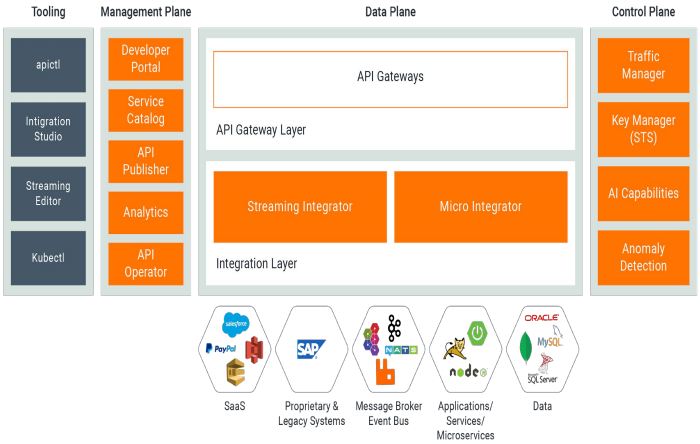
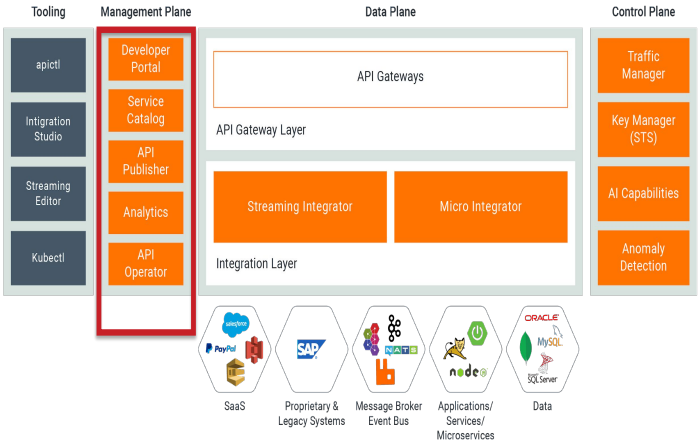

WSO2
WSO2 API Manager 4.1.0
Fundamentals
Architecture
WSO2 Training
CC by 4.0

Components



WSO2 API Manager consists of an API gateway and collaboration space where API publishers meet API consumers. The collaboration space consists of an API Devportal and API Publisher.

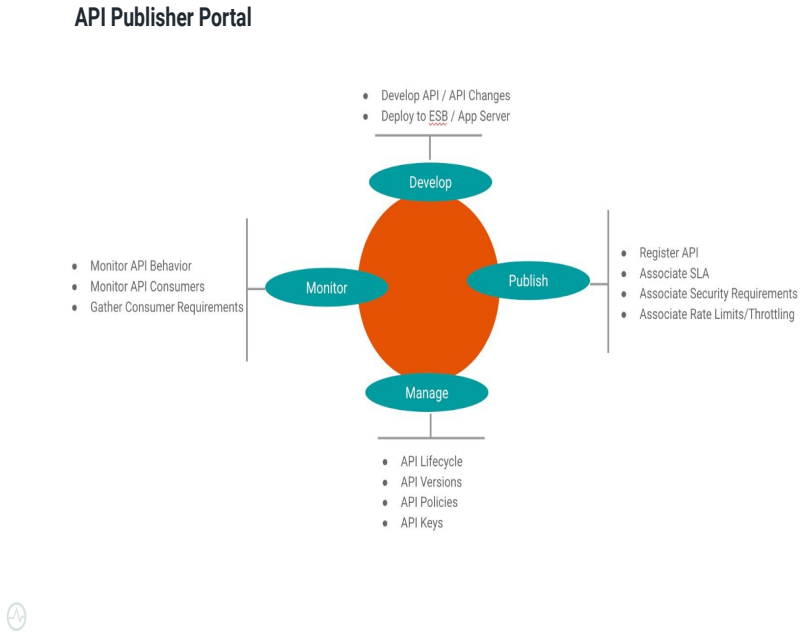
- API Gateways: enables you to secure, protect, manage, and scale API calls.
- API Publisher: enables API providers to easily publish their APIs, share documentation, provision API keys, and gather feedback on an API's features, quality and usage.
- API Developer Portal: provides space for consumers to discover API functionality, subscribe to APIs, evaluate them and interact with API publishers.
- Key Manager: Manages all clients, security and access token-related operations.
- Traffic Manager: The Traffic Manager helps users to regulate API traffic, make APIs and applications available to consumers at different service levels, and secure APIs against security attacks. The Traffic Manager features a dynamic rate limiting engine to process rate limiting policies in real-time, including rate limiting of API requests
- Analytics: Additionally, monitoring and analytics are provided by the analytics component, WSO2 API Manager Analytics. This component provides a host of statistical graphs, an alerting mechanism on predetermined events and a log analyzer. For more information



API Manager front end consists of two main components.

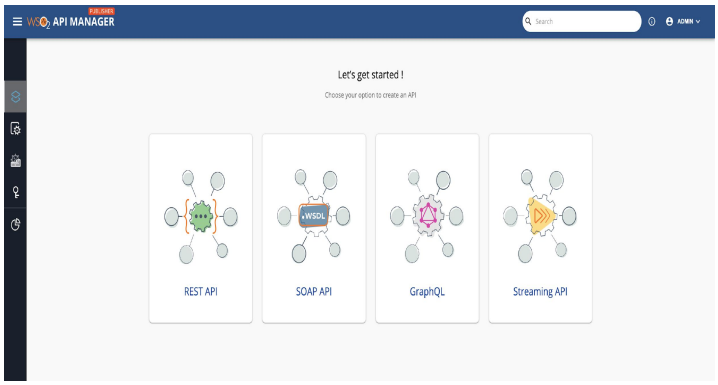
API Publisher : For creating and publishing APIs
Developer Portal : For API Consumers to discover, subscribe and test the Published APIs.

API Publisher Portal



Using the Publisher you can first develop the API and then publish it while managing the lifecycle, versions and so on, and monitor how it is used.

API Publisher

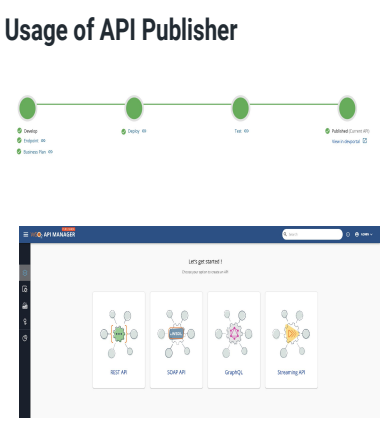


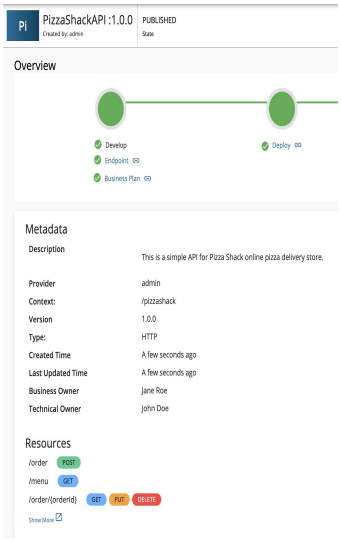
Link - API Publisher

API Manager uses a simplified Web interface called WSO2 API Publisher for API development, publication and management. It is a structured GUI designed for API creators to develop, document, scale and version APIs, while also facilitating more API management-related tasks such as publishing(making the API visible to API marketplace/devportal) and deploying(making the API invokable) API, monetization, analyzing statistics, quality and usage and promoting and encouraging potential consumers and partners to adopt the API in their solutions.

Collaborative Web interface to

- Create APIs/API documentation
- Deploy and advertise/publish APIs
- Manage API lifecycle
- Receive community feedback
- Monetize usage
- Analyze statistics

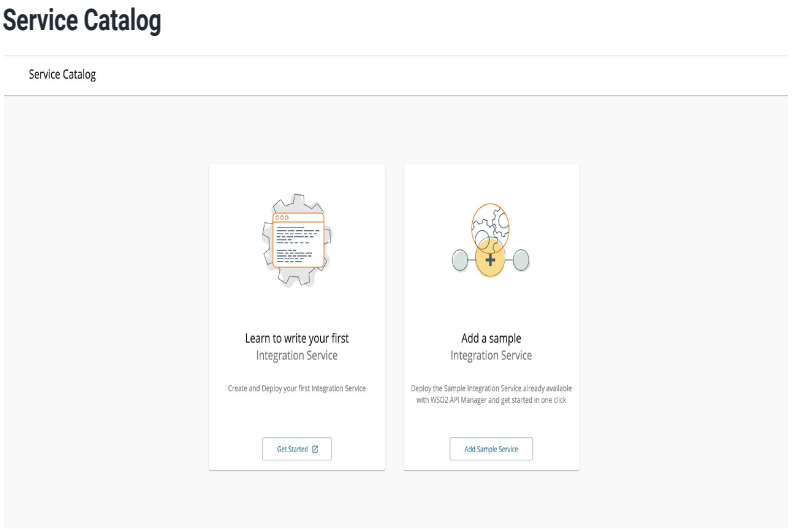




The API Publisher can be used to:

- Publish APIs to external consumers and partners, as well as to internal users
- Maintain API revisions
- Ability to deploy APIs to a selected set of gateways in a multi-gateway environment
- Support enforcement of corporate policies for actions like subscriptions, application creation, etc. via customizable workflows
- Manage API visibility and restrict access to specific partners or customers
- Manage API lifecycle from cradle to grave: create, publish, block, deprecate, and retire
- Publish both production and sandbox keys for APIs to enable easy developer testing
- Manage API versions and deployment status by version
- One-click deployment to API gateway for immediate publishing

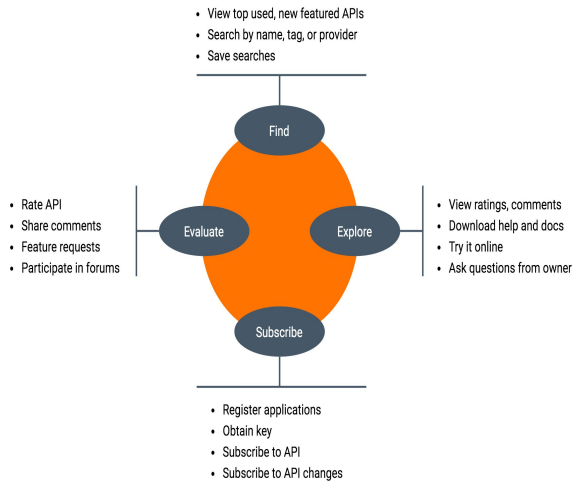
Service Catalog



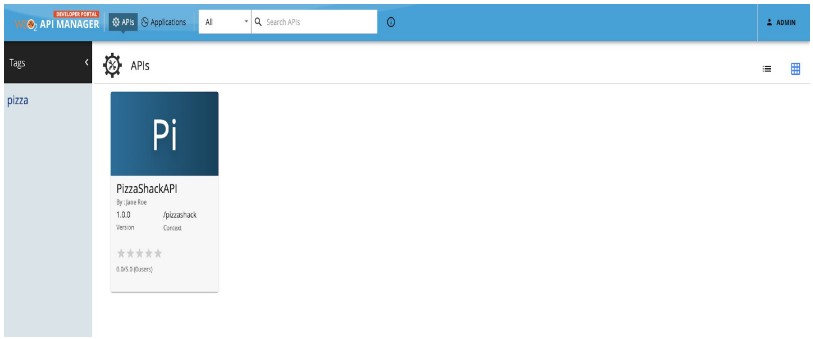
Service Catalog is one of the main attributes that enable the API-first Integration in WSO2 API Manager. Through the Service Catalog, integration services are made discoverable to the API Management layer so that API proxies can directly be created using them.

API Developer Portal

API Developer Portal



API Developer Portal



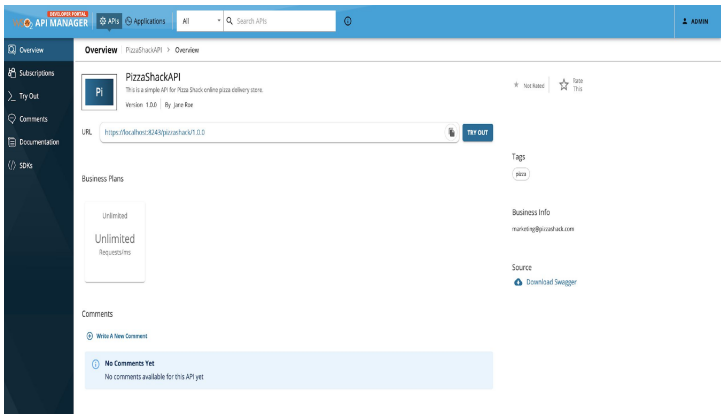
Link - API Devportal

API Manager provides a structured Web interface called the WSO2 Developer Portal to host published APIs. API consumers and partners can self-register to it on-demand to find, explore and subscribe to APIs, evaluate available resources and collaboration channels. The API Developer Portal is where the interaction between potential API consumers and API providers happen. Its simplified UI reduces time and effort taken when evaluating enterprise-grade, secure, protected, authenticated API resources.

A collaborative Web interface to

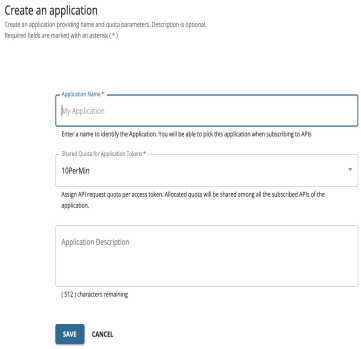
- Self sign-up
- Subscribe to advertised APIs
- Get access token to invoke APIs
- Invoke APIs
- Engage with the API community (commenting/rating)

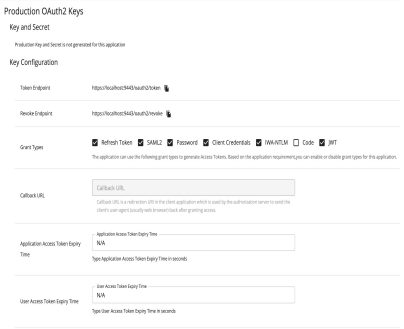
Usage of API Developer Portal



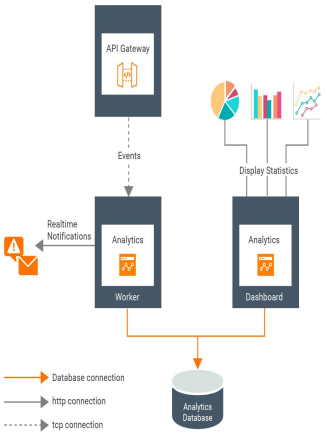
- Graphical experience similar to popular applications stores. Try APIs directly from the portal
- Browse and search APIs by provider, tags, or name
- Self-registration for developer community to subscribe to APIs
- Subscribe to APIs (including key provisioning) and manage subscriptions on per-application basis (at different service tiers)
- Common view of the store for users registered under same organization
- Developer interaction with APIs via forums, comments, and ratings
- View API consumer analytics

Usage of API Developer Portal





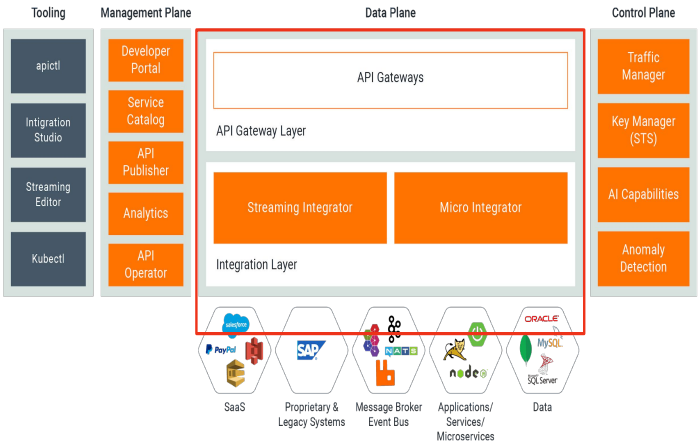
Analytics



Analytics

Analyses API requests for statistical reporting and abnormality detection

Link - Analytics


Tooling
Management Plane
Data Plane
Control Plane
apictl
Intigration Studio
Streaming Editor
Kubectl
Developer Portal
Service Catalog
API Publisher
Analytics
API Operator
API Gateways
API Gateway Layer
Streaming Integrator
Micro Integrator
Integration Layer
Traffic Manager
Key Manager (STS)
AI Capabilities
Anomaly Detection
SaaS
Proprietary & Legacy Systems
Message Broker Event Bus
Applications/ Services/ Microservices
Data

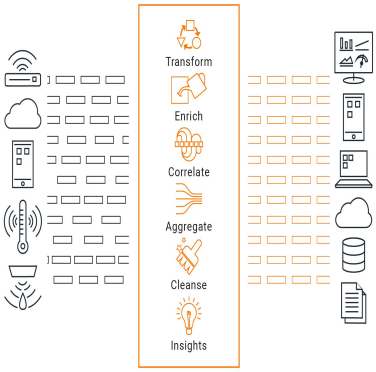
Link - API Gateway

The API Gateway is the runtime, backend component developed using the WSO2 EI, which is proven for its performance capability. API Gateway secures, manages, and scales API calls. It is a simple API proxy that intercepts API requests and applies policies such as rate limiting and security checks and also instrumental in gathering API usage statistics.

Gateway uses a set of handlers for security validation, rate limiting purposes and to publish statistics. Upon validation, it passes Web service calls to the actual back-end. If the service call is a token request call, API Gateway passes it directly to the API Key Manager Server to handle it. By default, there's a single Gateway instance (deployed either externally or embedded within the Publisher), but it is also possible to set up multiple Gateways to handle production and sandbox requests separately.

(https://apim.docs.wso2.com/en/4.0.0/learn/api-gateway/maintaining-separate-production-and-sandbox-gateways/#maintaining-separate-production-and-sandbox-gateways)

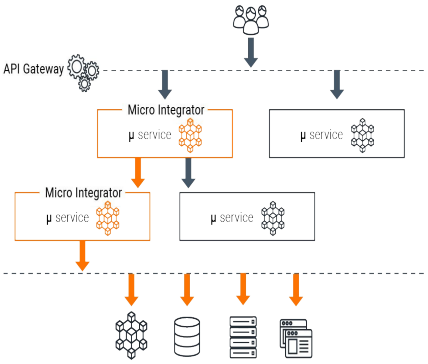
Streaming Integrator



A cloud-native, lightweight component that understands, captures, analyzes, processes, and acts upon streaming data and events in real-time.

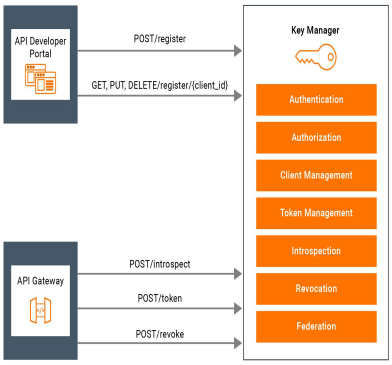
Link - Streaming Integrator

Micro Integrator



Allows you to leverage the comprehensive enterprise messaging capabilities of the Micro Integrator in your decentralized, cloud-native integrations.

Link - Micro Integrator



Key Manager

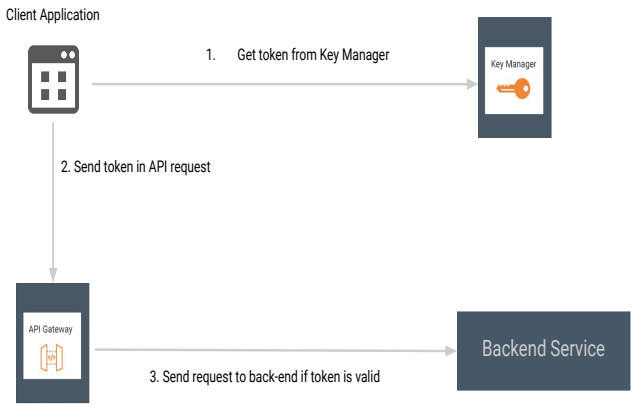
Used by the API Gateway to handle security and access tokens (Internal or External Key Manager)

Link - Key Manager

The Key Manager manages all clients, security and access token-related operations. The Gateway connects with the Key Manager to check the validity of OAuth tokens, subscriptions and API invocations.

When a subscriber creates an application and generates an access token to the application using the API Developer portal, it makes a call to the API Gateway, which in turn connects with the Key Manager to create an OAuth client and obtain an access token. Similarly, to validate a token, the API Gateway calls the Key Manager, which fetches and validates the token details from the database.

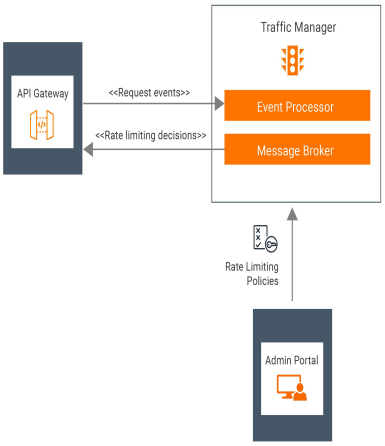
Key Manager Flow



1. Client Application first registers an OAuth application in Key Manager and request an access token.
2. Client invokes the API with the obtained access token in the Authorization header. (Authorization: Bearer <access_token>)

Gateway extracts the token from the header. If the token is opaque type (reference token), sends to the key manager to validate the token. For JWT, the gateway validates the token.

1. Upon the validation response, gateway either sends the request to the backend or responds back to the client with 401 - Unauthorized if the validation failed.



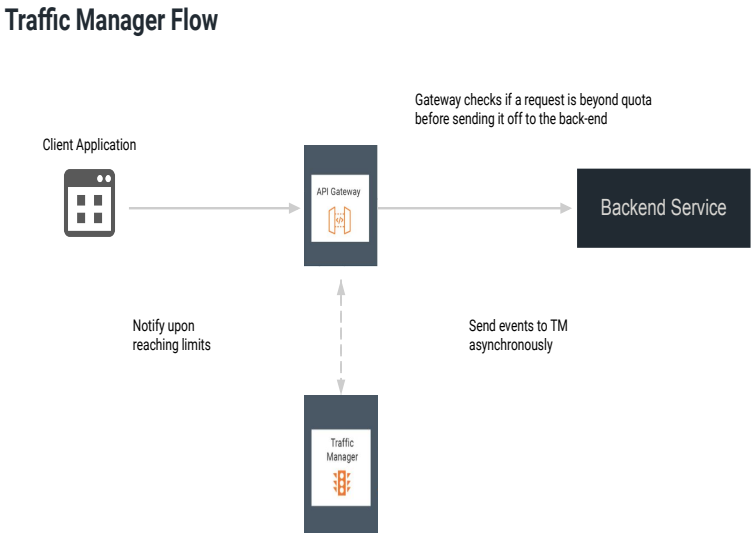
Traffic Manager

Keeps a count of API requests flowing through the API Gateways against various policies and ensures the rate limits (quotas) are met.

Link - Traffic Manager

Traffic Manager manages all the request traffic that is served by the gateway. It does this with the use of rate limiting policies which defines the allowed number of requests for particular resource/ the request rate etc. The Traffic manager is the enforcement point for these rate limiting policies.

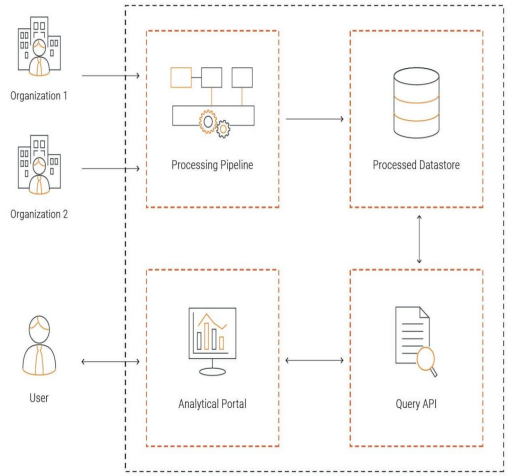
Rate Limiting policies can be created using the API Manager Admin Portal. Once created, Admin Portal publishes the policy in Traffic Manager instance.



When the client invokes the API, the gateway publishes the request details to the Traffic Manager as asynchronous events through JMS. The Traffic Manager evaluates the applied rate limiting policy for a particular API. Also it maintains a counter to count the number of requests received for that API. After evaluating the policy, the policy decision is sent to the Gateway. Policy decision says whether the defined limit for the API is reached.

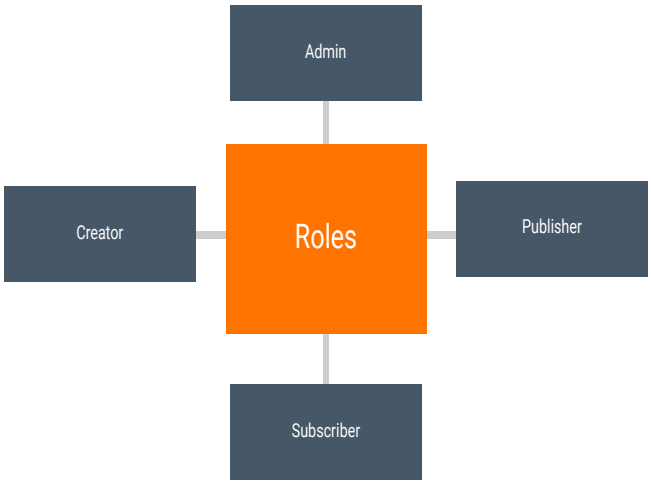
The Gateway limits the following requests to the API based on the policy decision.

Analytics Flow



API Manager offers analytics as a cloud service.
As depicted above, the Gateways will publish analytics statistics directly to the Analytics Cloud over the internet. The Analytics Cloud will have regional deployments to reduce publishing latencies and honor data privacy.

Common User Roles



These user roles are common and standard in a typical enterprise setting. But they are not mandatory. Users can change them according to their unique requirements.

- Admin: hosts and manages the server/s
- Creator: technical role that develops APIs
- Publisher: management role that publishes APIs, controls API lifecycle, monetization etc.
- Subscriber: App developers who consume APIs

Each role has different levels of permission

Let's try it out!

Getting Started with WSO2 API Manager

Defining Users and Roles