

Recommendations for an API Economy Center of Excellence



Table of Contents

3	Goals and Guiding Principles	10	Center of Excellence(CoE) Processes and Concerns
4	Organizational Structure	13	Real Center of Excellence examples
	<ul style="list-style-type: none">• Perspective, Business Drivers, and Goals• API Maturity Level• Existing Organizational Structure• Recommended API CoE Structure• Multiple Lines of Business	16	Recommendations – Do’s and Don’ts
7	Core Team Roles	17	About the authors
	<ul style="list-style-type: none">• API Product Manager• API Developer• Operations• API Initiative Leader		
9	Related Stakeholder Roles		
	<ul style="list-style-type: none">• Internal (and External) App Developer• Executive Steering Committee• Business Domain Owners• Integration Architects• Service Owners• Security		

Business APIs are here to stay. Companies big and small are adopting API technology for faster business growth. The API hype is long passed, and APIs are now the foundational building blocks for new initiatives such as Digital Transformation, Journey to Cloud, Artificial Intelligence, and Blockchain – to name just a few. With these critical initiatives relying on your ability to provide a solid foundation, it is imperative that you are positioned for success with your API initiative. Organizations frequently ask what roles are required to drive a successful API initiative, how should this fit in the current organization, and how do these roles relate to existing roles in the company.

In this paper we provide recommendations for an API Economy Center of Excellence to guide your API initiative. Topics include:

- Goals and Guiding Principles
- Organizational Structure
- Core Team Roles
- Related Stakeholder Roles
- Center of Excellence (CoE) Processes and Concerns
- Real Center of Excellence Examples

Goals and Guiding Principles

While the initial project to embark on an API journey is usually a pilot or a getting started set of API use cases, this usually grows over time sometimes rather too quickly as more API opportunities are identified, and line of business teams adopt APIs.

At this point there is a critical decision on governance of APIs. On one side there is the desire for speed to market and driving rapid API adoption, but this may result in overlapping API functionality, and inconsistent and poor API design. On the other side, there is a desire to drive consistency, reuse, and control which could result in bottlenecks and failure to deliver the desired speed to market. The answer needs to be somewhere in between. The goal for the API [Center of Excellence](#) is to allow the business to maximize the impact of its API strategy by driving the use of APIs across the organization to achieve the desired speed for new and enhanced business solutions.

The growth of API adoption requires a consistent approach for a successful API strategy and business outcomes. This requires defining uniform API discovery and API metadata requirements as well as driving uniform development practices while also supporting different organizational cultures coordinating enterprise alignment on their needs for API development and strategy.

Although the notion of defining an API Center of Excellence is quite appealing, it is something that is better to evolve into as you go from initial APIs to wide spread adoption across the enterprise. The initial set of APIs usually get defined by small teams which collectively could form an API guild. This could grow over time and lead towards creating an API Center of Excellence based on wide number of factors as shown in the diagram below:

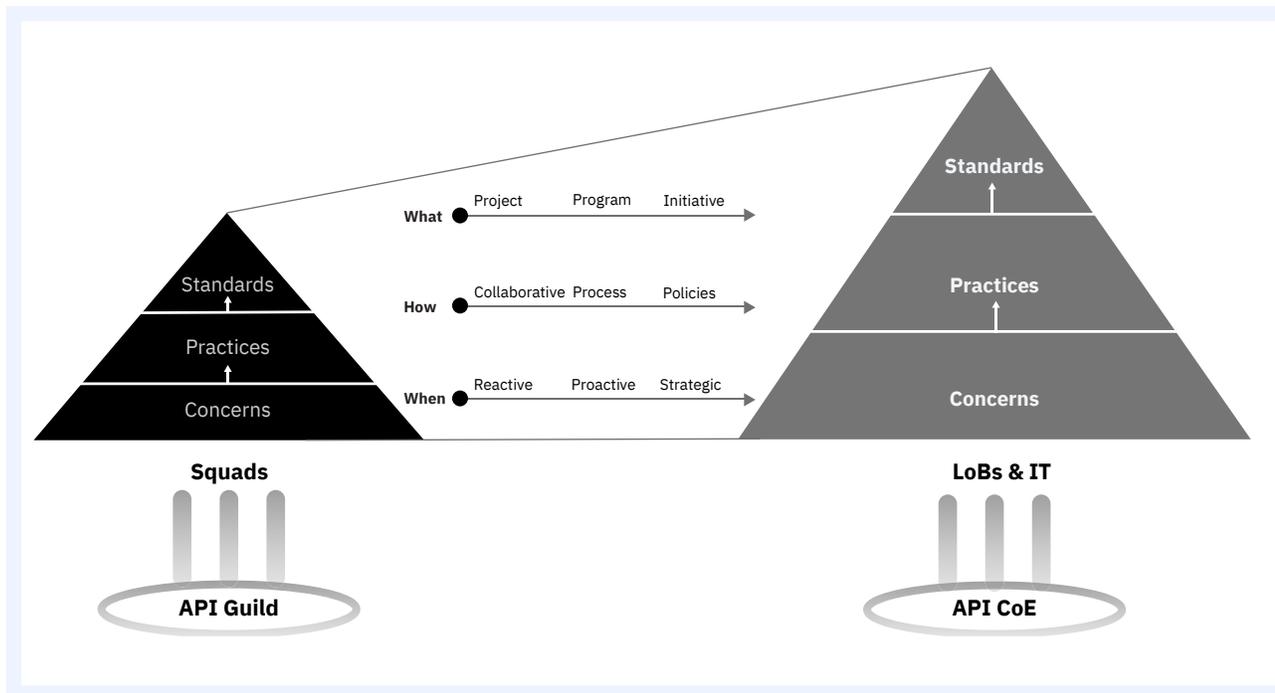


Figure 1. When do you start a API CoE
Anytime, Small and Iteratively

As described in the sections that follow, the CoE organization and roles combine both business and IT responsibilities and often span multiple organizations within the business and sometimes reach outside the business with extended partnerships with API consumers. There is no one “right” way that fits all companies but focus on definition of the principles and guidelines and not creating a centralized bureaucracy is a must.

Organizational Structure

There is more than one way to organize. What is best for one company may not be for another. In this section we provide some guidelines and share some observed best practices to help you decide what is right for your enterprise.

There are several factors to consider when thinking about how an API initiative might be positioned in your company:

- Your perspective, business drivers, and goals for the API initiative
- Your API maturity level
- Your existing organizational structure

Most likely changes are required to the organization. There are some new roles to be staffed and there is significant benefit to putting a focal point in place to help drive this as an ongoing **digital transformation** and not just another IT project.

Perspective, Business Drivers, and Goals:

The most successful businesses using APIs are viewing their API initiative as a [new channel to market](#). Their goals are to empower the business to execute faster, reach new customers, and provide business innovation. Companies with this perspective are incented to change the organization to drive this success.

Other companies are recognizing that there is a new ground swell around business APIs. They are beginning to learn about APIs but have not yet developed a strategy or identified business use cases. Companies in this situation are probably not ready to commit to organizational changes. So, working the API initiative into something that does not require significant organizational change is probably more likely – at least initially.

API Maturity Level:

IBM introduced an [API Journey Map](#) as a maturity model for the API Economy. In this we recognized as companies begin their API journey the initiative tends to be led by IT. However, as the initiative matures it becomes a partnership between the business and IT moving to a business led paradigm. Where are you on your journey and do you have the backing to push the initiative to the next level?

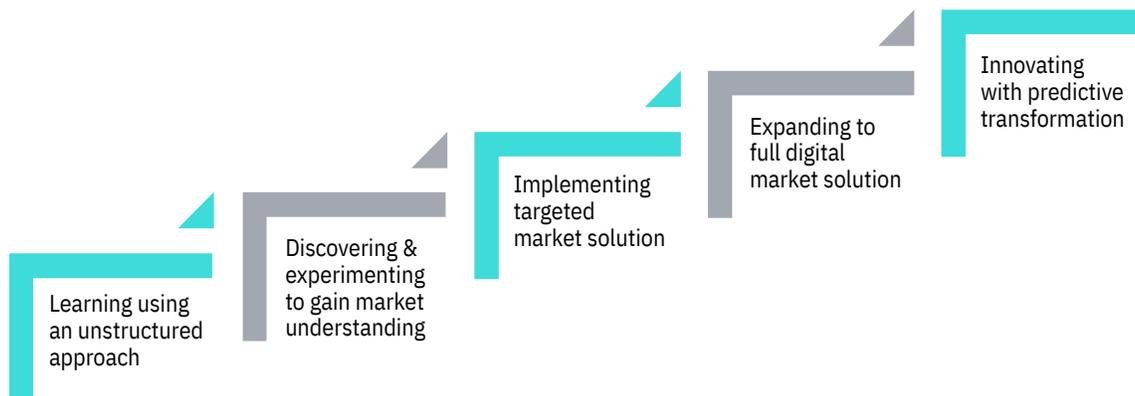


Figure 2. 5 Stages of maturity

Think about the early days of the web (if you are old enough to remember). IT led the initiative deciding what the company web page would look like. What does your web site look like now? The business is probably driving this as a channel to market for your company and IT is in a supporting role. This same transition will occur over time in the API Economy.

Involving the business is extremely important to driving initiative value, but in the earliest maturity levels there tends to be little business involvement. The quicker you progress in making this a business initiative the more value the initiative will bring. If you are in the early stages, recognize the path to success and bring the business on-board. The building of the Center of Excellence (CoE) and the appropriate role assignments can help move this forward.

Existing Organizational Structure

While APIs are about disrupting the market, most businesses want to minimize the disruption inside the company. The existing organization will be considered as part of the API initiative and adjusted as the initiative matures. How independent are the Lines of Business? Is there a separate IT organization for each LoB or a centralized IT (or something in between)? How well do the business and IT work together? Answering these questions can help determine the structure for the CoE.

Recommended API CoE Structure

If you believe that APIs are a channel to market, you should begin to position the organization to execute on this channel.

IT owns the infrastructure and processes to enable the channel. This includes establishing the API Management solution (e.g. IBM API Connect) either on the cloud, on premise, or hybrid. The architecture and security teams ensure the implementation is integrated with the current enterprise solutions. Processes such as lifecycle management are driven by IT. IT also performs the API Developer and Operations roles (see all role definitions in the following sections). Often the API Developer roles are initially centralized and frequently are related to prior IT integration centers of competence. But this may change to LoB driven API developers if this is your current organizational model and as new LoBs come on-board. If your company executes with a decentralized IT, then it is not critical to change this organization.

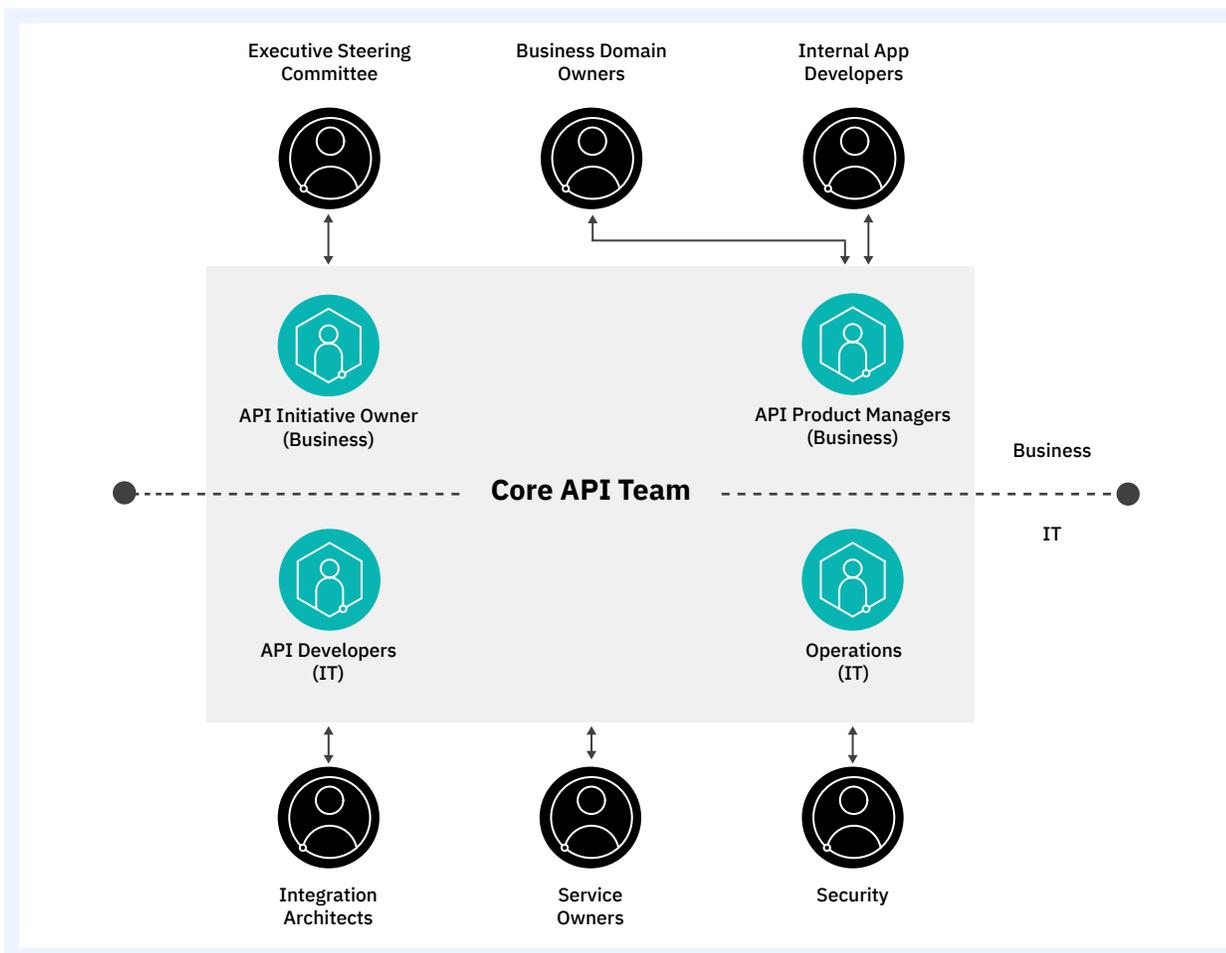


Figure 3. API CoE structure

The Business owns the API products that are to be delivered. This includes the API initiative owner and the API product manager roles. To drive success, API offerings should be treated as products. This includes appropriate market research to determine a desired target audience (internal, external, or both), defining APIs by the audience's business need - not by current IT infrastructure implementation, and marketing of the API product once it is created. API Products contain one or multiple individual APIs and the packaging is determined based on what makes it easiest for the consumer. Driving the success of the API product in the market and measuring usage is also a business concern. The API Economy is about revenue generation not IT cost reduction, so Business involvement is critical. Many companies either do not staff the Product Manager role or keep it contained in IT. Both choices limit the success of your initiative. For internal use cases (e.g. Mobile), the Business often owns the IT resources in the App developer role (API Consumer).

Multiple Lines of Business

Many companies have multiple Lines of Business (LoBs) that tend to operate independently. Almost universally initial API efforts start in one LoB and there is one Product Manager assigned from that LoB. As additional LoBs begin their efforts additional Product Managers are added. From an organizational perspective the API Product Manager role is aligned to each LoB driving their needs. However, the initiative leader role is across all the LoBs.

As you are establishing your organization to support your API initiative, remember one of the primary goals for most API initiatives is to improve speed of execution for business initiatives. Is your organization supporting this goal? If not, change is probably required.

Core API Team Roles

Core API team roles are the members of the API Center of Excellence. As shown in the organizational structure diagram in the previous section, there are 4 core team roles: API Product Manager, API Developer, Operations, and the API Initiative Leader.

The actual number of people in these roles vary. Early in the initiative a person may wear many hats executing all the tasks defined within a role and perhaps even cross role boundaries. But as the initiative grows and becomes more engrained in how the company executes, the number of personnel participating increases. It is entirely likely that there are multiple people in each role dividing the duties described by technical or business competencies and as the initiative grows additional resources may be assigned for each of the business' LoBs. It is best to think of each of the roles as a team of people, not just one individual. Thus, when establishing the governance and processes you will account for many people interacting to drive the success of the API initiative.

API Product Manager: This is the person/people who identify business needs to be addressed by an API. API Product managers identify desired audiences to be addressed by the APIs and work with these audiences to define and verify their requirements.

Following the initial requirement identification this role works with the API developer role to specify the API, set the terms and conditions for accessing the API (rate limits, security, monetization), and then markets (communicates/demonstrates) the existence and value of the API to the intended audience once the API product is available. They also, of course, care about the success of their APIs. So, they are viewing analytics about their APIs to see what is working and which ones might need some additional attention to rectify poor adoption or lower than expected use.

Finally, they gather metrics for reporting to demonstrate success to the business and the attainment of desired goals.

For most businesses this is a new role and therefore requires funding and staffing. There has not been a prior role that easily maps to this role. Traditional product manager or business analysts are the most common places businesses look to staff the correct skills.

API Developer: The focus of this role is the technical implementation of the API. This includes creation of the API itself and potentially other assets invoked by the API as required. While thinking of this as one role, there are often several skills that are required, and this may lead to several personnel being assigned with each specific skill. There is an API interface developer who deals with the Open API standards (Swagger, REST/JSON) and/or SOAP/XML. Since integration technologies greatly compliment the API Economy, many organizations have chosen to couple the API COE as a partner to their Integration COE. API developers need to interact with integration experts to connect to internal and external resources including assets such as Web Services, databases, packaged applications (e.g. SAP), cloud applications (e.g. SalesForce), or other cloud APIs (e.g. Google Maps). And, there also may be Microservice coders who introduce new or changed business logic in languages such as Node.JS, Java, Swift, or others. Assets accessed by an API may be in various locations inside the company and on one or more clouds. Integration architects and the API Developer need to work together to support this hybrid integration architecture.

Other technical roles related to API creation and lifecycle management are also included in this larger “API Developer” role category. This may include roles defining API standards, naming conventions, API architecture, Lifecycle governance, etc. Early on, many of these tasks will be executed with few people, but over time the tasks may be spread across more specific role definitions.

The API Developer role is often staffed with previously skilled integration developers who have worked with web services, ESBs, or other integration technologies. While the technology is new, the skilled integration developer can easily adapt to the new API environment capabilities.

Important Note – Frequently the API Developer role is asked to take on the duties described for the API Product Manager because of the lack of funding for the new API Product Manager position. This is a big mistake! API Developers are great at technical tasks but do not usually do market research well. This drives poorly defined APIs which can affect the success of the entire initiative.

See [“The 7 Biggest Mistakes Companies Make on their API Initiatives”](#).

Operations: The operations role handles API Management platform build and maintenance, organization definitions for consumers and providers of APIs, operations, and ongoing support for the execution of the API Management product. The role is concerned about scalability, availability, performance, etc. and as such establishing the environments and monitoring them is a key task.

Another task set for this role deals with the DevOps aspects of the API lifecycle. This includes promoting APIs into production and automating or handling transitions in the API Lifecycle and with asset management repositories.

Finally, this role may also handle administrative tasks including setting up the organizational structures for consumers and providers, setting up role access for the various other roles involved (Product manager, developer, etc.) and, managing the approval process for App Developers (consumers) attempting to sign up to use provided APIs. This role is often staffed by existing personnel from the operations organization in the company representing the operations aspects on the CoE.

API Initiative Leader: This is a second business-oriented role working at a higher level than the API product manager, driving the entire API initiative for the company. The role is looking strategically at the overall business goals, driving organizational change and governance, determining the strategy, measuring the overall success of the initiative, and reporting to the executive team. For this role leadership and communication skills are critical as they frequently need to influence parts of the organization that do not directly report to them in the management chain.

Frequently the API Product manager plays the initiative leader role in the earliest phases of the initiative. However, as the initiative grows and expands to multiple projects and LoBs, an overarching role is required as an initiative leader across the company working with the individual product managers.

Related Stakeholder Roles

There are many stakeholder roles that are very involved in the success of the API initiative but are not members of the CoE. There are frequent interactions between the CoE and this extended team. In this section we describe the part of the role as it relates to the API initiative. These roles may also execute many other tasks unrelated to APIs.

Internal (and External) App Developer: The App Developer role – internal or external – is the consumer audience for the APIs that are to be created. Internal developers are people who work for the company supplying the APIs either directly or as contractors. External developers work for another company, either a business partner or a previously unknown external company. It is important to recognize that even for internal App Developers the projects they deploy may send API traffic into the enterprise from outside the company – for example a mobile App. The designation of internal versus external is only about employment.

API Product Managers need to treat both internal and external App developer audiences as customers. The API Product manager needs to consult on API requirements before the API is defined and market the created APIs to this audience to demonstrate the value provided once the API is available.

From a role perspective, internal developers may be easier to consult for their needs and more easily act as part of the API initiative team. They are frequently the early target audiences and may have to deal with frequent API updates and re-versioning as the API Product Manager and API Developer are growing their skills.

External App Developers also need focus. Depending on your industry you may have supply side and sell side partners who would have different needs from your APIs. And, public APIs need special marketing to reach an audience with which you have no prior relationship.

Once again, for both API requirement gathering and marketing of the APIs after availability, the API Product manager needs to interact with the App developer audiences be they internal or external.

Executive Steering Committee: This is the executive team in the organization that has agreed to fund an API initiative and is providing resources for the initiative. They should also have defined the [business drivers](#) they are expecting the API initiative to deliver and the agreed upon metrics to define success. The API Initiative leader may be a member of the ESC or may report to them regarding achievement of the desired results.

Business Domain Owners: This is the management team in the business for the API Product Manager role. They own the APIs that are being created and own the assets that the APIs will expose. The API Product Manager is their representative in the initiative.

Integration Architects: APIs are usually part of a larger overall enterprise and integration architecture combining multiple forms of integration – events, messages, application integration, and files. As described in the API Developer role, the API Developer needs to work with the integration architects to fit within the overall integration architecture for the company and drive any necessary changes to this architecture as APIs are being introduced.

Service Owners: APIs need to access some asset inside or outside the enterprise to obtain data or invoke a transaction. For assets inside the enterprise, there is a technical owner who can provide access to the asset, needs to know how it is being used and how much, and provides information on how to invoke the asset - this is the Service Owner. The API Developer works with the service owner(s) to obtain access to the necessary assets to build their API and set the rate limits and security.

Security: A critical aspect to all API initiatives is security. The security team needs to ensure that the enterprise standards are met and understand the API authentication and access requirements and how security is being managed by the API management / gateway infrastructure.

Center of Excellence Processes and Concerns

As you start with initial APIs, it is important to think of defining consistency across a set of key areas and establish a few guidelines. These guidelines should be closely aligned with the business objectives for API adoption.

Here are the five areas you should consider establishing when starting an API journey. This could grow over time into a comprehensive set of governance guidelines and accelerate the maturity towards establishing an API Center of Excellence. The five areas are:

1. API use case identification

- Every API should have a consumer
- Determine API exposure – will it be internal, external, or both?
- Business objectives for the API. Defined success criteria

2. API Exposure

- Establish naming and API URI guidelines to enable ease of consumption and avoid duplicate or confusing naming
- Define API documentation guidelines for required information
- Create a consistent monetization plan and tiers e.g. freemium, silver, gold, platinum

3. API Security

- Understand security options and apply security consistently across APIs
- Ensure API consumers can simply understand and implement what is required for API security
- Establish specific resource access within API security e.g. some users can Get information, while others with more authority may be able to update the information

4. API Lifecycle

- Have a plan for how to version APIs – both backward compatible and non-backward compatible.
- Take advantage of Plans (multiple usage tiers) and Products (combinations of APIs) to provide simplicity for consumers and maintain controls on usage of resources.
- Establish API Ownership guidelines and who, how, and when updates will be rolled out.

5. API Taxonomy

- Organize and classify APIs for easy discovery
- Establish common terminology e.g. API vs Microservices vs Services. Terminology misuse can cause many issues!
- Provide documentation to API developers about how to conform to and take advantage of the taxonomy

This could be a starting set of guidelines for your initial APIs and can be expanded over time. The primary goal of these guidelines should be the continued success with APIs without being too overbearing on the innovation aspect of API adoption for business growth.

As the CoE matures and as needs arise additional areas for CoE consideration may appear. The diagram below outlines several areas of concern that may require guidance (but not burdensome) input from the CoE:

People 	Process 	Content 
Stakeholders/ Approval boards Personas, Roles and Responsibilities Accountability and ownership Access and Privileges Users and Organizations Identity access Management Human capital and skills requirement	Operations / Support Review / Approval process Collaboration tools Change Management API Lifecycle management Education training programs Billing and Payment Industry Compliance CoE Process and Project Office Development /Design	Metadata Management Documentations Quality Criteria Security and Access Control Metrics, KPI's and Analytics Packaging: Products and Plans Pricing and Metering Standards and Taxonomy

Five Common Best Practice Examples

While there are many areas that may be eventually built out, consider the following a “getting started” list of best practices to kick off your initiative.

API Identification: Control the granularity of products and plans:

- It is important to establish a clear definition within an organization on how API Developers and Product Managers should decide the granularity for a use case.
- A Plan is the subscription point that an App Developer will register for:
 - Focus on the 80/20 rules, and create Plans that include all the relevant APIs to complete a functional use case, instead of requiring the App Developer to subscribe to multiple separate plans
 - Think of billing: all the APIs within the Plan should have a similar business value to assure sensible entitlement levels can be defined.
- A product is a grouping of Plans with different usage tiers.
 - Think about managing their life cycle as a single entity
- An API is a collection of API Operations within the same context
 - In general, if two API operations are within two different functional areas, then these should be represented as two APIs.

API Exposure: Think of an approach to use the developer portal.

- The developer portal is the window to your brand. Some enterprises have a single brand and others have multiple brands
- Do you want to have separate portals for each of your LoBs?
- Do you have multiple consumer bases? Internal vs external developers
- Align this decision with other factors such as regulatory compliance, internal devops, backend business processes that are exposed as APIs, etc.?
- These decisions are led by the API initiative leader and product manager with the API developer implementing the desired approach.

API Security: Consider security first!

- Understand your users and their interactions
 - Will the APIs be used by trusted Business Partner applications?
 - Will the APIs be used by Third Party mobile applications, wanting to access your client's data?
 - Will the APIs be used by Lines of Business within your company?
- Clearly define different security approaches and involve your enterprise security teams
- API Product manager, Developer, and the security teams are all involved in this discussion.

API Lifecycle: Define an API Versioning Approach

- As your API strategy matures and the API consumption is wider and more diverse, the importance of a versioning approach will become more critical to the ongoing success of the API strategy.
- Defining your major / minor version approach and how this will be exposed to the user is an important first step in managing the agile release of new API functionality from your development teams.
- Typically, the API developer takes the lead on defining the strategy and implementing this in the lifecycle, however the API Manager and Operations are both involved in communicating the strategy to the target consumer audiences.

API Taxonomy: Align on common terminology across the enterprise

- API, Services, Microservices. Integration? What is your point of view? Understand and document the different constructs and how each is used and where it sits in the architecture.
- Without this single definition confusion will be introduced and many meetings will be wasted re-debating the same discussion.
- This is led by the API Developer role in conjunction with the Integration experts and service owners.

Center of Excellence Examples

We interviewed two companies – one in the automotive industry and one in Retail - who are doing very well on their API initiatives to get their insights into how they have organized their CoEs. The two interviews were held separately, so one answer did not influence the other.

Note: both organizations have participation from both Business and IT. The role names they mention may be different than what is documented here as each organization may define roles their own way and with their own naming conventions.

Here are their thoughts:

Please tell us about the structure of your API Center of Excellence – what roles are included, how often do you meet, what is the responsibility of your CoE, Is the business involved?

Automotive:

“Our CoE consists of API Architects, evangelists and SMEs to support API awareness and adoption, API lifecycle management, and governance. Our Business is involved in the CoE as one of the key stakeholders along with the IT folks. The IT folks are the API owners for integration services. However, any of the business APIs, for example, the window sticker on an automobile which is used by third-party dealer systems, are owned by the business and are responsible for approving the subscription.”

Retail:

“We have one CoE team that investigates the Business APIs, defines those Business APIs, and also looks into the (API) products. Once we create those products, we look at what can be common, what kind of issues we face, who’s portfolio does it fit in. We then look at a phased approach to creating the necessary APIs and how we can do a complete CI/CD pipeline for these APIs. This led to a lot of conversations and we have regular meetings to address this with myself and one other team that is looking into the CoE and then there is a practices team. So, three teams combine to address the issues as they come up and move forward with the initiative. Right now, the business is not completely involved in the CoE. There are business driven architects.”

Do you separate your API CoE from an SOA CoE (or other CoEs) if you have an SOA CoE?

Automotive:

“We started our journey with SOA, but we never created an SOA CoE. We started with an API strategy – defining the strategy, and as part of the execution, we realized that the CoE is the way to go. In the company we have other CoEs, and this model works because it is better suited to support across our enterprise process areas like finance, manufacturing, sales, and marketing. The CoE model helped us to scale much faster by cutting API platform learning curve and by leveraging the entire application team which supports 2000 plus applications. Our role on the CoE is more of an evangelist and guiding them. We have API architects on our team helping the application teams to build and use the APIs in the right way. that is, by treating APIs as a product, and focusing on building “consumer centric APIs”. We are doing this by adopting a product-based, API consumer-centric mindset for the organization’s APIs.”

Retail:

“We have a combined CoE for SOA and APIs.”

How involved is your CoE in the execution of your API Governance processes? Is it more about establishing standards and processes or is it more involved in the lifecycle process for all APIs in your enterprise? What other activities are involved in the CoE?

Automotive:

“Our API CoE is responsible for the entire lifecycle including the governance. We start with an awareness session for the application teams helping them to understand the use of an API, the right fit, all the way to the onboarding of the APIs. The CoE is responsible for configuring and publishing the API and then at a lower level, monitoring and supporting the API platform. The API owners will be responsible for the backend services as well as any subscription, managing the subscription and the backend system platform maintenance.”

Retail:

“It goes both directions. One is that you need to govern the APIs that are being created. We make sure that each portfolio comes up with their own APIs and then govern it in such a way that the product owners will work with their teams and govern it in their own portfolio. Then, when they come back to us, we will see that these are the 5 Business APIs. Then we look at how we can monitor it, monetize it, and how it can be used in applications that span the individual product lines.”

Based on your experiences in your API CoE, do you have any other suggestions or best practice advice for others?

Automotive:

“People still have confusion treating API as webservices. We have challenges explaining APIs from the business capabilities we are enabling. We need to answer a very important question, “what is in it for me?” How does the API help them or what is the value from using APIs? For example, we sit with the application teams who are responding to a request for enhancement that the integrations should be based on APIs. So, it is important to explain to them that if they don’t use an API then how do we do this? We show them “what if” scenarios. Then they immediately understand how the API platform simplifies the integration and reduces the cost. Also, very importantly, they won’t be owning a custom solution hosting somebody else’s data. With the APIs, they can only focus on the business functionalities and the components that are relevant to their business.”

Retail:

“First, creating the CoE and making sure the teams are aware of the capabilities. People are talking to the development teams and they come up with microservices. But how are the microservices aligned to APIs and how does this all get turned into business applications? These things need to be planned way ahead. Creating the CoE team earlier to guide this discussion helps a lot.”

Recommendations – Do's and Don'ts

Do's

- Do focus on a Community of Practice model where collaboration, inter-networking and fresh innovation are encouraged.
- Do practice servant leadership, provide a clear vision on the benefits of a common API initiative and build a support framework for the teams to formulate their own approach to achieve the vision.
- Do manage by exception and trust the teams in the implementation without dictating and micro managing. Intervene only when there are obvious flaws in the team decision or conflicting approaches within a team.
- Do provide for open feedback channels and frequent revision opportunities of the current standard practices.
- Do focus on team learning, enablement and skills growth around API and API management across the enterprise.

Don'ts

- Don't focus on an oversight supervisory model with decision and control concentrated within a small body or committee.
- Don't focus too much on standardization and rigid processes but do provide implementation guidelines, best practices and lesson learned.
- Don't focus on a SME model where there is reliance on a limited number of API champions and coaches who know everything on the domain.

Creating a Center of Excellence and staffing appropriate roles goes a long way to driving the success of an API initiative. While not all organizations are the same, there are many common practices that successful API initiatives have followed as outlined in this paper. API initiatives are different than traditional IT-only integration tasks that have gone before. Creating a CoE with both business and IT roles can reduce mistakes and help a company more rapidly move forward in their maturity and thereby drive successful business results sooner.

For more information:

To understand more about IBM's thoughts on Digital Business and the API Economy visit the [IBM API Economy website](#). IBM API Connect is IBM's complete foundation to Create, Secure, Manage, Test, and Monitor APIs. You can find more information about IBM API Connect at the [API Connect website](#). And you can also experience a [trial version](#) of API Connect.

About the authors:



[Alan Glickhouse](#) Digital Transformation and API Business Strategist, IBM.

Alan Glickhouse is a business strategist on the IBM API Connect offering management team. He joined IBM in 1981 and has held numerous positions in sales, technical sales, marketing, development and technical support. On the API Connect team, Alan assists clients in all industries, of all sizes, and in all geographies with their business strategy for Digital Transformation and APIs, understanding their business direction and existing environment (both business and technical), and helps businesses successfully adopt a strategy that fits their environment. Contact him at glick@us.ibm.com or [@ARGlick](#).



[Prasad Imandi](#), Senior Cloud Integration Architect - Client Technical Outreach, Integration, IBM Cloud and Cognitive Software.

Prasad Imandi is a Cloud Integration Architect in the Integration Modernization Client Technical Outreach team. He has expertise in strategic enablement of enterprises on APIs, Cloud and Kubernetes technologies for Digital Transformation. Prasad works with clients worldwide in helping them modernize integration with Cloud Pak for Integration. Contact him at imandi@us.ibm.com.



[Tien Nguyen](#), Distinguished Engineer - Cloud Garage Solution Engineering – Integration, IBM Cloud and Cognitive Software.

Tien Nguyen is an IBM Distinguished Engineer in the Garage & Solution Engineering team focusing on Enterprise Integration modernization solution architectures and methodologies. He provides technical leadership in key strategic client opportunities to ensure their implementation success. He leads technical workgroups to create deployment offerings that support the implementation of the IBM Cloud Pak for Integration and accelerate client time to value for our IBM cloud solutions.



For more information

Visit: <https://www.ibm.com/cloud/api-economy>
or cloud integration:
<https://www.ibm.com/cloud/integration>

Follow us



Connect with us



© 2020 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM’s products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at: www.ibm.com/legal/copytrade.shtml.