

# API-Led Transformation



## Introduction

When you think about how businesses were built in the past, it's easy to understand why they are so slow to adopt new technologies. In the past, many companies created their own software and hardware, which meant that there was a lot of time spent making sure that everything worked together.

Nowadays, companies realise they can save time and money by using Application Programming Interfaces (APIs) instead of building everything from scratch. From an architecture perspective, this decoupling allows companies to evolve their applications independently from one another, improving both agility and ability to scale. By using APIs, businesses can quickly access business-critical data and build resilient infrastructure in times of change. This shift towards API-led connectivity is exciting because it allows companies to future-proof their IT infrastructure.

## API Strategy and Approach

APIs break down the large self-contained and independent applications into microservices. They allow data and application components to be exposed to the wider enterprise including customers or partners, primarily using Internal, Open and Partner APIs. In the Digital Factory type of Operating Model, this allows pods to innovate without worrying about dependencies with other pods. This approach allows businesses to structure these building blocks across different systems, processes and experience layers to achieve greater organisational agility and control over their technology strategy.

API-led connectivity is not just a technology trend. It is a new way of doing business. Implementing an API strategy involves planning, designing, and managing the development and deployment of APIs to achieve specific business goals. As part of the company's API Strategy, you will need to design and document your APIs, Implement API Management and reviewed your security and authentication mechanisms. A key area to address is developer engagement and support, as you need to ensure that all teams now begin to expose their data and functionality through APIs and the communication is now through these interfaces. How they are implemented is down to your design work but the insistence that they are used must be communicated and governed accordingly.

Once you have an API Strategy in place and the support of all Stakeholders, you need to lay the technological foundation and define the required standards and policies. The same granularity that microservices bring to application design can be brought to integration, with similar benefits. Therefore, many companies begin their API journey by migrating the services from legacy ESB to microservices accessible by APIs. Alongside this they also tackle the common platform components that may not be directly business related. This foundation will stimulate the API programme and then you can begin to prioritise the microservices based on the overall digital transformation programme of work. This will provide insight into how you will need to decouple each IT system or platform and understand the cost of doing so.

## Making Your API Programme A Success

The work around Operating Model and Governance will now become crucial to the success of the API programme. How you organise platform pods to focus on building the APIs will be important to achieving the goals. Alongside this the necessary communication and education will be deployed so there is awareness across all developers, partners, suppliers and business owners to drive API adoption. The choice of technology will be made to enable both API Management and the Developer portal. Without these foundational steps, the budget to deliver and the right talent in place, the programme will stall. Once you have established the initial API's, demand will increase, and pace of delivery will become more of an issue.

## ServiceNow API-Led Connectivity

An excellent example of how API-led connectivity can improve business operations is when it comes to accessing business-critical data. Many companies have now adopted ServiceNow to support the delivery of critical business and I.T. processes. Still, in many cases, enterprise customers have other applications that need access to this data. In some cases, new applications can be built using the existing ServiceNow APIs, reducing the need to duplicate or replicate data in the enterprise.

ServiceNow API integration tools integrate important resources from both inside and outside of the organisation, including IaaS, SaaS, ERP, on-premises applications, legacy systems, data sources, and databases. This makes it possible to address business problems more quickly while also supporting extensive integration, applying automated workflows and continually making service improvements.

## ServiceNow Layered Approach to API Development

By Adopting the layered approach organisations can enhance their business operations, streamline processes and achieve better outcomes:

### System APIs:

A system API layer provides an interface to one or more underlying systems of records. These APIs can be further subdivided based on the business domain they represent, such as a product or a customer system.

A System API for a domain can contain resources that are accessed via the API and methods like GET, POST, PUT, and DELETE (CRUD) and the related schemas JSON and responses (200, 400, 500, etc).

An example of a system layer API would be to Create, Read, Update, and Delete an object inside of ServiceNow, for example, creating a new contact record or updating a CMDB record.

### Process APIs:

The Process layer orchestrates various data by calling multiple System APIs. Process APIs aim to encapsulate the business processes for a specific service independent of the source systems from which the data originates.

Process layer APIs include:

- **Collation API:** This API takes care of merging incoming data into a single stream.
- **Divide API:** This API divides incoming data into smaller streams.
- **Filtering API:** This API filters out specific messages based on certain conditions.
- **Routing API:** This API routes messages to different destinations based on certain conditions.

An example of a Process Layer API in ServiceNow would be to use the Automation Center API to deliver a new workflow across two different APIs or external systems. For instance, this could involve moving documents or extracting content from emails or forms.

### Experience APIs:

The Experience Layer is the topmost layer of the API stack. It is what the end-user sees. This layer provides an interface to expose data, processes and experiences across various consumer systems with different formats.

For example, our business scenario has a purchase order API that exposes data in JSON format. But we may have another consumer application which only accepts CSV or XML format. In this case, we need to transform the JSON format into CSV or XML format before exposing it to our consumer application.

An example of an Experience layer API in Service Now, is the Contact API which could be integrated with a chat application that could be a new digital product or solution.

## Conclusion

An API-led connectivity approach to delivering new services ensures you are on time and within budget, but you will have built the reusable assets by following this approach. The other key benefit of using reusable assets in the API-led approach is the ability to provide built-in visibility, compliance and governance standards. This enables you to move fast on your new and existing API projects.

The expectation for delivering digital transformation has dropped from years and months to weeks and, sometimes days. The delay in developing a new service might be the difference between gaining a unique market opportunity vs missing out and letting your competitors into this specific market.

For any questions, please call: 0344 846 3333

Option 1 - Public Sector | Option 2 - Sales

Insight 

[uk.insight.com](https://uk.insight.com)