Ministry of Electronics and Information Technology
Government of India

Open API Policy Implementation

17-Jan-19
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

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Registered** | • Data which falls under the ambit of Right to Information Act i.e. public  
                   • API definition is visible to all  
                   • Accessed by developer community and agencies having license key (provided by API provider) for innovative usage |
| **Protected** | • Data falls under the registered NDSAP classification  
                   • Only a specific set of authorized agencies shall get access e.g. Aadhaar auth APIs for use by DBT agencies  
                   • Shall result in reduced integration costs |
| **Restricted** | • Data falls under restricted NDSAP classification  
                   • APIs are used for development within the government agency or to users authorized by API owner  
                   • Shall lead to cost savings and operational efficiency |
API Lifecycle

I. API Development
II. API Registration
III. API Publishing
IV. API Promotion
V. API Deprecation and Versioning
VI. API Retirement
Steps for API Access

1. Register Developer
2. Register Application
3. Create License Key
4. APIs are accessed

Source: https://docs.apigee.com/api-platform/get-started/what-apigee-edge
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 GoI standards, quality standards, legal issues, time sensitivity, keywords, usage guidelines e.g. for PII)
- 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 Manager 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

<table>
<thead>
<tr>
<th>Component</th>
<th>Reads only from</th>
<th>Writes to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publisher</td>
<td>Permissions DB, Analytics DB</td>
<td>Registry, APIM DB</td>
</tr>
<tr>
<td>Store</td>
<td>Permissions DB, Analytics DB,</td>
<td>APIM DB</td>
</tr>
<tr>
<td></td>
<td>Registry</td>
<td></td>
</tr>
<tr>
<td>Key Manager</td>
<td></td>
<td>Permissions DB, APIM DB, API Registry</td>
</tr>
<tr>
<td>Traffic Manager</td>
<td>Permissions DB</td>
<td></td>
</tr>
</tbody>
</table>
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_3bo1bcb8_ob14_4abf_b6f2_c1bfd384ba69](https://data.gov.in/resources/real-time-air-quality-index-various-locations/api#/Resource/get_resource_3bo1bcb8_ob14_4abf_b6f2_c1bfd384ba69)

- [https://www.smartnation.sg/about/Smart-Nation](https://www.smartnation.sg/about/Smart-Nation)
  - [https://data.gov.sg/developer](https://data.gov.sg/developer)

- API management portal with configuration capability
  - [https://docs.apigee.com/api-platform/get-started/what-apigee-edge](https://docs.apigee.com/api-platform/get-started/what-apigee-edge)