

# Mastering Open Banking: How the 'Masters in openness' create value

[innopay.com](http://innopay.com)  
[openbankingmonitor.com](http://openbankingmonitor.com)

WHITEPAPER 2018

# Table of Contents

Management summary	3
1. Introduction: INNOPAY Open Banking Monitor shows that Open Banking is gaining traction	4
2. API Catalogue	7
3. API Documentation	11
4. Developer Usability	12
5. Developer Community	14
6. Five actions to execute on your Open Banking strategy	16

# Management summary

**Senior bank executives are starting to understand that Open Banking will have key implications on their future competitive positioning and related digital transformation activities. We define Open Banking as a business approach in which value creation results from sharing, providing and leveraging access to bank resources through application programming interfaces (APIs). This way data, processes and other business capabilities of banks are made available to an ecosystem of (selected) third parties (e.g. fintechs, technology vendors, corporate customers).**

Open Banking is set to transform digital experiences through compelling value propositions developed by third parties leveraging access to bank resources, ultimately adding value and putting the customer more in control. Banks that are able to put the required capabilities in place to effectively and seamlessly engage with third parties will benefit from an early mover advantage. In this article we assess **four core API Developer Portal capabilities** of 50+ banks and define **five strategic actions** that banks can undertake to execute their Open Banking strategy. In the capability assessment we focus on specific aspects of the Open Banking strategy, that is, the functional richness of APIs offered (i.e. Functional Scope) and the extent to which third parties are able to interact with these APIs in a seamless manner (i.e. Developer Experience). The bank's API Developer Portal is where these aspects come together.

## **Four core API Developer Portal capabilities**

Open Banking is definitely not a business model fit for all types of banks. The extent to which an Open Banking play will be successful depends on many different aspects that banks need to get right. This includes its Open Banking strategy, taking into account existing product offering, competitive positioning and size of customer base, and the bank's ability to execute on that strategy.

Many banks are taking action to engage and support external developers through an API Developer Portal. However, the level of maturity differs considerably across banks, as we assess in the **INNOPY Open Banking Monitor (OBM)**. **Banks differ on four core capabilities: API Catalogue, API Documentation, Developer Usability and Developer Community.** While the majority of banks is still mainly working on 'getting the basics right' of their Developer Portal, we also observe that others are gradually expanding the functional scope of their API portfolio.

## **Five strategic actions to execute on your Open Banking strategy**

With many banks across the globe establishing the basics of their API Developer Portal, there is a strong incentive towards differentiation in the emerging Open Banking landscape. To ensure banks are prepared for this new landscape, we have defined five strategic actions: 1) Learn from global API best

practices across industries, 2) Develop API rationale and strategy for your business to create new avenues for revenue growth, 3) Identify and prioritise the value that can be captured with APIs, 4) Manage API monetisation actively by determining if, what, how and who to charge in a transparent manner, and 5) Drive usage and adoption of your APIs to accelerate network effects and gain scale.

Open Banking should be approached as a business strategy and model in its own right, requiring an alternative way of thinking and working in product development. Combined with powerful execution capabilities and a successful and scaled partnership ecosystem banks will be able to future-proof their competitive position in the Open Banking era.

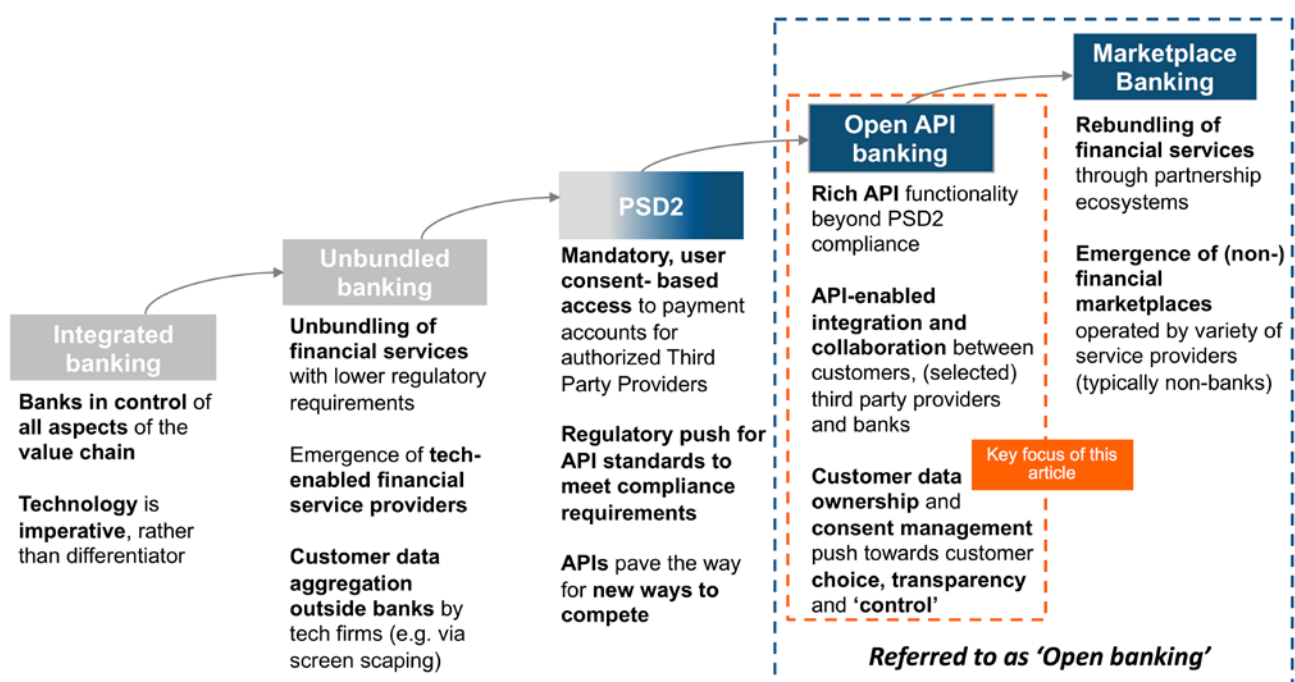




# 1 Introduction: INNOPAY Open Banking Monitor shows that Open Banking is gaining traction

Banks across the globe are starting to understand that Open Banking will have key implications on their future competitive positioning and related digital transformation activities. Open Banking has reached the boardroom agenda and strategic investments are being made or at least considered.

The evolutionary journey towards Open Banking is driven by ongoing digitisation of financial services, as depicted in figure 1.



Source: INNOPAY analysis

Figure 1: Evolutionary journey towards Open Banking

In this article we define and focus on Open Banking as a business approach in which value creation results from sharing, providing and leveraging access to bank resources. This in contrast to just owning these resources and being closed. Data, processes and other business capabilities of banks are made available to an ecosystem of (selected) 3rd parties (e.g. fintechs, technology vendors, corporate customers) through application programming interfaces (APIs).

Open Banking is set to transform digital experiences by enabling third parties to develop compelling value propositions while leveraging access to bank resources and putting the customer more in control. As the benefits materialise at scale we will witness an accelerated shift

towards Open Banking platforms. These platforms enable banks to effectively and securely interact and co-create with an ecosystem of service providers through APIs. Both banks and these service providers can create benefits for their mutual customers, strengthen their competitive position in the API economy and potentially establish new avenues for revenue growth. For banks this could offset competitive pressure resulting from the increasing openness in payments and banking introduced by PSD2. Indeed, in Europe, we already observe that banks are starting to experiment with offering APIs beyond the (perceived) mandatory functionality under PSD2.

### Open Banking is not fit for all banks

Open Banking is definitely not a business model fit for all types of banks. The extent to which an Open Banking play will be successful depends on many different aspects that banks need to get right. This includes its Open Banking strategy, taking into account existing product portfolio, competitive positioning and size of customer base, and the bank's ability to execute on that strategy. In this article, we focus on specific aspects of the strategy, that is, the functional richness of APIs offered and the extent to which third parties are able to interact with these APIs in a seamless manner. The level of maturity differs considerably across banks on these aspects, as we have assessed in a prior release of the [INNOPAY Open Banking Monitor \(OBM\) "Who are the Masters in Openness?"](#). Note that the level of openness of a bank is relative to the bank's product portfolio, that is, larger banks tend to have a more comprehensive product catalogue. As this study measures absolute openness, these elements (i.e. reach and product portfolio) should be kept in mind.

### Strong API Developer Portal capabilities are key to winning in Open Banking

A winning strategy is quintessential for banks to effectively participate in the Open Banking platform game. While there are little precedents in banking, banks can learn from open business models that have revolutionised other industries. Indeed, a selected number of progressive banks are starting to engage by publicly launching their own Developer Portals including APIs and sandbox environments. These capabilities allow banks to offer secure and controlled access to third parties to interact and use the bank's functionality and customer's data to create next generation financial services. Banks that are able to put the required capabilities in place to effectively and seamlessly **engage with third parties**, and facilitate an Open Banking ecosystem through its platform, will benefit from an early mover advantage. This will in turn strengthen the bank's API offering and a supportive ecosystem of third parties that drive customer value creation. Many banks are taking action to engage and support external developers through a comprehensive Developer Portal to facilitate effective interaction.



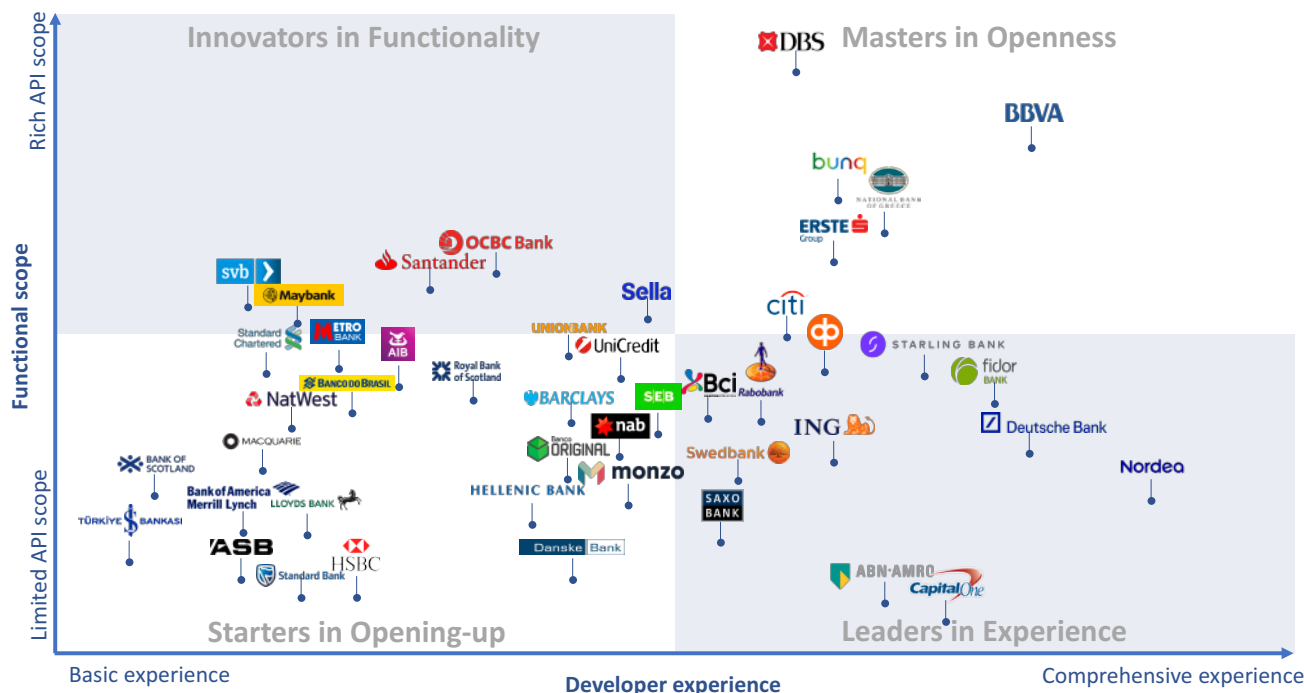
Open Banking is definitely not a business model fit for all types of banks

## INNOPY Open Banking Monitor assesses API Developer Portal Capabilities

The initial OBM assessment conducted early March 2018 included Developer Portals across the globe and triggered many positive reactions from various banks and financial institutions worldwide. The OBM has proven to be an accessible and intuitive tool providing a snapshot of the current state of play regarding API Developer Portals and insight in a bank's relative position. In this initial release, we have seen that the majority of banks mainly worked on

'getting the basics right' of their Developer Portal, rather than the Functional Scope of their API portfolio.

In this second release, 'OBM 2.0', INNOPY's assessment has been enriched with new banks, new API functionality and new features that drive the Developer Experience and nurture the use of APIs to accelerate innovation in financial services. Figure 2 below depicts the updated benchmark results.



INNOPY Open Banking Monitor (OBM) 2.0 – Developer Portal benchmark (update Sept 2018)

Note: If your bank is not in the OBM yet, go to [openbankingmonitor.com](http://openbankingmonitor.com) and let us know!

INNOPY

Figure 2: INNOPY Open Banking Monitor 2.0 - update September 2018



OBM 2.0 evaluates the relative position of banks across four core Open Banking platform capabilities, as depicted in figure 3 below. The state of play and best practices across these core capabilities will be further elaborated in the remainder of this paper.

Figure 3: INNOPY Developer Portal Capability Model (Grey coloured capabilities not assessed in this OBM release)

### Key messages on API Catalogue:

- Becoming a Master in Openness is about relative openness rather than absolute openness, meaning challenger banks and incumbent banks can only open up the resources they have. Therefore, the functionality the API enables is a better indicator of openness, rather than the number of APIs
- Current Open banking approach will lead to fragmented API Catalogues roadmap; guidelines in API design could improve the growth of the Open Banking ecosystem, increasing scalability and cohesion between banks and 3rd parties
- The design of API functionality varies with the granularity offered and can range from “do it yourself” to “ready to assemble” functionality

The API Catalogue is referring to all the products banks are exposing through APIs. In Europe, many banks are responding to the PSD2 compliance challenge by offering APIs enabling the mandatory services (i.e. Payment initiation, Account information and Confirmation of funds availability). We already observe some leading banks that are extending their offering by exposing more API functionalities to serve 3rd parties and corporate customers directly. Banks outside Europe are also starting to open up, seeking to expose functionality and data through APIs to add value to their Open Banking ecosystem.

### Current Open banking approach will lead to fragmented API Catalogues

API functionality can be designed and built in various ways, and the decision to expose certain APIs is determined by the bank's strategy. There seems to be no general structure on how the various banks define and set-up the Functional Scope of their API offering (i.e. API Roadmap). Common API standards for the Functional Scope could, however, promote **growth of the Open Banking ecosystem**. Currently, both the content (what is actually offered) and the delivery (the way in which it is offered) differs to a large extent per bank, increasing the risk of fragmentation. In Europe, however, we do see some early signs of convergence with numerous banks offering PSD2 inspired functionality (e.g. account information services and payment initiation services) according to the **NextGenPSD2 API framework** of the Berlin group. While this framework provides for a good start, NextGenPSD2 is an API framework and not a single standard such as **Open Banking UK**. Put simply, the API framework provides a toolkit for banks to build their own PSD2 API standard, allowing for various degree of freedom on certain API design aspects. Creating common API standards in an early stage for a community of (small) banks in a particular region could contribute to a faster growing ecosystem and increased cross-fertilisation.

Figure 4 below shows the division of the number of measured API functionalities per category currently observed in the Open Banking landscape. Just over 50 banks with publicly available Developer Portals (in the English language) were examined, spanning different types of banks (i.e. majority incumbent and one fifth challenger banks) and types of

business (i.e. retail and wholesale) to create an insightful overview of the current state of play in Open Banking. To define API functionality, we compared corresponding APIs of different banks with the possibilities they offer. One API can hold one or more functionalities, next paragraph will elaborate on this.

The functionality the API enables is a better indicator of openness, rather than the number of APIs

On the right side the categories are explained and the top 3 most common API functionalities per category are shown. This top 3 provides insight on which functionalities are most commonly offered across banks. Most offered functionalities are related to reading information (e.g. GET Account Balance) from the user's account instead of writing (e.g. POST SEPA Credit Transfer). As banks grow accustomed to Open Banking, more write functionalities are expected to emerge in parallel.

There is also a range of miscellaneous API functionalities that is offered by a single or very few banks, which are not taken into account in figure 4. These API functionalities vary greatly and are still in an emerging state. If these offerings mature they can be reported in a future OBM release.



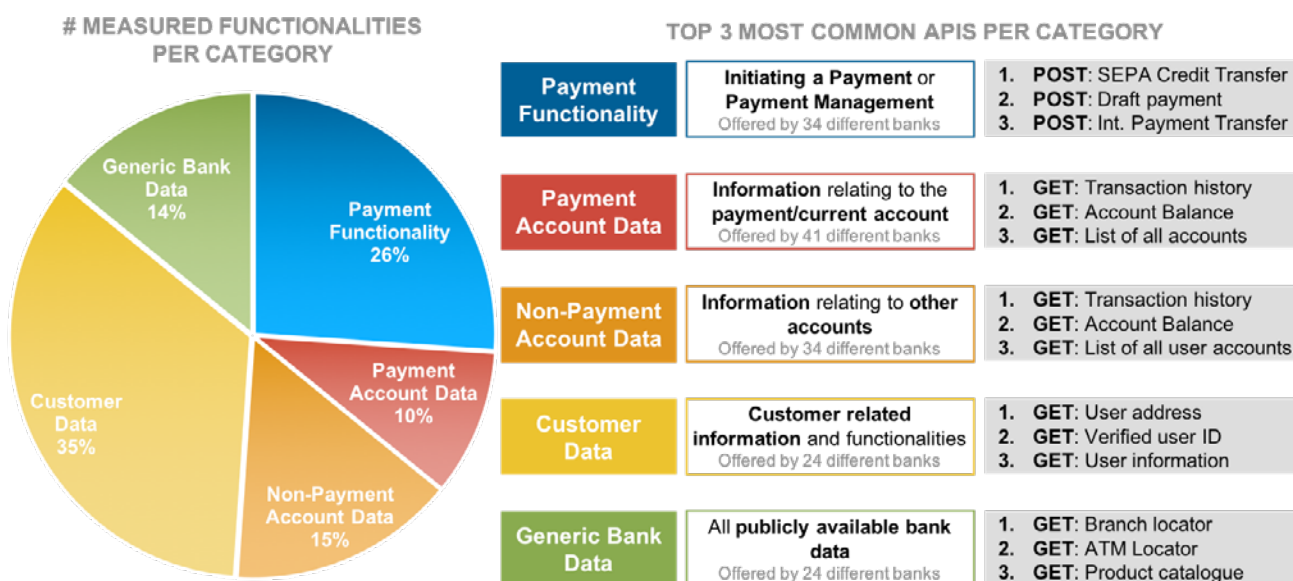


Figure 4: Number of measured API functionalities per category including top 3 APIs

### API functionality is a better indicator for openness, than the number of APIs

The various banks with a Developer Portal are often ranked by the number of APIs they are exposing. In our research, we are using the number of API functionalities instead, because due to the fact that an API can have one or more functionalities, comparing number of APIs would not give a clear representation of what the bank actually offers. Our analysis shows that a particular 'Bank A' can have a single comprehensive API for transaction history incorporating various functionalities, where 'Bank B' offers a single API for transaction history of payment accounts, another API for card payment transactions, another API for sent transactions and a separate API for incoming transactions. While both banks are offering the same functionality, Bank B would (unfairly) score higher when number of APIs would be considered a leading indicator for the extent of openness.

### Becoming a Master in Openness is about relative openness, not absolute openness

Challenger banks and incumbent banks can only open up the resources they have. Being a true Master in Openness is more about relative openness (which % of functionality does the respective bank open up), rather than absolute openness (how many functionalities does the respective bank open up). The Open Banking Monitor measures absolute openness, therefore the results of challenger banks need to be interpreted with caution especially when comparing these to incumbent banks.

Where, in our previous release of the OBM, we observed many challenger banks leading the ranks on Functional Scope

(i.e. Bunq, Starling and Fidor), we observe that incumbents are catching up. The top performers on API Catalogue, i.e. Functional Scope, in this release are large banks with a clear focus on Open Banking, such as DBS, BBVA and ERSTE Group.

Being a true Master in Openness is more about relative openness rather than absolute openness

BBVA offers a very comprehensive account functionality spanning multiple account types (e.g. savings, checking etc.). DBS offers five different ways of payment/transfer methods (including instant payment) and extensive payment management options (e.g. merchant checkout, corporate bill payments and refund/chargeback management options).





Figure 5: Number of API functionalities per region grouped in categories

#### Different regions show a preference for certain categories

Figure 5 above shows an overview of the various API functionalities that are available across certain regions. The figure shows that based on our research Europe is leading the Open Banking development in general and embracing Open Banking even beyond the mandatory PSD2 APIs. It seems that Oceania is experimenting with Open Banking by offering APIs like “Branch locator” and “Product catalogue”. Singapore seems to show high numbers in the category “Generic Bank Data”, although since the number of participating banks in Singapore is rather low, it is hard to make any reasonable statements on the Asia region. Overall, Oceania and US seem to be lagging behind in the variation of API functionalities in comparison to the offering of banks in other regions.

Figure 6 below shows a more detailed view of the number of API functionalities per category offered by the top 10 banks in the Open Banking landscape. Banks in Singapore are embracing Open Banking and offering the most functionality. As stated above and emphasised by the marginally presence of only two challenger banks in the top 10; challengers are lacking in Functional Scope, presumably due to their minimal product offering. There seems to be great variation in the offering of functionality. Some offer fine grained functionalities (i.e. Bunq), some offer full serviced products (i.e. BBVA), this will be further elaborated in the next paragraph elaborating on API design.

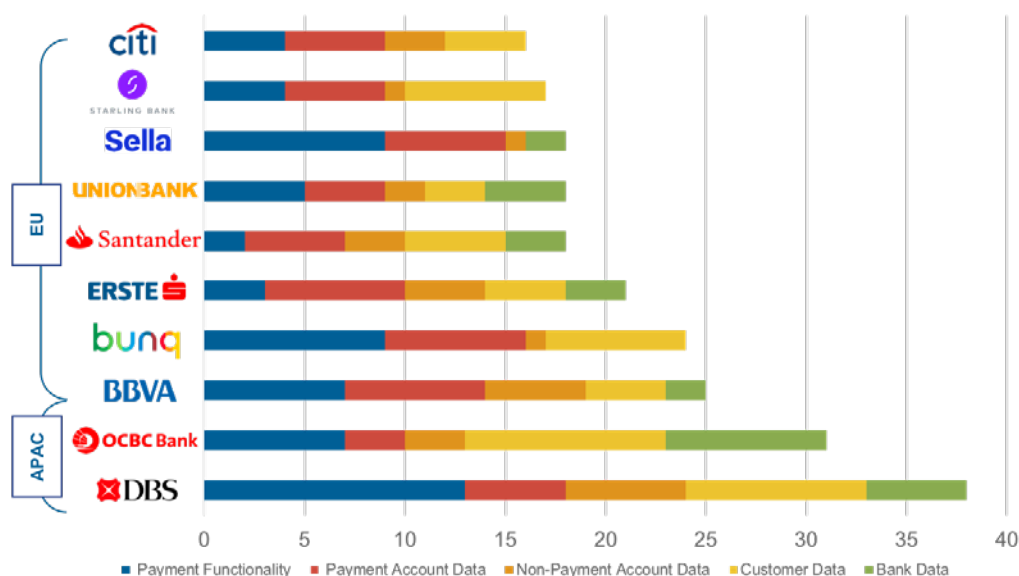




Figure 6: Number of API functionalities grouped per category of the top 10 banks

## Design of APIs varies with the granularity offered

We observe a great variety and granularity in API functionality offered by banks, as shown in Table 1 below. The table outlines the approach banks can have on building

their API offering. These approaches range from “do it yourself” to “ready to assemble” APIs on the other end of the spectrum with potentially many hybrid forms in between.

	Do it yourself	Ready-to-assemble
		
<b>Description</b>	<ul style="list-style-type: none"> <li>Start without a pre-developed plan</li> <li>Everything needs to be designed, sorted and built</li> <li>Starting with pieces of wood, a saw and pipes will be the equivalent of the stripped functionality like schedule and capture payment</li> <li>Mostly single functionality per API</li> </ul>	<ul style="list-style-type: none"> <li>Build according to the banks plan, using building blocks</li> <li>There is a structured plan for every single cabinet or drawer, however, the total kitchen needs to be designed</li> <li>Building kit reduces the possibilities compared to DIY however, less self-inventing will be needed</li> <li>Most APIs hold multiple functionalities</li> </ul>
<b>API Consumer pros</b>	<ul style="list-style-type: none"> <li>Increased flexibilities using combinations (parts of) of APIs</li> <li>Efficient APIs can be built, by incorporating only the necessary single functionality</li> </ul>	<ul style="list-style-type: none"> <li>More possibilities with less creativity</li> <li>Ready to use off the shelf APIs</li> </ul>
<b>API Consumer cons</b>	<ul style="list-style-type: none"> <li>Insights in bank processes is required to build APIs (e.g. the steps in the payment process)</li> <li>More work to create apps, since several functionalities need to be combined</li> </ul>	<ul style="list-style-type: none"> <li>Very dependent of the design choices made by the bank</li> <li>Reduced performance, due to the fact that a single functionality cannot be called separately</li> </ul>
<b>Example</b>	<ul style="list-style-type: none"> <li>Schedule-payment from Bunq</li> </ul>	<ul style="list-style-type: none"> <li>PayLah from DBS</li> </ul>

**Table 1:** API design approaches range ‘do it yourself’ to ‘ready to assemble’

For banks, it is relevant to determine who the target group is that will be consuming the API and for which purpose. The API fit gives a representation of the type of bank and the desired granularity of the API. Assessing the desired granularity of the functionality will allow banks to conclude which design and structure will be most suitable for their APIs.

## 3 API Documentation

### Key messages on API Documentation:

- Clear and unambiguous API Documentation is essential to enable API consumers to build efficient connections and facilitate self-service
- Banks differ in quality of API Documentation offered, with main difference in accuracy and comprehensiveness
- Good API Documentation will support the marketing of APIs

The core capability “API Documentation” refers to the quality, comprehensiveness and (logical) structure of the documentation of the complete API offering of a particular bank. API Documentation is needed for developers to understand the structure of the API, which data fields are used and, which parameters can be used to use an API functionality.

### API Documentation shows considerable difference in structure and quality

As with the previous release of the OBM, there are considerable differences between the way documentation is offered and functionality is being added for developers to get acquainted with the bank’s APIs more quickly. Although

it is obvious that all APIs and their functionalities need to be properly documented in order to drive usage, banks seem to be struggling to get this right. The top 3 banks in API Documentation, BBVA, Nordea and ING, all have elaborate explanations of all attributes used in the APIs. Version history of the APIs seems to be missing for some banks, but this could be explained by the fact that their Developer Portals are only just recently launched.

Figure 7 below shows a comparison of two different Developer Portals offering API Documentation for a ‘GET Transaction history’ API. This example illustrates opposite ends of a spectrum of how API documentation is structured by banks.

Bank A				
Field	Object type	Description	Example	Optional
id	string	Identifies a reference to a corresponding transaction	"61331236"	optional
account_id	string	Internal account id (not account number)	"91203682"	optional
transaction_type	string	Type of the transaction	"creditcard_payout"	optional
remote_bic	string	BIC / Swift code of bank	"DEUTNL2N"	required
remote_iban	string	International Bank Account Number (IBAN)	"DE89 3704 0044 0532 0130"	required
amount	integer	The amount in account currency, in minor units (e.g., 1 EUR is 100)	12000	optional
currency	string	Currency of Account or Amount. ISO 4217 Alpha-3 - 3 (e.g., EUR)	"EUR"	optional
is_verified	boolean	Indicates whether the transaction is verified or not	Yes	optional
Bank B				
<pre>{   "id": "string"   "transactionType": "string"   "bicCode": "string"   "ibanCode": "string"   "amount": 0   "currency": "string" }</pre>				

Figure 7: Comparison of API Documentation of two banks for API ‘GET: Transaction history’

Main differences between the API documentation of bank A and bank B is the overall structure and the description of each field. Bank A has clearly defined which fields are returned, by offering a comprehensive explanation of each parameter; what its object type is, the description of its contents, an example value and whether the field is required or returned optionally. Bank B gives little to no description of the returned values, leaving it up to the developer to guess what values he is actually receiving. It can be stated that Bank A helps developers to get started more quickly, since the returned attributes are clearly documented and therefore the developer knows how to use it and what to expect.



## 4 Developer Usability

### Key messages on API Documentation:

- Clear and unambiguous API Documentation is essential to enable API consumers to build efficient connections and facilitate self-service
- Banks differ in quality of API Documentation offered, with main difference in accuracy and comprehensiveness
- Good API Documentation will support the marketing of APIs

Developer Usability refers to the tools, guides and experience provided by the bank to the developer to interact with the available APIs. The usability indicates the ease of use of the portal in general, how effective and efficient developers can find their way around the portal. Developer Usability starts with the onboarding of the developer, the GUI that is presented, the toolset that is being offered and the ability for developers to manage their apps. The range of usability varies greatly where some Developer Portals offer guidance or help by performing any action (e.g. automatic authentication in the sandbox), others introduce new ways to test API calls with Telegram (i.e. BBVA), where other (starting) open banks miss out on these opportunities to interact with developers.

As stated earlier, Open Banking is in an emerging state, this is also being confirmed by the updated benchmark. More banks have launched their Developer Portal but more importantly, several banks have updated their Developer Portal looking for better ways to service and interact with developers and increase the overall Developer Experience.

### Various approaches to Developer Usability

The top performing banks, respectively Nordea, ERSTE Group and Fidor, have comprehensive portal usability, app management and sandbox environment. The analysis shows great variance in the offering of a sandbox. The top banks cover the complete API offering in a sandbox and guide developers through the process, having the sandbox integrated and with extended help functionality. Other banks do not offer a sandbox or without a GUI, leaving the developer to only get access to the sandbox through a terminal.

Bunq however, has a deviant approach by offering a large set of useful developer tools and accompanying documentation, including an Android app that connects to a personal test account in the Bunq Sandbox environment. Although this might take some extra time in the initial set-up of the APIs and getting familiar with the Developer Portal, the presence of the available tools (e.g. offering SDK's with the most different (script) languages) seems to make up for it on the long run. Such approach might be a good way of binding with developers, that is, when developers are over the steep

learning curve, chances are that they will return to use the respective bank's APIs.

The depth of app management differs substantially across portals from only basic key management functionality to comprehensive management of app permissions, team management (incl. roles) and even app analytics. These are good examples to improve a Developer Portal focussing on how developers are being served by the bank through its Developer Portal. These extended features can offer a big advantage to the developers, especially when third parties want to offer many different APIs, working with large development teams.

Several banks have updated their Developer Portal looking for better ways to service and interact with developers

As shown in figure 8, most fluctuation is seen in the offering of SDK's and other developer tools, with Bunq leading in SDK offering and Nordea with additional developer tools. BBVA has the most consistent offering on each category in Developer Usability, by dividing their attention and scoring far above average in each category. Nordea is the clear winner with great Portal Usability and a lot of additional documentation (e.g. many tutorials and guides) to help developers get started.

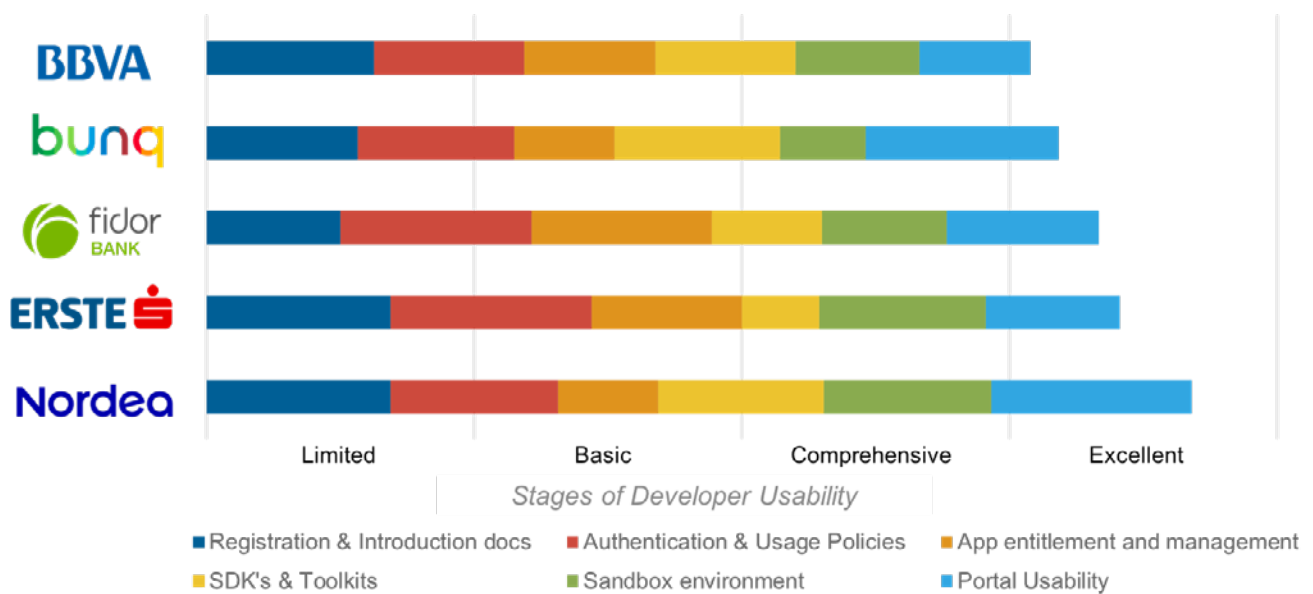


Figure 8: Top 5 banks in Developer Usability rated on each of the six capabilities

### First interaction with developers is key

Additionally, the way the first interaction with developers entering the Developer Portal is shaped, could create a barrier for developers to get engaged. The research shows large differences in 'getting started guides' and 'extended how to's' for developers to get acquainted with the portal and its way of working. Also, for Developer Usability, next to API design, a common set of guidelines for all portals could help developers to get up to speed more quickly. A progressive example would be the Open Banking Project in Nigeria<sup>1</sup>. While this initiative is still in an early stage and mainly focused on API documentation, various elements of Developer

Usability are taken into account (e.g. authentication and a sandbox). Creating common guidelines in an early stage for a community of (small) banks in a particular region could contribute to a faster growing ecosystem and increased cross-fertilisation.

### Banks tend to excel in a single capability of Developer Usability

The figure below shows a representation of the best performing banks in each of the six Developer Usability capabilities.

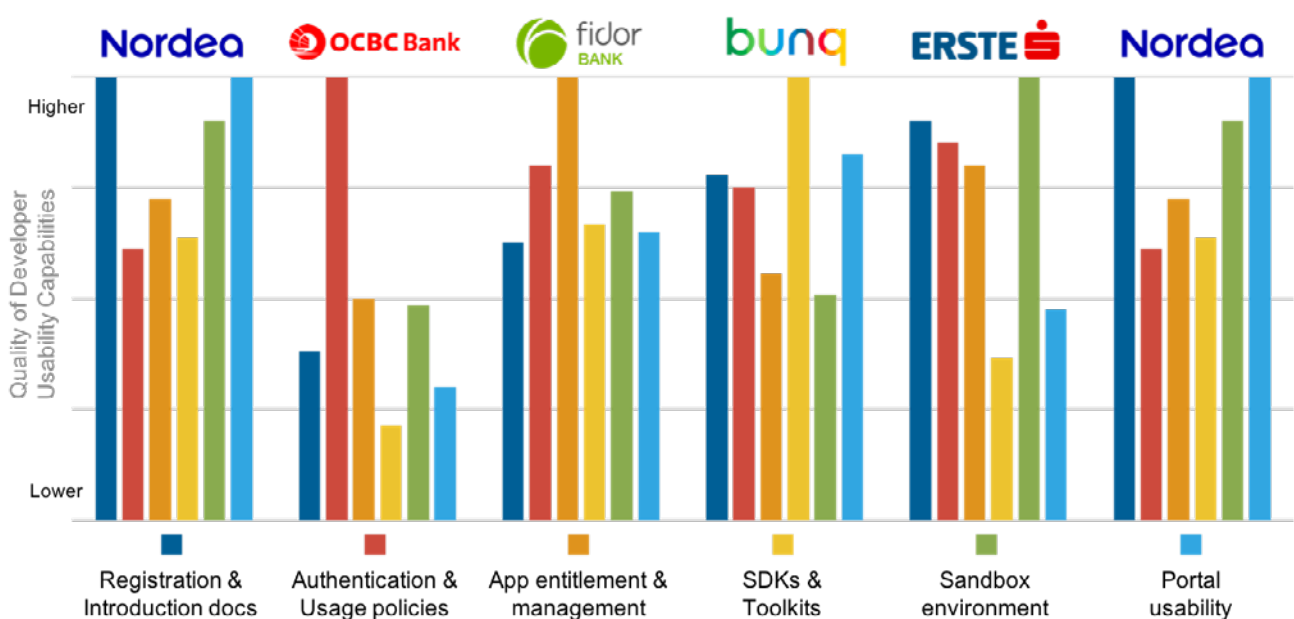


Figure 9: The best performing bank across six Developer Usability capabilities

<sup>1</sup> Open API Standards for Banking in Nigeria - <https://openbanking.ng/>

The data in figure 9 shows that most banks tend to excel in a single capability of Developer Usability. Nordea, however, is the top performing bank in Developer Usability achieving high scores on two capabilities: 'Registration & Introduction documentation' and 'Sandbox environment'. Nordea's sandbox is intuitive to use and has clear and well-structured documentation. Onboarding is quick and easy with the guidance of their "Developer Portal Starter guide", setting-up an account requires minimal effort. Only two banks (i.e. SEB Group and ING) are offering federated login functionality enabling developers to create their account in just a matter of seconds. Banks, in general, can further improve their Developer Usability by adding 'App entitlement and management' and 'SDK's start-up toolkits' to their Developer Portal.

There seem to be only very few banks (e.g. Fidor, Erste and Capital One) which are focussing on 'App entitlement and

management', where a large group of banks offer virtually no related functionality. Considering this is mainly of importance when working with multiple developers on an app, most banks have not met that maturity level on their Developer Portal yet. As stated above this can, however, be a great advantage in serving developers.

The fact that the quality of these capabilities substantially fluctuates across banks emphasises again that Open Banking is in an emerging state. The different capabilities currently being measured will probably be extended in a subsequent release of the OBM. Most likely, the fluctuation of the quality will decrease when Open Banking will achieve a more mature state, leaving fewer different banks reinventing the elements of the Developer Portal as they learn from best practices.

## 5 Developer Community

### Key messages of Developer Community:

- More banks are starting to see the potential of building a Developer Community to strengthen their position as an Open Bank in the ecosystem
- Critical mass is key for enabling a community around an Open Banking ecosystem
- Banks differ in the sophistication of shaping their Developer Community ranging from relatively simple support functions to full-fledged collaboration approaches embedded in other communities

Developer Community refers to the way banks inform actively engage developers to interact with the bank's Developer Portal. Certain banks are actively engaging with developers by creating direct channels to let 3<sup>rd</sup> party developers get in touch with the bank's developers. Other banks are organising events like hackatons to build and engage the Developer Community.

### Relevance of the Developer Community

The community of developers which is allied to the Developer Portal of the bank can play an important role for the banks position in the Open Banking environment. As the Developer Community increases, most likely production of API consuming apps will also increase. Incentives of developers joining the community might vary from a large customer base

the bank is offering, the experience of the Developer Portal, to a functionality which is solely offered by the respective bank. Setting up, maintaining and growing a community around the Developer Portal and/or participating in other's communities is likely to strengthen the bank's position by encouraging 3rd parties to drive innovation and to offer a greater variety of apps in a faster time period.

### Three stages of Developer Community sophistication

We separate three stages in which the level of community engagement differs with the level of sophistication, respectively 'support', 'manage' and 'collaborate', shown in figure 10.



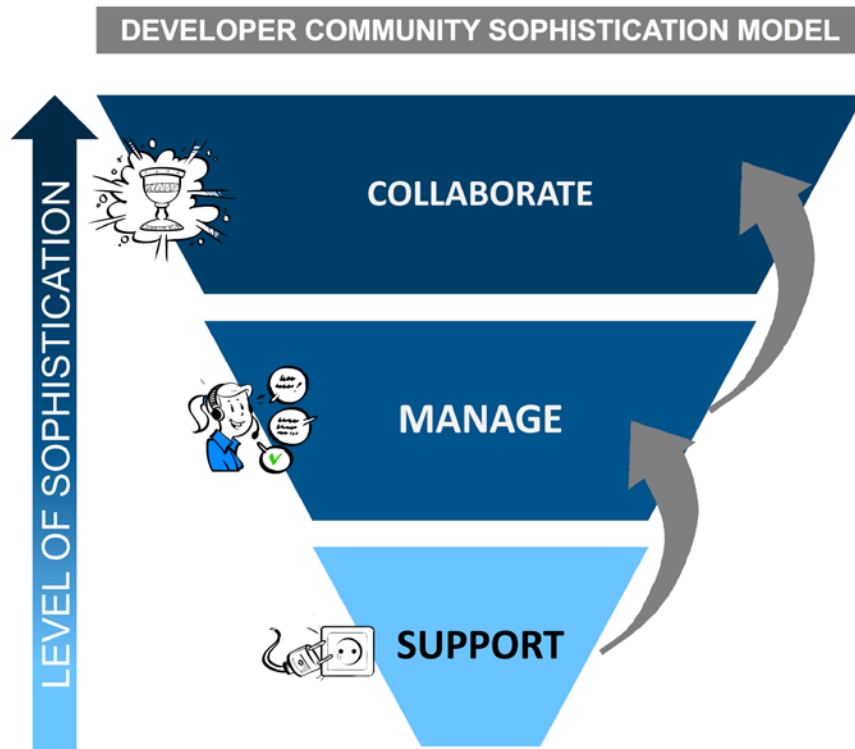


Figure 10: The three stages of Developer Community Sophistication

The 'support' stage can be defined as providing a Developer Portal with a toolset for developers to find their way around. This, over time, will be the smallest investment for the bank, however this will also have the least effect on growing the size of the developer community and cross-developer collaboration. Examples of banks in this stage would be Standard Chartered, BAML and Lloyds Bank. Most of the banks in this stage are "Starters in Opening-up" gradually working to improve the developer experience of their Developer Portal.

Moving up to the 'manage' stage, banks actively provide 3rd parties the ability to get in touch with the banks' developers, answering questions and establishing online discussions. Guidance through the development process can be actively stimulated by the banks' developers through dedicated communication channels and messengers (e.g. Slack or Telegram). Offering the ability to subscribe to updates on certain topics or specific APIs will keep developers informed of any changes or new insights in a suitable manner. Getting traction on a more mature level can also involve crowdsourcing idea generation for new APIs and online presence of commonly used forums (e.g. Github or Stack Overflow). Examples of banks in this stage are Swedbank, NAB and Erste Group.

The highest level of sophistication is the 'collaborate' stage in which banks actively bind with developers by organising events and hackathons to share experiences and insights. If these events are used adequately, it could lead to

strengthening the bank's position on the Functional Scope of APIs as well as the Developer Experience. Developers can share insights on the usability of the Developer Portal and experiences of the tools can be gathered at first hand. On the same note, new ideas can be generated for an app or a functionality to expose. If these new ideas are used in an updated version of the Developer Portal, developers will feel heard which will increase the likelihood that they will return. This will eventually generate a sustainable community around a banks Developer Portal. Examples of banks who are actively creating a Developer Community are Nordea, Monzo and Starling.

Nordea, Monzo and Starling are actively creating a developer community

With many banks across the globe establishing the basics of their Open Banking API platforms, there is a strong incentive towards differentiation in the emerging Open Banking landscape.

A “one-size fits all approach” will most likely not lead to success, as banks need to make strategic decisions on the four core capabilities, API catalogue, documentation, usability and community. Different types of banks are likely

to reap different benefits and experience different drawbacks from engaging in the Open Banking play. Moving forward, it is inevitable, however, that we will witness an explosion of Open Banking APIs.

To support banks in the execution of their Open Banking strategy, we have defined five strategic actions that banks can initiate today, as visualised in figure 11.

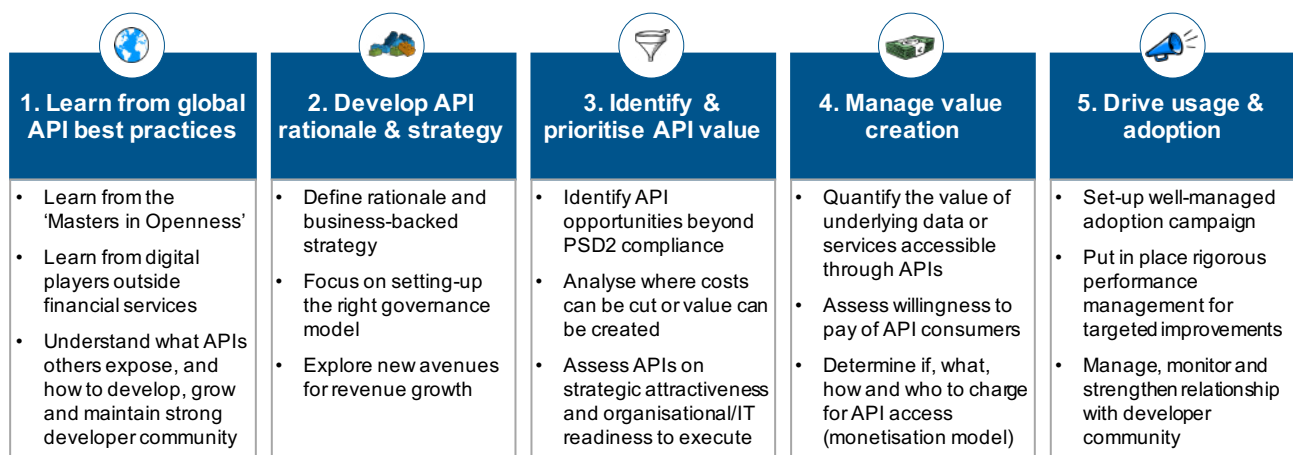


Figure 11: Five strategic Open Banking actions

#### Learn from global API best practices;

Learn from the 'Masters in Openness' in the Open Banking Monitor, and from digital players outside the financial services industry. This will provide insight in 1) what APIs other players expose, 2) how these APIs are distributed and potentially monetised and 3) how to create the most compelling developer experience to attract, grow and maintain a strong developer community.

#### Develop an API rationale and strategy for your business;

Open Banking in general and API monetisation in particular are definitely not a business model fit for all types of banks. Moving beyond PSD2 compliance APIs, requires solid understanding and decision making on the strategic attractiveness of APIs and organisational and technical readiness to execute. Banks pursuing an “API first” mentality can generate various benefits both for internal functions and externally, but financial institutions first need to understand if and where best to apply APIs.

It requires deliberate decision making from banks to 1) define a business-backed strategy for different customer segments (e.g. retail, corporate, SMEs, technology players, fintechs), 2) focus on setting up the right governance model to support effective execution of the strategy, and 3) explore

ways to create powerful new avenues for revenue growth by assessing if and how potential monetisation models could work in your specific context.

#### Identify and prioritise the value that can be captured with APIs;

With a clear strategy in place, banks need to focus on what they need to implement in order to capture the value they have identified. Banks continue their journey by detailing further where 1) value can be created, then they 2) estimate the potential impact in terms of revenue, customer experience, productivity and 3) determine efficiency gains by reducing operational or technology costs through simplified and accelerated development.

#### Manage value creation actively;

Banks need to determine if, what, how and whom to charge in a transparent manner. This requires quantifying the value of the underlying data or service that is accessible through an API (e.g. how proprietary is it and what is its role in generating value). In addition, banks need to assess how much API consumers and/or end-users might be willing to pay to access those APIs, to obtain insights in the revenue streams the APIs will open up.

In determining which monetisation approach to use, banks should 1) think about how their data and 2) how APIs can add distinctive value for different customer segments and 3) determine the most appropriate pricing strategy. Those insights can help banks make an informed decision on monetisation arrangements to pursue with different partners and/or end-users.

**Drive usage and adoption to accelerate network effects and gain scale;**

Like any product or service, a successful Open Banking API strategy requires a well-managed adoption campaign backed by rigorous performance management. Generally successful API first approaches start with engagement of selected API consumers and/or end-users to explore what benefits the use of APIs brings. Along the way functional and technical requirements are updated to fix issues, while related business, legal and operational arrangements are put in place to govern relationships. Once this is in place, banks proceed with driving wider-adoption to achieve critical mass among API consumers.

Combined with rigorous, ongoing performance measurement focused on relevant usage and traffic metrics, banks can obtain the needed insights to make targeted improvements and validate desired strategic and customer outcomes. Indeed, delivering innovation through an Open Banking API platform requires banks to build capabilities to 1) manage, 2) monitor and 3) strengthen their relationship with diverse segments of API consumers.

In essence, Open Banking should be approached as a business strategy and business model in its own right, requiring an alternative way of thinking and working in product development. Combined with powerful execution capabilities and a successful and scaled partnership ecosystem banks will be able to future-proof their competitive position in the Open Banking era. INNOPAY's experience and services portfolio can support banks to design, launch, and scale their Open Banking API platform strategy.



**Mounaim Cortet**

Senior Manager Strategy, Lead PSD2 & Open banking  
mounaim.cortet@innopay.com

**Art Stevens**

Consultant, Open Banking Specialist  
art.stevens@innopay.com





World Trade Center F-tower  
Strawinskylaan 381  
1077 XX AMSTERDAM  
The Netherlands  
T: +31 (0)20 65 80 651

Taunustor 1  
(TaunusTurm)  
60310 Frankfurt a.M.  
Germany  
T: +49 (0) 69 50 50 60 4350

[info@innopay.com](mailto:info@innopay.com)  
[www.innopay.com](http://www.innopay.com)

© 2018 Innopay

#### About INNOPAY

INNOPAY is a consultancy firm specialised in digital transactions. We help organisations worldwide to optimally utilize the opportunities presented by the digital transaction ecosystem. Since being founded in 2002, INNOPAY has evolved into an international player with a growing number of offices, including in Amsterdam, Frankfurt and Berlin. Our head office is in the Netherlands, where we are the market leader. We are a founding member of the Holland FinTech hub which interconnects fintech organisations from across Europe, the USA, the Middle East and Asia. Our team comprises over 50 experts in the areas of payments, digital identity and e-business.