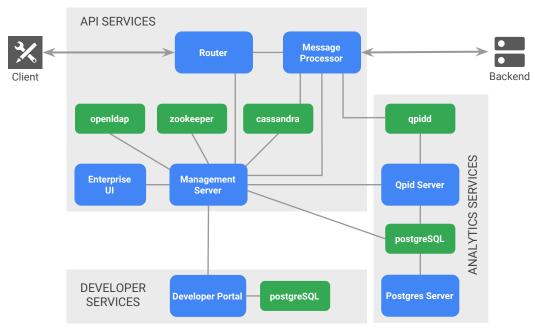
apigee

Terminology and organizational structure

Components view

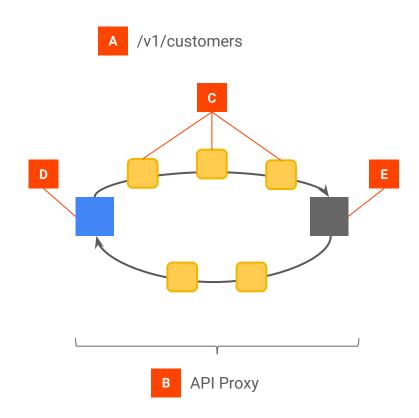
Each box represents a process. These processes placed on placed on dedicated dedicated VMs or grouped to share the same host.





API and API Proxy

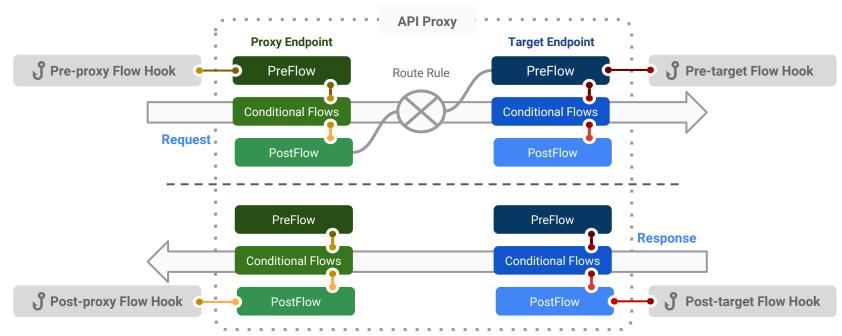
- A. **API** An API is an interface that makes it easy for one application to 'consume' capabilities or data from another application.
- B. API proxy An API proxy consists of a bundle of XML configuration files and optional additional resources such as Javascript files. The bundle describes the request and response cycle for a given API. The API behavior is model the application of Policies that allow you to control aspects such as security, mediation, transformation, enrichment, among many others.
- C. **Policy** A policy is like a module that implements a specific, limited management function as part of the proxy request/response flow.
- D. Virtual Host Virtual hosts are similar to Apache Virtual hosts and route traffic to environments based on ports and domain names.
- E. Target Server TargetServer configurations decouple concrete endpoint URLs from TargetEndpoint configurations. Each TargetServer is referenced by name in a TargetEndpoint HTTPConnection. Instead of defining concrete URL in the configuration, you can configure one or more named TargetServers.





API Proxy - Request/Response Cycle

Segment each part of the flow, including default and non-default paths, as well as pre-flow and post-flow for both the request and response.



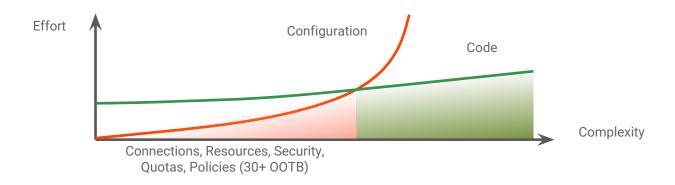


Policies

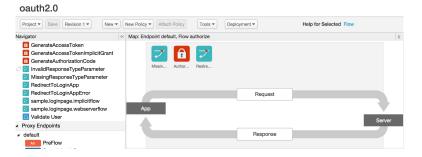
Traffic management policies	Mediation policies	Security policies	Extension policies
Traffic management policies let you configure cache, control traffic quotas and spikes, set concurrent rate limits, and so on.	Mediation policies let you perform message transformation, parsing, and validation, as well as raise faults and alerts.	Security policies let you control access to your APIs with OAuth, API key validation, and other threat protection features.	Extension policies let you provide custom policy functionality, with support for such features as service callout, message data collection, and calling Java, JavaScript, and Python behavior you have created.
 Cache policies Concurrent Rate Limit policy Quota policy Reset Quota policy Spike Arrest policy 	 Access Entity policy Assign Message policy Extract Variables policy JSON to XML policy Key Value Map	 Access Control policy Basic Authentication policy JSON Threat Protection policy LDAP policy *† OAuth v2.0 policies OAuth v1.0a policy Regular Expression Protection policy SAML Assertion policies Verify API Key policy XML Threat Protection policy 	 Java Callout policy JavaScript policy Message Logging policy Python Script policy Service Callout policy Statistics Collector policy



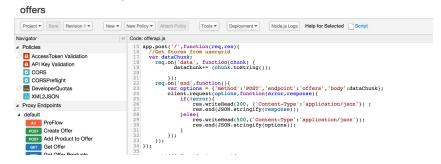
Policies and Extensions



Configuration Driven Proxy Definition

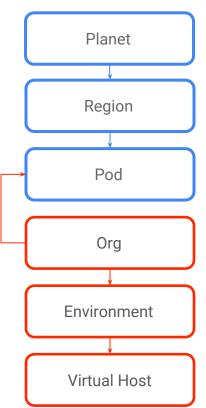


Code Driven Proxy Definition





Multitenancy



Planets represent an entire physical installation and it can encompass multiple regions and pods

Regional Services are shared by many pods across a single geographical region and maintain state and provide an API that works across the pods in a region

Pods are a collection of servers that share logical functions such as a Gateway Pod or an Analytics Pod.

Orgs provide logical grouping to secure access to API management services. Orgs are associated with pods for servers – they can either share hardware or be isolated

Environments are virtual routes that allow API bundles to be deployed and tested within and Org. Environments can be associated with pods for servers independently of the Org.

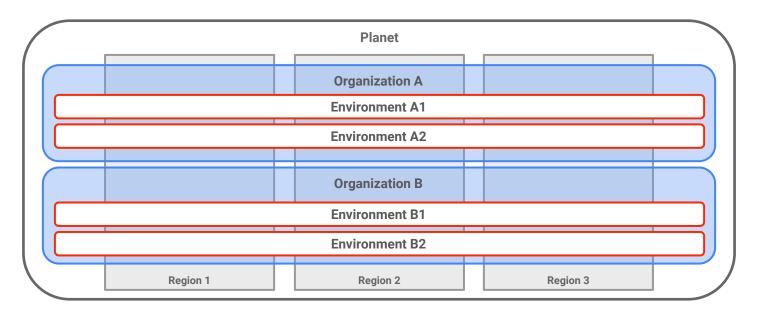
Virtual Hosts are similar to Apache Virtual hosts and route traffic to environments based on ports and domain names.



Multitenancy

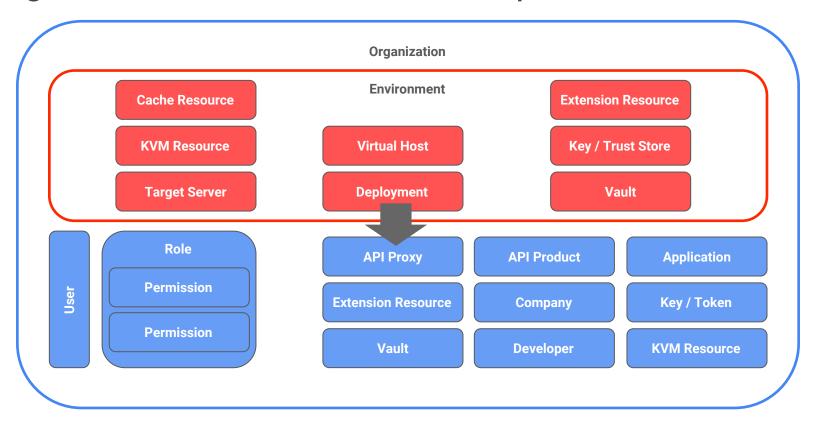
A Planet may contain multiple DCs. Organization and Environment expands across the planet.

A planet may contain multiple Organizations. Organizations represent tenants. An Organization can have multiple environments. Organization and environment represent conceptual boundaries driving logical data, and in some cases, processing partitioning.





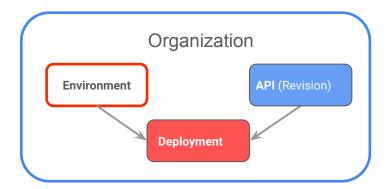
Organization & Environment scope





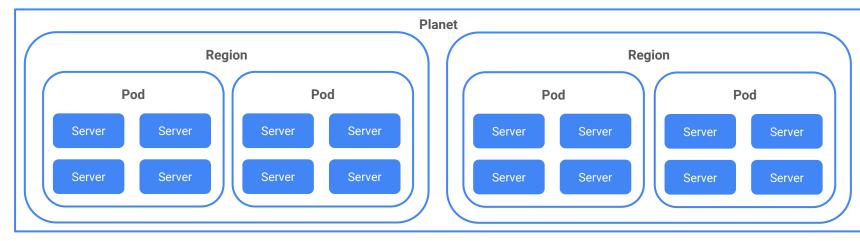
Environments, API Proxies and Deployments

- API proxies process API traffic and are created or imported into organizations.
- API proxies revisions are deployed into environments in order to become active.
- Deployments are the association between an API proxy revision and an environment.
- An API proxy may be deployed across one or more environments.





Regions, Pods and Servers



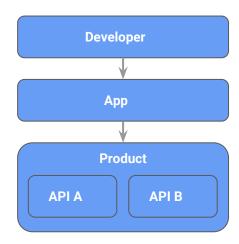
- Servers are grouped into pods, and pods are grouped into regions.
- · Regions represent geographically separate data centers
- · Pods groups servers by functions.
- 3 Pods types are used. Gateway Pod, Central Pod and Analytics Pod.
- Servers are registered with the management server at install time. Uniquely identified by UUID.

- Functions
 - API processing: Router, Message Processor.
 - System management: Management Server, Enterprise UI
 - Service management: Postgres Server, Qpid Server.
- Registration is the setup step that adds a server to the system and describes what functions the server offers to other components.
 - Examples: management-server, user-settings-datastore.



Products, Apps and Developers

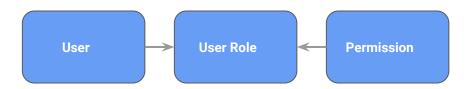
- Products bundle together API proxies to offer them to Developers.
- Developers are external API users who interact with your APIs by developing applications that use them. Developers may be grouped into Companies.
- Apps are associations between Developers and Products.





Users and Roles

- Users are accounts with some level of access on Edge
 - Uniquely identified by email address
 - Used to log into the user interface and issue calls to the management API
- Roles are groups that connect users to permissions in the system
- Both users and roles may be created at the global or organizational levels
 - Global users may be members of global and organizational roles
 - Organizational users may only be members of organizational roles





apigee Thank You