



## Introduction

“In the long history of humankind (and animal kind, too) those who learned to collaborate and improvise most effectively have prevailed.”

Current trends in mobility and collaboration driven business models make Darwin's erstwhile view on evolution still relevant. As per latest assessment by Cisco, data traffic via smart phones has increased 18x in the last five years. And this change is clearly visible in our lifestyle as well. Today, to call a cab, people in India are not looking at roads and waving hands but they are tapping away on their smartphones to use Ola and Uber apps. Groceries, medicines, doctor connect, ride sharing, maid services are some of the examples of App based services that are now part of our day to day life which we had not imagined few years ago. This is evolution! and only those who are ready to collaborate and are ready to change can survive.

Welcome to the world of APIs - a step towards collaboration with new age technology, a new step towards next generation of application development.

In business seminars experts are talking about the benefits of new service-based product designing. In technology fests, tech gurus are talking about API-based frameworks, business analysts and application managers are experimenting with proof of concepts (POC) and are able to create some APIs. But, over time they have realized that

merely creating APIs is not enough. They need a strategy around creating APIs and have to transform their existing design and development model to reap the real benefits of this approach. This paper discusses such new developments and possibilities, and attempts to establish a thought process that could help organizations plan their API strategy.

## Challenges and Solutions

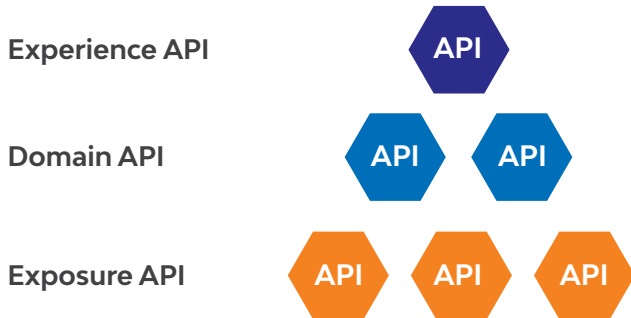
Organizations are gradually realizing the benefits of an API economy but there are many challenges and myths. Let's discuss about these challenges:

### Difference between product designing and API designing

It is a myth that product designing must be done by business analysts and API designing by developers. With growing digitalization of services, IT and business are coming closer and now business has started thinking like an IT expert. APIs are the gateway and are the product, it cannot be decided by IT alone. It's a joint effort - initiated by business analyst/product owner and implemented by IT developers.

## Grouping/Packaging APIs together

Grouping or packaging of APIs gives product designers and business analysts the flexibility to create multiple versions of the service offering. This allows them to design customized products for targeted customers. API experts recommend three-layer architecture – Exposure API, Domain API and Experience API. Experience API is the API layer that is exposed to end users, or, in other words, is the wrapper API that is launched as a product.



The Domain API is a collection made to cater to one particular domain or set of services. It is often called as Process API layer. Exposure API is the gateway that connects with the services exposed through APIs. For example, in a payment processing engine, the online funds transfer API can be considered as Experience API, while Account Management and Background Checks can be considered as Domain API. Account Posting Risk Check, Balance Check, AML check and Name Check can be considered as Exposure API.

Looking at the layered architecture, the Exposure API layer is the gateway of the services which are more from a service-based architecture. They should, hence, be designed by IT professionals while Domain API and Experience API are purely a product designing jugglery and should be handled by product owners and business analysts. Organizations can choose to implement what is best suited to them.

## Ownership of API. Business or Technology?

It's an eternal conflict between business and technology. There may be differences in opinion but with the growing invasion of technology in day to day life, the line separating business from technology is diminishing and in some cases, it has become non-existent. In API led architecture, APIs are to be designed and obviously owned by both business and technology teams.

Referring to the three layered architecture, APIs for the Experience layer should be owned by technology while the APIs for Domain and Exposure layers should be owned by business.

## API Lifecycle Management

API lifecycles are different from typical software/application lifecycle because the concept of API is essentially service-based. In a service-based approach, services are modified i.e. new features are added and removed and thus the APIs are replaced accordingly.

API lifecycle management is an important part of API management. Careful planning and execution could avoid many pitfalls caused by ignoring the proper lifecycle management principals.



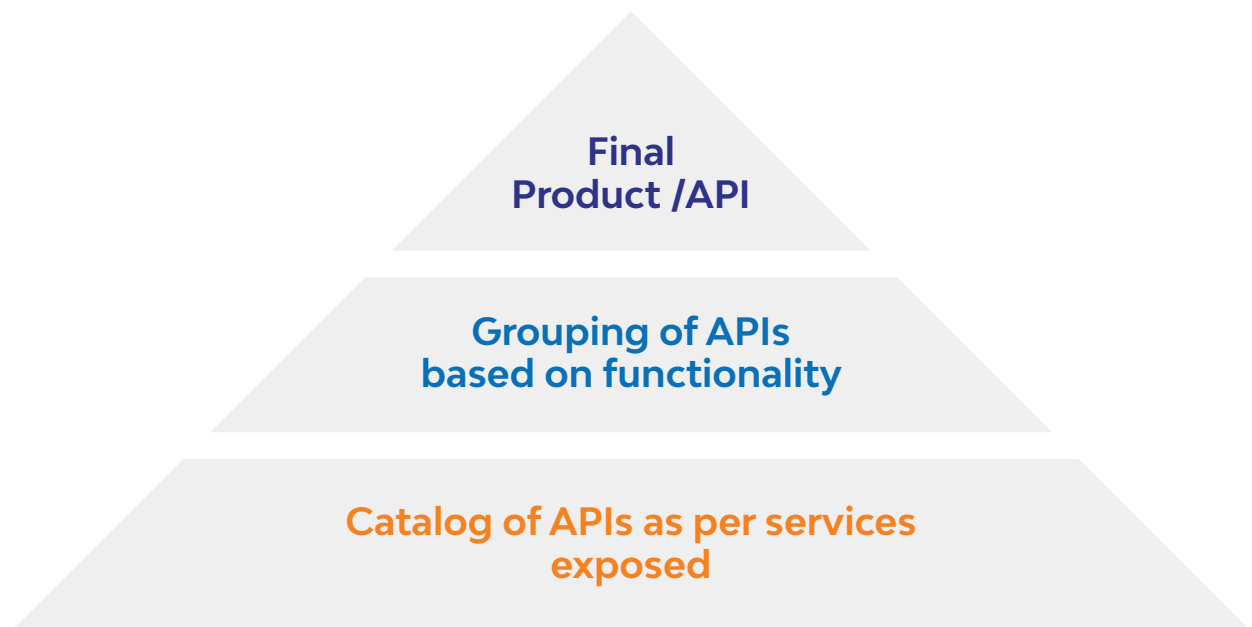
The lifecycle of an API passes through the following phases:

- Requirement Analysis
- Defining API – high level functionality
- Designing API – input/output and documentation
- Develop API – creation of API
- Test API using simulators or using end-to-end applications
- Publish or launch the API in a public/private domain – inform all users about the new version
- Monitor the usage pattern – capture and analyze information
- Retire – any change in the API plan retirement; informing all users about its possible retirement.

## Aligning API Development With Business Goals

Business analysis takes a top down approach while API development is bottom up. This approach looks more logical when we associate it with the recommended three-layer API strategy. At the top of the pyramid (refer figure), is the final product/API and at the bottom is the catalog of all APIs created for each services.

The technology team focuses on API creation per service and then passes that to business to aggregate as per domain or functionality. The business team then uses those catalogues to plan different combinations of service offerings/products.



## Identification/Packaging Of A Public API

Packaging of APIs that we are calling a product, is a very challenging task. This is the product which defines the complete API strategy of the organization. The realization of API economy is based on how carefully this packaging is done. Although this is purely driven by business, the following considerations are still worth revisiting:

### **Identify the target user**

API designers must know their target users. They should have a long-term plan to keep users engaged. Public APIs are a matter of realization and that in some cases, may take time. API product owners must have a strong marketing strategy to position the API in open market.

### **Focus on delivering value to end users**

Long and short-term value addition must be the driving force behind creating new APIs.

### **API documentation should be simple and easy to understand**

This is one of the most important factors for an API strategy to be successful. Although we often ignore documentation part but for the ease of integrating API with apps clear and crisp documentation is very important. This makes APIs popular among the developer community which hates reading complex and verbose documentations.

### **APIs should be independent and few sample subscriptions should be made to get started**

API should be self-sufficient and should not be dependent on other APIs.

### **Request and response must be captured at API management layer level (in order to capture usage patterns)**

For public APIs, each hit on the API must be captured. This data can be used to understand the pattern of API usages that ultimately helps design a mature API.

### **Exposure of third-party APIs**

Third-party APIs should be wrapped around internal APIs. Pre-processing and post-processing of messages/calls should be done to capture the usage and to apply any required policy. Third-party APIs must be part of the overall product strategy. A special caution is required when we use third party APIs. Data privacy and data masking becomes very important when we share business critical data with third party. A set of micro-services can be used to provide pre-processing and post processing of messages.



## Identification Of Internal APIs

Internal APIs are used to integrate internal applications. Reusability and uniformity of communication are the key drivers behind setting up internal APIs. For large multinational organizations private APIs play a very important role. The main focus of the API is to re-use services across the organization. Services which are common and can be reused in multiple workflows/applications are exposed through the API. For example, one API for approval is consumed by any new application where approval sub process is required.

For internal APIs following points should be considered:

- APIs must be strictly marked for internal use only .
- They often represent internal services that are used across applications
- Often these APIs work as middleware – connecting producers and consumers
- APIs for exposing data – usually master data for internal use.

There are many other type of APIs such as B2B APIs, Open Web APIs, Product APIs which are not covered in this article.

## Policies and Policy Management

Policy management is an important aspect of the overall API strategy. Success of APIs depends not only on the API but also on the type of policies being configured around the API.

Some important API policies are:

- SLA configuration for different sets of users – This should be defined as per business goals and must be driven by strategic marketing policy
- Quota policy: required to limit unwanted over use of APIs.
- Cache policy: Helps improve response time of the API
- Spike Arrest policy: This helps prevent backend systems from flooding and ultimately helps improve user experience

Note: If number of APIs are high then policy configuration become a repetitive task. LTI has developed one Excel-based accelerator that helps define policy in Excel and then migrates that into APIGEE in one go. This reduces time to configure API policy as per defined process and at the same time it gives more control on APIs.

## Monetization of APIs

The ultimate goal of public API story is to realize the tangible benefit of this exercise. Exposing services through API in public domain opens up an opportunity to make money on its usage. To make monetization successful, a mature and robust ecosystem is required. This ecosystem should contain systems to record each request and response and should have mechanisms to invoice the customer as per usages and policies defined. Nearly all established API Management tools such as APIGEE and WSO2 provide separate modules for monetization.

## Team Structure

Here's an ideal team structure for a successful API strategy.

**Marketing Experts:** This is the team involved in market analysis in terms of potential monetization options and assessment of potential revenue generation for each product designed by product team.

### Responsibilities:

- Analysis of existing marketing trend
- Monitor and analyze current usage trend of the API
- Market assessment of packaging of APIs/products
- Planning different monetization options
- Execution of monetization options
- Discover and innovate
- Responsible for success of the API program
- Share recommendations with the Product team

**Product Team:** This team comprises of business analysts and domain experts who evaluate and assess each product (API) proposed. The assessment should be based on varied criteria such as value addition, service offering, potential to grow, market presence etc.

### Responsibilities:

- Owner of the API
- Designing/packaging the API
- Evaluating recommendations received from marketing experts
- Main decision maker, takes decision on all change requests
- Decision on tools and technologies

**Technology Team:** Technology team is to be divided into two groups. Group one is a team of architects responsible for high level design, evaluation of all possible options, and a review of solutions. The overall objective of this group is to keep development aligned with long-term business goals.

The second group is a team of designers, developers and testers. This team is responsible for development and testing of APIs, micro-services, and making modifications in existing systems to make them compatible with an API-centric framework.



### Responsibilities of Architects:

- Must define end-to-end architecture
- Define standards for design and development
- Review each change and ensures adherence to standards
- Select tools and technologies
- Responsible for all technical decisions
- Monitor development

### Responsibilities of the Development Team:

- Development of API, configuration of policies and production support
- Testing and implementation

## References

1

[Survey →](#)

2

LTI Mosaic: LTI's Mosaic Things is a comprehensive IoT platform, which enables companies to capture, standardize and analyze data to arrive at actionable business insights. Enterprises can leverage our deep domain expertise to develop tailored digitization roadmaps, with the right technology architecture, channels and integration support. LTI's out-of-the box IoT solutions will help companies establish a firm competitive differentiation across the business value chain.

[Link →](#)

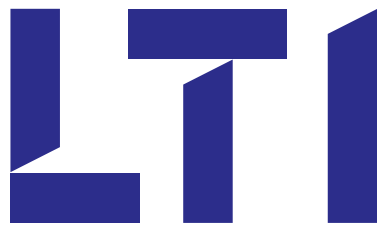
## About the Author



**Manish Ranjan**

Senior Technical Architect, LTI Whitefield, Bangalore, India

Manish has more than 16 years of experience in Analytics and Enterprise Integration. He is extensively working on API led micro-service architecture and helping clients in their APIfication journey.



# Let's Solve

LTI (NSE: LTI, BSE: 540005) is a global technology consulting and digital solutions Company helping more than 300 clients succeed in a converging world. With operations in 30 countries, we go the extra mile for our clients and accelerate their digital transformation with LTI's Mosaic platform enabling their mobile, social, analytics, IoT and cloud journeys. Founded in 1997 as a subsidiary of Larsen & Toubro Limited, our unique heritage gives us unrivaled real-world expertise to solve the most complex challenges of enterprises across all industries. Each day, our team of more than 25,000 LTItes enable our clients to improve the effectiveness of their business and technology operations, and deliver value to their customers, employees and shareholders. Find more at [www.Ltinfotech.com](http://www.Ltinfotech.com) or follow us at [@LTI\\_Global](https://twitter.com/LTI_Global)

[info@Ltinfotech.com](mailto:info@Ltinfotech.com)



A Larsen & Toubro  
Group Company