



सत्यमेव जयते

Ministry of Electronics and Information Technology
Government of India

Open API Policy Implementation

17-Jan-19



Digital India
Power To Empower



AADHAAR



UMANG



DigiLocker
Your documents anytime, anywhere



BHIM
Bharat Interface for Money



MeghRaj CLOUD INITIATIVE
GOVERNMENT OF INDIA



OpenForge

Background

- ❑ Ministry of Electronics and IT in 2015 had notified the "Policy on Open Application Programming Interfaces (APIs) for Government of India". *The policy intended to promote efficient sharing of data among data owners and **inter-and-intra governmental agencies** to achieve the objective of interoperable systems in order to deliver services in an integrated manner.*
- ❑ Meeting with Secretary MeitY on 7th September, 2018, regarding implementation of Open API Policy
- ❑ Points raised by Shri J Satyanarayana:
 - ❑ Classification of Open APIs based on openness to be taken up
 - ❑ Whether separate Security policy in place for Open APIs?
 - ❑ Whether the API software shall be designed on the lines of GitHub or ap.gov.in?
 - ❑ Designing of templates to facilitate and standardize API documentation
 - ❑ Whether and when to integrate using Enterprise service Bus (ESB) or take the API route
 - ❑ Approach for API policy implementation in brownfield vs. greenfield projects.

Objectives of Open API Policy

- ❑ Ensure that **APIs are published** by all Government organisations **for all eGovernance applications** and systems.
- ❑ **Enable quick and transparent integration** with other e-Governance applications and systems.
- ❑ **Enable safe and reliable sharing of information** and data across various e-Governance applications and systems.
- ❑ Promote and **expedite innovation** through the availability of data from eGovernance applications and systems to the **public**.
- ❑ **Provide guidance** to Government organizations in **developing, publishing and implementation** using these Open APIs.

Expected Outcomes of Open API Policy Implementation

Lower cost of
universal
service

Consistent
design

Low cost of
maintenance

Low cost of
integration

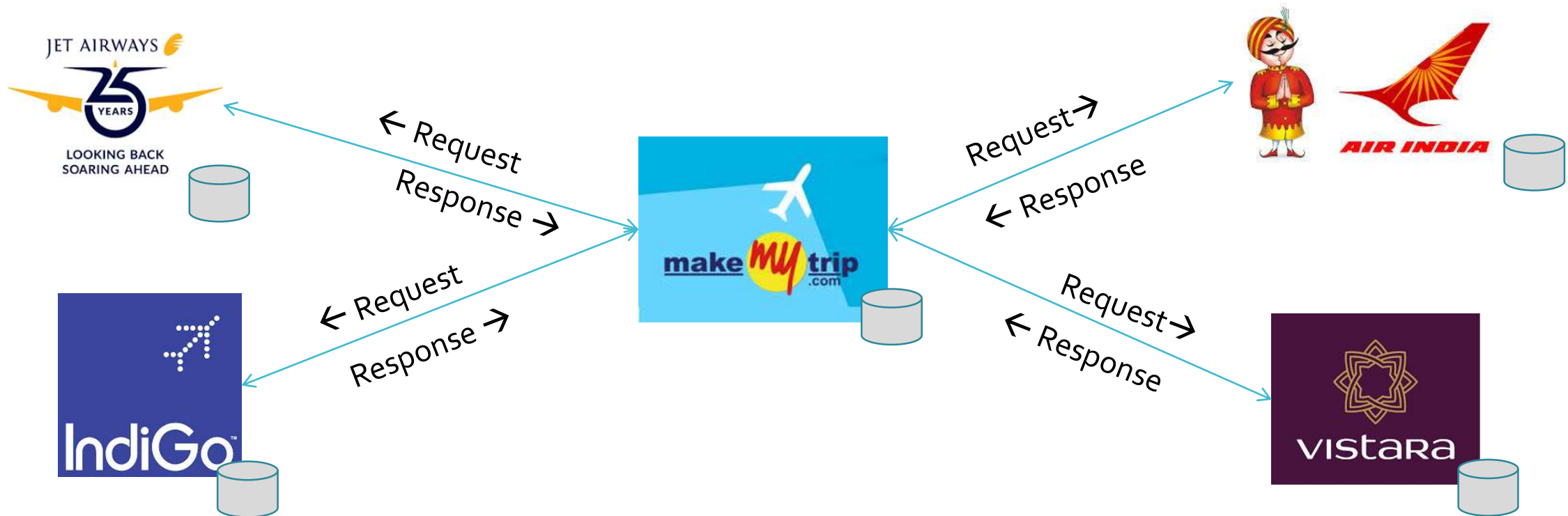
Higher security

Higher
availability

Data availability
to industry /
public

What is an API?

- Application Programming Interface i.e. **a specification of remote calls (MESSENGER)**



- <https://www.youtube.com/watch?v=s7wmiS2mSXY>

More about APIs

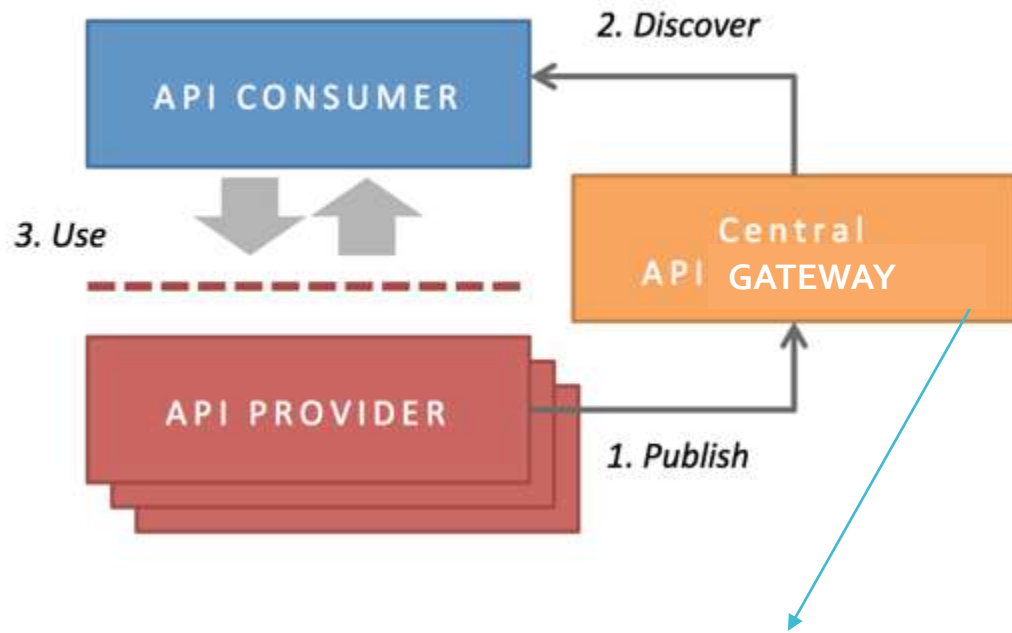
- ❑ In general terms, APIs are a set of clearly defined methods of communication between application programs to interact with each other and share data.
- ❑ APIs do not depend on the host application programming language
- ❑ All **Mobile First** apps have in-built APIs
- ❑ An **Open API** is a publicly available API that provides developers with **programmatic access to a proprietary software** application or web service.
- ❑ **Web Service** is an API wrapped in HTTP. A Web Service needs a network while an API may not mandatorily require a network for its operation. All web-services are APIs, although all APIs may not be web-services.
- ❑ **Brownfield projects**, build on SOA architecture should share their data **through web-services** or APIs
- ❑ When the **API first approach** is used for **architecting** an application, the service gets composed of many loosely coupled **micro-services or APIs**. **Greenfield projects** should build their applications using the API first approach.

ESB vs Web-services vs APIs vs Micro-services

Sl. no.	ESB	Web-Services	APIs	Micro-services
1	Is an additional piece of software	Is an additional piece of software	May be integrated with host application or as an additional piece of software	Multiple micro-services culminate into one application
2	Exposed via bus messaging protocol	Exposed via http	Can be exposed via JAR, DLL, XML over http, JSON over http etc.	Exposed via APIs
3	Communication via SOAP, REST, XML, WSDL etc.	Uses SOAP, REST and XML-RPC as means of communication	Apart from SOAP, REST and XML, it may communicate via curl calls	REST, http or thrift APIs
4	Style of integration architecture (SOA based)	Is a communication protocol (SOA based)	Is a communication protocol	Is a type of application architecture

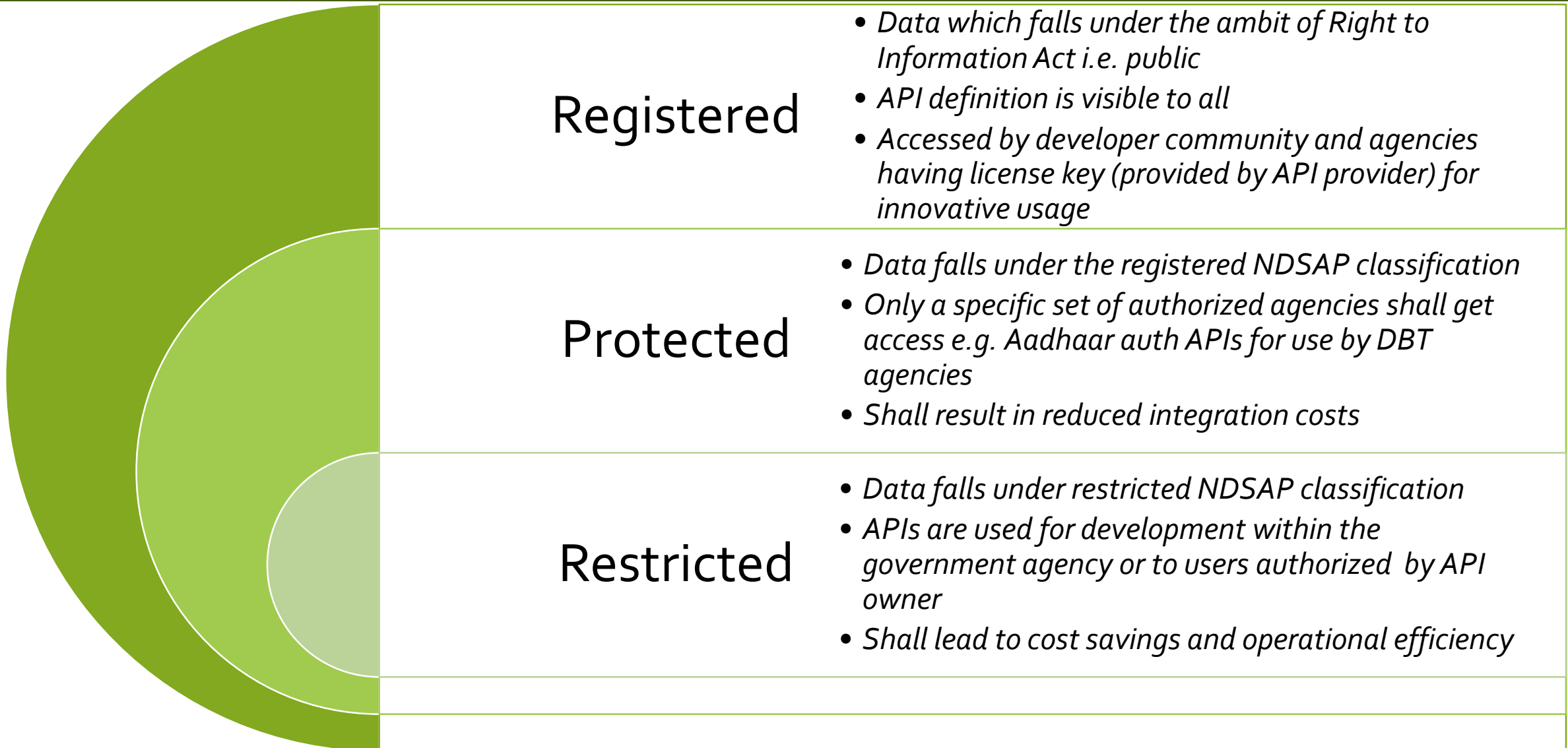
User Roles: API - Management

- ❑ **API Owner/Provider:** Government organisation or other organisation who is the ultimate owner of this API .
- ❑ **API Directory / Gateway:** An organization (**NeGD**) that is responsible for maintaining the central API directory, consent, sector wise API access etc.
- ❑ **API Consumer:** An API consumer might be another **government agency** or **organization / private agency** or an **individual** with a registered application and license key

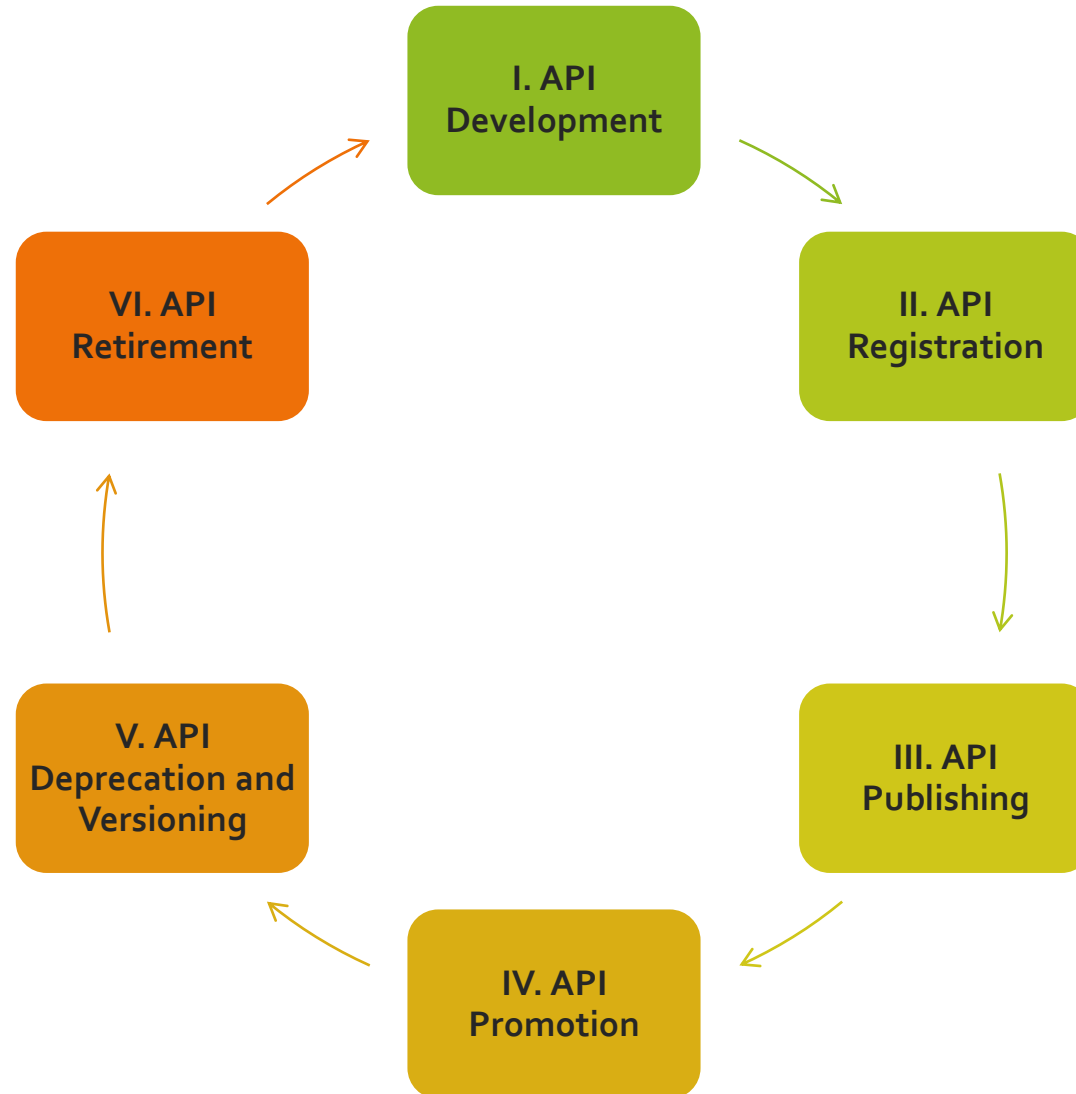


Open API Registry – every Open API is registered in this Repository

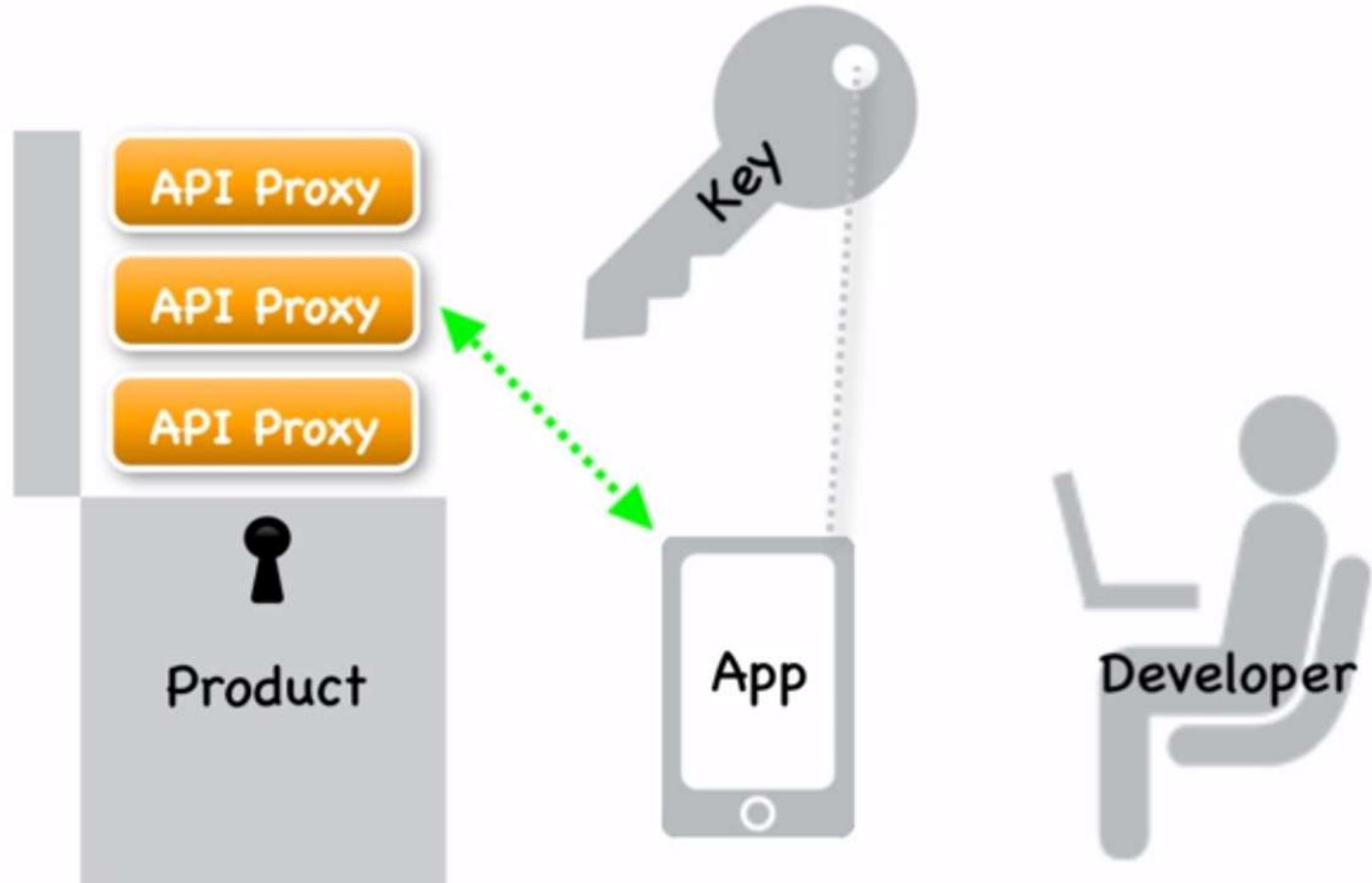
Classification of Open APIs



API Lifecycle



Steps for API Access



Software to Manage API Policy Implementation

- ❑ Manage **workflows** for
 - ❑ **Registration /Deregistration** of Application and Requesters
 - ❑ **STQC shall ensure compliance with security policy**
 - ❑ **API Publishing** (ensure compliance with Gov standards, quality standards, legal issues, time sensitivity, keywords, usage guidelines e.g. for PII's)
 - ❑ **Deprecation** and **version** control of APIs
- ❑ Allow **users** and consumers to **demand new APIs** from API Owners via API provider
- ❑ **Communities** component to allow member Government agencies, industry and citizens to discuss API usage and obtain necessary **technical cooperation**.
- ❑ Provide **unified support** mechanism through **email and chat** to facilitate **usage** of API repository, and receive enhancement requests
- ❑ Support both **REST and SOAP** web services to provide flexibility for data sharing in multiple format i.e. XML (Extensive Markup Language), JSON, KML (Key-Hole Markup Language used for maps), GML (Geography Markup language), RSS/ATOM (fast changing data hourly / daily) and RDF (Resource Description Framework) formats.

Software to Manage API Policy Implementation

- ❑ Provide **non scalable roles** of **publisher, store, admin** and **traffic manager**
- ❑ Provide **scalable roles** for **gateway** and **key manager**
- ❑ Storage types
 - ❑ **Registry Database**-API meta-data storage (Dublin Core standards) , Tenant Key Stores, Documents, Tags
 - ❑ **API Manger Database**- Stores API Runtime data, application data, token data etc.
 - ❑ **Permissions DB**- Stores role to permission and user to permission mappings
 - ❑ **Analytics Summary DB**- Stores API / Application usage summary

Component	Reads only from	Writes to
Publisher	Permissions DB, Analytics DB	Registry, APIM DB
Store	Permissions DB, Analytics DB, Registry	APIM DB
Key Manager		Permissions DB, APIM DB, API Registry
Traffic Manager	Permissions DB	

Proposed Responsibilities of Open API Cell

- ❑ **Coordinate with Government Agencies to:**
 - ❑ **Identify & publish indicative list of APIs** for release e.g.
 - ❑ **Prioritize** and Prepare **schedule** for API release in the next one year
 - ❑ Provide **free-of-cost** technical expertise for **API development**
 - ❑ Help in **drafting SLAs (TAT and update frequency)** related to APIs
 - ❑ Provide required **hand-holding** support
- ❑ **Empanel** agencies for API development funded by MeitY
- ❑ **Promote participation of industry and academia** for free API development
- ❑ Publicize **advantages of API first approach** during services development (**cost savings**)
- ❑ Release **manual for development, nomenclature, classification** (Open, Registered, Restricted) , **registration and numbering** of APIs

Responsibilities of Open API Cell

- ❑ Promote innovative usage of API repository with industry, academia and citizens e.g. [Clear Tax](#)
- ❑ Create a business model for sustenance of API repository
- ❑ Prepare manual for publishing, versioning, deprecation, retirement of APIs and API life cycle management
- ❑ Receive enhancement requests.
- ❑ Create and manage analytics dashboards for visualization of publishing, usage and feedback analysis for monitoring Open API implementation
- ❑ Collate and include existing APIs from applications like Data Gov, eSangam, Umang, [DigiLocker](#) etc.
- ❑ Capacity building
- ❑ Grievance Handling

Responsibilities of Security Cell

- ❑ **Publish - guidelines** on how to design, implement, and consume Open APIs in compliance with the Open API Policy
- ❑ To provide **consultancy** on all security related matters
- ❑ **Empanel** agencies to test security compliance of developed APIs

Risks & Mitigation

Risks

- Insufficient security consideration e.g. Lack of type checking, improper error handling, vulnerability to SQL injections, and inefficient memory overflow handling e.t.c may provide hackers with just enough information to sneak in and steal reams of data.*
- Privacy of customers' data and the potential for fraudulent use of data*
- Insufficient use of encryption at the transport layer may enable an eavesdropper to read and tamper with the data*
- Hackers can exploit the process of license key validations with phony certificates and programs used illegally to grab user credentials and data.*
- Business logic flaws and insecure endpoints*

Mitigation Strategy

- Thus, for successful usage of Open APIs, and mitigate the above risks we must ensure that the API:*
 - Does not contain software bugs;*
 - Does not perform poorly;*
 - Implementation of TSL certificate encryption at the transport layer*
 - Does not contain security flaws; and*
 - Does not leak any private data.*

THANK YOU

Examples

- https://data.gov.in/resources/real-time-air-quality-index-various-locations/api#/Resource/get_resource_3b01bcb8_ob14_4abf_b6f2_c1bfd384ba69
- <https://www.smartnation.sg/about/Smart-Nation>
 - <https://data.gov.sg/developer>
- API management portal with configuration capability
 - <https://docs.apigee.com/api-platform/get-started/what-apigee-edge>
 - <https://wso2.com/library/conference/2017/2/wso2con-usa-2017-building-an-effective-api-architecture/>