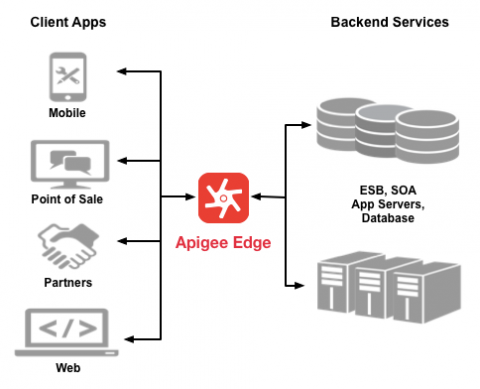
**Apigee**

Apigee Edge is a platform for developing and managing APIs with a proxy layer.

Companies today want to provide their backend API’s to make their service available.so Client app developers then make HTTP requests to these endpoints to consume that.

Apigee Edge enables you to provide secure access to your services with a well-defined API that is consistent across all of your services, regardless of service implementation.

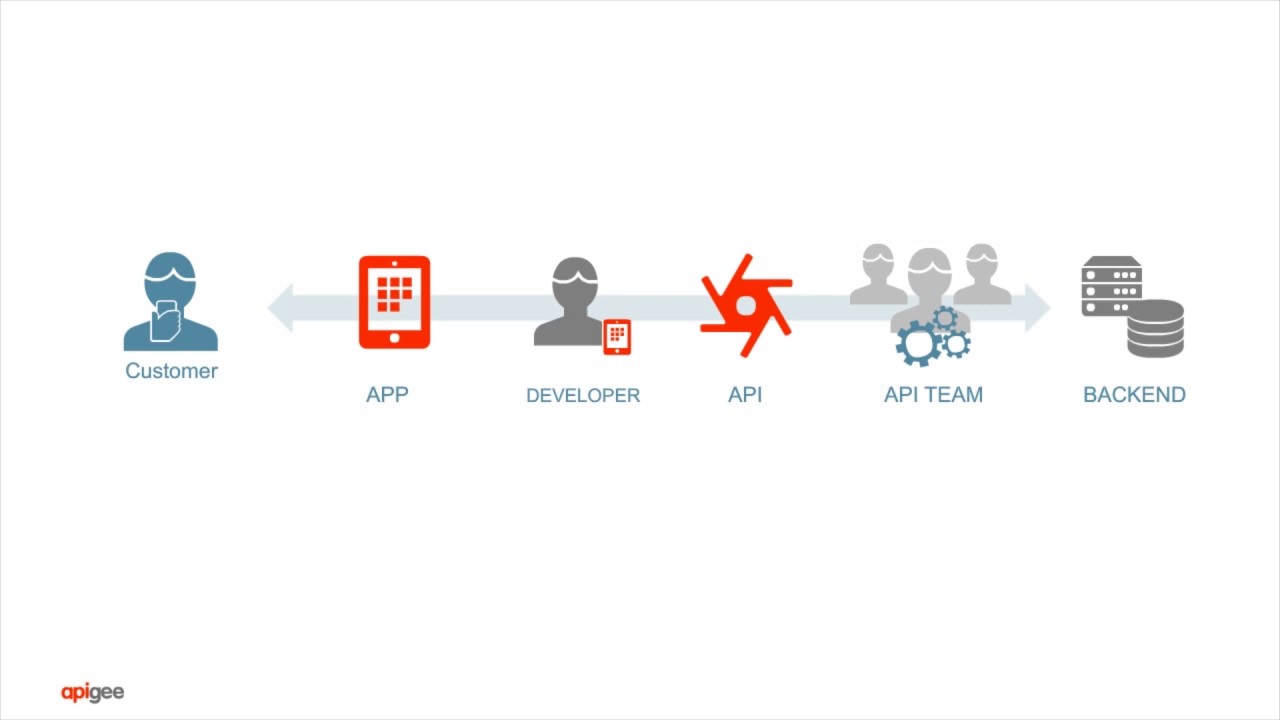
* Makes it easy for app developers to consume your services.
* Enables you to change the backend service implementation without affecting the public API.
* Enables you to take advantage of the analytics, monetization, developer portal, and other features built into Edge.



Rather than having app developers consume your services directly, they access an API proxy created on Edge.

The API proxy isolates the app developer from your backend service. Therefore you are free to change the service implementation as long as the public proxy API remains consistent.

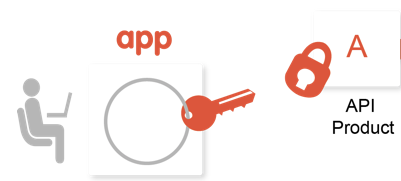
By maintaining a consistent frontend API, existing client apps will continue to work regardless of changes on the backend.

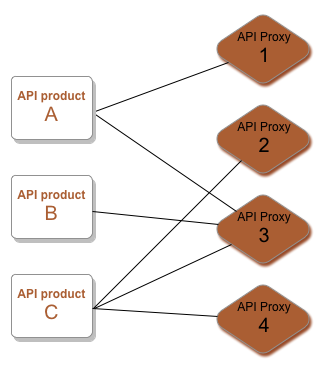
You can use policies on the API proxy to add functionality to a service without having to make any changes to the backend service. For example, you can add policies to your proxy to perform data transformations and filtering, add security, execute conditional logic or custom code, and to perform many other actions. The important thing to remember is you implement policies on Edge, not on your backend server.

**API Product:**

An API product is a bundle of API proxies combined with a service plan.

That service plan can set access limits on API proxies, provide security, allow monitoring and analytics.

Multiple API products can share the same API proxy.



We can set multiple Api products and access keys for specific Api products.Clients can access our proxy apis with that keys so we can monitor clients usage of our api.That will be useful in the billing process.

**Portal :**

We can add multiple Api Products based on customer requirements into a portal and provide portal url to consumers.

Out-of-the-box, the developer portal enables your portal users to create a developer account, sign in, and manage the apps that will consume your APIs.

Policies:

There are 30+ Policies are available in Apigee edge for security, Traffic management and mediation.

Some of the policies are

# ResponseCache policy

Caches data from a backend resource, reducing the number of requests to the resource. As apps make requests to the same URI, you can use this policy to return cached responses instead of forwarding those requests to the backend server. The ResponseCache policy can improve your API's performance through reduced latency and network traffic.

# ResetQuota policy

For example, the target Quota policy limits a developer to 1000 requests per week.

# SpikeArrest policy

This element throttles the number of requests processed by an API proxy and sent to a backend, protecting against performance lags and downtime.

# JSONtoXML policy

This policy converts messages from the JavaScript Object Notation (JSON) format to extensible markup language (XML), giving you several options for controlling how messages are converted.

# XMLtoJSON policy

This policy converts messages from the extensible markup language (XML) format to JavaScript Object Notation (JSON), giving you several options for controlling how messages are converted.

OAuth v2 policies:

# VerifyAPIKey policy

|  |
| --- |
| Extension policies |

Let you define custom policy functionality, such as service callout, message data collection, and calling Java, JavaScript, and Python scripts.