API Management Solution at MyGov

Publish, manage, secure and analyze APIs in minutes

Edited by
MOHAN DAS VISWAM

The API Management Solution deployed at MyGov facilitates the connection of data and business functions. MyGov uses Tyk as its API Gateway and API Dashboard.

ALKA MISHRA
Deputy Director General
amishra@nic.in

DURGA PRASAD MISRA
Sr. Technical Director
dpmisra@nic.in

NARENDER KUMAR JAIN
Scientist-D
nk.jain@nic.in

RAVI KUMAR
Scientist-C
ravi.k@nic.in

A good Application Programming Interface (API) management system provides a simple yet robust platform for creating, annotating and publishing APIs. It also simplifies the process of finding, understanding and using APIs, along with the processes and services behind them. Such an API Management Solution facilitates the connection of data and business functions so that organizations can enable access to applications via mobile, cloud, and on-premise environments. MyGov uses Tyk as its API Gateway and API Dashboard.

Manage All your APIs at One Place

- Expose all APIs behind a single IP and domain
- View near real-time usage, performance and health analytics
- Automate management and integrate using REST API
- Provision API Management and scale it on demand

Components

- MyGov API Gateway
- MyGov API Dashboard

Technology Brief

API Gateway on MyGov is an open source solution that is fast, scalable and modern. Basically, it is an implementation of gateway that sits in front of APIs and acts as a single point of entry for a defined group of micro-services or APIs. In addition to accommodating direct requests, it is also being used to invoke multiple back-end services and aggregate results.

Gateway performs the Authentication, Security, Traffic Control, Caching, Logging, Analytics, Monitoring and much more. It can run completely independently, requiring only MongoDB and Redis to be effective, and it can scale horizontally.

Important Features of Gateway

- Quotas and rate limiting
- Authentication
- On-the-fly transforms
- Caching
- Logging

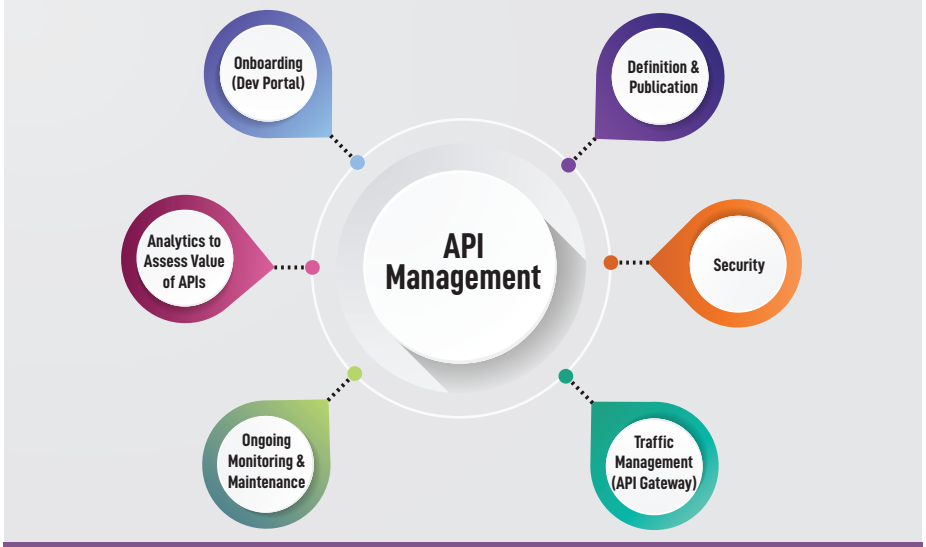

Fig. 1: API Management Life Cycle

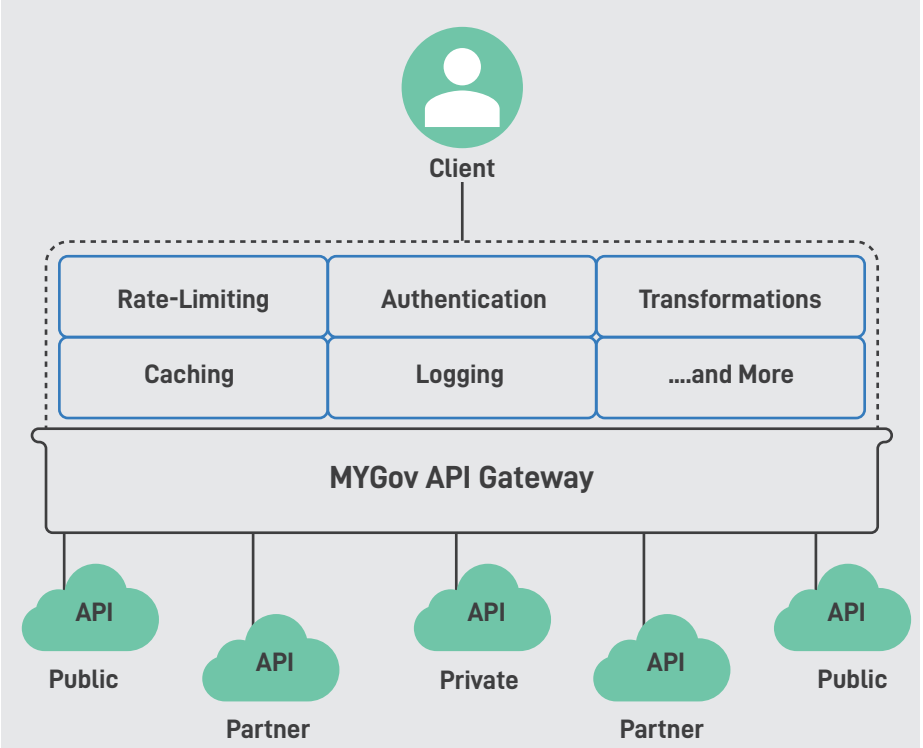

Fig. 2: API Gateway Components of MyGov

- Virtual endpoints
- Notifications and events

The Gateway is Capable of:

- Managing session objects.
- Managing and listing policies.
- Managing and listing API definitions.
- Hot reloading a cluster configuration (configuration changes can be pushed without any downtime).
- OAuth client creation.

API Dashboard on MyGov is used for creating, publishing, maintaining, monitoring, and securing REST APIs at any scale. It provides maximized scalability and reliability by offloading API Governance to the edge. It is the visual Graphical User Interface (GUI) and analytics platform for managing the APIs.

Dashboard allows to maintain, manage, promote and protect the APIs quickly and easily, using the open source gateway combined with a sleek user interface control panel. It provides an easy-to-use management interface for managing the installation as well as clear and granular analytics.

Important features of Dashboard

- Design endpoints with a GUI
- Interrogate API usage with powerful analytics
- Unify API management
- A user- friendly GUI

Benefits

- It encapsulates the internal structure of the application. Rather than having to invoke specific services, clients simply talk to the gateway.
- The API Gateway provides each client with a specific API. This reduces the number of round trips between the client and application.
- It also simplifies the client code.

Drawbacks

- It is yet another highly available component that must be developed, deployed and managed.
- There is also a risk that the API Gateway becomes a development bottleneck. Developers must update the API Gateway in order to expose each micro-service endpoints.
- It is important that the process for updating the API Gateway be as lightweight as possible. Otherwise, developers will be forced to wait in line in order to update the gateway.

How to Configure an API?

Creating an API Endpoint - It consists of Base and Advance Configuration to create any API Endpoint.

Base Configuration - Define the API Name, API Slug, Target URL (having provision of Round Robin Load Balancing), Policies (assign the policy), Service Discovery and Authentication Mode (choose one of the modes i.e., Token/ HMAC/ Oauth2.0/ Keyless) to create a new API Endpoint.

Advance Configuration - Define the version, designer (to manage processes

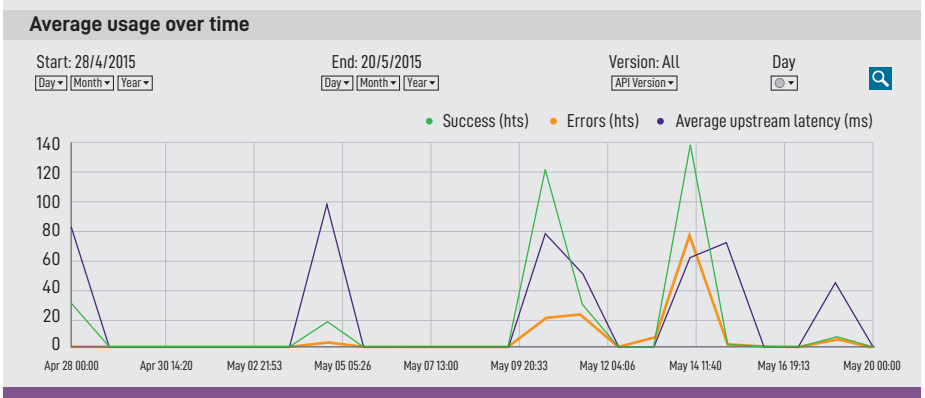

Fig. 3: API Dashboard Activity

specific endpoints by default, it will proxy all requests), cache options, CORS, batch requests, segment tags, white listed IPs and WebHooks.

Uptime Tests - Enabling this option will cause gateway to check the target URL against hosts in the uptime tests for downtime. This should only be used with round-robin load balancing.

Creating an API Key - A key can be created and the access permissions can be set up for a specific key. Multiple settings can be enabled including quota settings, rate limits and access rules for specific API versions. It is running in hashed key mode. This means that keys available on the system may not be listed, since keys have been encrypted. If the API key details are known, it is still possible to edit, delete and modify API keys.

Defining Policies - Policies are a way to enforce a standard set of rate limits, quotas and access rules on a set of keys. When used in conjunction with the portal, developers who enroll for API access will be given a key that is attached to a specific policy. The policy settings are refreshed every time a key attempts access, meaning that updating a policy will have an effect across any keys that are attached to it.

Security policy incorporates several security options that can be applied to an API key. It acts as a template that can override individual sections of an API key (or identity) in Tyk. For example, if 10,000 API keys were issued, how would it be ensured that all 10,000 users received an upgraded quota or accessed a new API that was published?

All 10,000 keys could be modified manually, or a policy could be applied to each of those keys when they are created, and then the policy is modified once.

Policies can set:

- Access lists for API and versions.
- Access lists for method and path (granular control).
- Rate limit for a user.
- Quota for a user.

Creating WebHooks - WebHooks are a great way to let external applications know about the status of a user, an API or an event that has occurred in the Tyk gateway. New WebHooks can be created here and re-used in API Definitions. They can be assigned to different events such as quota violations and rate-limiting violations.

A WebHook that can be reused across API definitions can be set up. WebHooks support custom headers and can be deployed using GET, POST, PUT, DELETE and PATCH HTTP verbs.

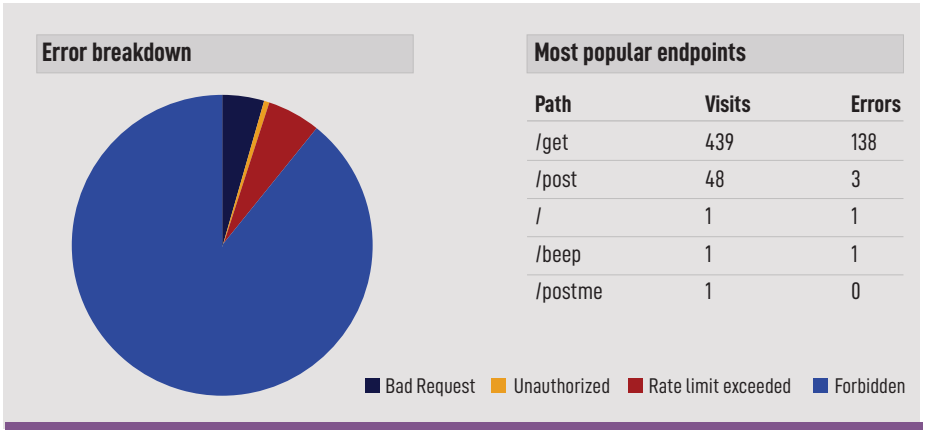

Fig. 4: API Dashboard Activity

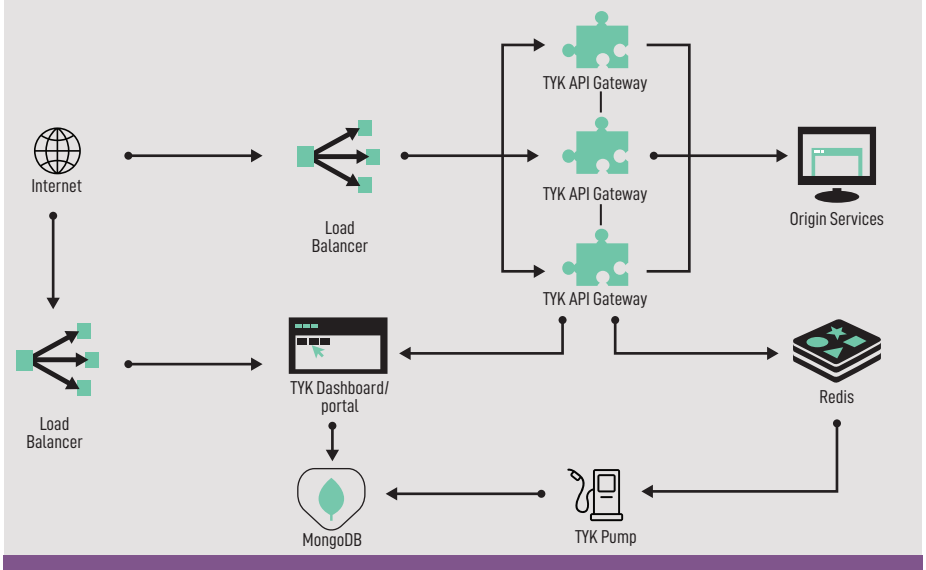

Fig.5: Overall Architecture of API Gateway and Dashboard

Summary

For most micro-services based applications, it makes sense to implement an API Management Solution, which acts as a single entry point into a system. The API Gateway is responsible for request routing, composition and protocol translation. It provides each of the application's clients with a custom API. The API Gateway can also mask failures in the backend services by returning cached or default data. ■

For further information, please contact:

ALKA MISHRA
Deputy Director General
Open Data, Web & Cloud Technology Group
A4B4, 3rd Floor
NIC HQ, A-Block, CGO Complex
NEW DELHI - 110003

Email: amishra@nic.in
Phone: 011-24305395