NeST-GDL-OAPI.01
Version 1.0
February 2020
Status: Released

Implementation guidelines for Open API policy for e-Governance

(National Data Highway)

Government of India

Ministry of Electronics & Information Technology

Metadata of the Document

S. No.	Data elements	Values
1.	**Title**	Implementation guidelines for Open API Platform (National Data Highway)
2.	**Title Alternative**	Implementation guidelines for Open API Platform (National Data Highway)
3.	**Document Identifier** *(To be allocated at the time of release of final document)*	NeST-GDL-OAPI.01
4.	**Document Version, month, year of release** *(To be allocated at the time of release of final document)*	V1.0, February, 2020
5.	**Present Status**	Released
6.	**Publisher**	Ministry of Electronics and Information Technology
7.	**Date of Publishing**	04/03/2020
8.	**Type of Standard Document** *(Policy / Technical Specification/ Best Practice /Guideline/ Process)*	*Guideline*
9.	**Enforcement Category** *(Mandatory/ Recommended)*	*Recommended*
10.	**Creator** *(An entity primarily responsible for making the resource)*	Ministry of Electronics and Information Technology
11.	**Contributor** *(An entity responsible for making contributions to the resource)*	NeGD, STQC and MeitY
12.	**Brief Description**	This document provides the guidelines on how to design, implement, and consume API-based e-Governance systems in compliance with the Open API Policy.

13.	**Target Audience** *(Who would be referring / using the document)*	This document is intended for: technical architects, solution architects, software development teams, hosting and operations teams, and in general any entity that needs to interact with an API-based e-Governance system.
14.	**Owner of approved standard**	Ministry of Electronics and Information Technology.
15.	**Subject** *(Major Area of Standardization)*	API for governance systems
16.	**Subject. Category** *(Sub Area within major area)*	Guideline Document
17.	**Coverage Spatial**	All Ministries /Central /State Government/ PSUs
18.	**Format**	PDF
19.	**Language** *(To be translated in other Indian languages later)*	English *(To be translated in other Indian languages later)*
20.	**Copyrights**	Ministry of Electronics and Information Technology
21.	**Source***((Reference to the resource from which present resource is derived)*	Based on Policy on Open Application Programming Interfaces (APIs) for Government of India
22.	**Relation** *(Relation with other e-Governance standards notified by MeitY)*	Policy on Open Application Programming Interfaces (APIs) for Government of India

Contents

1. Summary

This document acts as a supporting document to the "Policy on Open Application Programming Interfaces (APIs) for Government of India" notified on 21st May 2015, with the objective of assisting organizations for rapid and effective adoption of the policy to their benefit. It provides guidelines on how to design, implement, and consume Open API-based e-Governance systems known as National Data Highway (**NDH**). It's intended audience includes both those who charter and fund APIs as well as technical architects, solution architects, software development teams, hosting and operations teams, and in general any entity that needs to setup and interact with an API-based e-Governance system. The policy is available at

http://egovstandards.gov.in/sites/default/files/Policy%20on%20Open%20Application%20Programming%20Interfaces%20%28APIs%29%20for%20Government%20of%20India.pdf

The document will evolve over time, incorporating the experience gained from several implementations of the policy. The reader is therefore advised to refer to the latest available version for up-to-date content.

2. Introduction

Government of India (GoI) is implementing the Digital India programme to prepare India for knowledge based transformation through digitally empowered Society. Under the program, all Government services should be digitally accessible to citizens through multiple channels, such as web, mobile and common service centres. To meet this objective, there is a need for an interoperable ecosystem of data, applications and processes which will make the right information available to the right user at the right time. In this context, it is important to ensure universal access and interoperability among various e-Governance systems to upgrade the quality and effectiveness of service delivery. In order to encourage the formal use of APIs in Government organizations and to foster the self-sustaining sectoral ecosystems, the India Enterprise Architecture (IndEA) framework specifies API gateway as a core IndEA infrastructure application which can be used by all Ministries/States/Departments and other organizations

An API is an alternative to User Interface (UI), mainly for the purpose of providing a digital service. An API acts as the contract between independently created software entities that interact with each other, without any user intervention. In addition, any number of apps and user interfaces can be created on top of the API. An Open API is one where the API, its definition, design and compliance data is all open to all entities permitted by law.

The underlying vision of the Open API policy is to provide transparency of data, promote participation of the citizens by providing access to data and services offered by various government departments and creating an ecosystem of interoperable systems, processes, workflow and data, leveraging which Indian innovators will create applications that broaden and deepen the reach of Digital India to each and every Indian. The core objectives and corresponding guidelines of the Open API Policy are described in the Open API architecture below. This furthers the core e-Governance objectives of *Transparency, Universality, Legacy-free, Security and Innovation*, seeks to with minimum Government effort and expenditure, create an innovative ecosystem of apps with the dual goals reaching each and every Indian composed in the form of solutions they need. In parallel this approach is designed to empower the most innovative of Indian enterprise system builders and application makers who will create both the APIs as well as the Applications on top.

3. Purpose of document

— Used as a guideline by various Government of India departments to develop, publish and implement the APIs for their e-Governance Systems (services, e-Governance applications and systems)

— To help quick implementation and transparent monitoring of APIs and its usage.

— To ensure all participating APIs conform to the vision enshrined in the policy and recipes encoded in this document with minimal effort and total cost, while reaching every Indian

4. Structural Realities

The following structural realities are accounted while proposing a broad architecture:

1. **Centre and State**: e-Governance systems will be built both by Centre and States at their own pace. The Centre will sometimes lead the way and sometimes a state. Some states have needs that not every other part of the country has. So, the API architecture must allow for bottom up local innovation while ensuring that such innovation fits into the national data and service fabric
2. **Portability**: Indian citizens are mass migrating across cities and from rural to urban areas. At the bottom levels of the pyramid, such migrants are often deeply disempowered. A modern eService system must continue a citizen's identity, services and rights seamlessly across geographies subject to local law.
3. **Technology Churn**: Technology changes and must change constantly to keep up with best patterns, lower costs and to improve quality, speed, scale and ease of use of services. Any proposal hence by definition must think of these changes but stay away from baking in any existing architecture as writ, as this slows India and prevents innovation.
4. **Security**: Governance both citizen facing and otherwise will become increasingly digital and connected. Integrity, Security, Safety and Privacy of information + citizens

are then a paramount criterion and must be addressed at every level including in this architecture

5. **Legacy**: A myriad of systems and services exist already. These guidelines and blueprint provide a way for new systems to come up in the new paradigm while older ones have a way to migrate gently and get connected to the new fabric at a gentle but time bound pace.

5. Benefits of an API to e-Government initiatives

While there are several benefits of a good API implementation, here are a few important ones:

1. **Ease of development**–An API approach helps separate frontend components (user facing views like web applications, Apps and so on) from the development of scalable backend software (servers, data systems), thus easing development process itself. An API therefore provides the necessary separation between two pieces of software and creates independence in their respective development, innovation and release management.
2. **Lower Cost of Universal Service**-With the above noted separation, onus of creating each and every interface required for every possible need shifts to the private innovator realm, freeing up the Government to focus on core system and API creation. This has a dual benefit of lower total cost to Government while increasing the reach and usefulness to citizens and businesses significantly.
3. **Consistent design**–An API-first approach helps in bringing inconsistencies in design and access by other applications, upfront significantly increasing reusability both in India and globally.
4. **Low cost of maintenance**–Maintenance (or fixes to software) do not spread across multiple parts of software separated by API (unless the API itself changes); thus limiting the cost of software maintenance, as one can be sure about how far effect of changes would be felt. This contains ripple effects of changing any one system simplifying and speeding up innovation. E.g. UPI servers don't need to even be aware if Aadhaar servers are significantly changed.
5. **Low cost of integration**– Lower cost of integration is an expected benefit, as APIs are meant to act as a pre-agreed contract between two integrating applications. Well-designed APIs foster interoperability and integration.
6. **Higher security**–An API becomes a well-defined entry point across different applications, isolating the internal implementation from the world, making it easier to implement and test necessary checks and controls.

7. **Higher availability**–It is generally easier to monitor and ensure availability standards of a service when the access point between two applications (the service and the app using the service) happens to be through an API. This assures scalability and technology independence.
8. **Data availability to public**– APIs are a means for both data collection by Government Organisations and to make Government information and data available to public consumption.

6. Open API TULSI Principles and Architecture of National Data Highway

6.1 Open API TULSI Principles

The following core principles define and strengthen APIs and are fundamental to a successful implementation of a large scale API ecosystem which grows and evolves over time. All APIs must conform to these principles and if possible improve upon the recommended practices but at least ensure the practices so suggested at a minimum.

API protocol is an Information architecture for digital governance of India. It consists of: *Transparent Universal Legacy-free Secure and Innovative.* These are core principles which will be used to shape an information architecture for India:

1. **Transparent:** All public information, processes easily accessible to every Indian
2. **Universal:** Empower all GoI organizations to participate in information sharing and equip all citizens with information governance
3. **Legacy-free:** Future proof e-Governance systems by designing them for constant change and evolution
4. **Secure:** Ensure the services are secure, data is protected to the fullest extent of the law and privacy of Indian citizens, residents and organizations are fully honoured.
5. **Innovative:** Empower Indian innovators to build, deploy user, government friendly apps. Significantly increase the speed of building, deploying of India's e-Services, apps

6.2 High level Architecture

The NDH architecture consists of API Directory, API Gateway and API Portal which shall help in monitoring Key management and enable Publishers and Consumers to connect with each other for consumption of API in a balanced and secure manner, (figure 1).

The API service consists of a predefined Request-Response in the form of XML / JSON as defined. A Request sent by an API consumer to an API gateway generates a response consisting of a predefined collection of data corresponding to the content of the Request. The entity that responds to the Request is the API Owner. Every request contains a header which consists of information regarding the individual (API consumer) making the Request, institution of the individual making the Request, etc. In addition to the standard API service described above, the NDH can be used to transfer documents, files and large structured datasets.

API Owners and Consumer that join NDH will have to register their users and applications on the NDH Portal. API consumers will be able to access APIs and API Owners will be given access to dashboard and rights to publish their APIs on the portal for consumption.

The NDH infrastructure consists of security servers, NDH gateway servers, certificate (key management) servers and monitoring servers (figure 1). The NDH security server is the first point of contact between the API consumer/owner and portal/gateway, Consumer request will be processed only with API key provided by API key management system. All data travelling over the NDH is transported via the secure servers. The NDH Gateway Server shall manage API request/responses across multiple servers in a federated manner. Security servers encrypt/decrypt data, generate usage logs, control usage rights to services and prevent unauthorized access. The use of security servers ensures institutions that their data is travelling securely over the internet.

The NDH uses monitoring servers to monitor the status of servers sharing APIs and gather usage statistics. (Figure 1)

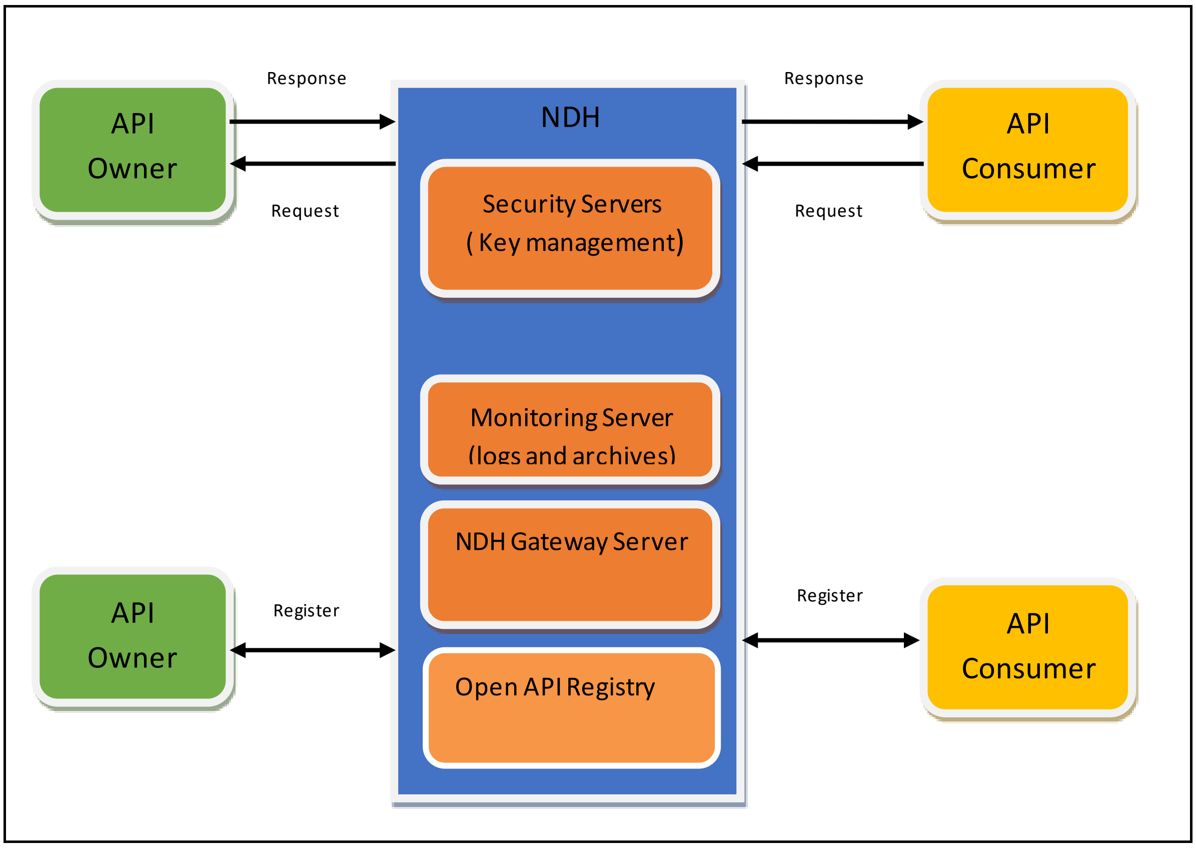

Figure 1: Server and API Architecture

The NDH High-level architecture is described in the following drawing. (Figure 2)

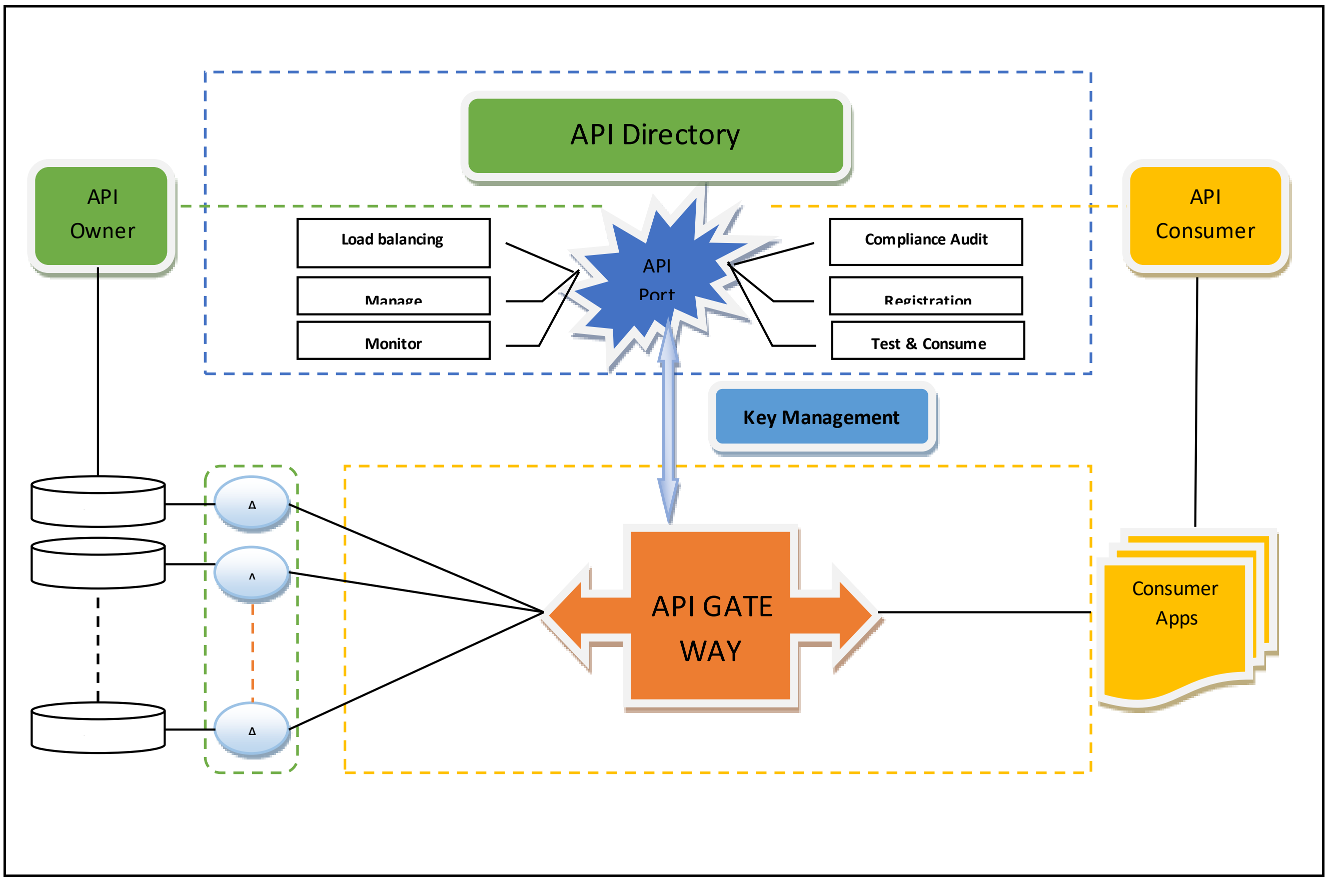

Figure 2: High level Architecture

7. Roles and Responsibilities

7.1 Targeted Stakeholders

- Public agencies (including policy decision-makers and governing bodies like NeGD, NIC, CDAC, NISG)
- Owners of e-Services (Central Ministries and State Government Departments)
- Citizens/Public at large which are users of e-Governance Services
- ICT industry (playing the roles of suppliers, developers, implementers and maintainers like various industry technology bodies)

In the context of an API architecture, there are three distinct roles: the API Owner, API Gateway, and the API Consumer as illustrated in the diagram below.

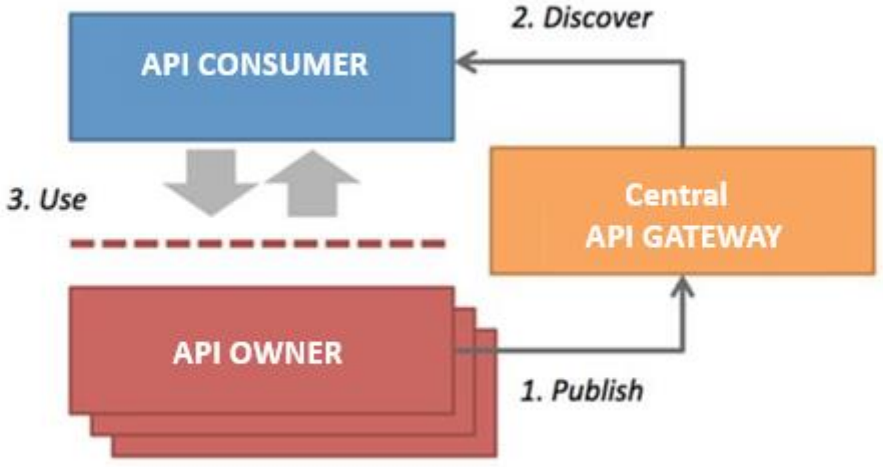

Figure 3. Roles of API-driven Integrated Applications

7.1.1. API Owner

The Government organisation or other organisation who is the ultimate owner of this API - who is responsible for the underlying governance function and is also responsible for defining, ensuring and chartering their vendors if any.
An organization that is responsible for implementing the software and exposing its API for others to use.

- Define and publish terms of use (including SLA) to restrict the way the API is consumed in accordance with this guidelines document
- Modify the implementation of the API at any time, as long as the new implementation complies with the API definition and the agreed SLAs are met including compliance with this guidelines document. NDH implementation team will itself undertake to build a mechanism to ensure compliance of every API exposed via NDH with Open API Specification 3.0+ and all other recommendations in the implementation guidelines so that the burden of specification on individual API Owners is reduced.

7.1.2. Open API Registry

Every API owner may register details of their APIs on the Open API registry. This will be adhered to even in case of the API not currently available on NDH Gateway

7.1.3. API Gateway

API gateway will comprise of Central servers, load balancers, key manager. Its responsibility to ensure smooth registration process, compliance audits, API life cycle management, API monitoring on the basis of agreed upon KPIs and SLAs and provide test beds for handshaking between API Owner and Consumer. It will serve as a secure channel for consumption of API service provided by API owner. Communication between the 2 Consumer and Owner will happen via predefined request response formats. It is highly recommended that the APIs provided by the API Owners be published in Open API Specification 3.0+ format although legacy APIs that are already available may be published on the NDH API Directory. All the APIs that have been successfully integrated to be available via NDH will be deemed compliant to Open API Standards. The APIs published by API Owners may also be accessed directly by API consumers from outside of NDH Gateway ecosystem. In other words the access channel for these APIs is a direct peer to peer integration between API Owner and API consumer or an integration via the NDH Gateway.

Refer Annexure II: API Specification Documentation template

Refer Annexure III: API XML Template Sample definitions

7.1.4. API Consumer

An API consumer might be another organization or government agency or an individual citizen, resident or other user or an Indian organisation or other organisation with a registered application for use of an API. API consumers are free to use the published APIs in compliance with the terms of use subject to authentication and authorisation under which the API use is permitted. API consumers can rely on the published API definition without worrying about the specific implementation details (including implementation platform, internal architecture, etc.).

API Consumer has the responsibility to strictly adhere to the terms of service as defined in these API guidelines.

7.2Governance Bodies

The API Protocol which includes the architecture also builds in technology governance bodies which will in turn aid creation of a strong, secure and nimble API ecosystem which uses and evolves globally best practices and technology. The governance of this protocol is primarily the responsibility of the API Owner where compliance is ensured with rigorous but light-touch data driven, self-certification based principles.

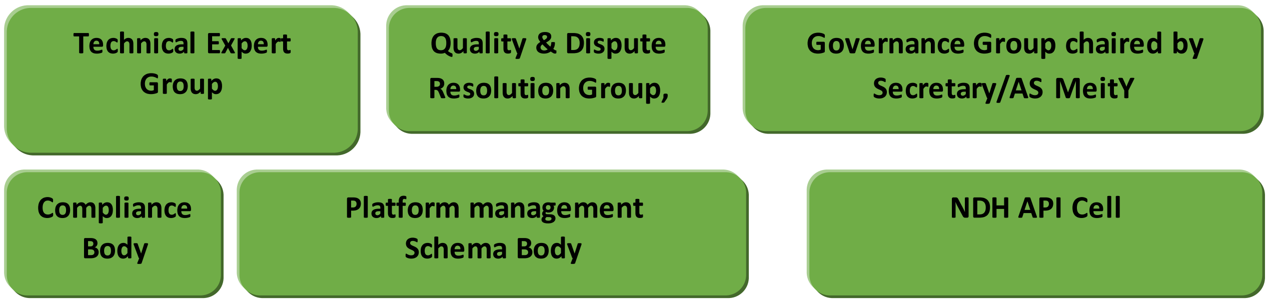
Technical Expert Group
Quality & Dispute Resolution Group,
Governance Group chaired by Secretary/AS MeitY
Compliance Body
Platform management Schema Body
NDH API Cell

Figure 4. Governance bodies

7.2.1 Governance group

Governance group will be setup under the chairman ship of Secretary/ AS MeitY and consists of Nodal officers from respective ministries and other members to monitor progress, facilitate issue resolution and provide overall guidance. NDH API cell will assist this group.

7.2.2 NDH API Cell (PMU)

NDH API Cell will be responsible for administering the management of the API ecosystem. The NDH API Cell will create three independent governing bodies which are in turn chartered by Meity to help evolve and ensure compliance with the guidelines. These are the Schema Body, the Security Body and the API Dispute Resolution Group.

The NDH API Cell will coordinate with Government Agencies to:

- ❑ Identify, Classify, Prioritize & publish the list of APIs for release
- ❑ Prioritize and Prepare schedule for API release in the next one year
- ❑ Help in drafting SLAs (TAT and update frequency) related to APIs
- ❑ Empanel agencies for API development
- ❑ Promote participation of academia and industry including Indian Software Product Companies as identified under National Software Product Policy for prioritized API development

- ❑ Publicize advantages of API first approach during services development (cost savings)
- ❑ Maintain Implementation guidelines for NDH
- ❑ Awareness and communication
- ❑ Support the Platform management and schema body in marketing and resolution of implementation issues
- ❑ Explore and create a business model for long term sustenance of NDH
- ❑ Provide support to the primary governing bodies – Governance group, Platform management and Schema Body, Compliance Body, Dispute Resolution Group.
- ❑ Generate and publish a quarterly NDH API highlights and SLA report.
- ❑ Reward early adopters among Ministries/ Departments/ States for excellence in design or implementation via public, national recognition as best in class governance innovators and technologists.

7.2.3 Platform Management and Schema Body

Schema is the language of any API. This includes the digital language used by the APIs to initiate, communicate and represent various messages. The charter of this body is to encourage both API Owners as well as the Indian software community to propose creation or improvement of schema definitions of entities, critical service messages, transactions and operations.

The key responsibilities of the Platform Management and Schema Body is to:

- ❑ Create and Maintain the API Directory.
- ❑ Marketing of NDH platform
- ❑ Coordinate with compliance audit body and ensure timely publication of APIs.
- ❑ Create Gateway interface for Registration, API life cycle Management, API Monitoring dashboards, key management, and load balancing.
- ❑ Create communities to encourage expert participation and crowdsourcing
- ❑ Create separate workflows for Owners to publish and manage API and one for Consumers to discover and consume API to be deployed.
- ❑ Provide technical support and test bed for Owners as well as consumers for seamless use of the portal and integration with Gateway
- ❑ Provide technical expertise for API development and hand-holding support
- ❑ Release manual for development, schema, classification (Open, Registered, Restricted) , registration and numbering of APIs refer Annexure IV: Detailed Guidelines for API Owner
- ❑ Implement and maintain any common infrastructure as detailed (e.g. NDH API Directory)

- ❑ Monitor quality and ensure consistency and that overall quality keeps improving using reports from the governing bodies, especially the Dispute Resolution Group - focused on user feedback (both by using usage levels and stickiness as a proxy as well as from direct user feedback).
- ❑ Raise all implementation issues to relevant stake holders and governing bodies
- ❑ Maintain security operation centre to ensure privacy and security of data sharing at all times.
- ❑ Ensure unified support mechanism through email and chat to facilitate usage of API directory and receive enhancement requests

7.2.4 Compliance Body

This Body's domain is to evolve security requirements, suggestions, checklists and testing tools such that these are informed by global best practices but Indigenous using scientific principles.

Compliance Body should take a holistic picture of security to make its recommendations to provide the best possible end-to-end security with minimal intervention and reasonable effort, suggesting changes to existing system if required. This body shall do a compliance audit for physical server, legal security, localisation security, network security, service security, user authentication and authorisation service security and impregnability. This body's role is to use a set of technical processes and tools to conduct a continuous monitoring and best effort certification of compliance of APIs to their relevant guidelines.

This Group may also over time recommend the development of a launch checklist and Engineering practice (LCE). For low risk and low volume APIs, self-certification by a technically sound API Owner is sufficient. For moderate risk and moderate volume APIs, self-certification and a fully complete checklist by a technically sound and experienced API Owner is sufficient. For high risk or high volume APIs, the LCE checklist must be completed and reviewed by competent organisation. Such organisations need to be empanelled to conduct LCE reviews as well as compliance audits using both advanced tools and technical experts.

7.2.5 Quality & Dispute Resolution Group

Resolution of this body shall resolve disputes related to data sharing amongst ministries government agencies, States, industry, communities, and Individuals. The Dispute Resolution Group can seek advice of technical advisory group. Grievances related to API forwarded by competent authorities and related legal matters will be handled by this group. This group shall include experts from cyber security.

7.2.4 Technical Expert Group

Purpose of the Technical Expert Group (TEG) is to give guidance and consultation with respect to the technical, standards and security related aspect in Implementation of NDH as well as overall Open API ecosystem. The TEG is a volunteer group adjunct to the Compliance Body. This group comprises of global experts, experts from state, NIC, Academia, CDAC, communities including Cyber Security and industry.

The Body's charter is in addition to helping with selecting exceptional members into the Compliance Body is to suo moto advise on any, all matters, guidelines, practices and other work, output of the Compliance Body and Dispute Resolution Group. It will also develop supporting tools for the API builders, suggest SLAs, assist Schema Body in formulation and evolution of schema standards, resolve any schema disputes. The TEG is further empowered to crowd source using tools of its choice, input from the broader technology community. This input may be in the form of ideas, improvements, and threat discovery and newly identified potential bugs or hacks that compromise NDH. The consultants from the API cell shall assist the advisory body to ensure all this effort is organised, its output summarised, prioritised by the body and communicated to the relevant stake holders transparently in a timely manner. TEG may suggest mechanism to maximise value of Open API ecosystem, NDH to the Indian economy.

The TEG will be given access to all logs data and dashboards of the NDH and those provided by Open API Owners, and may ask for and should be provided further reports as soon as available from the other Open API bodies to analyse and formulate their advise as per their charter.

The Expert Group will also advise on matters, guidelines, practices and other work related to security. The Technical Expert Group is further empowered to crowdsource using tools of its choice, input from the broader technology community. This input may be in the form of ideas, improvements, threat discovery and newly identified potential bugs or hacks that compromise eServices.

All members of the TEG must be of Indian origin (passport or OCI holders) where more than half must be residents of India. This body is a pure volunteer body where their only obligation is to review material sent to them, to suo moto alert the Compliance Body of issues they find and to respond to specific questions and request for opinion, especially during potential security events or emergencies. It is suggested that the members of this body meet and present to the Compliance Body at their convenience at least once a year.

8 Risk and Mitigation Stratergy

Risks

Insufficient security consideration e.g. Lack of type checking, improper error handling, vulnerability to SQL injections, and inefficient memory overflow handling etc. 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

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 complies with OWASP guidelines, and does not leak any private data.

9 Security Protocol - Principles and Architecture of Core and five Security Levels

Ensure eServices are secure, data is protected to the fullest extent of the law and privacy of Indian citizens, residents and organizations is fully honoured. The TEG is responsible for creating a sub-body within the TEG called the Security Body, where the members are a subset of the members of the TEG, appointed and managed by the TEG to allow TEG to provide a unified advisory as good security is designed into every layer of a system.

Core Security Framework

All new eServices must confirm to these security guidelines. The Security protocol is satisfied by the sub-protocols of System Security, Authentication and Testability.

System Security

Five levels of security are defined as noted below. Any given eService is assigned a security level by the eService Owner using a spreadsheet model developed and published by the TEG with further consultation with the Dispute resolution group if they so choose. Exceptions to lower security level vs the model generated security level must be approved by the Dispute resolution group and sent to the Security Body for an optional review if they so choose.

Legal Compliance - All implementation, data safety and processes must comply with the relevant laws which must be identified by the eService Owner at the time of specification and registered in the OSR. The measures taken by the API Owner must be reviewed and published openly.

All of the above will be defined, maintained and made available by the TEG.

As a starting point the following are the minimum security requirements across all levels:

•All API or inter-server Communication and messages must be encrypted via HTTPS + PFS and at least with the best standards that popular modern browsers or servers support

•All servers used must be firewalled and inaccessible to the outside world except via a specific set of servers such as a DMZ resident proxy via the published API addresses and request messages.

•Firewalls, TLS 1.3+, HTTPS certificates and data-centers / cloud services are either pre-certified by the Dispute resolution group or specifically reviewed and certified to comply with Security Body guidelines

•All Server to Server communication must be either via HTTPS + PFS using TLS 1.3+ and TLS 1.3+ for non-web layers

•Further, all server to server (including API to API) messages carrying sensitive data as defined by the security body should be secured using TLS 1.3+ (or equivalent as approved by the Security Body).

•All data that is not declared public shall be encrypted when stored to disk.

•All data of value whose integrity is important (even if public) shall have an audit chain - i.e. change history include who changed it, when, what was the prior state, with what/ whose approval and so on. For high levels, this audit chain shall be tamper proof using a private block chain or analogous systems as approved by the TEG.

•All private data shall have logs of access to such data. Depending on the level of security as prescribed by the TEG, the access logs must themselves be tamperproof using approved approaches or technology. Further, as defined by the relevant levels of security, alerts of access to private data - authorised or unauthorised must be recorded, disclosed to the owner and custodians of that data whose role (again defined by Security Body) is to act proactively on behalf of the ultimate owner and ensure the said access was legitimate and when not, take suitable remedial action which may include notifying the relevant judicial body pro-actively.

•The TEG may require that both private key and encrypted data must never be persistent in memory at the same time for certain levels of security.

•All access to the servers, certificates and private keys must be via listed, authorised engineers and manager only and all access to these must be logged for at least 36 months - this may be softened or strengthened for different levels of security.

•All messages sent by/ to API/ eService Consumers must be screened for js, html, and script injection unless the schema explicitly permits this for specific fields

•All session based (e.g. logged in user) services must be protected against common x-site scripting, phishing and other common identity theft practices.

•To minimise cost, all security certificates can be self-generated at least until level three of four using technology and procedures or systems defined by the Security Body. All paid certificates or security systems if any must be using systems and IP owned by Indian Companies. The Security Body shall endeavour to ensure at least two providers of any private services or tools prescribed it exist unless a significant innovation reduces overall costs or at the same cost significantly increases quality of security - in which case, an exception may be made for up to five years.

•All security protocols and standards must be open, free and public. Any proprietary standard or technology if prescribed or suggested must be of rare use and only when it's financial and IP ownership is Indian to ensure long term integrity. And for these rare inclusions a full voting by eService Owners, eService Consumers and eService Owners is required with special attention paid to suggested alternatives, where final decision vests with combined voting by the Security Body and Security Advisory Body.

It is suggested to make the security guidelines encouraging - i.e. to reward eService Owners/ Implementers with higher rating as well as better compensation subject to their demonstrated and measured quality of security over the life of the service. Punitive measures such as fines, payment withholding may be applied only in the case of gross negligence, malicious intent or failure to comply after repeated requests by the Dispute resolution group or eService Owner or within agreed upon time after notifying them of a clearly identified, repeatable, notified security issue. This creates an environment of open collaboration. Very strong punitive measures without giving due leeway and consultation are detrimental to system security.

When an implementer or owner claims that their eService even if not compliant in letter with the noted guidelines is however in spirit at least as secure or more secure, that matter is referred by the Dispute resolution group to the Security Advisory Body who then makes its recommendations along with suggestions on how to ratify this claim. Such claims when ratified by the Security Body with assistance from the Dispute resolution group (for measurement and other due diligence), lead to three actions - approval to that implementer, suggested modifications to existing guidelines to allow this innovation (subject to due quiet period and at least two more beta implementations) and public recognition of the innovating individual or organisation.

Security Levels and Corresponding Framework

1. Low risk or low volume (100K active users) or low transaction size/ volume or open data: Self-certification by Implementer is sufficient

2. Moderate risk or moderate volume (1M active users) or moderate transaction size/ volume or personal data: Self-certification and a fully complete checklist by the implementer signed by the technical lead of the implementation team and their manager.

3. High risk or high volume (10M active) or high transaction size/ volume or sensitive personal data: The Launch checklist must be completed and reviewed by an empanelled private Indian Company with world class experts and expertise in digital system security. Such companies need to be empanelled to conduct Launch Security reviews as well as compliance audits using both advanced tools and technical experts.

4. Very high risk or eService essential for state / metro level essential services or very high volume (100M active) or very high transaction size/ volume or extremely sensitive or critical data: In addition to above, an architecture review and cataloguing what if scenarios by empanelled national scale system security reviews which must be successfully addressed by the implementer with time bound requirements for corrections if any. Logging of identified critical events including a list of events noted by the Security body (e.g. DOS attacks, phishing incidents) and publishing of this data to Dispute resolution group.

5. National security impact eService or National Scale (500+M active) or National level risk or eService essential for national scale essential services or National scale transaction system or National Scale privacy/ authentication/ identification system: Same as above. In addition create bug bounties and hackathons to probe and discover weaknesses in premier academic institutions open to all Indian experts conducted by the Security Advisory Body.

Apart from increasing level of scrutiny and rigor for each security level noted above, the Security Body is chartered to create quantifiable guidelines encoded in a spreadsheet or similar software to accurately identify the recommended level for any proposed or existing eService. And further, to codify the checklist of security requirements, practices, logging, quality bar and processes (including code management and audit processes) for each level.

These per security level checklists and minimum limits so encoded in the checklists will be the pivotal compliance tool for meeting the minimum bar for said security level. The checklists shall be online, electronic and can include aggregated, statistical data from logs as well as data gathered by the Dispute resolution group. The checklist can be updated by the implementer, owner and Dispute resolution group with suitable notice and review at any time including from real-time aggregation of log data. This checklist will then emit a signal indicating both the current measured security level of the eService along with compliance strength. If either of these is out of bounds on the negative side, the Dispute resolution group, Owner and Implementer are simultaneously alerted, where the onus is on the Owner and Implementer to respond to the alert suitably to bring the eService back into compliance (e.g. confirming to a higher level if the traffic has increased substantially)

Authentication and Authorisation

All eServices requiring sign-in must be accessible via a common, Security Body published Authentication and Authorisation interface. A Single sign-on protocol designed jointly by the Schema and Security Bodies using globally best, secure and simple will be used to implement single sign-on systems. The single-sign on authentication mechanism should explicitly allow for interoperability across all common platforms, all eServices and APIs and if possible, with the broader internet ecosystem. Re-authorisation during a session should only be required using the lightest weight approach required during sensitive access or transactions (e.g. Payments) as defined by the API Owner, as well as refresh authentication protocols reviewed by the Security Body. eServices that serve purely, globally public data can optionally require the lightest weight authentication and only to prevent large scale hacking or DoS attacks. Cross API and service Authorisation will be via the OAuth 2.0 framework. This mechanism is to be used by all eServices and APIs to allow Composability across multiple APIs and applications to access data on behest of the user (delegated authorization or scope grants).

Concerned Administration shall decide the Authentication and identity framework with input from the Schema Body and technical community. OpenID Connect is suggested as the authentication framework where IDs registered or salted with the user's Aadhar ID or VID or Passport number or DIN or any allowed, permanent Indian ID as determined by concerned administrations eSign platform as a valid KYC.

No such registration is needed for accessing completely public data and services such as consulate pages, applications for the said IDs, open public services like railways and so on - as defined by the API Owners. In case the API is serving purely public services, a relaxed restriction is fine where the user can be authenticated via their phone number or via a global auth token such as Google, MS or Facebook. For purely public data, anonymous users are acceptable if approved by the API Owner.

This entire system shall be reviewed and ratified by the Security Body.

Testability

The eService Owner must define and enter into the OSR with each registered eService, the level of security that the API must comply with using the process noted above. As discussed, the Security Level will be chosen jointly by the eService Owner and eService Implementer after completing a checklist published by the Security Body, with open questions to the Dispute resolution group if necessary - the data and final levels of which shall be entered into the OSR. The Dispute resolution group may choose to review or question this choice based on uploaded data and other data they may receive. Compliance and conformity to the level of security of an eService is then the responsibility of the eService Owner ensured by the following mechanism:

•Self declaration and completion of a checklist published along with each level

•Auditing of accuracy of self-declaration if and as noted by each level's prescription

•Black-box testing using software and services published by the Security Body for each level of security defined.

•Instrument their logs to check for potential breaches and attacks

All data from testing, breaches or attacks generated from logs or reported to the API Owner shall be entered into the OSR for review by the community and Dispute resolution group.

The checklist as well as the black-box software will continuously evolve and a reasonable time will be given to the API Owner to keep up with new recommendations. Both these will evolve using an open, peer reviewed process governed by the Security Body.

Annexure

Annexure I: Definitions and acronyms

API (Application Programming Interface) - is a set of functions, data definitions and protocols which programmers can use when building software that uses an e-Governance system. In software terms, an API definition becomes a pre-established contract between two software applications for exchange of data or functionality.

API Consumer is an organization that builds Apps that use an e-Governance API. Sometimes, this document uses API Consumer to refer to the actual App itself.

API Owner is the Government organisation or other organisation who is the ultimate owner of this API - who is responsible for the underlying governance function and is also responsible for defining, ensuring and chartering the API Owner

API Publisher is an organization that builds the e-Governance application implementing the API and provides as a service (Web Service) for integration with Apps built by an API Consumer. Sometimes, this document uses API Owner to refer to the actual Application or service itself.

App is a general term used[1] for a mobile or web application that uses an e-Governance system.

HTTP (Hyper Text Transfer Protocol) is the popular communication protocol used by web sites and applications.

Open API is an application with a published programming interface by the service owner organization (e-Governance application) that can be used in building integrated systems. Documentation and access to an Open API are available for eco-system app developers to utilize during their app development.

REST (Representational State Transfer) is a simple stateless architecture that generally runs over HTTP and hence platform neutral. REST is a popular approach to development of web services and used by most popular web services around the world. When Web services use REST architecture, they are called RESTful APIs (or REST API for short).

SLA (Service Level Agreement) is a commitment between organizations that integrate their software through an API. An SLA is designed by the service owner organization that

[1]While App normally refers to a mobile application, this document uses it as a common term for both mobile and web applications.

specifically defines what the user of API will receive. In some cases, it might be simple "Terms of Use" and in others it could be elaborate API subscription plan including rate limits, pricing, uptime considerations, etc.

SOA (Service Oriented Architecture) is a technique that involves the interaction between loosely coupled services that function independently. Such an approach is used to create software designed on creating and using of services. Here, services (such as RESTful Web services) carry out some small function, such as producing data, validating a customer, or providing simple analytical services.

SOAP (Simple Object Access Protocol) is an alternative approach to REST, for building web services. REST's decoupled architecture and lightweight communication protocol has made it more popular than SOAP.

OAS (OAS 3.0+) – OpenAPI Specification is an API description format for REST APIs. It allows you to describe the entire API, including API endpoints, operations, input and output parameters and authentication methods..

Web Service – A web service (or service, in short) is a running e-Governance application that provides data/functionality access through a published API to other software applications.

NDH - National Data Highway

NDSAP - National Data Sharing and Accessibility Policy

LCE- Launch Checklist and Engineering practices

OSR – Open Security Registry

Annexure II: API Specification Documentation Template

This is a set of example schemas to use for common interactions of Open APIs which may be jointly evolved in the first few meetings of the Schema Body and TEG

A. Methods

1. login

Authenticate the user with the system and obtain the auth_token

REQUEST

Method	URL
POST	api/login/

Type	Params	Values
HEAD	api_key	string
POST	username	string
POST	password	string

api_key

api_key must be sent with all client requests. The api_key helps the server to validate the request source.

RESPONSE

Status	Response
200	{ "auth_key": <auth_key> } auth_key(**string**) - all requests to
403	{"error":"API key is missing."}
400	{"error":"Please provide username."}
400	{"error":"Please provide password."}
401	{"error":"Invalid API key."}

401	{"error":"Incorrect username or password."}
500	{"error":"Something went wrong. Please try again later."}

2. get updates

Get the new updates

REQUEST

Method	URL
POST	api/updates/

Type	Params	Values
HEAD POST	auth_key version	string number

auth_key

The auth_key that was given in response to /api/login

version

The current version of internal recipe database. Each time when updates are pulled from the server through the web service, the internal database version is incremented.

RESPONSE

Status	Response

200	**Response will be an object containing the list of recipes (array) as well as the updated recipe database. Each item in the recipe array has the following structure.**
	
	{
	    "recipe_id": 10,
	    "title": "Green Chilly Salad",

	```
	    "category": 1,
	    "ingredients": {
	        "Green Chilly": "1 kg",
	        "Salt": "0.5 tbsp"
	    },
	    "steps": [
	        "First clean and cut the chillies",
	        "Now you can eat."
	    ],
	    "remarks": "serves 2 people"
	}
	```
	 An example response is:-
	```
	{
	    "recipes": [
	        {
	            "recipe_id": 10,
	            "title": "Green Leaf Curry",
	            "category": 1,
	            "ingredients": {
	                "Green leaf": "1 kg",
	                "Salt": "0.5 tbsp"
	            },
	            "steps": [
	                "First clean and cut the leaves",
	                "Now you can eat."
	            ],
	            "remarks": "serves 2 people"
	        }
	    ],
	    "version": "4"
	}
	```

400	{"error":"Please specify database version."}
400	{"error":"Invalid database version."}
401	{"error":"Invalid API key."}
500	{"error":"Something went wrong. Please try again later."}

3. deletions

Get the recipes that were deleted from the web interface, so that they can be deleted from the internal database also.

REQUEST

Method	URL
POST	api/deletions/

Type	Params	Values
HEAD POST	auth_key version	string number

version

The current version of internal database. Each time when updates are pulled from the API, the internal database version increases.

RESPONSE

Status	Response
200	**An array containing the ID's of recipes to delete is given** Example response:- {"deletions":[10,11,40], "version":"5"}
400	{"error":"Please specify database version."}
400	{"error":"Invalid database version."}
401	{"error":"InvalidAuth key."}
500	{"error":"Something went wrong. Please try again later."}

4. get recipe image

Get more information on a particular recipe

REQUEST

Method	URL

GET	api/image/<recipe_id>/

Type	Params	Values
HEAD URL_PARAM	auth_key <recipe_id>	string number

recipe_id

Id of the recipe you want the image of.

RESPONSE

Status	Response
200	**An array containing the ID's of recipes to delete is given** Example response:- {"image":"http:\/\/example.com\/recipe-5-image.jpg"}
400	{"error":"Please provide recipe_id."}
400	{"error":"Invalidrecipe_id."}
401	{"error":"InvalidAuth key."}
500	{"error":"Something went wrong. Please try again later."}

B. Glossary

1. Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All response are in JSON format.
- All request parameters are mandatory unless explicitly marked as [optional]
- The type of values accepted for a *request* parameter are shown the the values column

like this [**10**|<any number>] .The | symbol means *OR*. If the parameter is [optional], the default value is shown in blue bold text, as **10** is written in [**10**|<any number>].

2. Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX -Success of some kind

4XX -Error occurred in client's part

5XX -Error occurred in server's part

Status Code	**Description**
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

Annexure III: API XML Template Sample definitions

Below is a sample Request response format for getting documents for student education certificate from University

Sample Data Request Format

```
<?xml xmlns=”http://meity.gov.in” version=“1.0” encoding=“UTF-8”?>
<VerifyDataRequest   ver=“1.0”   ts=“YYYY-MM-DDThh:mm:ssZ+/-n.n”   txn=““
verifierId=““ keyhash=“SHA256(API Key+ts)” >
    <DataConsumer id=“in.gov.mea”/>
    <DataProvider id=“in.gov.cbse”/>
    <Data docType=“HSCER” format=“xml/pdf/both”/>

    <Citizen uid=““ fullName=““ dob=““ mobile=““ email=““/>

    <Parameters>
      <Param key=“URI” value=“in.gov.cbse-HSCER-1234567”/>
      <Param key=““ value=““/>
      <Param key=““ value=““/>
                .
                .
      <Param key=““ value=““/>
    </Parameters>

    <Consent id=““ src=“CF/VF” purpose=““/>

     <Signature> Digital Signature of requester </Signature>

  </VerifyDataRequest>
```

Various elements/attributes in the request are described below-

Element: VerifyDataRequest The envelope element for the verify XML request.		
ver	M	API version.

ts	M	A timestamp value. This will be used to decode the keySign element described below. YYYY-MM-DDThh:mm:ssZ+/-n.n (derived from ISO 8601). Time Zone should not be specified and is automatically defaulted to IST (UTC +5:30).
txn	M	A ***unique*** transaction id provided by the verifier. This will be used to uniquely identify the request by the service endpoint. It is recommended the verifier application prefix a unique sequence.
verifierId	M	verifierId is the id provided to the Verifier by DigiLocker application.
keyHash	M	Provide SHA-256 hash value of the API key and the timestamp values concatenated together in this sequence. The verifier can obtain the API key from their verifier account on Partner portal of DigiLocker application. You must use the same timestamp value that you have specified in the *ts* element described above.

Element: DataConsumer
Contains the details about the verifier organization.

Attribute	Mandatory (M)/ Optional (O)	Description
Id	M	The verifier id provided by DigiLocker application upon registration on DigiLocker Partner portal. This id is available in your verifier account on DigiLocker

Sample Data Response Format

```
<?xml xmlns=”http://meity.gov.in” version="1.0" encoding="utf-8"?>

<VerifyDataResponse  status="1"  statusMsg=””  ts=”YYYY-MM-DDThh:mm:ss+/-
nn:nn” txn=""/>

<DocDetails>

                <DocContent>

                        //PDF content encoded in Base64 in string format

                </DocContent>

                <DataContent>

                        //Certificate data in XML format

                </DataContent>

        </DocDetails>

</VerifyDataResponse>
```

Element: VerifyDataResponse The envelope element for the verify XML response.		
status	M	Response status 1 for success 0 for error
statusMsg	O	Descriptive status message in case of error
ts	M	A timestamp value as sent in the original request.
txn	M	The transaction id as passed in the request.
Element: DocDetails This is the enveloping element for PDF and XML certificate data.		

Annexure IV: Detailed Guidelines for API owners

These guidelines are to be maintained by Platform management and schema body in consultation with NDH API Cell and peridocally reviewed by TEG.

Planning

Planning is a key responsibility of the API Owner, and such planning mainly involves decisions regarding what APIs are to be exposed and how they are expected to benefit the API consumers. The API Product Owner is encouraged to interact with the platform management and Schema Body and virtually access global technical expert volunteers registered with and empanelled by the platform management and Schema Body who can assist in this high level design process.

Government departments need to take an information-centric approach where all content needs to be treated as data, turn required unstructured content into structured data, then ensure all structured data are associated with valid metadata - Schema. The data would then be accessed via APIs.

The API Product Owner works with the Platform Schema Body and its volunteers and if in place, the API Owner to understand guidelines, standards, and best practices for improved interoperability among various API providing and Consuming Organizations. Adhering to various e-Governance standards helps achieve this goal. Planning includes the design of the APIs. Though platform management and Schema Body and volunteers can advise and review, very rarely can they engage in full-fledged Schema and API design.

NDH APIs should be architected for openness and expose high-value data and content as APIs at a discrete and digestible level of granularity[2]. Under a presumption of openness, departments must evaluate the information contained within their systems for classification and release to other departments/ministries and the public, publish it in a timely manner and make it easily accessible for external use.

Selecting the APIs to open

The services and data which may be frequently demanded and used by citizens, small organisations, other departments or App Developers should be identified at the beginning phase

[2] A bad example would be a single API call providing all possible data, with hundreds of parameters.

of building APIs. API Planning includes the decision on policy and methodology to be adopted for sharing data with external systems. APIs may be picked for opening for three main reasons:

For Common Services & Workflows

The API Publishing Organization should identify and make available APIs for common services, which can used by various applications, address verification using the EPIC database or PAN verification. Such APIs of common services and workflows help in standardizing the process, reducing the development time and maintenance effort for other applications. It is very important to create compensation models and structures to reward API Owners who design or implement APIs to create high usage APIs that are used in myriad ways.

For Data Sharing

National Data Sharing and Accessibility Policy (NDSAP-2012[3]) was designed in year 2012, with the aim of promoting sharing of non-sensitive data in digital form. The Open API Policy clearly refers the API Owner to the National Data Sharing and Accessibility Policy (NDSAP), for specifics of what data can be exposed and what cannot be. Of course, subsequent significant laws or Supreme Court rulings such as right to privacy must be applied. The NDH API Cell will publish and continuously update the list of applicable policies and laws along with their best reading of implications.

For Data Generation

Sharing data along with APIs which allow the public or relevant actors to improve, contribute related data is vital to creating and maintaining high quality, national data repositories. Such APIs which embody workflows to moderate, curate and ensure data quality are to be encouraged for data of broad value.

Based on Rules of IT act Government information can be classified as below,

Information Classification as Specified in Information Technology (Certifying Authorities) Rules of IT ACT

Top Secret: It shall be applied to information unauthorized disclosure of which could be expected to cause exceptionally grave damage to national security or national interest. This category is reserved for Nation's closest secrets and to be used with great reserve.

Secret: This shall be applied to information unauthorized disclosure of which could be expected to cause serious damage to the national security or national interest or cause serious

[3] Refer https://nsdiindia.gov.in/nsdi/nsdiportal/meetings/NDSAP-30Jan2012.pdf

embarrassment in its functioning. This classification should be used for highly important information and is the highest classification normally used

<u>Confidential</u>: This shall be applied to information which is essentially meant for official use only and which would not be published or communicate to anyone except for official purpose

<u>Unclassified</u>: This is the classification of information that requires protection against disclosure

The API can be categorised on the basis of envisaged openness, i.e. on the basis of need to share and need to know.

Negative List

APIs which are meant to share data which is top secret, secret or confidential can be a part of the negative list. These APIs need not be made a part of the API repository.

Sensitive API List

APIs which are meant to share restricted data that falls under sensitive NDSAP classification. These APIs can only be accessed by internal government ministries for internal development and ease of business. Still follow best practices like load balancing etc, but no public routes to these APIs. e.g: Aadhaar auth APIs for use by DBT agencies.

Protected API List

APIs which are meant to share unclassified data that falls under restricted NDSAP classification. APIs are used for development by organizations which belong to a particular sector or to users authorised by API owner. e.g. APIs for data sharing among banks. This will increase operational efficiency and ensure data is safe at the same time.

Public APIs

APIs which are meant to share unclassified data that falls under sharable NDSAP classification. These APIs should have public routes and can be accessed by registered users from anywhere. These APIs may be monetized in agreement with the API owners.

Visibility

API Directory contains API Definition, API Description, API Arguments, Responses, and Error Codes, Sample Source Code, Mashups and Library & SDK. Visibility for Government Ministries and Departments before API subsription:-

Classification	**Sensitive**	**Protected**	**Public**
API Definition	Yes	Yes	Yes
API Description	Yes	Yes	Yes
API Arguments, Responses and Error Codes	No	No	Yes
Sample Source Code	No	No	Yes
Mashups	No	Yes	Yes
Library & SDK	No	No	Yes

For Organizations and Industry before API subscription :-

Classification	**Protected**	**Public and Indian Software Product Companies**
API Definition	Yes	Yes
API Description	Yes	Yes
API Arguments, Responses and Error Codes	No	Yes
Sample Source Code	No	Yes
Mashups	Yes	Yes
Library & SDK	No	Yes

Identification of Stakeholders

Knowing the API Consuming Organization and their purpose of using the API, would help the API Product Owner make key decisions on the design of the API, its publishing timing and the operational sizing of its implementation.

Relating Internal KPIs to SLAs

The API Owner is responsible for providing first analysis and later statistics to the API Owner when SLAs for services powered by internal teams are falling short and along with that data, identify the primary bottlenecks if possible. The best practices of API Management translate API performance parameters into KPIs of roles within the service organisation. Open publication of API graphs in realtime for uptime, failure rates, latency, etc and self-certification that they comply to the SLA for these and other parameters - these graphs and data will be available to the public and regularly reviewed by the Quality and dispute resolution group.

Further API-first Approach

NDH APIs should use an API-first approach where the API is defined first even before its implementation is done for green field projects. Once the API passes rigorous review as defined, the server side and applications (as shown in Figure 5.) can be developed in near-parallel and independent manner saving significant time and costs.

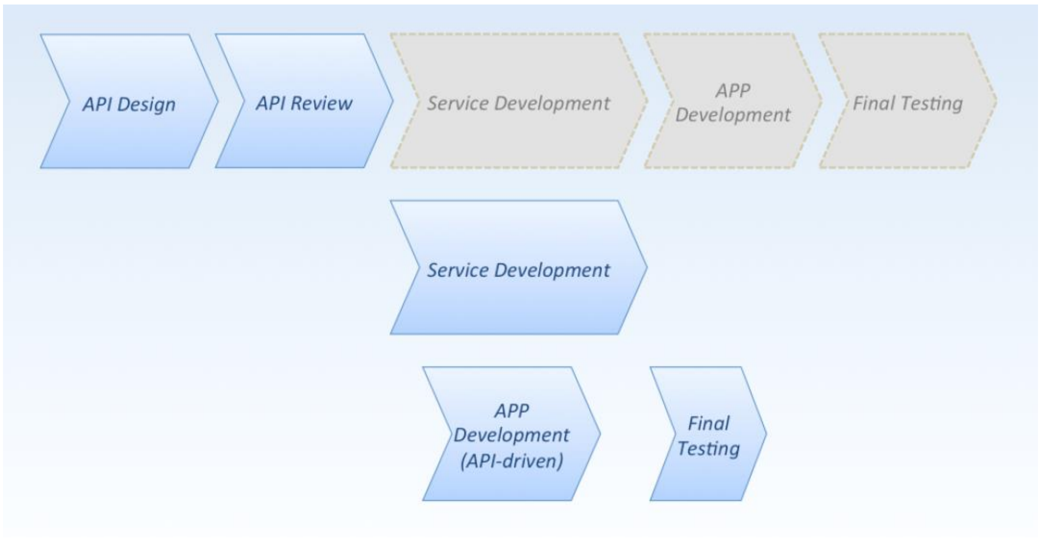

Figure 4. Near-parallel development using API-First approach

The miniaturisation as well as the above separation simplifies, helps user experience be consistent[4], while digitally documenting the service in a readable structure[5] at a level of detail never before done while making all this accessible to every stakeholder in real-time.

API Lifecycle and Management

The overall development process of a typical API-integrated system is depicted in the figure below. Each of the steps in this process is further elaborated - but specified in such a way that the API Owner can drive the bulk of the process by themselves with little oversight from MeitY except when help is asked for.

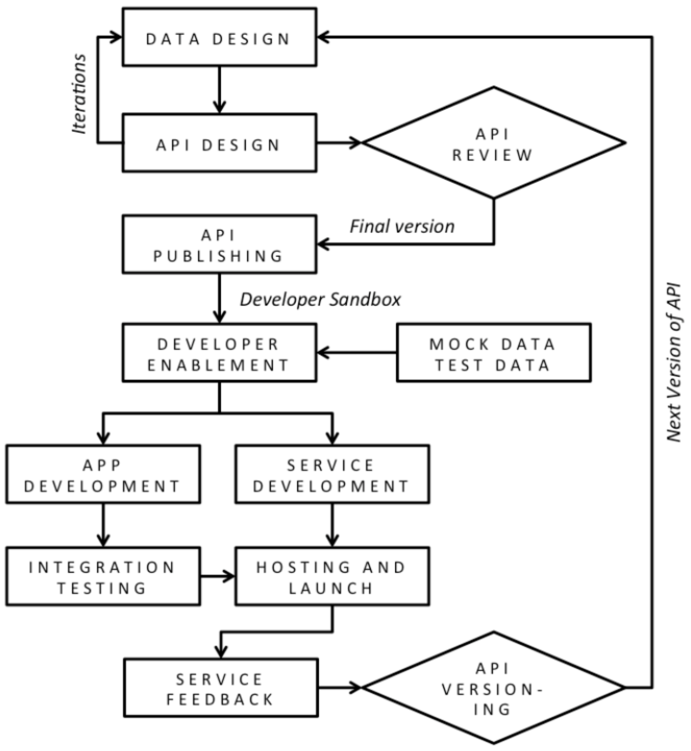

Figure 5: API-driven Development Process

[4]"Design interactions on top of your API, not the other way around." — api-first.com

[5]Description of an API in Open API Specification compliant format also contain definition of data structures

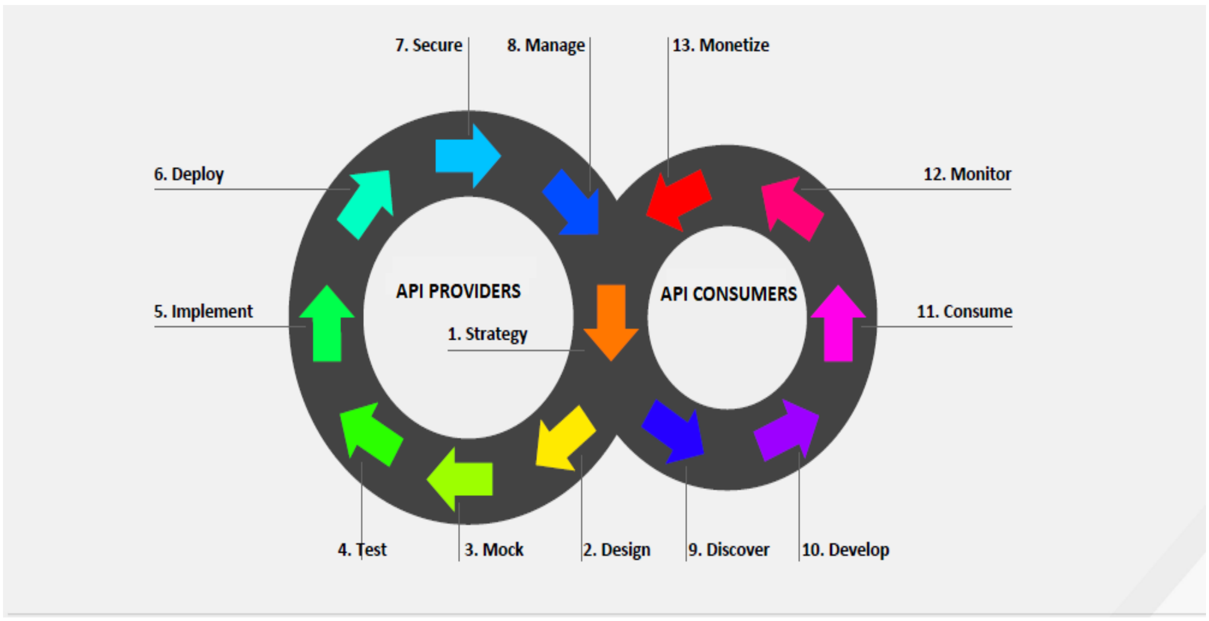

Figure 6. API-Life cycle

Schema Design

The Schema (aka Data) design is the first step followed by the service (and along with it, API) design. The Schema definitions both in draft and once final must be uploaded to the API Directory for public comment as well as review and signoff by Schema Body. It is recommended that this schema definition be made available with sample, test data in Open API 3.0+ compliant format via the API Directory. Once Schema of core entities is finalised, API definitions may be created as long as the API is purely a service level definition and is in no way defining a new common entity or common service - in which case such definitions go through a Schema Body review. The simplest way to design Schema is to reuse (and if really needed, extend) existing Schema. This is also required when the entity in question matches the meaning of an existing Schema's underlying real world entity.

All schema shall be aligned both with "Meta Data and Data Standards" (MDDS)[6], as well as the WWW backed schema.org (or the relevant global standard) and then published in Open API 3.0+ compliant format. If the entity does not exist in MDDS but does in Schema.org (or similar), it may be adopted after refining it to be machine readable where the refined proposal will be published back to schema.org. Similarly if the entity is represented in MDDS but not machine readable or canonical as determined by the Schema Body, it will be upgraded and aligned with Schema.org. The Schema Body will make available tools and a helpdesk to

[6] More information at https://egovstandards.gov.in/faq/domain-specific-mdds

simplify finding, extending, creating of schema. Aligning the designed schema with schema.org (or the relevant global standard) is the responsibility of the Schema Body. Publishing or modifying the relevant Schema in MDDS to be machine readable or to improve it is the responsibility of the API Owner with details and justification provided by the Schema Body. When reusing a schema, the Schema definition must not be copied over. It will be referenced using the schema address in the OAR along with timestamp of registration.

If a new Schema is proposed but the community or Schema Body points out an existing schema that is semantically similar and serves the purpose as per the Schema Body, reuse is always a must. And permanence and auditability of these discussions and those re transparency disputes as well as the API registry data and change history each captured as a ticket in a ticketing system maintained using the same persistent, deep history, inerasable contract mechanism that underlies the OAR.

API Design and Review

When Schema design is complete, the API design starts, with designing of REST resources and methods, documented in the form of individual method calls. There are many sources of good practices with API-design[7] on the web that could be useful for new developers.

OAR registry entries will have metadata noting status of the API - draft for comment, final draft. A separate API exists for comments, suggestions, complaints and response threads along with final resolution of each comment (which can point to a master comment). Metadata of an API also includes published + data, revision + date, live + revision No. that went live + date (for each release).

At all stages, the API drafts are available to both experts and prospective users (API Consumers) via the API Registry to obtain their feedback on usefulness, usability and completeness of the API. Such feedback might lead to interactions and further tweaking of the API before it is finalized after approval by Schema Body for publishing.

Critical Success Factors

An API that is not well used is useless. Hence, the API Owners have to undertake every step to make their API used. Here are a few important ones:

1. **Making it easy to discover and consume**: This is achieved by registering the API and documenting good quality examples, sandbox and FAQs. It is also important to spread the

[7] A very useful collection of articles are listed at http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api#docs

news about the upcoming API to the ecosystem of start-ups and entrepreneurs - and allow every Indian company to be a consumer if legally possible. The less the friction the better.

2. **Listen and Iterate**: It is very important that the API Owner to listens to API Consumers, the Governance Bodies and improve the API to meet needs of a wide range of uses.
3. **Launching the API**: A good launch involves potential API Consumers participating in trial projects, beta programs and at launch hackathons conducted by event manager's expert in handling hackathons, preferably with recognition and prizes.
4. **Engaging the Ecosystem**: Effective engagement of the ecosystem involves numerous small things that build confidence and trust in the API. Some examples are: creating regular blogposts and tweeting these to the ecosystem informing upcoming improvements to API and Service, showcasing new apps and use cases served by the API, etc.
5. **Stability of the APIs**: APIs unless its required by a court order should remain stable and backward **compatible**. **Else, Backward incompatible** improvements or **suspension of any API** should only be with sufficient notice - 3 months for nascent, low use APIs, 12-18 months for highly used or livelihood critical or nationally important APIs. **Typically backward incompatible changes may be implemented by having the older API being marked deprecated but still available and the newer one run in parallel for 12-18 months, prohibit new usages of the deprecated API, send suitable periodic alerts to existing owners of deprecated APIs and then sunset a deprecated API as scheduled.**

"Getting the word out is key for any product, and APIs are no exception. Connecting with developers by sponsoring a Hackathon has yielded tremendous success for companies like American Express and FourSquare." – Programmable Web[8]

[8]"MARKETING YOUR API AS A PRODUCT" Read more on The Programmable Web at: http://www.programmableweb.com/news/marketing-your-api-product/2012/04/24

Annexure V: API Management

Consumers/Owners are to first register on NDH. On registration they may be consumer or owner or both partner type. They should set up their profile and get Access credentials.

Security body shall ensure compliance with security policy for these partners. Once approved they are allowed to access the dashboard they can discover API's they are entitled to view as decided by Cell during registration. API Owners can also Publish their API (ensuring compliance with GoI standards, quality standards, legal issues, time sensitivity, keywords, usage guidelines e.g. : PIIs). A provision for deprecation and version control to be provided to API Owners.

Once API is discovered and tested with test data and test API key, a formal request for usage can be provided through portal for establishing license and tracking.

Any API in the NDH and all certified Open APIs across states and ministries are uniformly available to all Indian Software Product Companies and Indian Institutions subject to a graded eligibility criteria which will be evolved jointly by the Compliance Body and TEG, aligned with Open API Security levels. For Open APIs at each level of Security, a specific set of eligibility criteria to access the API will be defined focused on organisational reputation or team credentials and prior services or Apps of similar or better traffic magnitude they have successfully launched and maintained. Financial size will not be a criteria. The organisations stability may be required for accessing APIs at the highest security level. When a Indian Software Product Company or Indian Institution meets the criteria for a given security level, all registered Open APIs of that level and below shall be accessible for building composed Apps and services on top.

Allow users and consumers to demand new APIs from API Owners. 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.

Annexure VI: Terms of Service

Terms of Service

Thank you for using NDH's APIs, other developer services, and associated software (collectively, "APIs"). By accessing or using our APIs, you agree to the terms below. You agree to comply with the Terms and that the Terms governs your relationship with us. So please read all the Terms carefully.

Under the Terms, "NDH" a Govt. of India initiative to ensure universal access and interoperability among various e-Governance systems to upgrade the quality and effectiveness of service delivery, developed and hosted by Ministry of Electronics & IT, Government of India. We may refer to "NDH" as "we", "our", or "us" in the Terms.

Applicable Law

These terms and conditions shall be governed by and construed in accordance with the Indian Law. Any dispute arising under these terms and conditions shall be subjected to the exclusive jurisdiction of the courts of India.

Registration

In order to access certain APIs you may be required to provide certain information (such as identification or contact details) as part of the registration process for the APIs, or as part of your continued use of the APIs. Any registration information you give to NDH will always be accurate and up to date and you'll inform us promptly of any updates.

Your End Users

You will require your end users to comply with applicable law, regulation, and the provision to access NDH services under these Terms of Service.

Compliance with Law, Third Party Rights, and Other NDH Terms of Service

You will comply with all applicable law, regulation, and third party rights (including without limitation laws regarding the import or export of data or software, privacy, and local laws). You will not use the APIs to encourage or promote illegal activity or violation of third party rights. You will not violate any other terms of service with NDH platform and its issuers and requesters.

Permitted Access

You will only access (or attempt to access) an API by the means described in the documentation of that API. NDH assigns you partner credentials on registration, you must use them with the applicable APIs. You will not misrepresent or mask your API Client's credentials when using the APIs or partner accounts.

API Prohibitions

When using the APIs, you may not (or allow those acting on your behalf to):

1. Charge any fee for NDH services.

2. Access the APIs unless the access is initiated by the end user or you obtained the explicit user consent.

3. Sublicense an API for use by a third party. Consequently, you will not create an API Client that functions substantially the same as the APIs and offer it for use by third parties.

4. Perform an action with the intent of introducing to NDH products and services any viruses, worms, defects, Trojan horses, malware, or any items of a destructive nature.

5. Defame, abuse, harass, stalk, or threaten others.

6. Interfere with or disrupt the APIs or the servers or networks providing the APIs.

7. Approach any NDH user in an unsolicited manner. Reverse engineer or attempt to extract the source code from any API or any related software, except to the extent that this restriction is expressly prohibited by applicable law.

Confidential Matters

Developer credentials (such as passwords, API key) are intended to be used by you and identify your API Client. You will keep your credentials confidential and make reasonable efforts to prevent and discourage other API Clients from using your credentials.

Submission of Content

You and your end users are responsible for contents shared with NDH and consequences thereof. NDH does not actively pre-screen content, but it reserves the right to refuse or remove any content or account that in its sole discretion is found to be unlawful, offensive, threatening, promoting violence, defamatory, pornographic, or violating any NDH Terms of Services and Privacy Policy or violating any other user or party's intellectual property rights.

Retrieval of content

When a user's content is obtained through the APIs, you may not expose that content to other users or to third parties.

Prohibitions on Content

You will not, and will not permit your end users or others acting on your behalf to, do the following with content returned from the APIs:

1. Scrape, build databases, or otherwise create permanent copies of such content, or keep cached copies longer than permitted by the cache header;

2. Copy, translate, modify, create a derivative work of, sell, lease, lend, convey, distribute, publicly display, or sublicense to any third party;

3. Misrepresent the source or ownership; or

4. Remove, obscure, or alter any copyright, trademark, or other proprietary rights notices; or falsify or delete any author attributions, legal notices, or other labels of the origin or source of material.

Branding

You agree to use NDH brand while providing services using these APIs and you will not rebrand these APIs under any other name. You agree to display any attribution(s) such as NDH logo or name to your users at the point of service. NDH hereby grants to you a nontransferable, non-sublicense able, non-exclusive license while the Terms are in effect to display NDH's logo and name for the purpose of promoting or advertising that you use the APIs.

Privacy and Copyright Protection

By using our APIs, NDH may use submitted information in accordance with our privacy policy (https://digilocker.gov.in/privacypolicy.php).

Termination

NDH reserves the right at any time to modify or discontinue, temporarily or permanently, the Service (or any part of it), with or without notice. NDH in its sole discretion reserves the right to suspend or terminate any partner account and refuse any current or future use of the platform for any reason at any time. Such termination may result in the deactivation or deletion of the account, and the loss of all the content hosted therein.

General Provisions Modification

We may modify the Terms or any portion to, for example, reflect changes to the law or changes to our APIs. You should look at the Terms regularly. Addressing new functions for an API or changes made for legal reasons will be effective immediately. If you do not agree to the modified Terms for an API, you should discontinue your use of that API. Your continued use of the API constitutes your acceptance of the modified Terms.