process to identify security concerns and can serve as a type of framework for such a process.
- The Gramm-Leach-Blilely Act is a US federal act that requires financial institutions to ensure the security of both

financial and personal data. This act is the foundation of the data Safeguards Rule as issued by the FTC, which requires an information security plan that incorporates risk assessments as part of a total security auditing process.

- PCI-DSS is a regulatory standard that requires vulnerability scanning and source code review to ensure that payment card industry data and processes meet the stringent security processes needed by providers and payment vendors. PCI-DSS is critical for many online organizations, especially those managing payment processing for a product. As such, any API that has a monetization approach that integrates card payment natively should be very aware of PCI-DSS and its requirements.
- Sarbanes-Oxley requires that internal controls report to the provider as a means to track the accuracy and security of financial data. It includes rather stringent audit mechanisms for internal controls and demands the examination of IT assets, software, and solutions for resistance to data breaches and exposures.

This is only a small selection of the most common and top-level regulatory standards. Depending on where you regionally do business, regulations can be narrower, and these variations can become even more marked in separate fields within the same industry. That being said, understanding these top-level regulatory structures can help frame the conversation moving forward.

Identifying Vital Data

We also need to identify what specific data is protected. We can sort our data into one of three categories: Personal Data, Financial Data, and Transactional Data.

Personal Data is exactly what it sounds like. This data type is specific to each user, like Social Security Numbers or National Identification systems, addresses, names, and more. This type of data isn't necessarily dangerous to expose on its own (though Social Security Numbers and other related content are often federally significant). In combination with other data on the system, it can be used for personalized and directed attacks.

Financial Data is all data related to the accounts held by Personal Data, and typically has more to do with the funds available to a person, their standing in the bank, credit information, and other data. Again, while not specifically dangerous on its own, in combination with Personal Data, it can quickly become a huge security concern for the user.

Transactional Data is intimately related to Financial Data, but it is separated here due to the nature of its data generation. Transactional Data is the data that dictates transfers of funds, purchases, movements from account to account, and more. This data is often required to be kept for organized crime prevention and fraud investigations, but could also be used to capture authorization codes for replay attacks, track purchases for fraud and blackmail, or even be used as a means of monitoring an individual's approximate location.

It should be noted here that individually, none of this data is necessarily dangerous on its own (though certainly, the potential for single-vertical damage is indeed present). While they should always be secured as stringent as possible, knowing someone's address is a security exposure for sure, but not one that could result in the kind of massive crimes that are possible in financial theft.

The danger comes when these data types are combined – knowing account holdings, addresses, names, and more can result in massive fraud and damage to the user. As such, the highest weakness should be considered allowing any data at all to leak,

regardless of perceived importance or value.

Potential Vulnerabilities

While financial data has its own set of caveats and considerations, it is ultimately just data, and as such, can be considered the same as any set of data would be regarded as when it comes to potential vulnerabilities. Every data set is governed by three simple letters – CIA.

As any security professional can tell you, **CIA** is an acronym that represents the security scope when it comes to secured systems – **C**onfidentiality, **I**ntegrity, and **A**vailability. Understanding these three concepts will help highlight vulnerabilities, and can expose serious issues that are often hidden away from more top-level audits.

In **Confidentiality**, the concern is keeping data private and exposing it only to those who have the right to view it. This is a significant part of financial protection, as the account holders will not be the only people accessing the data. Investigators, auditors, bank managers, employees, and others may have to access bank data from time to time. As such, ensuring different levels of Confidentiality is key to ensuring vulnerabilities are kept to a minimum.

Integrity is concerned with data being stored and unaltered between uses. This is majorly important for banks, as data in the ledger states the value of holdings – allowing amounts, points of origin, transactional information, and more to be altered would nullify the entire purpose of a financial institution and would result in absolute abject confusion and panic. Thus, securing systems from incorrect write actions and backing up these ledgers in secure form for restoration is key to ensuring these vulnerabilities do not propagate.

Availability is an essential aspect of banking as well, as there is a significant amount of regulation concerned with ensuring users and investors have access to their systems and funds. Downtime is often associated with additional cost, loss of investor confidence, and exposure to systems to attack; thus, ensuring Availability by distributing server load and filtering fuzz traffic is key to any banking institution.

While there is a wide range of vulnerabilities that are not discussed herein, these vulnerabilities will broadly fall into one of these three categories.

API Security Methodologies

Now that we understand our vulnerabilities, we can begin to look for new solutions to mitigate them. No single solution is going to be perfect here. As such, each solution as presented is going to represent only a single part of a greater solution, and should be contextualized as such. No solution is going to apply to every level of the financial institution. This should be kept in mind while cost-averaging and contextualizing the solutions for their given use cases.

API Keys

API Keys are a fundamental mechanism of API security and are used for low-level and low-security APIs around the world. The concept is simple – generate a key for the user that identifies them to the system so that when a request is issued, the system itself can identify the key and note the user is who they claim to be.

The problem is that API Keys themselves are not entirely secure. API Keys are not meant to serve as a method of securing access,

but only as a method of proving the user is who they say they are. Compound this with the fact that API Keys lack control in granularity (often either being a "yes or no" option for identification), and you very quickly run into a situation where your API Key is not enough.

The vast problems with API Keys come when end users, not developers, start making API calls with these Keys, which more often than not expose your API to security and management risks. What it comes down to is that API Keys are, by nature, not a complete solution. While they may be perfectly fine for read-only purposes, they are too weak a solution to match the complexity of a high-use API system. Whenever you start integrating other functionality such as writing, modification, deletion, and more, you necessarily enter the realm of Identification, Authentication, and Authorization.

In other words, API Keys serve a particular purpose, and can be used to significant effect for read-only functions of publicly exposed endpoints on insecure data (such as verifying server status or beginning a more stringent authorization/authentication process), but should not be considered a "security solution."

OAuth and JWTs

For secure data, a combination of OAuth and JWTs can be very useful in securing data.

JWTs, or JSON Web Tokens, are very small, self-contained packages that contain all the information the server will need to process a request, and then later all the data generated by the server that the user needs. As such, the JWT represents a very efficient, lightweight communication methodology in the API network, and it offers all of the perks of JSON (both machine and human readability is a key perk of JSON-type structures).

JWTs are not encryption, however, and serve only the process

of encoding the data – as such, they are only as secure as the system they reside on. This is one caveat of JWTs. There is indeed a specification for JWT encryption, known as JOSE – JSON Object Signing and Encryption – that can offer greater security, especially when paired with more stringent encryption and security methods.

With our encoded package out of the way, we can begin to look at other methods of securing access to the system itself. OAuth is a token-based system for authentication and authorization first proposed in 2007 by various developers of Ma.gnolia and Twitter. Since the official publishment in 2010 of RFC 5849, which defined the standard, all Twitter applications – as well as a great many online applications in general – require the use of OAuth. RFC 6749 created OAuth 2.0, an updated framework evolution that is paired with a Bearer Token Usage definition in RFC 6750.

While it's true that OAuth is a system of "authentication and authorization," it should be said that OAuth by itself does not actually provide an authentication protocol. Instead, it sets a framework for authentication methodologies and decisions. Its native authorization protocol, or more specifically its authorization-through-delegation protocol, uses four general entities to control data access:

- A Resource Owner (RO) controls the data being exposed by the API and is considered the "owner" of said data.
- The Authorization Server (AS) is a Security Token Service (STS) that issues, controls, and revokes tokens for access – this is often also referred to as the OAuth Server in the internal system.
- The Client is the application, user, website, etc. that requests data on behalf of the resource owner.
- Finally, the Resource Server (RS) is the service that exposes and stores the data and is typically the API itself.

The “basic flow” of OAuth is the “implicit flow,” and functions like this. The Client requests access to a resource by contacting the Authorization Server. The AS then responds to this request with its own request for data, namely the user and password information. The AS passes this data, once received, to the Authentication system, with either responds to the AS with approval or denial. From here, if approved, the AS allows the client to access the Resource Server. This can be further extended using a Bearer Token, which essentially gives the client a revocable, often time-limited token that can be used to access resources over time with a given scope of rights.

OAuth has some great benefits for banking institutions, but the best aspect is that it fits with the natural organization of a financial institution. The Client functions as a teller or manager, the AS functions as an internal server verifying account ownership. The token functions as a sort of ID card verifying ownership and rights – in this way, at least insecure situations, OAuth is a great solution for mirroring bank organizational structures in the way it handles its own data.

OpenID Connect

OpenID Connect is a simple identity layer that is designed to lay over OAuth. OpenID Connect is actually the third revision of the OpenID approach. More specifically, it was built to overcome the earlier design limitations of OpenID 2.0 concerning Relying Parties, XML reliance, and more. OAuth 2.0 is the substrate for OpenID Connect and utilizes HTTPS (specifically TLS/SSL) infrastructure for data security. We’ll cover OpenID Connect more in-depth in the following chapter.

Security is The API Provider's Responsibility

It's important to discuss exactly where the responsibility for data security actually exists for the financial industry. In theory, responsibility for security should exist at all levels of the API space, with users securing their devices, clients implementing strong encryption, and so forth. While this is a lofty goal, the reality is often that someone, somewhere, is going to do something counter to the overall security of the system – and in the financial industry, this can be extremely damaging.

Accordingly, the best approach to security in the financial sector is to assume nobody is going to "do their job" besides the provider – and as such, the weight of security rests solely on the institution's shoulders. In other words, both in terms of compliance with regulations and in moral responsibility, security rests chiefly as a concern for the API provider.

Recent Exploits and Breaches

This isn't all just theoretical nonsense, either – failure to incorporate and internalize these issues has resulted in some major recent exploits and breaches from some rather large organizations.

An attack in 2014 was leveled against Chase bank, exposing the financial and personal information of 76 million households and 7 million small businesses. The breach was somewhat of a sore spot considering that Chase spends over $250 billion annually on cybersecurity, especially considering that the cost of the breach has been estimated at $1 Billion USD.

Not all breaches are just from financial institutions, either. Home

Depot was subject to a cyberattack in 2014 as well, with hackers using partner information from a vendor to expose a great amount of data from purchasers and businesses. The cost, before insurance reimbursements, is estimated at $80 Million USD. The Home Depot breach is a great example of how institutions can't necessarily trust others in the chain to provide adequate security. In this case, it was the vendor specifically that exposed the data internally.

Both of these cases are chump change compared to perhaps the biggest breach in recent years, the Equifax breach. In 2017, Equifax revealed that a cyberattack revealed credit card numbers and personally-identifying information of almost 143 million Americans – almost half of the entire country. The breach included names, Social Security Numbers, birth dates, addresses, driver's licenses, credit card numbers, and other financial information, leading to massive insurance claims and widespread fraud. As of September 2017, the breach has cost Equifax almost $4 Billion in lost market share, with insurance claims and punitive measures still being calculated.

Ultimately, these are just three of many breaches in recent years – and as technology dramatically advances, and the attacks leveled against institutions becomes more mature and complex, the number of breaches will likely increase.

Conclusion

Ultimately, API security is squarely on the shoulders of the API provider, and thereby the financial institution itself – failure to understand the vulnerabilities inherent in such business functions and the fundamental concerns of CIA within the greater ecosystem, can result in massive losses and exposures. If a financial institution internalizes these concerns, such functions can be provided securely and efficiently.

Is OAuth Enough for Financial-Grade API Security?

by **Art Anthony**

OAuth is a mature protocol sufficient for safe-guarding bank data in transit.

"If you think about where OAuth started, it was really about securing comments on blog posts, and now we're talking about enterprises, so it's a whole different class of security."

This is how Travis Spencer, CEO at the identity company Curity, opened his talk at our 2019 Austin API Summit. It's an astute summary of how many products (particularly in the tech scene) are tweaked or re-engineered beyond their original purpose.

As Spencer clarified in his talk, "when we say banking grade, we're not just talking about banks; we're talking about healthcare, government, and high security." In other words, financial grade security is relevant to any data-centric API.

It's often true that products struggle to adapt to tasks they were not originally designed for. In this post, we'll be looking at whether or not that's the case here. Knowing OAuth evolved from humble beginnings, is it competent for use in financial grade protection?

Can OAuth Make The Grade?

Early in his talk, Spencer provides a summary of some things you can do with your OAuth implementation to provide financial-grade security:

- Mutual TLS constrained tokens
- PKCE (Proof Key for Code Exchange)
- Consent
- Signed request/response objects
- Pairwise Pseudonymous Identifiers (PPIDs)
- Phantom tokens
- Strong authentication
- Prefix scopes
- Dynamic clients

There's no denying that's a pretty comprehensive list and suggests that OAuth is more than capable of holding its own when

it comes to security. In other words, OAuth shouldn't be seen as insufficient when it comes to securing data. In fact, Spencer outlines many ways in which it works very well for financial grade protection.

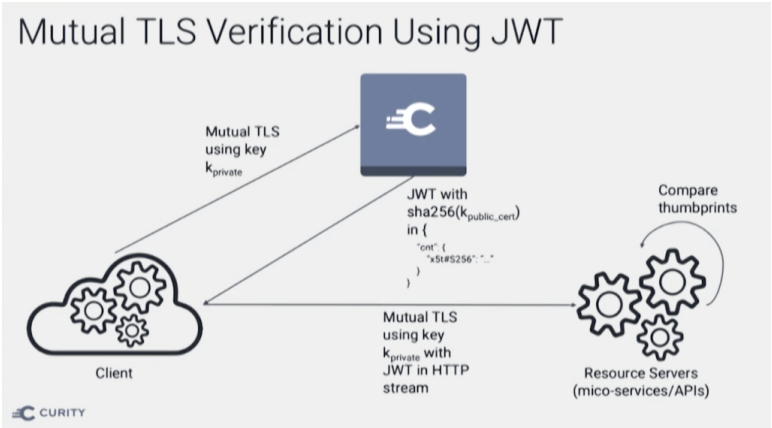
For example, Spencer mentions how PPIDs can be used to prevent collusion between separate apps while allowing interaction between, say, sites, and their associated mobile apps. He also describes how phantom tokens (vs., say, a JSON web token) allow for a smaller regulated space while preventing clients from accessing any Personally Identifiable Information (PII) and front-end clients from depending on the content of access tokens.

Some Tokens Are Unbearer-able

The way developers use OAuth isn't, however, always perfect. The two main vulnerabilities of OAuth, as seen by Spencer? Bearer tokens and the redirect. "Those two things are the primary attack vectors in OAuth."

Spencer likens the former to bearer bonds; "if you bear the bond, then…you can get a million dollars for each piece of paper and have a great life!" That's particularly problematic because, unlike bearer bonds, there's no way to lock tokens away in a physical safe.

"That's the Achilles heel because, if I can snatch someone else's token, I can become them." So what's a solution that can get around this? "The opposite of this is a holder of key token or proof of possession token, and that's what mutual TLS constraint tokens are."



From "Financial Grade APIs Using OAuth and OpenID Connect," by Travis Spencer, CEO at Curity.

In simple terms, the addition of a hash of a certificate to the token exchange process means that works something like using a cipher to decode a message; without the proper credentials, the token won't function properly when used for an API call.

Unfortunately, that isn't the end of the story…

Away With The PKCEs

When talking about redirects, Spencer refers to the vulnerability in callbacks that exists if…

- It's not confidential, i.e., not done over TLS
- There's no authentication done on the token endpoint
- The callback is not unique per client, or per client operator

"In cases where it's possible for a bad actor to intercept the credentials, they may be able to entirely sidestep the use of mutual TLS tokens outlined above." He also highlights that this is something particularly prevalent in mobile devices.

The solution? Proof Keys for Code Exchange (PKCE). “We can add a step 0 into this process…The legitimate application uses a proof key to prove that it is the actual application that started this flow, with the OAuth server refusing to finish it unless that proof is verified.”

You could think of this as a more robust version of Dennis Nedry’s “ah ah ah, you didn’t say the magic word” in Jurassic Park…

Signed, Sealed, Delivered

Spencer highlights the issue that, in most cases, OAuth flows go through the user agent. Malware installed in the browser, for example, on the user side, can undermine all of the measures taken above.

“How do we know that OAuth flows aren’t being tampered with in flight?” Spencer asks. “We sign the request from the client sent to the server and back…We can also sign the response that’s sent back to the client as well so they know nothing was tampered with on the way back.”

All of this signing and hashing is something that’s already proven its worth in recent years: read an article about data leaks by big companies, and you’ll notice, in cases where it’s been done, how eager they are to talk about hashing passwords.

Although hashing passwords, signing requests, etc. isn’t always 100% secure, there are cases, for example, in which weak hashing has been cracked. It is, at this point in time, about the best thing we can do.

What's Next For Financial Grade API Security?

It seems likely, for now at least, that the status quo will continue as the norm in the world of financial and financial grade APIs.

From PayPal and Stripe to Coinbase, and probably Facebook's Libra in the not too distant future, there are plenty of financial services utilizing APIs that are built around existing frameworks and services like OAuth.

If actual financial services are already relying on these, then there's very little incentive for other API developers to go beyond what already, in effect, constitutes financial grade API security…barring massive data leaks or the exposition of serious vulnerabilities.

In the meantime, we would expect to see OAuth continue to be one of those rare exceptions: a service built with something minor (securing blog comments) has effectively expanded to something much more robust than its original intent.

OpenID Connect: Overview of Financial-grade API (FAPI) Profile

by **Chris Wood**

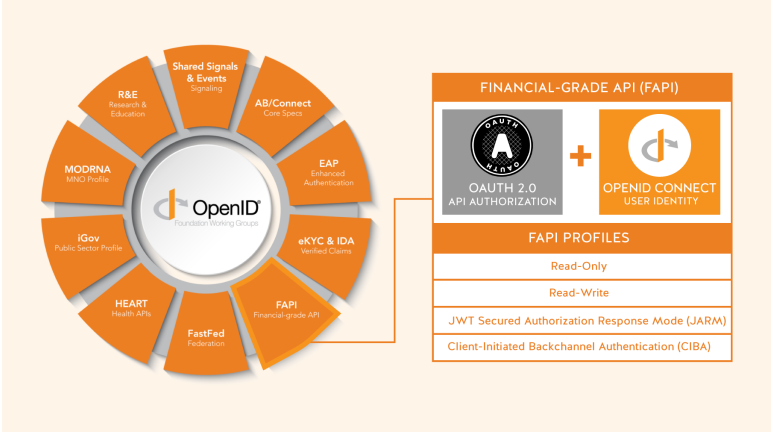
OpenID Connect, and the FAPI profile, will play a pivotal role in tracking identity to keep open banking architecture secure.

Open banking continues to be a subject of keen interest in financial services, reaching “buzz word” status over the last few years. We’ve covered the growth of the ecosystem on the blog several times before, but haven’t yet studied the standards emerging both within and across open banking markets. It comes as no surprise open banking initiatives have spurred several notable

advancements in standards. In this post, we'll look at a suite of standards that focus on improving API security: **The Financial-grade API (FAPI) Profile**.

What is FAPI?

FAPI is a working group of the OpenID Foundation, the body responsible for the development and maintenance of a family of protocol standards centered around OpenID Connect. FAPI was initiated in 2017 and sought to bring enhanced security to the new API standards being created to deliver PSD2 regulations across Europe, one of the key drivers in the buzz around open banking.



The scope, however, is not limited to PSD2 and the regulatory aspect of open banking. According to FAPI, its goal is to provide a "higher level of security than provided by standard OAuth or OpenID Connect." The standards also frequently refer to screen-scraping and the purpose of replacing this method with "an API model with structured data and a token model, such as OAuth."

To existing adopters of OAuth and OpenID Connect the benefits

might seem obvious, but for the uninitiated, here are some key points:

- One of the primary goals of OAuth is to ensure that the End User – a real human being – does not have to give their confidential credentials when sharing data with another app.
- By offering a delegated access model, OAuth ensures that the Client app – the "Relying Party" never sees those credentials but instead uses an access token.
- OpenID Connect adds to this model by providing proofs-of-authentication that the app both has delegated access to an API on behalf of the End User and proof that the real human being actually authenticated. As we covered in our introduction to OpenID Connect, not having proofs-of-authentication may be fine for Tweeting your best score on Wordscapes, but it's not so great when accepting a payment instruction from an app that may or may not be acting on behalf of an authenticated End User.

OAuth has its detractors, especially in how the redirection mechanism for authenticating the End User introduces friction. However, arguments for the security benefits are sound. FAPI, therefore, uses OAuth as its base profile for a good reason. FAPI currently incorporates four standards, all currently specified as Implementer's Drafts (in the parlance of standards an Implementer's Draft means "stable enough to implement, but not quite formally approved"). These are:

1. Read Only API Security Profile
2. Read & Write API Security Profile
3. JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)
4. Client-Initiated Backchannel Authentication (CIBA) Profile

Generally speaking, Parts 1 and 2 and JARM are concerned with hardening the implementation of OAuth 2.0 and OpenID Connect, while CIBA provides a new means of requesting the authentication of an End User. Each profile has implications for both the API Provider and Consumer, which we'll discuss below.

The Read-Only and Read-Write Profiles are effectively the "base" specifications for organizations wanting to attest to the level of security perceived as being required for operating financial services APIs. The entry criteria is OpenID Connect 1.0; both these standards use this as its core building block.

Adding Resilience: The Read-Only Profile

The Read-Only Profile is aimed at read-only access to an End User's account and introduces features that are carried into the other profiles. The goal is to specify behaviors that the actors will adopt when interacting to provide account-based data. These include:

- Enforcing strong levels of authentication for Clients, including Mutual Transport Layer Security (TLS) and a JSON Web Token (JWT) based mechanism for authenticating the Client at an application level.
- Enforcing levels of authentication for the End User, specifically at Level 2 from the ISO specified Entity authentication assurance framework. This is important for providing the Client application with certainty on how the End User authenticated.
- Implementing levels of cryptography strong enough to protect the parties involved, such as enforcing minimum key lengths of 2048 bits for RSA algorithms.
- Tightening up on other behaviors allowed inside the core OAuth 2.0 specification, such as requiring that all redirect_uri settings are pre-registered at the Authorization Server.

- Ensuring that short lifetimes are used for Access Tokens that are bearer tokens, given the number of different endpoints they may be exposed to in an account-access scenario.

At a very practical level, the Read-Only Profile also mandates the use of JSON in request/response payloads. It introduces several headers that allow consistent identification and tracking of events, with the subjects being fairly typical in financial services APIs:

- `x-fapi-interaction-id`: A UUID that provides a unique reference to be created for each request/response.
- `x-fapi-auth-date`: The date and time that the End User was last authenticated by the Client application.
- `x-fapi-customer-ip-address`: The IP address of the End User, if available.

The Read-Only Profile, therefore, provides standards for tightening the operating parameters to a degree deemed suitable for the sharing of account access.

Bullet-Proofing: The Read-Write Profile

The Read and Write Profile adds further constraints that are considered applicable for creating or updating account data through – for example – initiating a payment. Arguably the most important additional constraints introduced by the Read-Write Profile are:

- Enforcing the use of a signed-JWT (JWS) to encapsulate Request parameters when a Client application sends an End User to the Authorization Server to be authenticated

(the request parameter). We touched on this in our OpenID Connect introduction. This is important as it adds cryptographic proofs of the identity of the Client application, meaning the Authorization Server can verify it. This is an optional aspect of OpenID Connect that becomes mandatory with the Read and Write Profile. The standard also specifies a REST API that can be used to send parameters securely to the Authorization Server and then reference them using the request_uri parameter.

- Mandating that Client applications are proved to be a "holder of key." Key-based authentication mechanisms (for the Client) are permitted, through either Mutual TLS for OAuth or a JWS minted from a private key. This improves security as direct access to the private key must be sought by the miscreant in order to impersonate a valid Client and is enforced at both the Authorization Server and the Token Endpoint.
- Increasing the End User authentication assurance level to Level 3, enforcing multi-factor authentication which – in the context of PSD2 – means multiple things the user knows, possesses, or is part of them ("Inherence"), such as a biometric gesture.
- Restricting the cipher suites that can be used for Mutual TLS to reduce the risk of a compromised transport layer.

The Read-Only and Read and Write Profiles, therefore, provide important general additional protections to both the API Consumer and Provider. The JARM Profile looks to complement this by protecting one of the more vulnerable areas of OAuth, namely the exchange of authorization codes.

Improving OAuth 2.0: JWT-Secured Authorization Codes

The JWT Secured Authorization Response Mode (JARM) Profile does – largely speaking – what it says on the tin; it aims to secure authorization response parameters through the use of a signed and optionally encrypted JWT. In this mode, a secured authorization code is requested by the Client application, with the Authorization Server providing the response signed using a private key it owns. The Client application has access to the public certificate corresponding to this key so it can verify the signature (if the JWT is signed and encrypted, the Client will require access to both the signing and encryption public certificates).

The signing and optional encryption of the authorization code delivers additional protections for the Client against code replay style attacks in an OAuth-only implementation, similar to the benefits of the Request parameter in the Read/Write profile. Arguably it is the least pervasive standard within the suite, with Client-Initiated Backchannel Authentication breaking the newest ground.

Decoupling Authentication: Client-Initiated Backchannel Authentication

Client-Initiated Backchannel Authentication (CIBA) is the latest – and arguably most complex – of the FAPI profiles. It attempts to address two of the primary sticking points in the Open Banking ecosystem at the moment, namely:

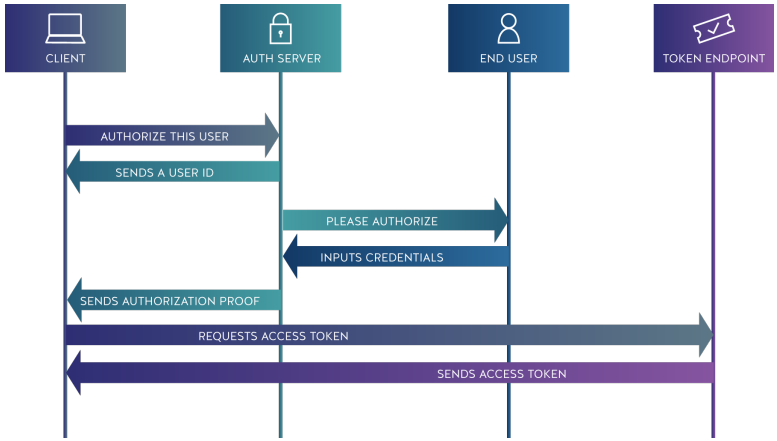
- To allow a user to authenticate and authorize an operation on a different device to the one they are using. For example, if an End User is using a desktop application but can only authenticate via a biometric stored on a mobile device, which is tied to the mobile operating system (rather than an open standard like Webauthn.)
- To allow for multiple real human beings to authorize the sharing of data or initiation of payment. This is important as the majority of corporate internet banking implementations adopt an "x-eyes" approach to authorizing sensitive operations, with associated workflows, queues, and notifications for the participants involved.

Both these needs are at odds with the majority of Open Banking implementations. Most processes expect to redirect a single user to perform the acts of authentication and authorization either within the browser or via a claimed URL to a mobile banking app. Some standards have loosely addressed some aspects of this; for example, the UK standards exposes a "MultiAuthorization" property in their Consent resource. However, the authorization workflow in this scenario is a facet of the API standards rather than the security protocol, which seems somewhat orthogonal and relies on the client polling an API to discover changes in the authorization state.

CIBA provides the flexibility to address these needs elegantly. Rather than expecting a synchronous or semi-synchronous response, CIBA anticipates an entirely asynchronous workflow:

- The Client application makes an Authentication request to the Authorization Server, using a hint to indicate the user they would like authenticated. The hint is provided through knowledge known to both parties – an email address, a shared identifier, or proofs-of-authentication previously offered, such as an ID Token.

- If the Authentication request is successful, the Authorization Server initiates backend workflows – either directly or through other components – to ask one or more End Users to authenticate themselves and authorize the request, through whatever means (mobile, desktop, etc.) are required by their internal application infrastructure.
- When all required End Users have completed authentication and authorization, the Authorization Server calls the Client application at a known webhook to indicate authorization is complete.
- The Client application can then visit the Token endpoint to retrieve a new token that allows them to access a protected resource.



The workflow for Client-Initiated Backchannel Authentication (CIBA), the latest FAPI profile.

There are permutations to this flow. For example, the Client application can poll the Authorization Server if no webhooks can be made available. But, the specification can offer Relying Parties a completely decoupled means to request user authentication and authorization. This massively increases usability and gives Clients many different options as to how they design their

user experience with redirection no longer being a fundamental requirement.

Final Thoughts

The prevalence of the FAPI is, of course, only as good as its adoption. There will be a lag between Implementers Drafts becoming available and their inclusion in supporting open source and vendor solutions. Hopefully, adoption will continue to increase as the suite of standards can only serve to provide more secure implementation in Open Banking ecosystems.

Developers cannot view security as a static notion that, once dealt with, can simply be marked "Done" and forgotten. Open Banking ecosystems will continue to evolve, especially as we move from solely banking-focussed APIs into the realms of Open Finance. As such, how we define and apply appropriate security protocols must also evolve. This is especially true when we consider the role of digital identity and its relationship with the OpenID Connect stack. The union of identity and "open everything" ecosystems means security profiles such as FAPI will only become more critical.

Case Study: Growing Internal API Consumption in Danske Bank

by **Thomas Bush**

Encourage adoption by improving the developer on-boarding process and evangelizing an internal API mindset.

An unfortunate truth of APIs is that they rarely sell themselves. Even internally, many organizations find themselves met with subpar adoption when they take a set-and-forget approach to rolling out new APIs. However, the good news is that there are plenty of ways to mitigate this, ensuring steady API consumption among both internal and external developers.

In this chapter, we'll look at three reasons an API development team at Danske Bank — the largest bank in Denmark — saw poor adoption of their internal APIs, and the steps they took to set growth on a much steeper gradient.

We based this chapter on a talk given by John Madsen, API Product Owner for Customer Information at Danske Bank, at the 2019 Platform Summit in Stockholm.

The Path Towards APIs

Before they adopted APIs, Danske Bank's technical architecture very much fit the stereotype of any bank: monolithic and slow-moving. With some systems that were over fifty years old — and continually expanding — it wasn't easy to improve, fix, or build anything. Not to mention, many of the systems were built on outdated technologies. John jokes that many issues could only be solved by *that* one specific person.

Naturally, this heavy IT architecture resulted in a poor speed of change. It could easily take half a year, a year, or even longer to develop new products and services. As a result, the organization decided enough was enough — and it was time to untangle the mess.

John briefly explains the process of how Danske Bank managed to silo their technical systems into API-ready subdomains. To start with, they divided their existing stack into three layers: a foundational layer (mainly consisting of a system of records), a process or function layer (providing meaningful data to consumers), and an experience layer (responsible for individual customer touchpoints such as apps).

Then, at the foundation layer, the bank broke down its system of records across object-oriented subdomains — for example,

customers. Each of these objects then became the basis for an API (or set of APIs).

Set-and-Forget Performance

Danske Bank was able to build an extensive suite of intuitive internal APIs after successfully abstracting away from the complexity of its mainframe. Having created these APIs, they expected consumption to grow, without all too much promotion quickly.

Unfortunately, the team was met with a slow, steady consumption graph — and not the hockey stick curve they had hoped for. John imagines that this is the case for many banks and financial institutions that took the same approach. But what did Danske Bank do wrong to be met with this sub-par internal API consumption?

Identifying Setbacks… and Addressing Them!

In hindsight, John identifies three particular setbacks in the consumption of their internal APIs: old habits, friction, and a lack of marketing. He explains each of these issues in-depth, also discussing some of the solutions used to tackle them.

Old Habits

Despite a suite of brand-new, sleek REST APIs, the team found that developers were unenthusiastic to change. This is because, elsewhere in the organization, separate teams were continuing to support alternative integration forms. The mentality of *if I can still use this point-to-point integration, why change?* was rife.

The Fixes

A definite solution to this reluctance for change was simply to deadline older integration methods. John found that developers were quick to react and adopt the new APIs by announcing that they would drop support for a specific point-to-point integration in three or six months' time. Of course, this harsh approach requires managers' buy-in, but the timeframes involved can always be used as a bargaining chip.

Perhaps a more accommodating fix, the team also pushed for changes in company culture. By encouraging others to build longer-term solutions with the APIs — and defending the importance of doing so with managers — the team was frequently able to nudge developers onto a more strategic course.

Friction

Developers who *did* want to adopt the new APIs were met with another issue: friction. To start with, there was friction in discovering the new APIs. Indeed, the team offered an API discovery tool, but John says that simply too much "stuff" had been pushed into there, leading to early frustration.

A subsequent cause of friction was security tokens. Developers were unsure how to create them, which should come as no surprise, as they knew that they were generated from the mainframe. This made the new REST APIs difficult to consume from separate platforms.

The Fixes

The natural solution to this friction was to improve the developer journey. One particular pain point, John says, was everything that led up to consuming the APIs: discovery, security tokens, and more. By focusing on these areas and making the

onboarding process easier, developers were more likely to pursue API-based solutions. One area Danske Bank did not struggle with was API-specific documentation, but others might.

Marketing

It should come as no surprise that the team — a group of techies "down in the dungeon" — completely overlooked the importance of marketing. Battling a tight schedule, they did almost no promotion for their new APIs. Naturally, this meant fewer developers knew the APIs existed in the first place.

The Fixes

The fix for this lack of marketing was simply: the team had to get out and sell their new APIs. This meant knocking on plenty of doors — development directors', engineers', architects,' CIOs,' and even business folks' — and sharing their latest developments. John found that with each meeting, the initiative gained momentum.

The Results

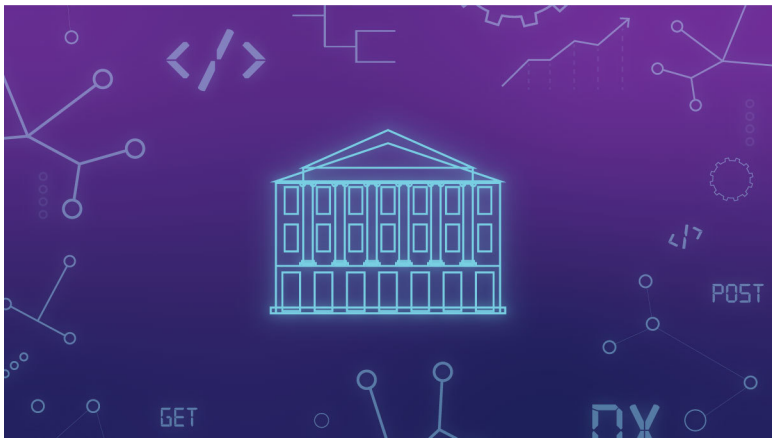
So, what were the results of Danske Bank's many maneuvers to improve API consumption? Fortunately, growth quickly sped up. While it wasn't the exponential growth John had hoped for, he could sense that the new APIs were gaining traction through the organization. Tell-tale signs of this new-found growth, he reminisces, were when developers began to call him — saying things like "I heard from *him/her* that you have some new APIs — or when business folk began to invite him to present the new APIs to others.

Summary

Even the best APIs can suffer from slow adoption. In this post, we looked at three reasons why Danske Bank's internal APIs were slow to gain traction. According to John Madsen, a Product Owner who fought on the frontline, old habits, onboarding friction, and a lack of promotion are major causes of poor API growth. With any luck, the solutions presented herein should help to minimize these issues, ensuring sure and steady adoption.

It Started With PSD2 and Personal Data

by **Chris Wood**



EU regulation unlocked consumer data, stimulating an open banking global response.

On the European FinTech stage, initiatives and regulations have fostered tremendous innovation in the banking sector of the API economy. The most significant of these is the Payment Services Directive 2 (PSD2), a regulation that was applied across the European Union and has resulted in a huge increase in the number of APIs for banking products.

Making banks programmable is significantly altering the engagement model for accessing a consumer's account. What is

less clear is how this may affect the consumer themselves, including their level of access to the data (that *in theory* they own), and their ability to use their data in any way they see fit. In this post we'll look at how PSD2 and the growth of APIs for banking are affectin affect personal data ownership.

The Status Quo

Most banks lock customer data away in internal systems with very limited access, restricted to tightly controlled channels. This state of banking data accessibility is widely viewed by industry commentators and even banks themselves as nothing short of a travesty, epitomizing the hegemony banking giants have held for a long period of time. According to Andres Wolberg-Stok, global head of emerging platforms and services at Citi, **APIs** present an opportunity to *"break a few windows to let free air and light in."*

While disrupters and innovators (especially challenger banks) will be prone to hyperbola, the common consensus is that banks should open up to provide APIs for a number of compelling reasons:

- **To enable consumer choice**: There are many consumers who want to be selective about banking products without having to choose a single banking provider, instead simply picking from a marketplace. APIs provide a mechanism to enable such selectivity of product. Supporting this notion is the great deal of coverage to the "narrowing" of banking (where the portfolio of bank activities is restricted), as well as the lack of trust millennials place in traditional banking establishments;
- **To unlock customer data**: Internet banking has become the key customer engagement point during the last two

decades, but the digitization of the customer experience has made it clear that the vast majority of banks consider this data to be a vault with only one entrance and two sets of keys — theirs and the customers. This is juxtaposed with the wants and needs of customers, who are keen to exploit what is essentially their data in any number of different ways: Personal Financial Management (PFM), credit checks, digital notarization, and more. However, banks simply don't provide facilities for third-parties to work at the delegated authority of the customer, forcing many third party solutions to use the customer's internet banking login credentials and web scraping to function (examples include Mint and Xero). While this provides customers with useful solutions, by sharing their credentials with a third party they may actually break the terms and conditions of their bank's internet banking platforms;

- **To unlock themselves**: Anyone who has worked in the IT department of a large bank will be familiar with the architecture and approach that typifies them — extreme risk adversity with large amounts of governance on top of legacy systems and monolithic applications. An API-based architecture, built incrementally with many small steps could help unlock and decouple these architectures, making them more accessible and providing an environment to foster innovation.

The core theme is that banks could be doing much more to open customer data up to new use cases and business models. Naturally there are disrupters and innovators who are having some success trumpeting down the walls of Jericho, and the Open Bank Project is clearly the poster boy for the open banking movement. Another example is Figo in Germany that delivers a single API that integrates with the German FinTS/HBCI banking network. These initiatives show what can be achieved without a standardized API network, but a lot of hard yards are involved

in creating the solutions with a myriad of different integrations across the banking ecosystem. Such efforts could be significantly reduced if each bank offered a standardized suite of APIs.

Regulatory Impact

The rationale for standard banking APIs is clear, and there has always been the potential for a large bank to "break cover" and offer a suite of APIs with access to customer data in advance of its rivals. However, in the absence of this early mover encouraging other banks to offer APIs by way of market competition, regulatory forces are now likely to coerce many into taking action.

PSD2 requires banks to allow third parties to access a given customer's data, where that third-party is acting as a data consumer or a delegated authority: The EU describes these as "third party providers (TPPs) [who] offer specific payment solutions or services to customers". As this could include making a payment on a customer's behalf, PSD2 grants third-parties considerable power. There is a great deal of debate how this might be implemented from a technical perspective, but an obvious solution for anyone familiar with the API economy is the use of APIs to facilitate access. Moreover, APIs could be coupled with a rich framework such as JSON Web Tokens under the guise of OpenID Connect to provide strong authentication and non-repudiation.

It is an oversimplification to say using APIs and OpenID Connect would *immediately* provide the framework for implementing PSD2, and some of the logical architecture has already been framed: it includes the introduction of Account Information Service Providers that will provide a single view across multiple customer accounts. However, with the right governance framework these technologies could provide the bedrock of a new open banking landscape, supporting a wide range of new entrants to the market.

The Open Banking (and Data) Landscape

The open banking landscape will allow customers to unlock their data and give delegated authority to their payment instruments, empowering them to share it with whomever they saw fit via an API. The value in this unlocked data isn't restricted to banking — clearly other solution providers see much value in it. For example, frameworks like Figo will become much more commonplace with initiatives like PSD2 and possibly draw in data sources that aren't financial in nature, again via APIs.

Customers will have a single window on all their data, financial or not: while solutions already exist in this space, such as Trunomi and Mecco, who offer federated personal data stores and mechanisms to produce different views of an individuals' personal data, the integration effort is fraught with difficulty. With standardized APIs there is an opportunity to view, understand, and control a unified view of our data.

The consolidation of our digital persona into a single construct is both a risk and an opportunity for individuals: a risk, because without the right governance we offer ourselves up to solicitation on a huge scale, but an opportunity because we can enable the genuine usage of our data ***for our own benefit***, truly empowering consumers to make insightful decisions. With such insight at our disposal concepts like the Secco Aura could become a reality for more people than just Secco bank customers, with a **broadcast of interests** possible across many different service providers, financial or otherwise. Such possibilities are so relevant in the data sphere that large organizations like Visa Europe are researching them, with their innovation lab currently running their "Me2B" theme. Howard Elsey, the innovation partner running this theme describes such a personal data network as:

> "the connective tissue and nervous system of the data economy. This has ramifications into all current areas of technical innovation, from big data to blockchain, identity, and IOT but is so interesting … because of the impact that it will have in overcoming something that computerization has managed lose — the personal nature and trust in the relationship between business and the individual and the disrespect of third party companies that exploit the value in an individual's data without returning value to the individual."

In architectural terms, APIs underpin this framework as both the personal data source, federated into what Mecco calls "the API of me" and the service delivery point. However, there is a new paradigm, the carrier for the broadcast of interests which, at face value looks and smells like massively distributed publish-subscribe network: the NATS technology for the personal data network. It is to this network that both consumers and providers broadcast their interests with APIs providing the delivery mechanism for data or services; requests can be serviced either directly from the consumer's data stores or via aggregators that pull together and consolidate the data in a myriad of different views. Some initiatives already exist that could grow into the backbone of this system, such as the Open Mustard Seed project, but however it comes about, the "network of me" will make the personal API economy truly revolutionary.

Final Thoughts

The opening of banking through APIs, whether through competitive pressure or regulatory enforcement, has massive implications across the data ecosystem:

- **for consumers** it empowers their ability to access products and their data;
- **for solutions and services providers** it allows them to engage with customers in a much more seamless fashion;
- **for IoT** it provides the network for devices to make autonomous actions based on an individual's preferences, wants, and needs.

It will take work to come to fruition but these are exciting times for us all as consumers. It will be fascinating to see how banks receive the PSD2 regulation, and how this personal data network develops into the future.

Nordic APIs Resources

More eBooks by Nordic APIs:

Visit our eBook page to download these eBooks for free!

GraphQL or Bust: Everything GraphQL! Explore the benefits of GraphQL, differences between it and REST, nuanced security concerns, extending GraphQL with additional tooling, GraphQL-specific consoles, and more.

How to Successfully Market an API: The bible for project managers, technical evangelists, or marketing aficionados in the process of promoting an API program.

The API Economy: APIs have given birth to a variety of unprecedented services. Learn how to get the most out of this new economy.

API Driven-DevOps: One important development in recent years has been the emergence of DevOps â€" a discipline at the crossroads between application development and system administration.

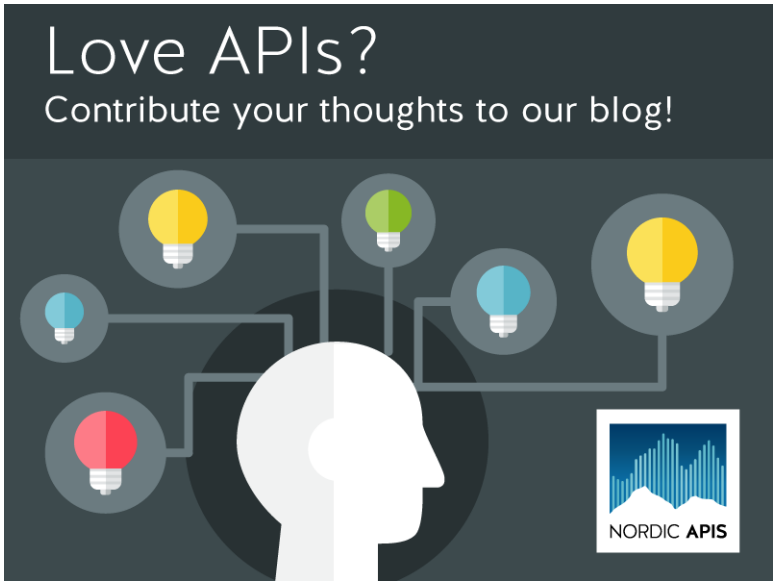
Securing the API Stronghold: The most comprehensive freely available deep dive into the core tenants of modern web API security, identity control, and access management.

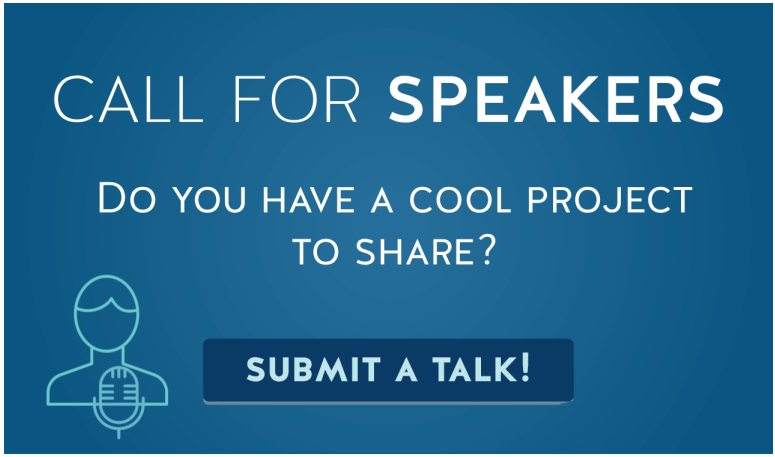
Developing The API Mindset: Distinguishes Public, Private, and Partner API business strategies with use cases from Nordic APIs events.

Visit our Youtube Page for Deep Dives

For more, watch full videos of high impact API-related talks on our Youtube Channel.

Create With Us




CALL FOR SPEAKERS
DO YOU HAVE A COOL PROJECT TO SHARE?
SUBMIT A TALK!

Nordic APIs is an independent blog and this publication has not been authorized, sponsored, or otherwise approved by any company mentioned in it. All trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

Nordic APIs AB ©

Facebook | Twitter | Linkedin | YouTube

Blog | Home | Newsletter