Transforming the future of digital banking with APIs and DataSecOps

Received (in revised form): 30th July, 2021

Peter Lancos
CEO and Co-Founder, eXate, UK

Peter Lancos is the CEO and Co-Founder of eXate, which provides a universal way to manage data privacy, accelerating access to sensitive data, automating enforcement of data policies and providing a secure platform for compliant data delivery globally. Before co-founding eXate in 2015, Peter worked in a variety of senior positions at HSBC, including those of Chief Operating Officer for EMEA Markets and the Chief of Staff for Corporate and Institutional Digital. Peter has 30 years of banking experience, primarily in the markets business. He is published as part of the RegTech Book, as well as in several technical journals.

eXate, 85 Great Portland Street, London, England, W1W 7LT, UK
Tel: +44 (0)7444 732834; E-mail: peter@exate.com

Abstract Digital banking involves high levels of process automation and web-based services and may include application programming interfaces (APIs) enabling cross-institutional service composition to deliver banking products and provide transactions. It enables users to have access to their financial data via their digital devices. APIs represent as great a threat and an opportunity to banks today as the advent of the digital era once did, opening the market to greater competition around customer loyalty and engagement. Banks should not consider these APIs as simply technical interfaces that expose data to third parties but rather as radical enablers of new and attractive customer experiences. APIs are already commonplace across many industry sectors, where they are, in fact, viewed as customer products. Banks will have to follow suit in order to remain competitive. In brief, APIs are the nucleus of digital transformation. Around 84.5 per cent of those working on digital transformation initiatives state that APIs are playing a significant role in those initiatives.[1] Given the risk of API-first architecture, these numbers are expected to continue to grow. There are risks, however, that companies face as a result of the increased reliance on APIs as the main driver of digital banking. Seventy one per cent of software engineers surveyed considered 'security' the most important factor to consider before integration with an API, which was tied for the top concern.[2] While software engineers are aware of this risk, there is a need to address it not only at a technology level, but at a risk and business level as well. In order to address these risks, the concept of Data Security Operations, or 'DataSecOps', has arisen. DataSecOps is a discipline that empowers software engineers, data scientists, governance risk and control, cybersecurity & operations teams to work together in a single application for safer and easier access, analysis, delivery and governance of data.[3] DataSecOps principles will become a critical component in addressing the security issues related to digital banking.

KEYWORDS: API, digital era, security, DataSecOps, financial services, privacy, identity and access management

THE RISE OF AN 'API-FIRST ARCHITECTURE'

With respect to companies that have confirmed that they are working on digital transformation information initiatives, 84.5 per cent reported that APIs were helping them do that. APIs have become such a critical component of digital transformation that an 'API-first' architecture has been developed. API-first architecture is an approach to software design that centres the API in order to create applications that can easily interface with one another. API-first creates ecosystems of applications that are modular, reusable and extensible, like Lego blocks. API-first is a different approach from 'code-first', in which developers design an application's functionality and then insert the API at the end. That code-first approach can be problematic if your application is not structured in a way that makes it easy for the API to access data. In API-first design, your development team works on API design before the rest of the application. By introducing new features as an independent service accessed by API, the rest of the app can be stitched together, along with any other future apps. To accomplish this, any successful API-first strategy hinges on creating excellent API documentation that all teams can use. This makes for a better developer experience and ensures that the design of the API is treated as a valuable company resource.[4]

An API-first approach to building products provides the following benefits:

1. Development teams can work in parallel — API-first involves establishing a standard API 'contract' that allows teams across an organisation to work on multiple APIs at the same time. Given the common language of speaking to/from an API, developers do not have to wait for updates to an API to be released before moving on to the next API. Teams can quickly build and test API dependencies on the basis of the established API definition.
2. Reduces the cost of developing apps — APIs and code can be reused on many different projects in an API-first architecture. When a development team wants to build a new application, they do not have to start from scratch, which is time-consuming and costly. API-first design also allows most problems to be solved before any code is even written, helping prevent problems when it is time to integrate APIs with applications.
3. Increases the speed to market — Given that much of the process of building APIs can be automated using existing tools in the market, the speed to market is dramatically reduced. In addition, API-first also makes it possible to add new services and technologies to applications without having to re-architect the entire system.
4. Ensures a good developer experience (DX) and reduces the risk of failure — Consumers of APIs are most often developers, and DX can strongly impact the success of an API. Well-designed, well-documented, consistent APIs provide positive DXs because it is easier to reuse code and onboard developers and reduces the learning curve. API-first reduces the risk of failure by ensuring that APIs are reliable, consistent and easy for developers to use.[5]

THE RISE OF APIs IN FINANCIAL SERVICES

APIs are the invisible ties binding together modern digital life. Whether you are turning on the lights with a smart voice device, paying for groceries with a swipe of your phone or using geolocation for delivery, APIs work in concert to make many customer experiences possible. At each point of connectivity, however, bad actors can exploit weak links in these digital handshakes and pry away your customers' personal information.[6]

Over the past five years, APIs have become the great enabler of digital ecosystems. Third-party developers can

now build applications and services around financial institutions (FIs), and FIs can select from a growing number of highly specialised FinTech offerings to enhance customer journeys, improve customer data intelligence and automate back-office processes. In other words, they are free to pick individual APIs to create new combinations of value. In the past, there was no way, for example, of plugging a fraud service bought from a third party into a payment process because everything was integrated into large, end-to-end monolithic systems.[7]

There are three typical strategies that a firm will incorporate with respect to their digital strategy: 1) a passive API strategy, which entails implementing compliance-based APIs for a limited number of use cases, and automation of high-cost sales channels. This strategy does not enable access to partners, ecosystems or new revenue pools; 2) an opportunistic API strategy that incorporates new partners and begins the journey to becoming an ecosystem over time, while gradually opening up new revenue pools or 3) an active API strategy that incorporates multiple partners in many ecosystems and allows increased access to new business models and revenue pools. Given that these ecosystems are not necessarily industry specific, it enables financial services firms to be an essential part of cross-industry digital marketplace platforms. Given that financial services products (providing cash transfers, financing and risk management) are required in most industries, the active API strategy can allow financial services firms to plug themselves into any industry's ecosystem and value chain, thereby increasing revenue pools.[8]

QUESTIONS TO BE CONSIDERED WITH RESPECT TO APIs

While APIs are a powerful driver of digital banking and transformation and allow new revenue pools to be accessed, there are also questions that need to be considered with respect to the data being processed through APIs:

1. Regulation with respect to data — Regulation, such as General Data Protection Regulation (GDPR), requires firms to have more control of their data than ever before. How will data that is being processed by APIs factor into this?
2. Fragmented protection — Data sources for API access are often implemented by different teams with different security needs, and the process is fragmented. How can the process be made standardised and consistent?
3. Legacy systems — Legacy data services may not have been built with API consumption either inside or outside an organisation being considered. How can these applications be brought into the digital age?
4. Onward distribution — When API data consumers receive data from an API, what is stopping them from distributing it themselves?
5. Who and where is your data — Once data has been processed by an API, do we know who has access to it? Is it being stored either inside or outside the organisation? If so, by whom and for how long?
6. Cross border data transfers — Are APIs sending data across geographic boundaries or from countries with data localisation laws? Does the data that is crossing have any legal restrictions or bans, such as Schrems II?
7. Auditability — Do you know what is happening with your data once it has been accessed by an API? What type of user entitlements are in place to protect it?

These questions suggest interesting challenges about what the potential risks are when it comes to digital transformation and the increased reliance on APIs, and they are the premise behind DataSecOps.

FRAUD DETECTION USING APIs

The finance sector can greatly benefit from an API ecosystem. Fraud detection is one of the top priorities for banks and FIs. Detection frauds can be addressed using machine learning models, which can be built to analyse the flow of data from data to and from APIs. Activities such as phishing and sniffing can be detected using APIs. They can also help to monitor and detect abnormalities in time series data. Using this, the API determines boundaries for anomaly detection, expected values and anomalous data points.

Considering the volume of transactions in the finance sector, fraud detection using APIs can be well thought of as a gold mine.

THE RISK OF APIs

As the API economy has been booming, so too have API data breaches: Equifax's breach cost US$1.14bn in 2019 alone. Capital One's hack saw 100 million customers affected at a cost of up to US$150m. Hostinger lost 14 million customer records in an API data breach, and an attack on Facebook's APIs affected 50 million accounts. Yet many companies still lack an API-specific security programme, and if they do it is ineffective.[9]

Organisations that utilise APIs to enhance their business offerings and grow revenues in the digital banking industry need to be aware of the possibilities for bad actors to take advantage of an API strategy. To date, such API attacks have largely flown under the radar of security professionals, who have been focused on other attack vectors. While 71 per cent of software engineers surveyed considered 'security' the most important factor in a decision to integrate with an API, which was tied for the top concern, many businesses, IT and security leaders fail to recognise that APIs inherently lack security, making them a prime target for the next big wave of cyberattacks.

Illustrating this point, the Open Web Application Security Project (OWASP), a non-profit foundation that works to improve the security of software, added 'under-protected APIs' to its proposed list of top ten application vulnerabilities in May 2020.

According to OWASP, many client applications do not communicate in the traditional architecture, talking directly to a database. Instead, most talk to an API. Protecting the API must be a priority for application security teams starting early in the development cycle. Developer teams must understand that calling the API, even when IP addresses are whitelisted, is obsolete at a time when bad actors use attack proxies capable of changing the request parameters, including those sent to the API.[10]

API security issues can be broken down into three categories[11]:

1. APIs can be used to expose sensitive data in a variety of ways:
   - Impersonating an application — Bad actors can reverse engineer an application and use this information to call an API by pretending to be the legitimate application. Given that the back-end servers may not be aware of this, they will freely interact with it.
   - Client-side phishing — Bad actors can redirect a legitimate applications API to a malicious site without the knowledge of a user. The user is then prompted to enter their credentials, which are then stolen.
   - Brute force and unauthorised access attacks — Bad actors can manipulate an applications API to attempt to gain access to the personal data of other application users from back-end servers.
   - Code injection attacks: Bad actors can use an API to inject code into an application's back-end servers. SQL injection, remote code execution and other exploitation attempts can be performed quite effectively through the APIs, much like traditional web application attacks.

2. API communications can be intercepted:
   - Insecure and unencrypted communications between an application and the back-end servers (via an API) may be visible on the open internet. Bad actors can either listen in on or intercept these API communications.
3. Denial of service attacks can take servers and applications offline through an API:
   - Single application, repetitive action — Bad actors can instruct a single application to make repetitive, resource-intensive API calls on the back-end servers, which will cause service outages or dramatically reduce the response time of the application.
   - Multiple applications, single action — Bad actors can instruct multiple applications to make a single resource-intensive API call on the back-end servers, which will cause service outages or dramatically reduce the response time of the application.

HOW TO MITIGATE A POTENTIAL API DATA BREACH

As API use has increased (today, APIs represent 83 per cent of all internet traffic by one measure), the need for a proven approach to API security has never been greater. There are important principles of API security that can help protect companies against what is becoming one of the most attacked digital surfaces on the internet.

As discussed earlier, there are multiple concerns while hosting large-scale APIs. Some of them are API overuse or abuse, lack of knowledge about who is using the API or the refresh of old APIs with new ones. Few of the solutions to the foregoing problems are authentication and authorisation, analytics and monitoring.

An API gateway is a crucial part of the API management system. They act as the major point of enforcement for API traffic. A good gateway will allow the authentication of traffic and the control and analysis of API usage. The API gateways enforce policies that control security aspects such as authentication, authorisation and traffic management. It is comparable to a security guard, protecting the underlying data. Put very simply, API gateway resource policies are rule lists that you attach to an API to control whether a specified user can invoke the API.

API gateway authentication is an important way to control the data that is allowed to be transmitted using an API. In all cases, it makes good sense to authenticate the end point before allowing it to transmit data via the API. This protects the organisation against malicious data exchanges and provides a layer of security. API gateways also implement industry-standard encryption and access control. While authentication practices can be susceptible to man-in-the-middle or brute force attacks, they have the potential to deter mass attacks and data breaches.

From the confluence of technologies at work today a new market segment called DataSecOps is emerging that directly addresses this issue. DataSecOps is a discipline that empowers software engineers, data scientists, governance risk and control, cybersecurity & operations teams to work together in a single application for safer and easier access, analysis, delivery and governance of data.[12.] The very crux of this domain is that the entire business and the associated processes must be involved in combating security issues.

The primary focus with respect to API protection is to create a culture of prevention and to enhance security, agility and speed without sacrificing the benefits of APIs. This is called 'Shift Left', or to find and prevent defects early in the software delivery process. The idea is to improve quality by moving tasks to the left as early in the life cycle as possible. Shift Left testing means testing earlier in the software development process.[13]

1. Data privacy should be simple — Currently, companies protect data at an

application level as it flows throughout the organisation; in other words, data is managed and controlled across hundreds, or even thousands, of applications and APIs — a costly, complex and inefficient process.[14]

2. Implement API protection at a proxy level — Implementing API protection at a proxy level with a centralised team ensures that there is consistency across the IT estate with respect to the protection. Having a central team enables continuity, which in turn helps members to become intimately familiar with the code and its capabilities. This also promotes reuse instead of creating new solutions. Fewer APIs translate into a smaller attack surface for bad actors. Full-time teams have the added benefit of understanding business processes and downstream systems, enabling them to make modifications to APIs quickly.
3. Implement 'as a service' offerings internally — Organisations need to shift their mentality from 'owning an operational service' to providing 'as a service' to internal customers. For example, if you are building an API gateway, then there should be an 'as a service' team that onboards development teams. This creates tested, secure and trustworthy protection that can be delivered at speed.
4. Train teams to best practices and install repeatable processes and standards — By training teams to best practices and by installing processes and standards, teams can Shift Left and create great products safely. By installing repeatable processes and standards, such as DataSecOps, you can help ensure developers stick to the best and most secure practices. DataSecOps principles should be followed and built into the IT workflow:
   - Instead of depending on manual intervention, data policies should be automatically applied to data attributes. The appropriate access, control and security need to begin at data level with the ability to revoke access — ensuring data is never mismanaged, misplaced or misused — in a straightforward process.
   - The principle of least privilege (PoLP) should always be applied in order to protect sensitive data and ensure only the correct, relevant information is provided to the right people. As such, the risk of over-distributing data should be eliminated.
   - The solution to unlocking the power of data is not just a 'technology' issue, but one that brings operations, governance risk management and compliance (GRC), business heads, and security together to facilitate policy automation and data access permissions. A seamless, collaborative environment between the administrators and the engineers who store, analyse, archive and deliver data must exist.
5. Productionise APIs — Because APIs are often built for integrations (eg websites and mobile application), they are sometimes treated as one-offs by development teams. It is a short-sighted mentality that breeds convenient designs, instead of effective productionised solutions. Productionisation increases the speed of consumption and liberates the organisation when it comes to the consumption of APIs. When there is a focus on repeatable and productionised processes, security becomes layered into the architecture.
6. Encapsulate to reduce complexity — By encapsulating all of the complexity for a capability behind the API, it provides the consumer the easiest interface possible. This makes the API not only simpler to use, but also more secure by hiding the explicit details of the underlying architecture.
7. Develop and automate pre-built/pre-approved components — Eliminate error and unnecessary work by using pre-built components with standardised

contracts and security mechanisms and by automating as much as possible. Automated activities should be pre-approved and identified as secure and best practices. Automation should also be implemented to ensure that proper security and practices are being used.

8. Enable automated governance and control — Incorporate automated governance and control processes that audit data usage to identify bad actors and poor practices as well as ensure that the standardisation that underpins your API security programme is as automated as possible and adhered to.

API security, fundamentally, requires a Shift Left in the culture of an organisation to build it around simplicity, repeatable processes, productionisation, governance and control, automation and DataSecOps principles as a practice.[15]

CHALLENGES OF DATASECOPS

Just as every coin has two sides, implementation of DataSecOps comes with its own share of challenges. While the aggregation of privacy enhancing techniques (PETs) is foundational to DataSecOps, the seamless incorporation and integration of all these PETs is quite challenging. Additionally, since the domain is still at a nascent stage, consistency in protecting data and API end points proves to be an area that requires careful attention.

CONCLUSION

Digital banking and the wider concept of digital transformation are changing the way that people interact with companies. At the heart of this change is the API. APIs are allowing firms to unlock new markets across multiple industries and to tap into new revenue pools. As with any enabler, there is another side that needs to be considered, namely the type of risks these new transformations introduce. These risks are both technical and non-technical and can be successfully mitigated through new concepts such as DataSecOps. When introducing the relevant risk and governance, the API economy can safely thrive.

References

1. Postman. (2020) '2020 State of the API report', available at: https://www.postman.com/state-of-api/ (accessed 4th March, 2021)
2. *Ibid*
3. (DataSecOps. (2020) 'DataSecOps', available at: https://www.datasecops.info/ (accessed 4th March, 2021)
4. Gontovnikas, M. (2020) 'The business value of API-first design' [Internet], *auth0*, 3rd September, available at: https://auth0.com/blog/the-business-value-of-api-first-design (accessed 4th March, 2021).
5. Wagner, J. (2020) 'Understanding the API-first approach to building products' [Internet], *Swagger—SmartBear*, available at: https://swagger.io/resources/articles/adopting-an-api-first-approach/ (accessed 4th March, 2021).
6. Powell, L. (2020) '10 Ways to prevent an API data breach' [Internet], *Forbes*, 13th July, available at: https://www.forbes.com/sites/forbestechcouncil/2020/07/13/10-ways-to-prevent-an-api-data-breach/?sh=778bf7dd43d0 (accessed 4th March, 2021).
7. Ernst & Young Global. (2020) 'How financial institutions can build a robust ecosystem strategy', *EY*, available at: https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/banking-and-capital-markets/ey-taking-a-strategic-view.pdf?download (accessed 4th March, 2021).
8. *Ibid*., ref. 4 above.
9. *Ibid*., ref. 3 above.
10. Curiel, J. (2020) 'OWASP's top ten update: Key changes your app sec team needs to understand' [Internet], *TechBeacon*, available at: https://techbeacon.com/security/owasps-top-ten-update-key-changes-your-app-sec-team-needs-understand (accessed 4th March, 2021).
11. Gates, S. (2021) 'Post-equifax: Why API security should be a priority' [Internet], *TechBeacon*, available at: https://techbeacon.com/app-dev-testing/post-equifax-why-api-security-should-be-priority (accessed 4th March, 2021).
12. DataSecOps. (2020) 'DataSecOps' [Internet], available at: https://www.datasecops.info/ (accessed 4th March, 2021).
13. Magowan, K. (2020) 'Shift left testing: What, why & how to shift left' [Internet], *bmc blogs*, 30th November, available at: https://www.bmc.com/blogs/what-is-shift-left-shift-left-testing-explained/ (accessed 4th March, 2021).
14. DataSecOps. (2020) 'DataSecOps: Statement of belief' [Internet], available at: https://www.datasecops.info/statementofbelief (accessed 4th March, 2021).
15. *Ibid*., ref. 3 above.