VERSION 1

The Imperative of API Modernisation in the Future of the API Economy and Open Banking

By: Andy Parsons (Digital Advisor, Financial Services), Brian O'Dwyer (Digital Advisor, Insurance) and Des Bohan (Senior Consultant, Digital, Data and Cloud)

Introduction

In the rapidly evolving landscape of digital finance and the broader API (Application Programming Interfaces) economy, the importance of well-designed, efficient, and secure APIs cannot be overstated.

As we advance further into an era dominated by open banking initiatives and API-driven business models, the need to modernise and optimise existing APIs has become increasingly crucial. This document explores why modernising APIs is not just a technical consideration, but a strategic imperative for future success in finance and beyond.

The numbers linked to this are astounding. The US and UK API management market size is estimated to be around:

$1.6 billion in 2023,
with projections to reach
$6.2 billion by 2028,
growing at a CAGR (Compound Annual Growth Rate) of approximately **20-25%** per year.

The Changing Landscape of the API Economy

The API economy has transformed how businesses operate, collaborate, and innovate. APIs have become the building blocks of digital transformation and the 'must have' capability, enabling companies to expose and consume services, data and functionality with partners, developers, and customers in a controlled and efficient manner. This has led to unprecedented levels of integration, innovation, and the creation of entirely new business models.

In the financial sector, the rise of open banking has been a game-changer. Regulatory initiatives like PSD2 in Europe and similar regulations worldwide have mandated banks to open up their data and services to third-party providers through APIs.

This has paved the way for fintech innovation, improved customer experiences, and increased competition in the financial services industry, **85%** of Insurance CIOs in Gartner's 2024 CIO and Technology Executive survey are increasing investment in APIs and integration technologies in order to lay the foundations for future business models. Cloudflare's 2024 API Security and Management Report reveals a striking statistic that **57% of internet traffic is now API requests.**

The US and UK markets are ripe for increasing reach with embedded journeys and the API economy.

Why Modernising APIs is Becoming Critical

1. Evolving Technical Standards

As the API economy matures, and the regulatory landscape continues to evolve technical standards, APIs that were designed even a few years ago may not adhere to the latest standards for security, performance and interoperability. Modernising APIs allows organisations to implement the most up-to-date protocols, data formats and authentication methods, ensuring they remain futureproofed for emerging technologies and platforms, backwards compatible and regulatory compliant.

At Version 1, we expect to see continued standardisation across the industry, potentially leading to more interoperable "super APIs" through the upcoming Regulations like PSD3 and PSR.

2. Scalability and Performance Demands

The last decade has seen an exponential growth in API usage, particularly in Open Banking scenarios since the PSD2 regulation and with the rise of the FinTech ecosystem, putting unprecedented strain on existing API infrastructures. We know from the work with our customers that the APIs that were originally designed for limited internal use struggle to handle the volume and complexity of requests in an open ecosystem.

Modernising APIs with scalability in mind, using microservices architectures, for example – can significantly improve performance and reliability under high load as well as delivering the increased payloads of modern demand. We expect to see even greater emphasis on real-time capabilities to support instant payments and decision-making as the new reality of payments transformation.

3. Security Imperatives

We are in an era of increasing cyber threats, data privacy concerns and stability where the security of APIs is paramount, especially in the financial sector with significant financial, reputational and regulatory risk associated with data breaches and non-compliance. Older APIs have vulnerabilities or may not incorporate the latest security practices, modernising APIs provides an opportunity to implement robust and futureproofed security measures, including advanced encryption, better access control, and improved monitoring and logging capabilities.

Older APIs will also use basic authentication or simple API keys, which are more susceptible to interception and theft, lacking support for modern, more secure methods like OAuth 2.0, JWT (JSON Web Tokens), or multi-factor authentication. Additionally, Older APIs might not use HTTPS by default or use outdated SSL/TLS versions that lack the support for the latest encryption standards enforced by Regulations like PSD3 or PSR, therefore making data transmissions more vulnerable to man-in-the-middle attacks.

4. Enhancing Business Model Flexibility

The API economy is enabling new business models and revenue streams. APIs that were originally designed for internal use or basic data sharing may not be well-suited to support complex pricing models, usage tracking, or differentiated service levels. Modernising APIs will allow organisations to build in the flexibility needed to monetise their APIs effectively, adapt to changing business requirements and evolve new business models leveraging modern composable architectures.

As AI and Machine Learning become more prevalent in financial services and other industries, APIs need to be designed to handle the unique requirements of these technologies. This might include support for large-scale data processing, real-time analytics, or integration with AI models.

5. Increased capability to handle modern payloads

Larger payloads take longer to transmit over the network, this directly increases the time between making an API request and receiving the response. For real-time or near real-time applications, this increased latency can significantly impact user experience, business decisioning and system responsiveness, this is typically where we see the most dropouts in a journey.

Both the client and server need to process the entire payload. Larger payloads require more CPU cycles and memory to parse, validate, and handle the data. This can lead to slower response times and increased resource utilisation on both ends. Modern APIs can reduce latency while driving operational efficiencies in the handling of these increased payloads.

Best practices for API modernisation

To maximise the benefits and minimise the risks of API modernisation, we consider the following to be best practices:

1.

Start with a Comprehensive API Audit: Assess your current API landscape to identify priorities for modernising:

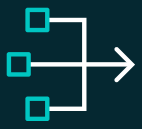
- We have used the 'Richardson Maturity model' which is one approach that is focused on looking at existing APIs
- For one customer, we looked at breaking down the monolith. This was where we had a single MVC application for Online self-service which we replaced with an angular application served by a series of APIs. We use a domain driven design approach to break the monolith up into bounded contexts
- For another customer, in addition to creating green field applications and APIs, we have also taken existing SOAP web services, required to expose functionality externally to third parties, and rewritten them as containerised spring boot rest APIs to improve scaling and performance

2.

Adopt API Design-First Approach, we recommend starting with clear API specifications before implementation to ensure consistency and quality. This means a clear understanding of:

- The purpose of each HTTP method (GET, POST, PUT, DELETE, PATCH ...) and which ones are supported
- The naming conventions for data fields camelCase, date, indicator fields
- The strategy for data formats e.g consistent format for date fields
- HTTP return codes and what they signify (200, 201, 204, 400, 500 ...)
- Error response format and ensuring it's consistent

3.



Consider a scalable application architecture:

- Adopt the 12-factor app methodology (https://12factor.net/)
- The twelve-factor methodology can be applied to apps written in any programming language, and which use any combination of backing services (database, queue, memory cache, etc)

4.

Implement a Versioning Strategy: Use clear versioning to manage transitions and maintain backward compatibility.

- Adopt a semantic versioning (https://semver.org/)
- We have tended to use URI versioning where the version number is in the URI there are other options such as header (special header field holding version) and body

5.

Focus on Documentation: Provide comprehensive, clear documentation to facilitate adoption of new API versions:

- We have used springdocs (https://springdoc.org/) to automate the generation of API documentation. (Note: Specific to Java Spring framework)
- This is more considered a code first approach as we had a detailed understanding of the API requirements

6.

Monitoring and Observability: Exposing Actuator endpoints to expose operational information about the API:

- Health
- Metrics
- Standardised logging approach to allow streaming of in consistent format to be interrogated
- Tracing to allow transactions to be traced across multiple APIs/ Application

7.

Embrace Automation: Implement automated testing and deployment processes to ensure quality and speed up the rewriting process. You want to be able to drastically reduce or eliminate the need for manual testing:

- Code Quality: Use static code analysis tools to identity poorly written code, security issues or Maintenace issues (SonarCloud)
- CI/CD pipelines that execute various testing phases and reports on the results:
- Unit Testing
- Performance Testing (JMeter)
- Functional Testing (Cucumber, Gherkin)

8.

Prioritise security: Make security a core consideration in the API rewriting process, not an afterthought. Again, with automation shift those security checks left in the development process. Adopting a DevSecOps approach:

- Container scanning (Twistlock, MS Defender for Containers)
- OSS (Open-Source Software) inventories. Modern API's leverage open-source libraries. You need to be aware of what they are and what licensing types are in place (WhiteSource, Synk, BlackDuck)
- Security testing (Penetration Testing with OWASP ZAP)

Conclusion

In the evolving landscape of the API economy and open banking, standing still is not an option. Organisations that proactively modernise and optimise their APIs will be better positioned to capitalise on new opportunities, meet regulatory requirements, and deliver superior experiences to their customers and partners.

While the process of modernising APIs can be challenging, the long-term benefits in terms of scalability, security, and business agility make it an essential investment for the future.

As we move forward, the ability to rapidly evolve and improve API infrastructure will become a key differentiator in the market. Those who embrace this challenge will be the ones shaping the future of digital finance and beyond. The time to start modernising your APIs is now – the future of your business in the API economy depends on it.

For more information, please visit version1.com

VERSION 1

Andy Parsons (Digital Advisor, Financial Services)

Brian O'Dwyer (Digital Advisor, Insurance)

Des Bohan (Senior Consultant, Digital, Data and Cloud)