Tink's guide to

open banking connectivity

What it takes to connect to open banking APIs – and why it's not as simple as it seems.

tink
A Visa Solution

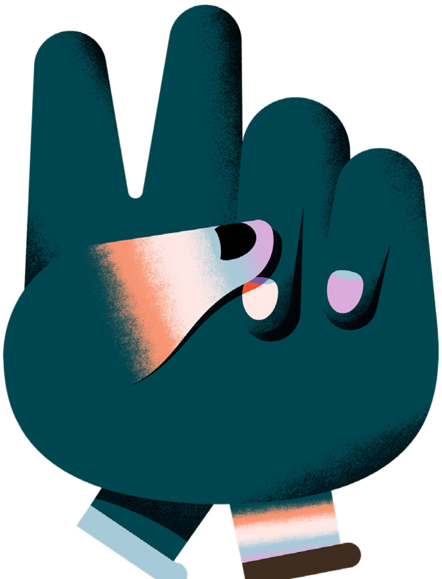
On the surface, connecting to an open banking API seems to be a fairly straightforward task. Since the PSD2 legislation came into effect in 2018, ASPSPs across Europe are obliged to provide free access to customer data through APIs. And once you've connected to that bank or financial institution, the connection should work flawlessly till the end of time. Right?

Wrong.

What's missing in this equation is the very simple fact that ASPSPs don't get any money for providing access to data through APIs. While it may be a long-term investment in the idea and vision of a more open financial system, it doesn't pay the bills here and now. This means that the ASPSPs teams responsible for managing these APIs are often understaffed.

Open banking platforms such as Tink form an important part of this new ecosystem. They work together with the ASPSPs to continuously monitor their connections, find bugs, report problems, and communicate with their engineering teams on a regular basis. This is an operational and technical challenge that requires massive investments in time, people and infrastructure.

This guide explains the entire process and shows why connecting to an open banking API is about much more than making the initial connection.

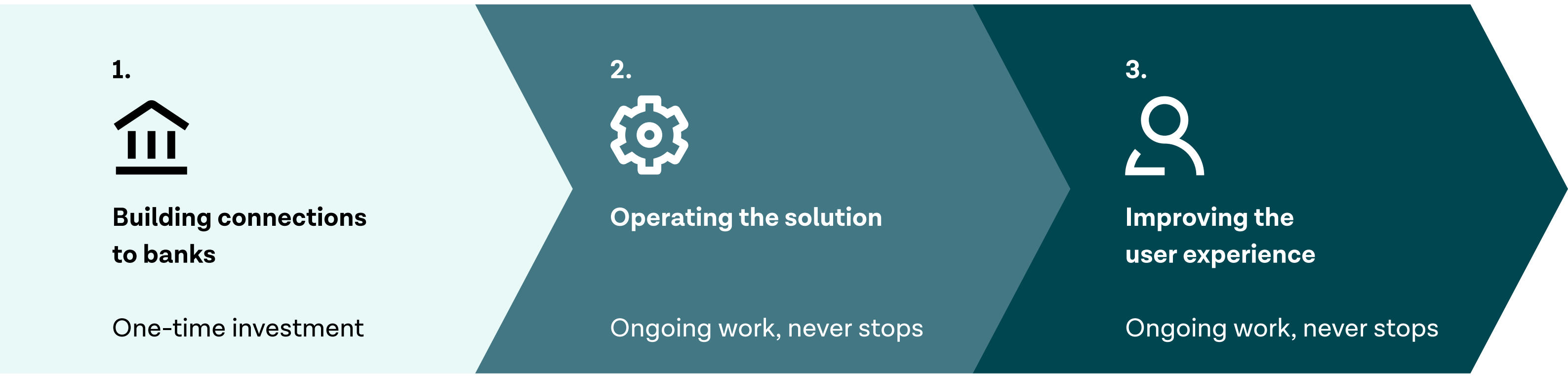


What you'll find in this guide

What does connecting to open banking APIs really mean?

Building out connectivity across multiple countries is a complex endeavour that requires significant investments, especially in engineering, operations and legal. To explain this properly, we will use the hypothetical example of a Third Party Provider (TPP) connecting to Account Servicing Payment Service Providers (ASPSPs) across Europe and take you through the entire process of optimising connections to bank APIs, where connecting to a bank is only the first step of the value chain.

Connecting to an open banking API



1. **Building connections to banks**

One-time investment

2. **Operating the solution**

Ongoing work, never stops

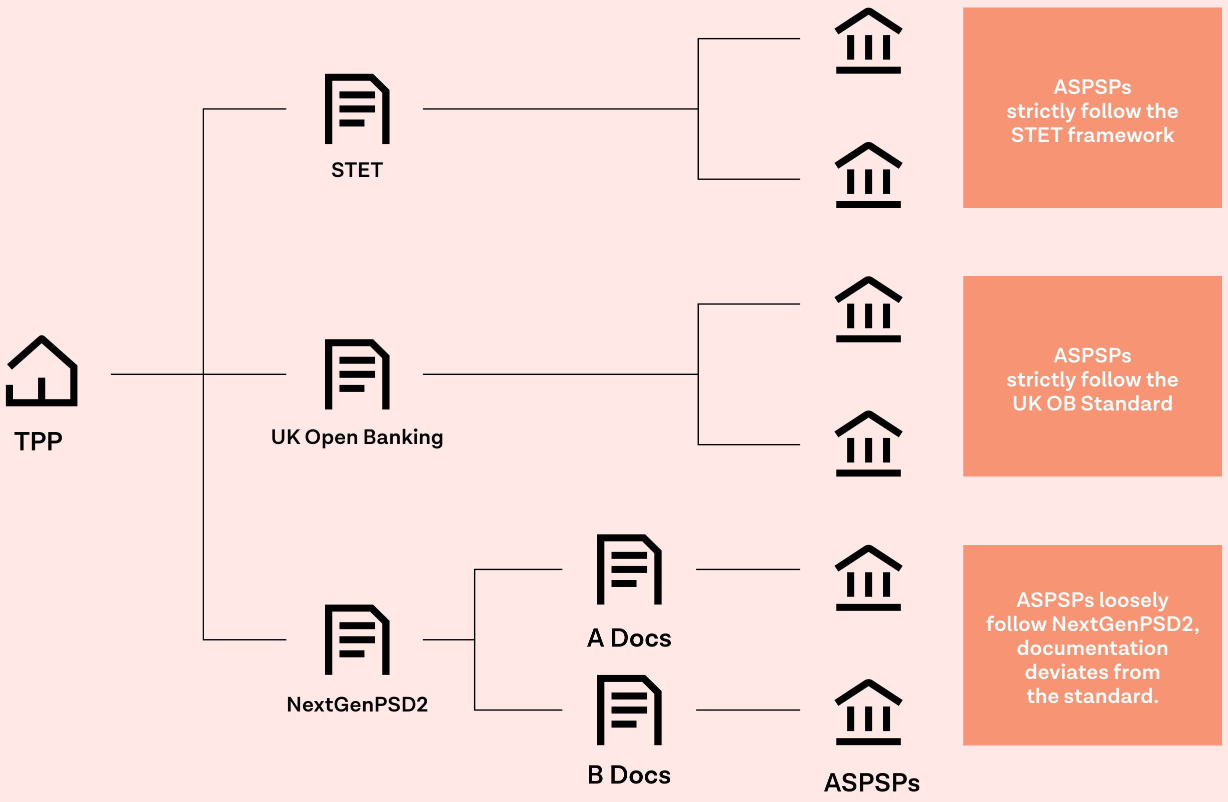
3. **Improving the user experience**

Ongoing work, never stops

Building connections to banks

1.1 Understanding European standards

Connecting to an open banking API demands a thorough understanding of standards that are used by banks across different countries. In Western Europe, there are three overarching frameworks:


STET
UK Open Banking
NextGenPSD2
TPP
A Docs
B Docs
ASPSPs
ASPSPs strictly follow the STET framework
ASPSPs strictly follow the UK OB Standard
ASPSPs loosely follow NextGenPSD2, documentation deviates from the standard.

STET: The STET PSD2 API framework is used mostly in France and is stricter than the NextGenPSD2 framework. The main difference lies in the signature request protocol, which is used to validate that the information has not been altered by the TPP.

UK Open Banking: The UK Open Banking Standard is the least flexible, with most UK ASPSPs following an almost direct implementation of it with very small variations. This is the most mature standard as it has been around since 2018, thoroughly tested by hundreds of TPPs and it’s development and implementation has been monitored by the CMA and FCA.

NextGenPSD2: The Berlin Group’s NextGenPSD2 Access to Account (XS2A) Framework Standard provides an architectural view of the API and allows for a flexible implementation, where ASPSPs have the possibility to deviate from the standard and implement it according to their own requirements. Although the Berlin Group requires ASPSPs to submit change requests to the committee governing the framework, unfortunately this is rarely done. In turn, this has resulted in high fragmentation of the implementation of the standard and high degree of variance between open banking APIs. This is the most commonly adopted framework in Europe.

1.2 Registering at ASPSPs

The second step is to register at ASPSPs and this can be done through either manual or dynamic registrations.

Manual registrations require communication between the TPP and the ASPSP, such as registering in the developer portal, creating an app, or sending back and forth emails with the ASPSPs to obtain access. This process doesn't require valuable developer time, but can take anything from a few days to months to complete, depending on the readiness of the PSD2 API. Unfortunately, manual registrations are the norm in Europe and only about 15% of European ASPSPs offer dynamic registrations.

Dynamic registrations are made through an API and if done correctly, the ASPSP registers the TPP automatically. Registering dynamically requires a large initial investment but is highly scalable, allowing for automatic registration without any further effort.

Tink has invested significant resources in building internal dynamic registration tools that allow us to plug-and-play with different registration APIs and onboard our clients almost instantly.

1.3 Making the connection

In theory, the process of making a connection is relatively straightforward as it requires the TPP to follow the documentation published by the ASPSP. The reality, however, can be much more complex.

The challenges of connecting to open banking APIs are different for every ASPSP, and in every country. Some ASPSPs are relatively easy to connect to, while others require a substantially larger time investment.

Luckily, most open banking APIs nowadays have matured to a point where these types of investments are no longer needed, and can be connected at much faster speeds. Nevertheless, here are some of the actions that TPPs should consider taking:

- → Set up a correspondence team that engages with the ASPSPs support organisation when their documentation is unclear or outdated.
- → Become familiar with technical specifications, both local (specific to ASPSP or market, e.g. API specifications in the UK) and broad (wider standards, e.g. JWT methods). This will help developers understand the underlying requirements of the programs that are being used and solve bugs quickly.
- → Design a flexible core architecture to support a variety of ASPSP connections. The technical standards mentioned in section 1.1 provide a foundation for what the open banking APIs look like, but the setup to gain access to the account depends on the ASPSP. The underlying structure is extremely important, as small alterations may impact other connections, resulting in breaking changes.
- → TPPs should support all cryptographic methods to minimise friction for users and plug them depending on the standard required by each ASPSP. For instance, UK Open Banking requires 5 different cryptographic methods that are broadly used across ASPSPs.
- → Invest in training developers and give them a thorough understanding of the standards, systems and processes. The amount of code needed to perform these tasks is expensive and increases with each new connection.

Tink's first large client utilising open banking APIs required a team of 5 developers, testing and remediating bugs across the platform in a single market for a total of 6 months. We exchanged over 1,500 emails and raised dozens of tickets with the banks before the solution was ready to be used.

Operating the solution

2.1 Testing and stabilising connections

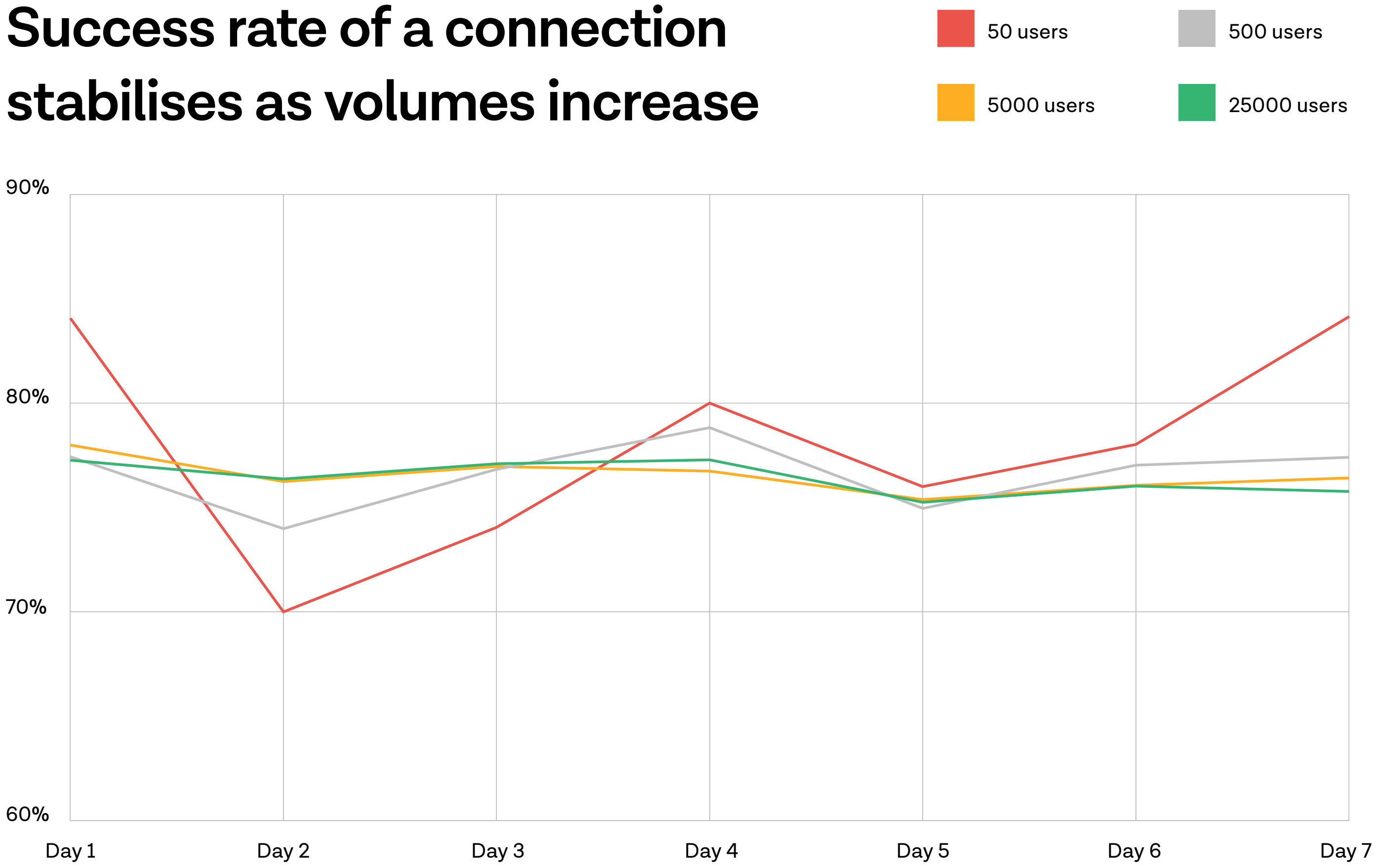
Once the connections have been built, they need to go through a series of rigorous tests to ensure that they are stable and working properly. A typical testing process would look like this:

1. Open bank accounts with every bank in a country, test the connection with real credentials and provide feedback to developers about the end-user experience.
2. After this initial round of testing is completed, work through all the edge cases with a larger number of users.
3. Record all end-user journeys and collect evidence in case a test fails.

If a TPP plans to support a variety of use cases, it is important to have a large amount of traffic across several clients and use cases, as this will give a more comprehensive picture of how a connection is performing. For instance, testing a connection on a digitally-savvy customer base that is able to overcome obstacles during authentication could be falsely interpreted as a well-functioning connection with high conversion rates. In the real world, a lot of users may not be able to work their way around these same obstacles, leading to a big drop-off in conversion rates. This requires TPPs to think about the ways in which all types of customers may be impacted and how to guide them through the journey.

In the graph below, we use live data to illustrate the value of increasing the number of users to test connection stability. The red line shows the volatility of success rates when there are 50 users, while the green line shows a relatively stable success rate once traffic goes beyond 25000 users.

Success rate of a connection stabilises as volumes increase



2.2 Day-to-day operations

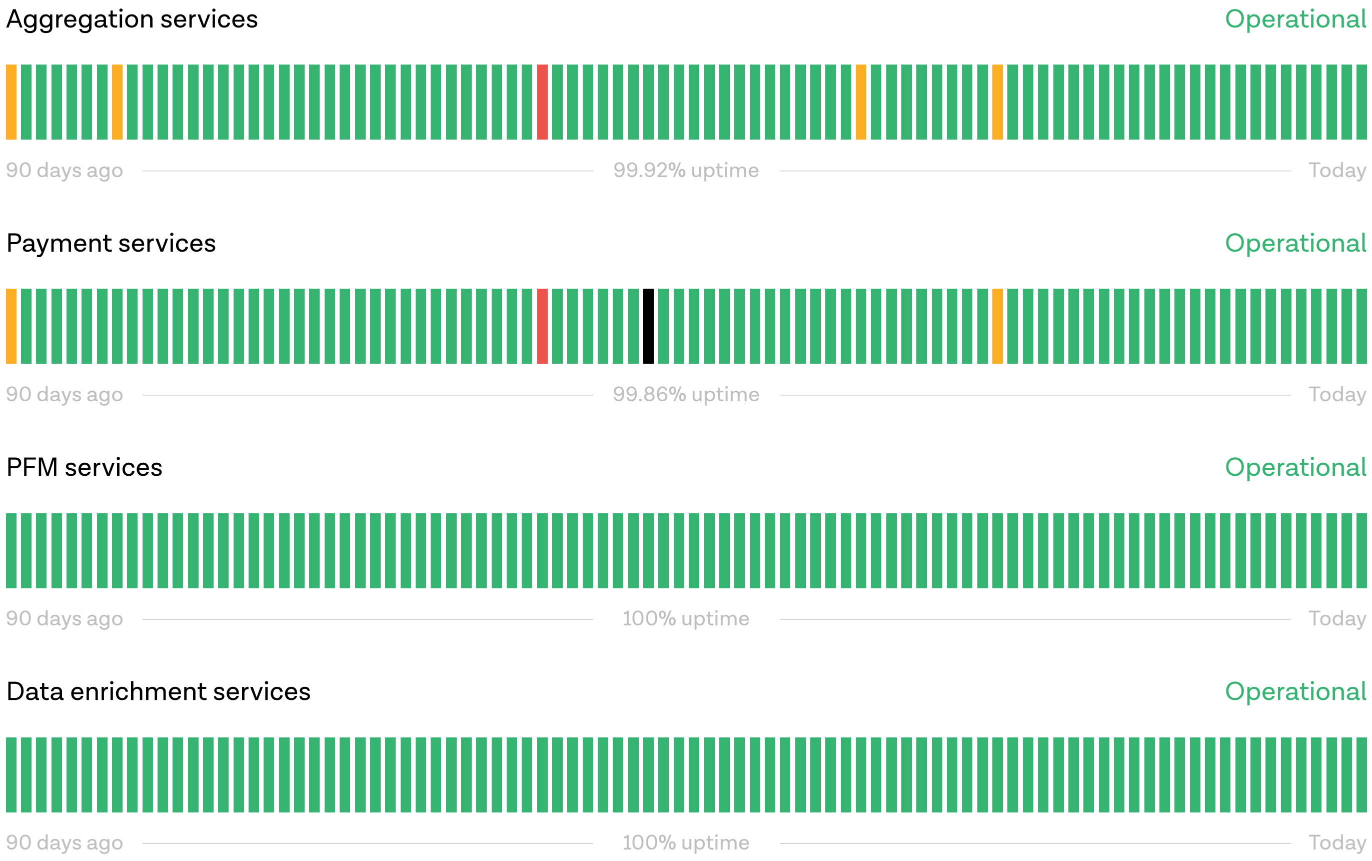
The next step is to ensure that connections work properly on a day-to-day basis, as well as resolve issues that arise in the long-term. To do this, TPPs should put in place a system to constantly measure the quality of each connection, generate automated alerts when connections do not meet quality standards and have on-call developers that are trained to respond to such situations.

Alerts do not always require immediate attention, or even coding, as occasionally ASPSPs services are down. However, TPPs should make sure to identify every incident and risk, have a clear understanding of the root cause of the issue and inform customers when necessary.

Operating a solution over a long period of time, with proper testing and day-to-day routines is a never-ending process and is often undervalued when deciding to build in-house.

At Tink, we have built a comprehensive alerts and on-call system to keep up with constant changes. Our alerts system has a range of priorities and thresholds that take into account the severity of the issue and number of users impacted. As soon as the number of errors crosses the threshold, a message is routed to the team responsible for the API to evaluate and prioritise depending on the severity of the issue.

Service status page on Tink.com



Aggregation services

Operational

90 days ago — 99.92% uptime — Today

Payment services

Operational

90 days ago — 99.86% uptime — Today

PFM services

Operational

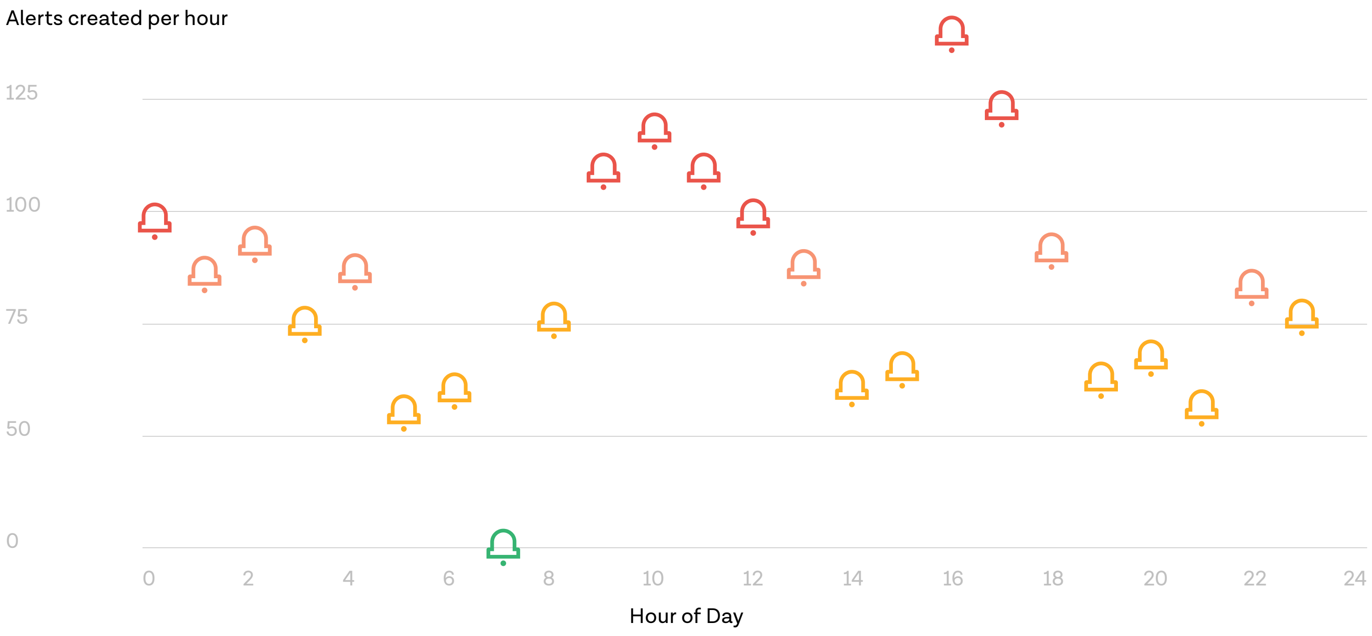
90 days ago — 100% uptime — Today

Data enrichment services

Operational

90 days ago — 100% uptime — Today

Alerts dashboard

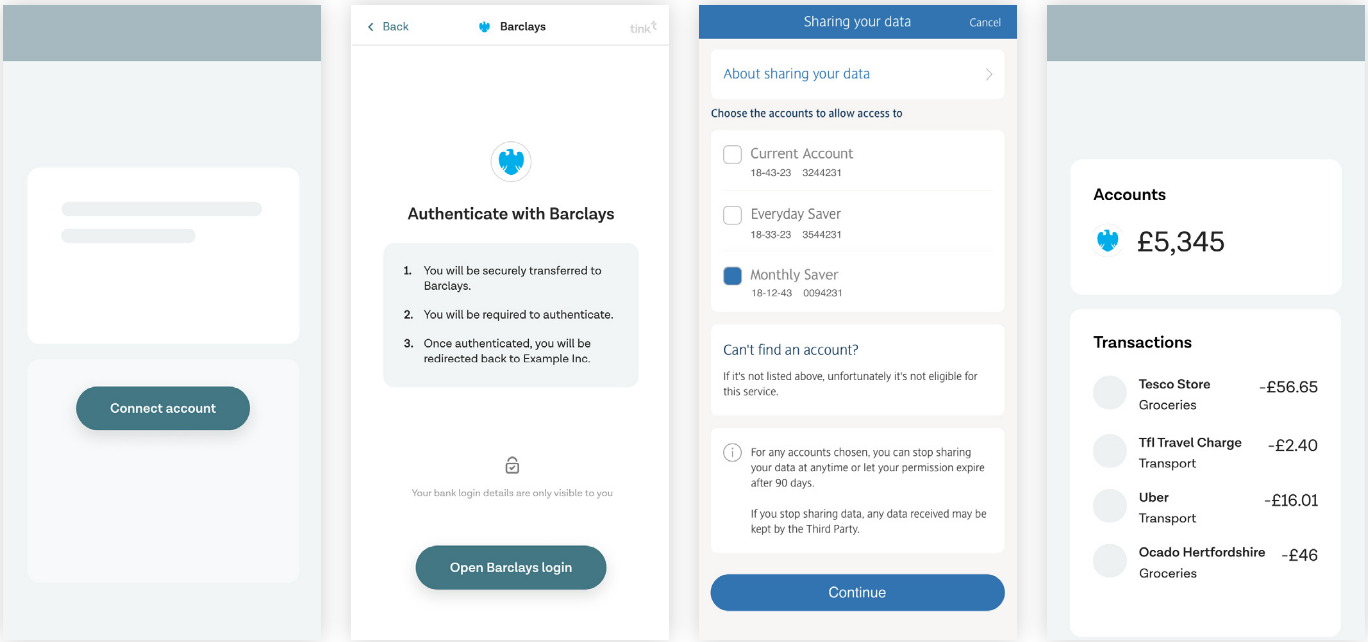


Improving the user experience

3

All the time and effort spent in building and operating bank connections leads to that all-important touchpoint where TPP's ask end-users for consent to access their bank accounts. This is where the user experience comes into play. Providing a good UX requires a combination of operational knowledge of authentication flows, hiding that complexity from the end-user and turning it into a beautifully simple and seamless journey.

At Tink, we do this through Tink Link, our front-end SDK to manage and optimise ASPSP authentication.



Authenticate with Barclays

1. You will be securely transferred to Barclays.
2. You will be required to authenticate.
3. Once authenticated, you will be redirected back to Example Inc.

Your bank login details are only visible to you

About sharing your data

Choose the accounts to allow access to

□ Current Account 18-43-23 3244231

□ Everyday Saver 18-33-23 3544231

■ Monthly Saver 18-12-43 0094231

Can't find an account?

If it's not listed above, unfortunately it's not eligible for this service.

For any accounts chosen, you can stop sharing your data at anytime or let your permission expire after 90 days.

If you stop sharing data, any data received may be kept by the Third Party.

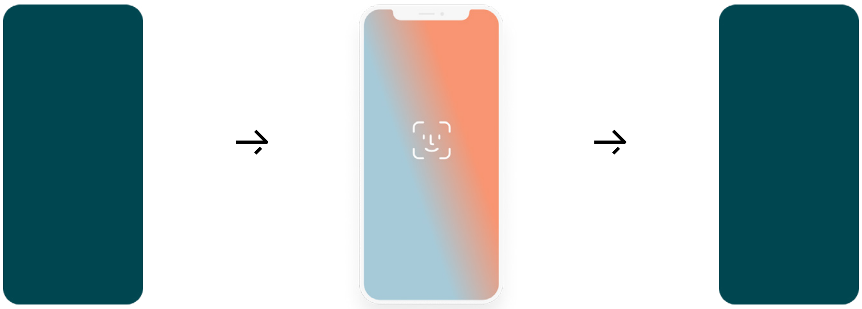
£5,345

Tesco Store Groceries	-£56.65
Tfl Travel Charge Transport	-£2.40
Uber Transport	-£16.01
Ocado Hertfordshire Groceries	-£46

The objective is to ensure that end-users fulfill Strong Customer Authentication requirements, which stipulate that a user must identify themselves through two out of three elements: possession (e.g: phone), knowledge (e.g: password) and inherence (e.g: biometrics). To most end users, this may look like a simple sequence of login screens. The reality is that there is a whole bunch of complexity in the background, such as connection type, device type, countries and languages, among other variables. But perhaps the most challenging part is that many steps in the user journey are outside of a TPPs control. Due to the lack of consensus within the industry for how ASPSPs should be verifying user identity, TPPs often need to build bespoke authentication flows, which creates a head-scratching number of combinations to consider. Typical scenarios include app-to-app, web, decoupled and embedded flows.

Common authentication flows across Europe

Decoupled
(Sweden)



TPP

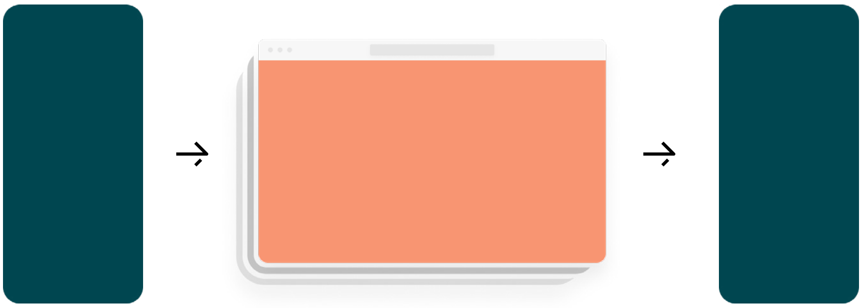
ASPSP
Authentication app
User fulfills SCA

TPP

Embedded
(Germany)

TPP
User fulfills SCA

Web redirect
(Spain, Italy)

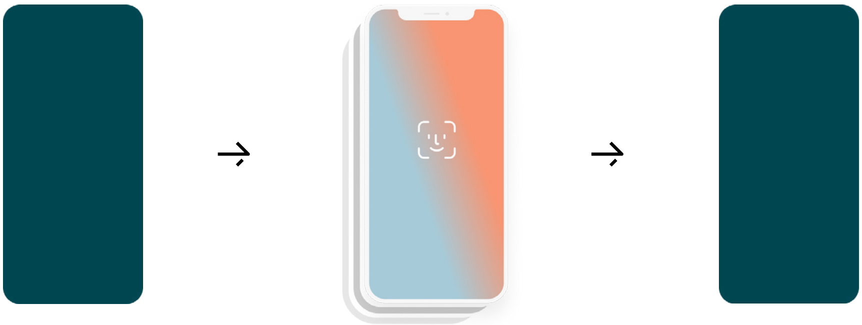


TPP

ASPSP Web environment

TPP

App redirect
(UK)



TPP

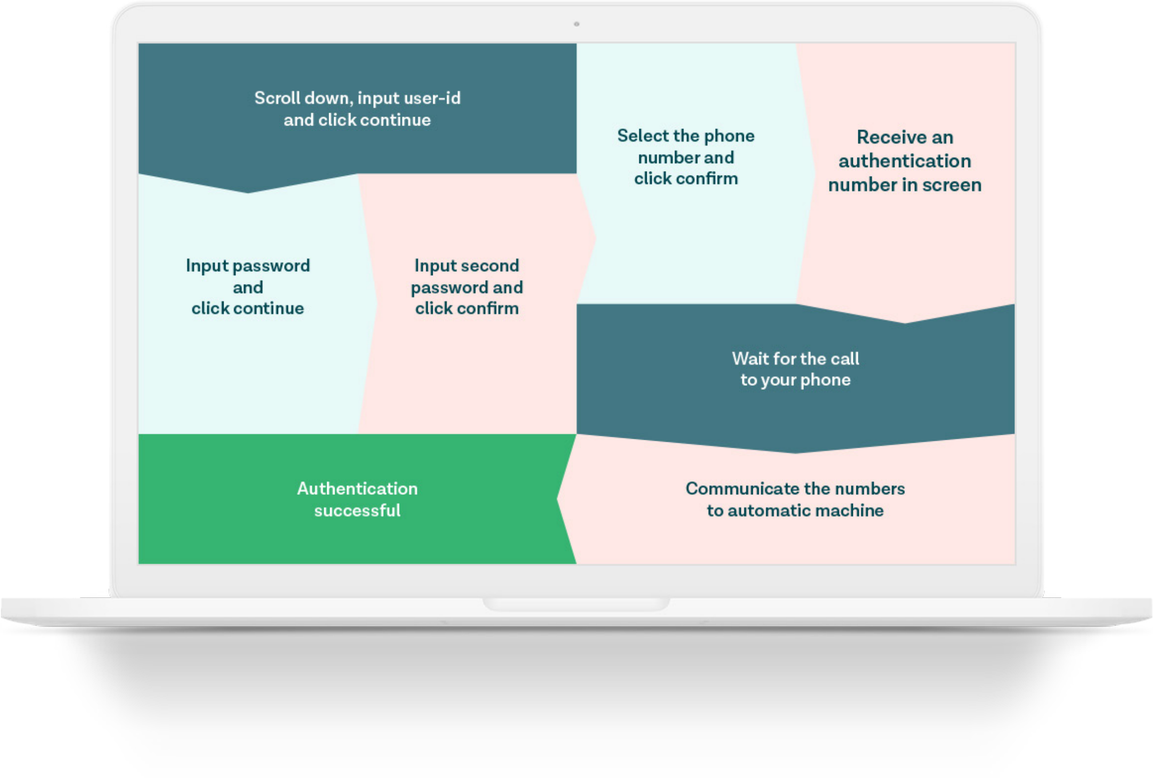
ASPSP app environment

TPP

Case study

To explain the complexities behind improving the end-user experience, let us take you through a recent example of a client who wanted to remove manual steps in their onboarding process through Tink. Improving user onboarding is a relatively well-established open banking use case and we've done it for a number of clients before. We authenticate users, fetch their account information, connect bank accounts and voilà! Our teams spent a couple of months on the integration work but once we rolled it out, we were surprised to find that the conversion rate of new users finishing the entire onboarding process was disappointingly low. So we started to drill into the data to understand what was going on.

Interestingly, the conversion rate of mobile users completing the onboarding process was as per internal benchmarks. The real drop-off was caused by starting and ending the authentication journey on a desktop computer. This is because most financial institutions in the UK have made big investments in improving their mobile banking journeys, but their desktop journeys are still quite poor.

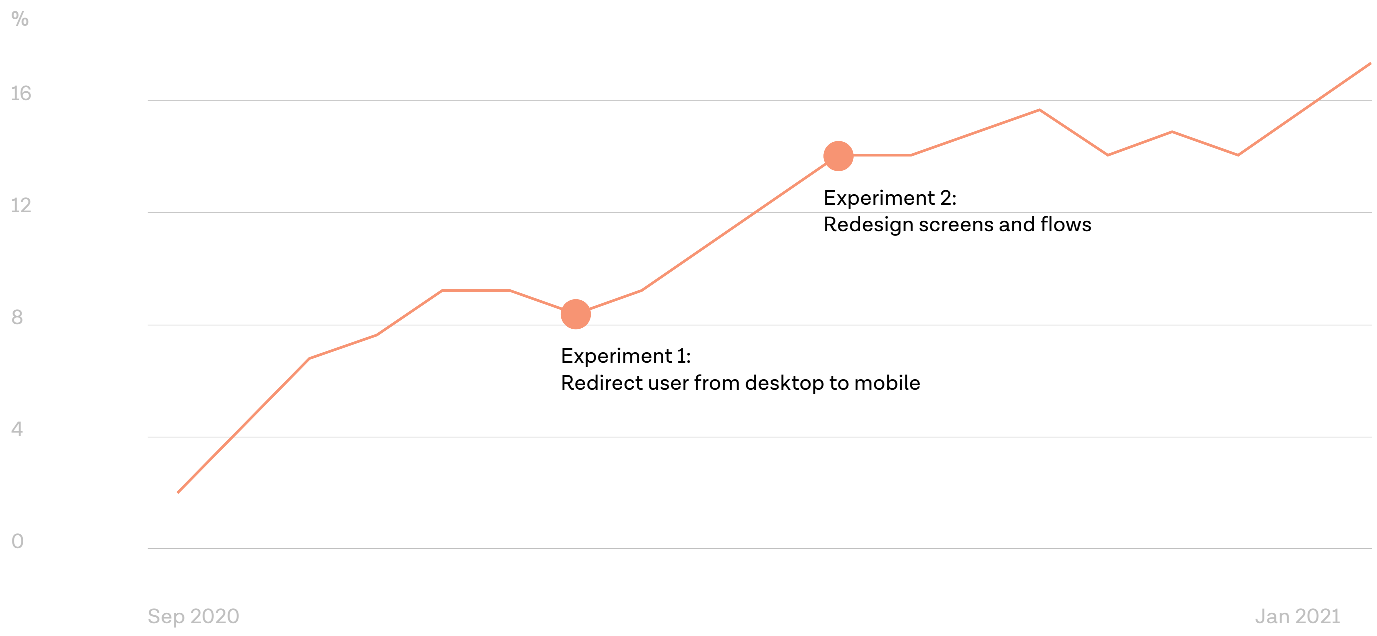

Scroll down, input user-id and click continue
Input password and click continue
Input second password and click confirm
Select the phone number and click confirm
Receive an authentication number in screen
Wait for the call to your phone
Communicate the numbers to automatic machine
Authentication successful

Understandably, the result was a very high drop-off rate. With this key insight, we started to gather login flows for every bank in that country, highlight pain-points, brainstorm ideas on how to improve them, and engage in discussions with the relevant banks as well as the UK OBIE. After several experiments, we found that the best solution was to use QR codes to redirect users from desktop to mobile to complete the authentication process and bypass all the hurdles that were present in the desktop journey.

The first iteration of this new “QR code handoff screen” was rolled out within a few days. We gathered our data analytics team and waited for their verdict: have success rates improved? The results were encouraging, but still did not meet our expectations. The next step was to bring in designers. Our product design team made several improvements, from reducing the QR code density to providing clearer instructions to the user. We also removed the option for the user to choose between desktop and mobile journeys and asked them to complete the authentication process entirely on their mobile device.

The final results were outstanding, with those utilising the QR code increasing their success rate by 62% over those staying in the desktop journey.

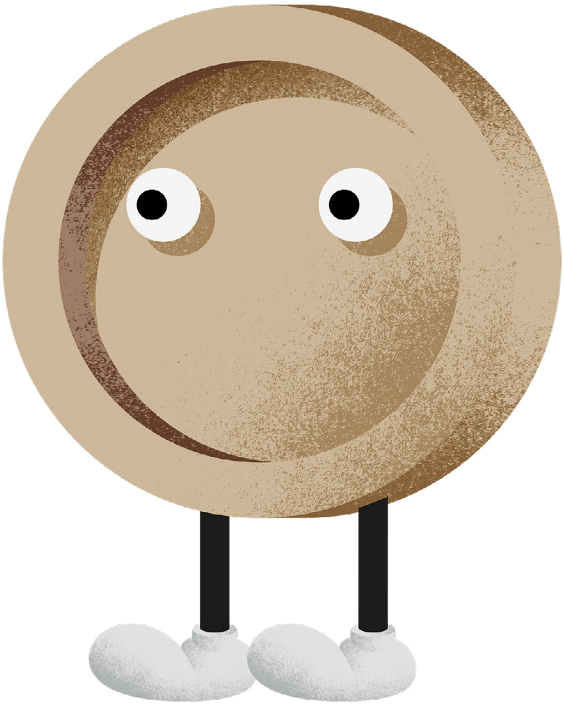
Relative % improvement in end-to-end success rates



This example highlights two things:

→ The transition from physical to digital banking is messy, requires a deep understanding of the market and in many cases, one-off solutions.
While this may not sound very scalable, it is where we are at the moment.

→ The importance of cross-functional disciplines in improving the overall user experience. The pillars of such improvement are a mixture of technical and design expertise, combined with a data-driven approach to understanding the pain-points experienced by users.



Driving the industry forward

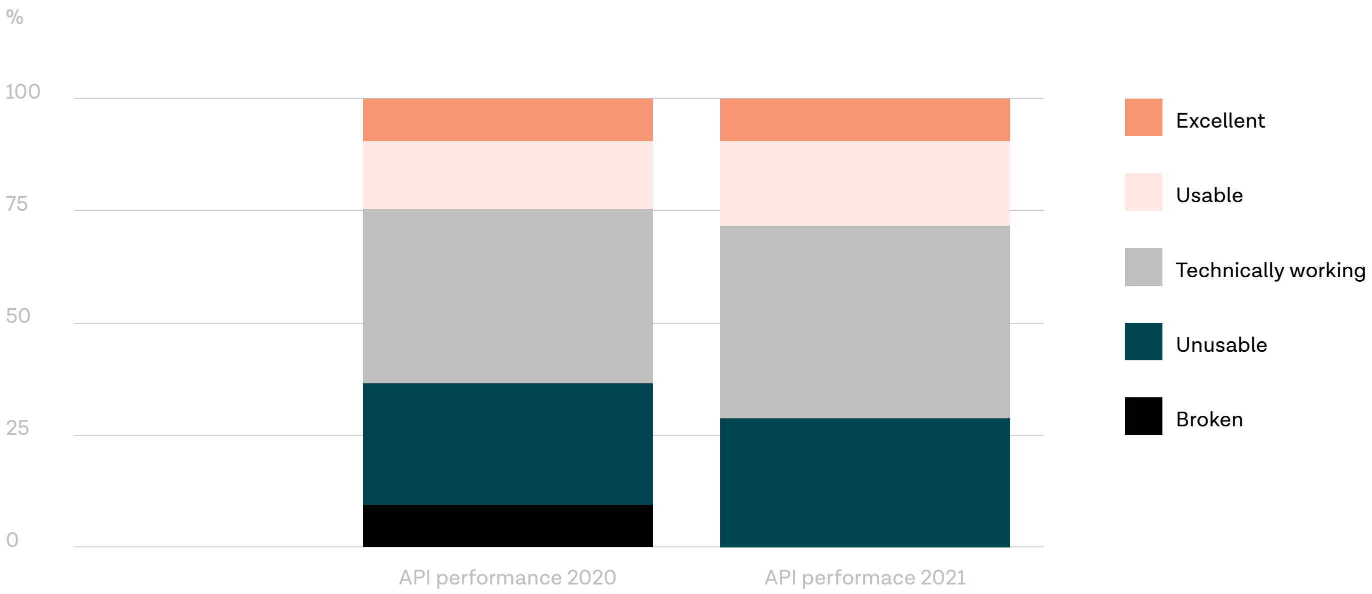
Driving the industry forward

Besides the technical side of building and optimising connections, TPPs also need to constantly engage with ASPSPs, regulators and financial authorities. Regulations such as PSD2 and RTS are complex, and TPPs play a valuable role in pushing the industry forward by highlighting short-term issues, holding ASPSPs accountable when regulations aren't met, as well as staying laser-focused on the long-term vision of open banking.

Short term

In the short-term, TPPs work together with ASPSPs to resolve issues that have an immediate impact such as unexpected changes to APIs, missing documentation, out-of-date test environments and server downtime without prior communication to TPPs. In most cases, direct communication with ASPSPs is enough, but in cases where no action is taken, TPPs have to escalate issues with the relevant authorities.

The good news is that these initial investments are starting to really bear fruit and over the past few years, API performance among ASPSPs has improved substantially.



The figure above shows the status of PSD2 APIs of major banks in 12 countries. We do these assessments periodically to measure the latest improvements in API performance.

→ **Broken:** The PSD2 API is non-existent or does not work.

→ **Unusable:** The PSD2 API works but the authentication journey is extremely cumbersome for the end-user, requires several non-intuitive steps, multiple SCAs and web-redirection.

→ **Technically working:** The PSD2 API works but the authentication journey does not offer app-to-app redirection and has unnecessary screens.

→ **Usable:** The PSD2 API is usable and provides app-to-app redirection. However, the authentication flow still contains unnecessary screens.

→ **Excellent:** The PSD2 API offers app-to-app redirection and has no unnecessary screens. The screens are optimised to enable the user to authenticate.

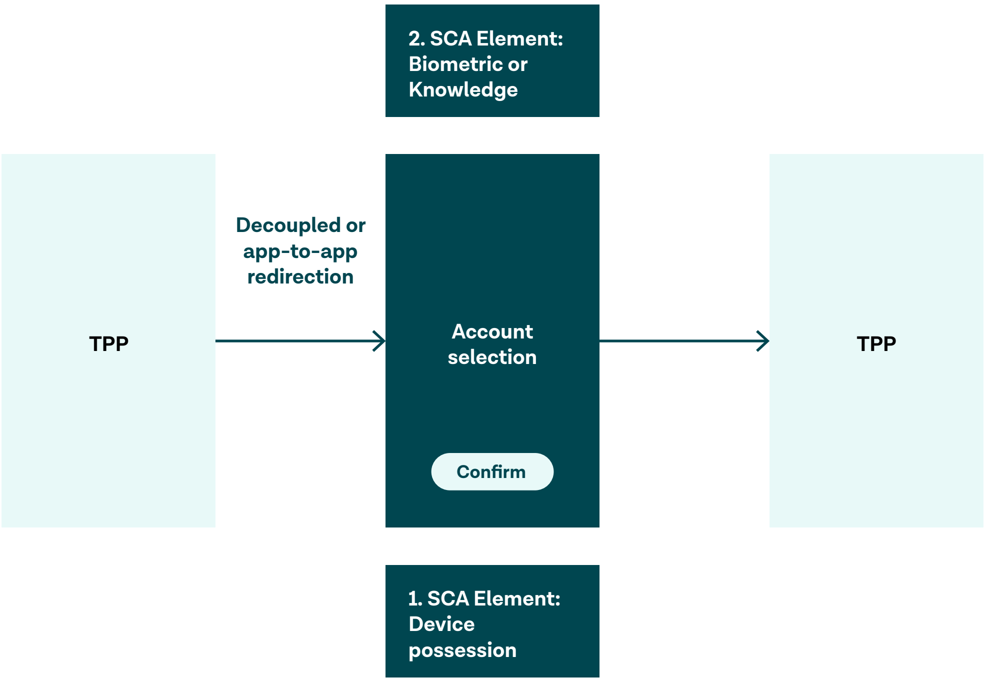
Long-term vision

Connecting to a bank API for the sake of establishing a connection means very little if the long-term vision of a more customer-centric, competitive and innovative financial industry gets lost in debates around regulation, technology and processes. This requires TPPs to have a clear view of the end-user experience, and invest time and resources in making this future a reality.

The best way to do this is through data. TPPs need to create a culture of constant testing, mapping of obstacles encountered when connecting to an API, and sharing insights and data with the ASPSP.

Utilising objective, quantitative data on the impact of sub-par user flows makes it easier to engage in constructive conversations with ASPSPs and regulators. Although this is a prisoner's dilemma, where a TPP may benefit from someone else actually making the lobbying investment, the long-term implications of better user-flows and end-user experience is of great benefit to the entire industry.

For example, a big share of Tink's lobbying efforts are focused on making the figure below the standard user flow in Europe.


2. SCA Element: Biometric or Knowledge
TPP
Decoupled or app-to-app redirection
Account selection
Confirm
TPP
1. SCA Element: Device possession

- ✓ All authentication procedures are provided by the ASPSP (Art 30.2 of RTS)
- ✓ The PSU is redirected to the ASPSP's authentication app (paragraph 16.1 of EBA Opinion)
- ✓ Strong Customer Authentication fulfilled (Art 97 of PSD2).

In order to influence the long-term vision of open banking, Tink actively participates in lobby groups and trade associations such as ETPPA, PSD2 SIG, PayBelgium, Fintech Norway, UK Finance and the European Payments Association.

Conclusion

If there is one key takeaway from Tink's experience, it would be that building out connectivity is a long-term strategic investment. The industry is moving very fast, and keeping up with the pace of change requires focus and dedicated resources.

With Tink, you don't have to do any of the heavy lifting. Our open banking platform lets you connect to over 3,400 banks and institutions across Europe and get enriched and categorised financial data – through 1 API. We focus on connectivity so you can focus on innovation and smart financial services.

Want to know more about how you can access real-time financial data? Our experts are always happy to answer your questions – just reach out:

partnerships@tink.com

tink.com