



# WSO2 API Manager 4.2.0 Fundamentals

Terminology and Background

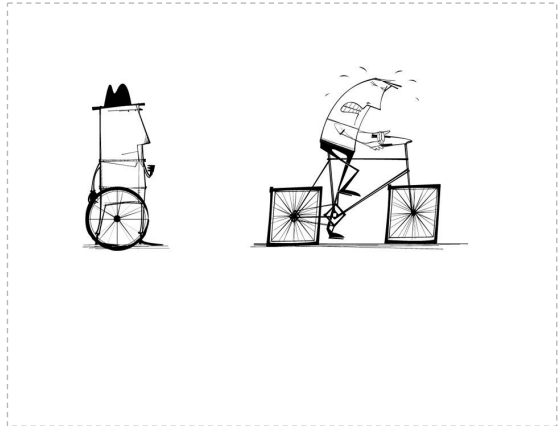


—WSO2 Training

CC by 4.0

## Why Do I Need APIs?

***“Don’t give me a file; give me an API”***



If we consider the Meteorological department as an example, we can get weather data for a particular time or place.

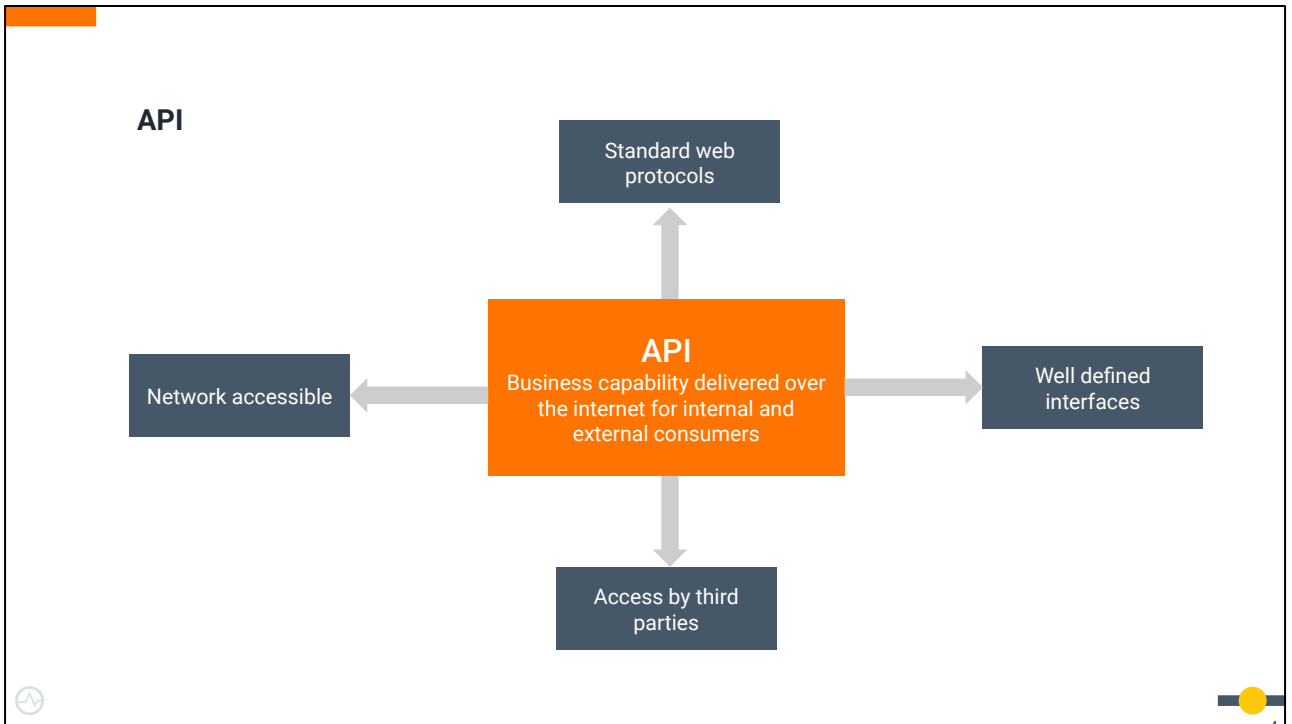
If the data is in a file, when we request the data, we get a large file with all the data in it.

But what if we need some specific set of data? Either we have to extract those from the file, or the department should provide a new file with the data.

If the Meteorological department has APIs for this, we can easily send the request and get the required set of data.

# What is an API?



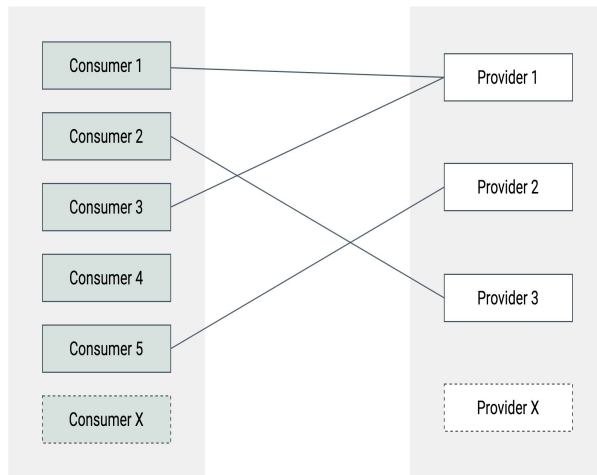


An API is a business capability delivered over the internet for internal and external users.

APIs:

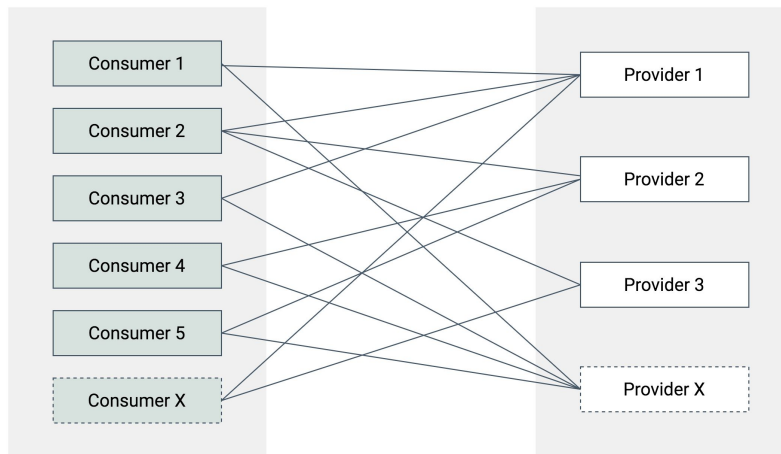
- Accessible over the network and uses standard web protocols. (HTTP, GRPC, WebSocket, etc.)
- Have well defined interfaces.
- If allowed, can be accessible by third party entities.

## Many Consumers and Providers



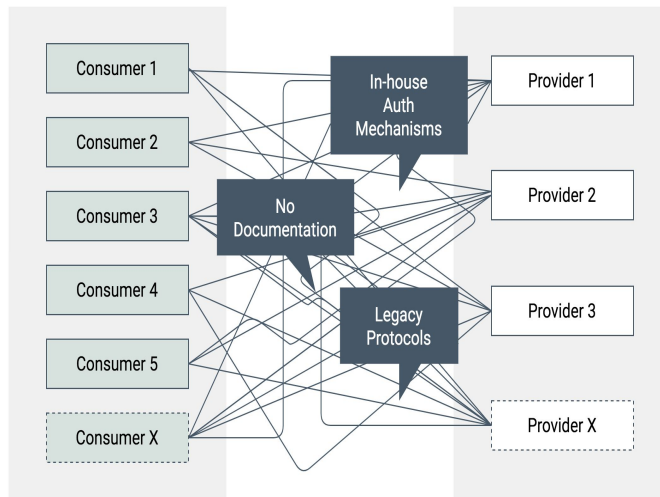
In a business/organization, at the initial phase of adopting APIs, there may be a small number of APIs (providers) that are used by the consumers.

## Many Consumers and Providers



When the business grows, the number of APIs (providers) increase and the number of consumers who use the APIs also increase.

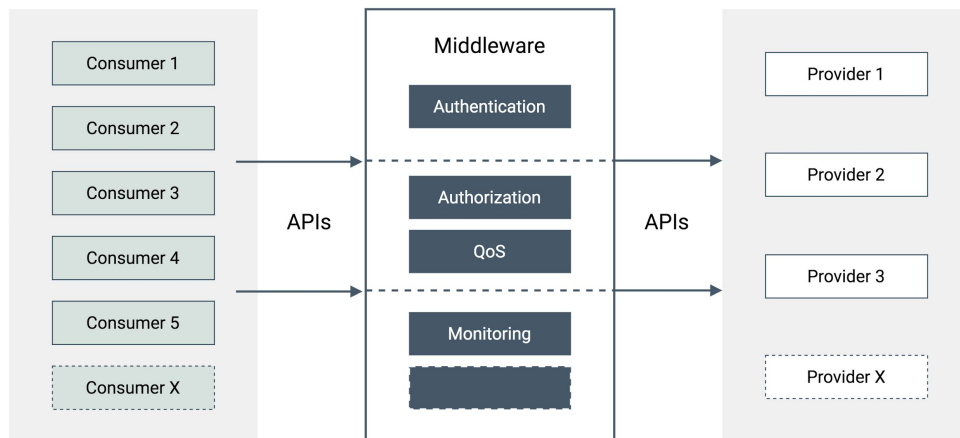
## Many Consumers and Providers



As a result, it becomes very hard to manage all these APIs due to,

1. Using different protocols
2. Using different authentication mechanisms
3. Lack of documentation
4. Legacy code

## Solution: Managed Provider/Consumer Interactions through APIs?



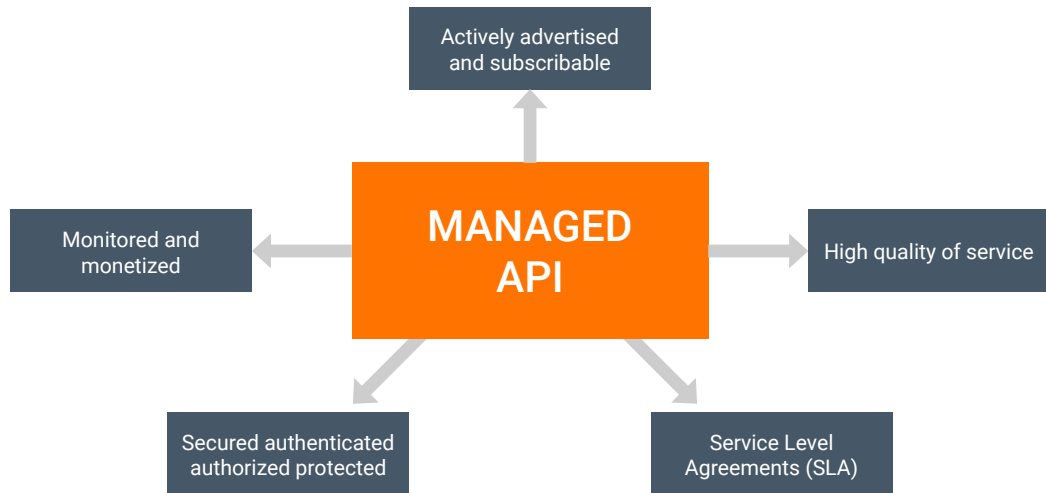
To solve this problem, a Middleware layer is introduced. Each (API) Provider is connected to the consumers through a managed interface where there are no individual point-to-point interactions.

This middle-layer manages all the traffic that flows through and it provides the following services:

1. Enforces security (Authentication and Authorization)
2. Provides QoS such as rate limiting, protecting the backend from sudden spikes of requests
3. Monitoring and analytics
4. Request/response transformation



## Managed APIs



## Advantages of Exposing APIs Publicly

Expose  
API  
to  
Public

Innovative ways to **build** new solutions

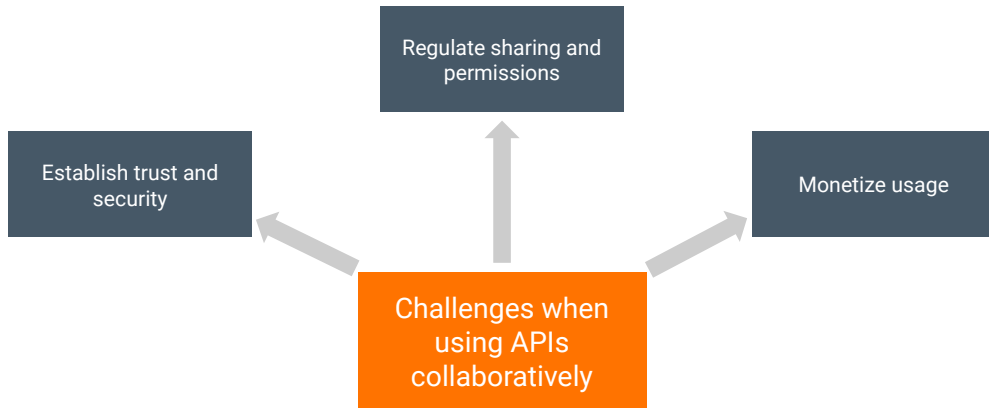
Increase **growth potential** and  
**partnership advancements**

Reduce **duplication** and **rework**

Increase **collaborative** development



## Why is API Management Important?



API development is usually done by someone who understands the technical aspects of the API, interfaces, documentation, versions etc., while API management is typically carried out by someone who understands the business aspects of the APIs.

API management brings on-demand, self-service development onboarding into the SOA and integration space. Developers can rapidly find relevant APIs, discover API functionality, test APIs online, subscribe to APIs, evaluate them, generate access keys, and interact with API publishers. API publishers can easily provision their APIs, share documentation, manage API keys, and gather feedback on an API's features, quality and usage.

Exposing core processes, data and services as APIs to the public allows you to,

- Let others mash up your APIs in innovative ways to build new solutions
- Increase your growth potential and partnership advancements
- Reduce duplication and rework
- Increase collaborative development

# Introduction to WSO2 API Manager



## What is WS02 API Manager?

### Problem



An increase in the number of consumers makes managing services/microservices hard such as security, access control etc.

### Solution



100% Open Source  
Full API Lifecycle Management  
Policy Enforcement  
Monetization

### What is the WSO2 API Manager?

#### Design

API Publisher



Design & Publish API

API Developer Portal



Consume APIs  
API discovery, read API docs, quick try out etc.

#### Runtime

Gateway/Microgateway



Intercepts all requests applies policies, mediates messages & feeds data to analytics engine

Key Manager



Issues/validates secure tokens for consuming devices/applications

Traffic Manager



Applies rate limiting policies on the gateway

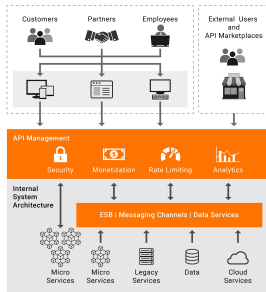
Analytics



Collects & processes data to give insights on the business

WSO2 API Manager is a fully open source solution for end to end API Management in the cloud, on-prem or in hybrid environments. It comes with an [Apache Software License Version 2.0](#) which makes it **free to use**. It allows API developers to design publish and manage the lifecycle of APIs and API product managers to create API products from one or more APIs. It hosts an application developer portal which helps in building and managing a developer community for your APIs. Its cloud native API gateway is used for securing, routing, controlling and monitoring your API traffic in a scalable manner.

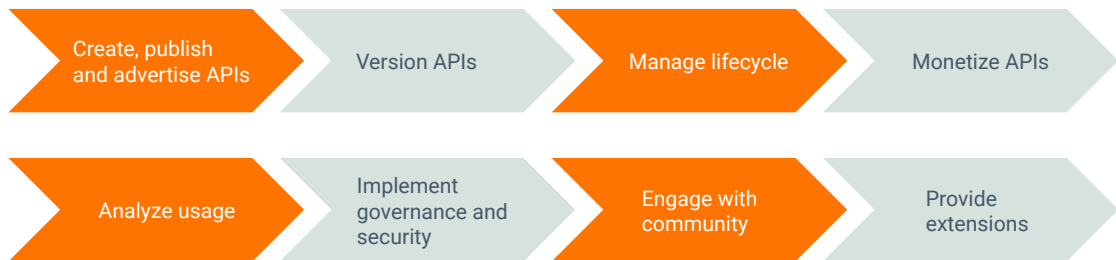
## API Management



- APIs are exposed via an API Management platform.
- Organizations can enforce security policies and rate limiting policies based on the user.
- APIs can be monetized.
- APIs can be exposed to external users and marketplaces, which expands business capabilities.



## Using WS02 API Manager...



WSO2 API Manager is a full lifecycle API Management platform which can be used to

- Create and publish APIs
- Advertise APIs in the storefront
- Version APIs
- Manage their lifecycles
- Monetize API usage
- Monitor and Analyze API usage
- Implement governance and security
- Facilitate community engagement and provide extension points

## API Manager Features

Design and Prototype APIs

Publish and Govern API Use

Control Access and Enforce Security

Create a Store of All Available APIs

Manage Developer Community

### Design and Prototype APIs

- Design APIs, gather developers' feedback before implementing (API First Design). Design can be done from the publishing interface or via importing an existing Swagger 2.0 definition
- Deploy a prototyped API, provides early access to APIs, and get early feedback
- Mock API implementation using JavaScript
- Supports publishing SOAP, REST, JSON, and XML style services as APIs
- Supports grouping of multiple APIs based on the version
- A sample API to try-out for a hassle-free first experience

### Publish and Govern API Use

- Publish APIs to external consumers and partners, as well as to internal users
- Ability to publish 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



## API Manager Features

Design and Prototype APIs

Publish and Govern API Use

Control Access and Enforce Security

Create a Store of All Available APIs

Manage Developer Community

### Control Access and Enforce Security

- Apply security policies to APIs (authentication, authorization)
- Rely on OAuth2 standard for API access (Implicit, Authorization code, Client, SAML, IWA Grant Type)
- Restrict API access tokens to domains/IPs
- Supports plugging in third-party key servers for application registration, token generation & token validation apart from the WS02 Key Manager
- Block a subscription and restrict a complete application
- Associate API available to system-defined service tiers
- Encodes JWT with Base64 and Base64URL
- Leverage XACML for entitlements management and fine-grain authorization
- Configure Single Sign-On (SSO) using SAML 2.0 for easy integration with existing web apps

### Create a Store of all Available APIs

- Graphical experience similar to popular applications stores
- Browse and search APIs by provider, tags, or name
- Provision API keys
- Subscribe to APIs and manage subscriptions on per-application basis
- Subscriptions can be at different service tiers based on expected usage levels
- Try APIs directly from the storefront
- Internationalization support
- Common view of the store for users registered under same organization

### Manage Developer Community

- Self-registration for developer community to subscribe to APIs
- Developer interaction with APIs via forums, comments, and ratings
- View API consumer analytics

## API Manager Features

Manage API Traffic

Monitor and Monetize

Pluggable, Extensible and Themable

Easily Deployable in Your Enterprise

WSO2 Platform Multitenancy Support

### Manage API Traffic

- API gateway can act as an SSL termination point
- Supports protocol transformation, data transformation, and API composition
- Maps between HTTP(s) and other protocols, such as JMS or writing to file systems
- Extremely high performance pass-through message routing with minimal latency
- Enforces rate limiting and rate limiting policies for APIs by consumer
- Rate limiting hard limit for API
- Horizontally scalable with easy deployment into cluster using proven routing infrastructure

### Monitor and Monetize

- All API usage is published to a pluggable analytics framework
- Out-of-the-box support for WSO2 Data Analytics Server and Google Analytics. Supports configuring WSO2 Data Analytics Server through a graphical interface.
- Track consumer analytics per API, per API version, per tiers, and per consumer
- Configurable payment schemes to monetize API usage
- Monitor SLA compliance
- Alerting, real-time dashboards
- Publish your own events and create your own dashboards
- OOB support for events based on rate limiting, faults, latency within and from WSO2 API Manager to target and approval/rejection of self-registration, subscription and app creation

## API Manager Features

Manage API Traffic

Monitor and Monetize

Pluggable, Extensible and Themable

Easily Deployable in Your Enterprise

WSO2 Platform Multitenancy Support

### Pluggable, Extensible, and Themeable

- All components are highly customizable through styling, theming, and code extensions
- Storefront implemented with React JS for easy customization
- REST API with an extensible security mechanism; OAuth 2.0 is used by default
- Pluggable to third-party analytics systems and billing systems
- Pluggable to existing user repositories including Microsoft Active Directory, LDAP, databases, or Apache Cassandra
- Components usable separately: API Developer Portal can be used to catalog APIs deployed in third-party gateways

### Easily Deployable in Your Enterprise

- Role-based access control for managing users and their authorization levels
- Storefront can be deployed in DMZ for external access with publisher inside the firewall for private control
- Different user stores for developer-focused storefront and internal operations in Publisher
- Integrates with enterprise identity systems including LDAP and Microsoft Active Directory
- Gateway can be deployed in DMZ with controlled access to WSO2 Identity Server (for authentication/authorization) and governance database behind firewall

### WSO2 Platform Multi-tenancy Support

- Run a single instance and provide API management to multiple customers, each in their own domain
- Share APIs between different departments in a large enterprise