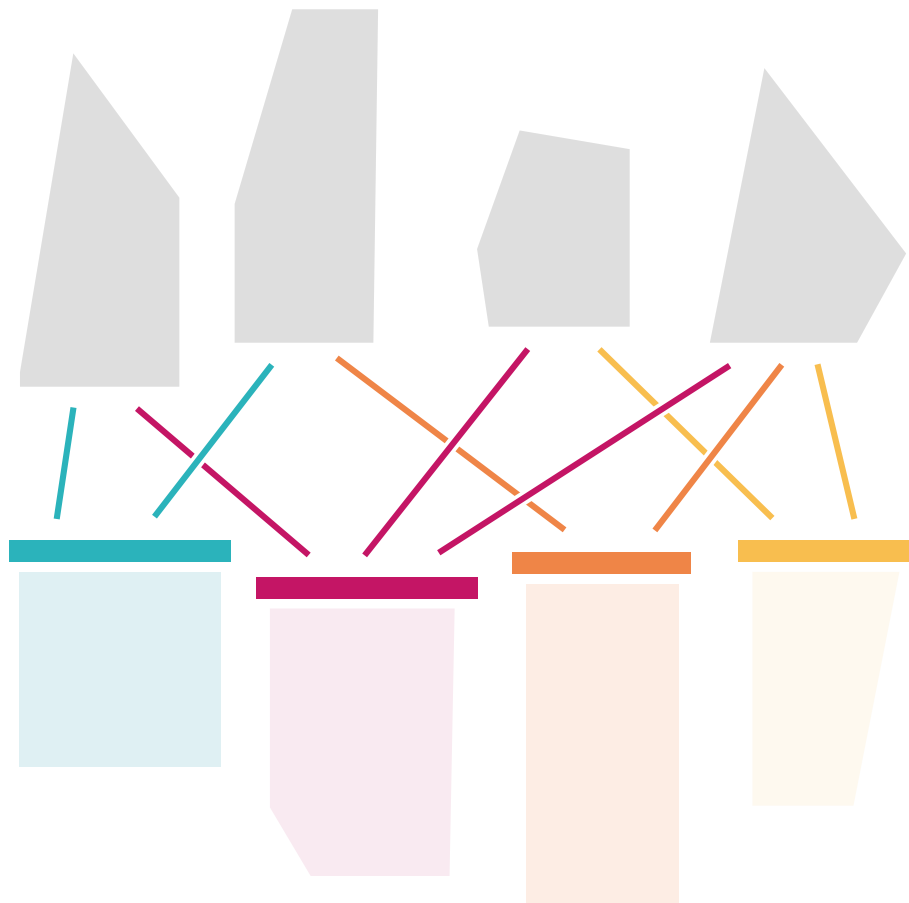


Open your eyes to APIs

A non-technical exploration of their oft-untapped potential

Embracing and understanding Application Programming Interfaces (APIs) across a whole organisation, rather than just within technical teams, is an important part of realising the full value of digital transformation. It is critical to move on from short-sighted API projects driven by technical necessity to strategically aligned efforts that unlock broader business benefits and opportunities. Organisations unable to make this shift are being dragged down by their legacy estate and left behind by the digital-native world.

A white paper by Graham Odds



Contents

Let machines talk to machines	3
Typical state of play	4
An API done well	6
<i>Case study:</i> Embracing your part in workflows	6
<i>Case study:</i> Digital transformation by regulation	9
The benefits of "proper" APIs	10
<i>Case study:</i> The value of unknown opportunities	11
Want us to help open your organisation's eyes to APIs?	12

Let machines talk to machines

Application Programming Interfaces (APIs) are a broad technical construct that have existed for almost as long as digital technology itself. They provide defined ways of exposing information, functionality or services, conceptually very much like User Interfaces (UIs). That is, an API or UI dictates what you can see and do with a system.

However, where UIs enable humans to interact with digital systems, APIs enable digital systems to interact with other digital systems – whether that be on the same device or across networks. In other words, people talk to machines using UIs and machines talk to machines using APIs.

While APIs are nothing new, their technical nature means that their purposes and capabilities are rarely understood or championed by non-technical people in organisations, despite their huge significance in our digital world. As a result, the development of APIs is typically driven by narrow-sighted technical necessity rather than broader strategic intent, meaning these efforts rarely unlock the true value they can provide if their business potential is fully embraced.

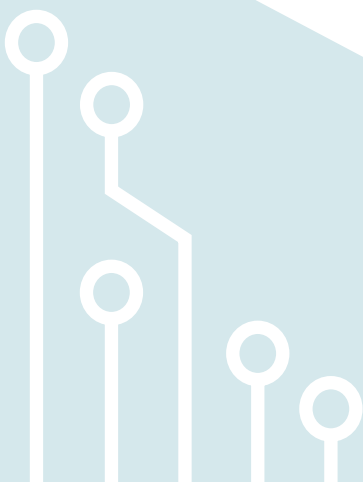
Furthermore, we frequently see naive digital transformation efforts – ones grounded in unquestioned historical assumptions of the need for human actors to participate in processes and workflows – leading to technology merely being a change in medium rather than an opportunity for reinvention. Think of how absurd it is to be emailed that form you are required to print out, fill in by hand, scan and then return by email...

While we must not leave behind [those who are not effectively engaging with the digital world](#), we must not use that as an excuse to take a lowest-common-denominator approach. Digitally capable organisations and individuals – the number of which is only ever increasing – should be offered digital services in forms that best enable them to further their interests; with increasing frequency, this is by means of an API rather than or as well as a UI. APIs can stream real-time information with precision and accuracy at scales no human can process. They can execute actions at rates faster than humans can react without tiring.

We should not fear or fight these capabilities, and we should not allow them to be ignored or dismissed through ignorance. Effective digital transformation is grounded in appropriately embracing the “superpowers” of the two available actors in processes:

Humans bring free-form creativity and thought at the expense of speed and scale; while machines bring narrow hyper-efficiency and accuracy at the expense of adaptability.

The rest of this paper is intended to help you better understand how to create the foundations to exploit machines’ “superpowers” to their utmost and realise a whole raft of latent business potential.



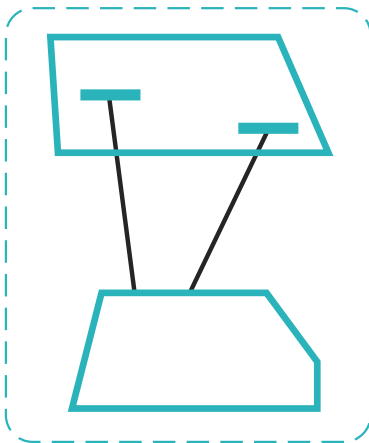
Typical state of play

APIs of various types exist at practically every layer of software, from the layer where code interacts with the operating system of the machine it is running on, right up to where websites and mobile apps connect across networks to their backend services and databases.

When people talk of integrating system X with system Y, they are typically referring to the technical act of hooking up the APIs of those systems to one another. In many respects, this is little different from how websites or mobile apps speak to their backends. When referring to APIs in this paper, we primarily mean these higher level, system-to-system APIs. It is an understanding of the potential of APIs at this level that is likely to yield the most benefit for your organisation.

It is important to understand that if you work in a large, long-standing organisation, there will undoubtedly be APIs of this sort in use and actively being developed. To better understand why they may not be on the corporate radar, we present a rough classification of the typical APIs that exist.

As already noted, APIs are conceptually comparable to UIs. Draw on your experience using similarly classifiable UIs to deepen your understanding of how they apply to APIs.



Internal APIs

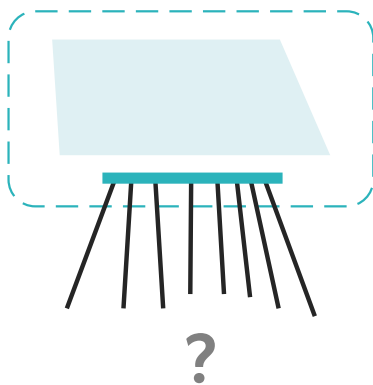
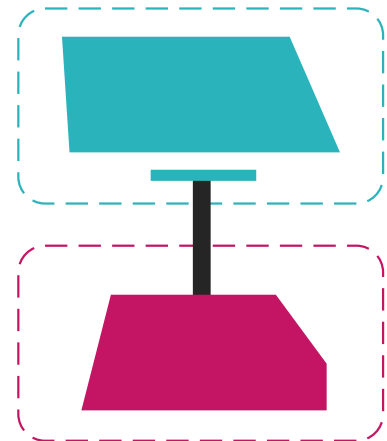
Internal APIs are the most prevalent class of organisational APIs since, by technical necessity, they are present in any integration between internal systems. As a result of growing and appearing organically, they tend to be numerous and messy and lacking in strategy or technical consistency. They typically form very narrow point-to-point connectivity between specific elements of systems to fulfil the specific integration needs of a particular point in time, resulting in brittle, assumption-riddled implementations. Much like those (typically grey-screen) internal UIs you have undoubtedly been compelled to use at some point, they are often esoteric, heavily reflecting any technological and conceptual idiosyncrasies of the underlying system.

These APIs, like their UI counterparts, are mostly developed on a shoestring, with a “just get it done as quickly as possible” drive. Consequently, they are rarely created to good standards, often inadvertently introducing security vulnerabilities or unforeseen load elsewhere. There is frequently little to no discoverable documentation of their existence, let alone documentation on how they work or the assumptions under which they were developed. Furthermore, these APIs are rarely identified as the internal “service” they actually are, meaning they are, at best, maintained as a reluctant side-of-desk activity and quickly become a frustrating and costly burden.

Partner APIs

Partner APIs often evolve from existing internal APIs that are made available to a small number of “known” external consumers. Typically, they take a more system-to-system form than the point-to-point integrations of internal APIs. However, they are still developed around a detailed understanding of both systems’ potentially arcane needs and behaviours, and corresponding assumptions. Comparing these to UIs you likely have experienced, think of those line-of-business tools that are a million miles from the quality of the digital tools you use in your personal life but are tolerable because they – just about – get the job done.

There is commonly some form of commercial or contractual agreement underpinning partner APIs, meaning that there are some kind of service expectations and operational efforts surrounding them. Similarly, there will likely be some form of documentation if you know where to look, but it tends to be very low-level and out of date as a result of being generated as a one-off exercise when the integration was initially established. Ultimately, the one-off nature of many partner APIs routinely locks both parties into a stalemate of change due to their bilateral overheads.



Public APIs

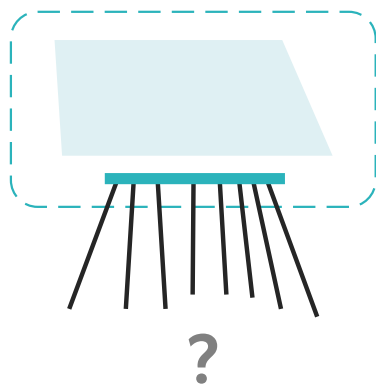
Public APIs are those that are publicly available – commercially or free – for any third party to use (such as all the case studies elsewhere in this paper). By their very nature, they are recognised as a formal service provided by your organisation. Consequently, they tend to be readily discoverable both publicly and internally, with thorough documentation. In this way, they are similar to UIs for publicly available websites or apps, and they typically receive similar care and attention.

In most cases, public APIs attempt to create as “easy” an interface as possible for consuming systems, both in technical and domain terms. That may be by dealing with the idiosyncrasies of any underlying systems internally or by using common technologies and standards to make the system as accessible as possible. The ultimate aim is to ensure the service is as universally and easily usable as possible, regardless of the consumer system.

Consequently, public APIs are developed with a defensive mindset. No assumptions are made about any connecting systems, resulting in services that robustly handle overuse, misuse or abuse, whether intentional or not. In these respects, the comparison to consumer websites and apps holds. When was the last time you had to worry about what’s happening under the covers of your email client? When was the last time you accessed your online banking without two-factor authentication?

An API done well

As with all products and services, APIs should be tackled with a user-centred mindset. By starting from a consumer perspective, you can initially focus on establishing a clear vision of what the API is trying to achieve for people. Subsequently, you can explore how best to achieve that vision as the producer of the service, while taking into consideration any internal constraints or ambitions.



All too often this approach is reversed due to the technical nature of APIs and the typically narrow strategic backing they receive. More commonly, the consumer perspective is barely even acknowledged.

Broadly speaking, public APIs provide the aspirational blueprint across all categories of APIs, as public APIs are effectively offering a service “blindly” to users – that is, without detailed knowledge of who they are or what purposes they have for consuming the services – who must be wooed rather than being forced to use them. This necessitates the user-centricity that is typically lacking from purely technical undertakings and lays the foundations on which true business value can be built.

Case Study

Embracing your part in workflows

[Stripe](#) is a business that provides “payments infrastructure for the internet”. Whether you realise it or not, if you have made any payments online in the last few years, there is an extremely good chance Stripe has played a role in that transaction. Stripe recognises that while payments are a complex problem, from an end user’s perspective they are only a small, functional component of a broader experience. As such, Stripe has resisted the temptation to impose itself on the end user – instead, it has ensured its business is primarily one of providing APIs for others to integrate into services. Contrast that with the occasions you have been redirected into the worlds of PayPal, WorldPay or SagePay.

Despite largely not owning the end-user interaction, Stripe’s most recent fundraising [values the business at \\$60 billion](#). Stripe treats its key points of interaction, its open APIs, as true first-class citizens, with a rounded user experience beyond what most technologists can dream of. Its APIs’ capabilities are intuitively representative of their respective domains, accessibly documented for both technical and non-technical audiences, and easily available for exploration and testing. This care and consideration is why people flock to Stripe over and above all its competitors.



The consumer perspective

Adopting a consumer lens, you quickly realise that, as with most services, different users will be approaching you at different stages of understanding and intent. New potential users need to gain a high-level understanding of what your service is (and/or is not!), while those who are “bought in” will want to get into the details of how to use it, try it out or actually use it in anger.

Across the spectrum, your objective is to be as “accessible” as possible, whether that be in domain or technical terms. Good APIs, like good UIs, provide a seemingly intuitive experience and help you achieve your goal. Poor ones require you to battle to understand the interface you are presented with or the underlying idiosyncrasies of the system you are trying to interact with. With both UIs and APIs, users are familiar with particular conventions and patterns, and deviation from those tends to cause frustration or confusion unless users are sufficiently handheld through the process.

Slicing the user base on a different dimension, there will inevitably be multiple user types with needs and desires that overlap or contradict each other. You will need to identify, navigate and prioritise these. For example, beyond the common needs and desires for robustness, security and reliability, consider the following API user archetypes (this is where APIs can really start to deviate from UIs!) and delve deeper into them within the specific context of your goals:

Regardless of whether the consumers of your API are internal or external to your organisation, being attentive to their needs and desires pays huge dividends, greatly improving the value and longevity of your service.

Developers

The individuals doing the technical work to hook up your API to another system are often the only identified “user” of the API. At the very least, they need to be able to: find and evaluate the service; understand how to interact with it in detail (ideally in a familiar, conventional way); and, easily develop and test against the API without affecting anything “real”.

Ops

The moment your API is being consumed by another live system, there is an associated on-going operational interest in that dependency. Support/maintenance teams need live visibility of the status of your API and, potentially, related operational metrics. Is your service down? When was the data last updated? What is the state of downstream services you are providing access to? Regardless of whether such insight is consumed manually or automatically, it is a critical foundation for anything built on top of the API you are providing.

Systems

While not a human user, the consuming system of the API also has “needs” and “desires”. In order to exploit its speed and precision “superpower”, the system’s interactions with the API service should have as few technical overheads as possible (something which can be at odds with developers’ desires for human-readability). Furthermore, its mode of interaction should match the service intentions. For example, real-time information should be “pushed” by the API rather than requiring the consuming system to “ask” constantly whether there is anything new.

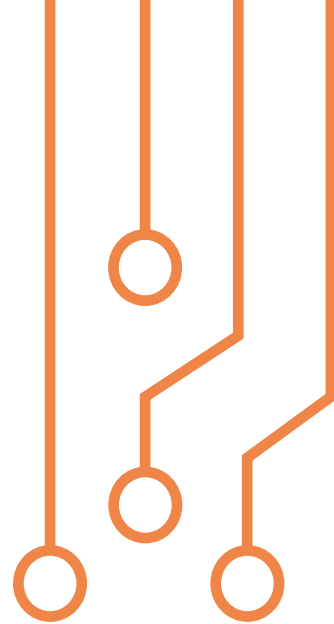
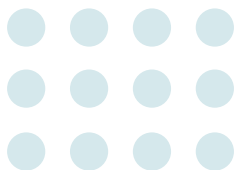
Non-technical users

Business analysts, UX designers, product managers and the like are frequently overlooked as users of APIs. However, such people play critical roles in shaping the products and services your API may be able to underpin or enhance. Like developers, they need to be able to find, evaluate and understand the service you provide, but unlike developers they will find walls of impenetrable technical detail extremely off-putting and unhelpful.

The producer perspective

Beyond simply creating purposeful, valuable services for users, an API producer must be concerned with the long-term viability and maintainability of that service. Fortunately, these concerns align well with what good looks like from the consumer's point of view. APIs, like good UIs, should be developed with a well-rounded view of robustness at their heart. In the short term, they need to be appropriately secure, scalable, fault-tolerant and resilient, and free of assumptions of any particular type of behaviour from consumers. In the longer term, they need to be expandable, maintainable and as loosely coupled to their underlying implementation as possible in order to support the iterative improvement that all on-going services warrant. If these aspects are not appropriately tackled, an API, like any service, will age and wither gracelessly, likely adding to the technical and non-technical foibles of your digital estate.

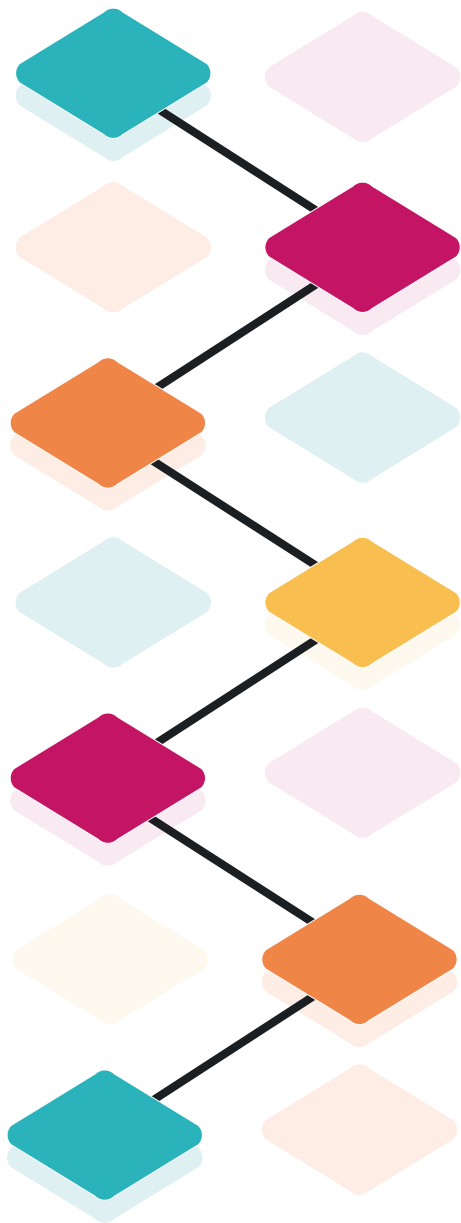
Many of these internal requirements cannot be meaningfully tackled in isolated cases – rather they require broader strategic efforts to be achieved in earnest. On the plus side, such efforts typically also greatly accelerate the creation of APIs, ensure higher quality and remove many of the technical distractions “for free”. Automated provisioning, deployment, documentation, testing, versioning and the like all help projects to focus attention on their core value – instead of the bureaucracy and painful technical hurdles that all too often demand most of the attention of API projects.



First-class citizens

All of this might sound like an onerous undertaking, and may not feel warranted for the immediate value offered by internal and partner APIs. However, we are not suggesting that all APIs should be implemented to the highest expectations of the public API blueprint. Rather, we strongly recommend anchoring your approach aspirationally to those standards and then taking informed, conscious decisions where and how to pare back from that ideal. For example, the Department for Work & Pensions' ability to react and respond swiftly and with clarity of intent during the COVID crisis was [hugely improved by a simple API catalogue](#); no need for some beautifully polished artefact! This overarching approach ensures the enormous strategic potential and capability of good APIs is not compromised in the way it naturally tends to be by the short-sighted, quick and dirty approach of the typical technically driven API project.

Treat APIs like the first-class citizens of your digital estate that they deserve to be. Ignore or abuse them and they will make you suffer; show love and care for them and they will help you prosper.



Case Study

Digital transformation by regulation

In January 2018, the Competition and Markets Authority brought in the [Open Banking](#) initiative as a regulatory driver to increase user-centred competition and innovation in the market. It requires UK banks to make customer information available to regulated third-party providers through prescribed (but evolving!), standardised, domain-focused APIs that place data security and customer consent at their heart. While the implementation is a detailed technical one, the overarching driver is improving user (and thereby longer-term business) value.

Like all other major banks, [NatWest Group \(then RBS\) started work towards this regulatory requirement in 2016](#). They quickly realised that while they had disparate relevant internal APIs, these were: hard to locate; esoteric in both technical and domain terms; and far from meeting the expectations of any public API, let alone those prescribed by Open Banking. Over the years since, the programme of work has not only met and exceeded regulatory requirements, but has also resulted in a significant cultural shift across the organisation from dismissive reluctance to committed ambition. It has established a mature, organisation-empowering approach to APIs for both internal and open consumption, and has opened whole new commercial opportunities for the bank. And these opportunities can be exploited at much greater speed, thanks to increased business responsiveness born of the resultant team empowerment and the bank's wealth of new API services. Take a look at [bankofapis.com](#) to see what this looks like!

We believe that Open Banking is but the first of many formal, sector-wide pushes towards open ecosystems. For example, various activities from Ofgem look set to drive similar changes in the energy sector over the coming years. All public services, utilities and markets seem likely to follow suit in the coming decade.



The benefits of “proper” APIs

When APIs are created purely to address technical necessity, they invisibly tie you down. You will end up having to unpick a tangled mess of impenetrable integrations at every turn, adding to any legacy burden. When APIs are used to their potential, they liberate you. You can think freely about your organisation and the value it can deliver, keeping pace with customer expectations in a world that is changing ever more rapidly around you.

Providing business agility and accelerating innovation

If you treat APIs like a first-class citizen, they encourage you to construct software in a better way. Rather than building monolithic single-use software, you build modular software for reuse. Think LEGO blocks rather than die-cast toys. You can swiftly rebuild to change direction, without throwing away all your good work, breeding broader confidence and agility across the organisation.

And the more “proper” API services your organisation creates, the more each new product or feature you develop gets a head start, with some of its technical challenges already solved. You are developing a silo-less platform for innovation that accelerates your time to market, and focuses your energy on testing and validating your proposition rather than creating solutions to technical challenges that have been solved already.

Giving you choice (and the ability to change your mind)

APIs done properly make sure all their implementation details (and potential nastiness) are hidden from consumers. The resulting neatness and containment of any choices around those details significantly reduces any associated risks and knock-on effects. As a result, proper APIs allow you to shape a more agile, incremental strategy. They can form a key element of a new modular approach to your technology landscape that avoids lock-in. In this infrastructure, there will eventually be no such thing as a legacy system as you understand it today – but rather a collection of much smaller services that can be updated, replaced or retired quickly and inexpensively in line with the evolving needs of your organisation.

With this strategy, technology problems and decisions are smaller and can be made in isolation. You can choose to continue to derive value from your legacy systems without that impacting your ability to incrementally address other priorities. The API services you develop can connect legacy systems to new services and applications that offer the modern user experience your business and customers expect.

Focusing your efforts

Frequently, organisations try to be all things to their customers – and what’s more they strive to anticipate and stay ahead of customer demand. With the digital revolution, this is simply no longer tenable; the speed of technological innovation is too great – as is the attendant evolution of customer expectations.

By embracing APIs for what they are, you have the option of focusing your energies on those parts of your proposition where your organisation provides unique value. You do this by opening up secure API services to a wider ecosystem of third parties who can reach customers you can’t and can meet customer needs that you would struggle to address. This may require a fundamental shift in organisational mindset, and a reconception of your place in the world. But the rewards can be vast.

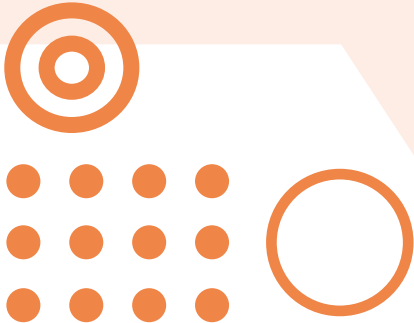


Case Study

The value of unknown opportunities

For over ten years, [Transport for London](#) has been making significant amounts of its non-personal data – such as timetables, service status and disruption – available for anyone to use free of charge through public APIs. Its original hope was that partners would produce new products and services and bring them to market quickly, thereby extending the reach of TfL’s own information channels.

Deloitte undertook [an independent review to understand the economic value and social benefits of this approach](#). It quantified that the release of open data by TfL was generating annual economic benefits and savings of up to £130m for travellers, London and TfL itself. These benefits and savings range from: saving time, effort and cost for direct and indirect travellers; producing a virtuous cycle that results in better information for travellers and TfL alike; creating high-value jobs; and, adding gross value to the economy. The public API approach has allowed TfL to focus on its core business and capabilities rather than having to produce and support an ever-expanding portfolio of digital apps. Furthermore, it has resulted in the organic creation of whole new commercial opportunities.



Want us to help open your organisation's eyes to APIs?

At Scott Logic, we have deep expertise in APIs and many years of experience in helping organisations like yours drive value from them.

If you'd like to discuss how we can help your organisation tap into the full potential of APIs, we're always happy to chat.

Please contact Graham Odds on:

+44 333 101 0020

graham@scottlogic.com



6th Floor, The Lumen
St James Boulevard
Newcastle Helix
Newcastle upon Tyne
NE4 5BZ

+44 333 101 0020

[scottlogic.com](https://www.scottlogic.com)

Jan 2024 © Scott Logic Limited. All rights reserved.