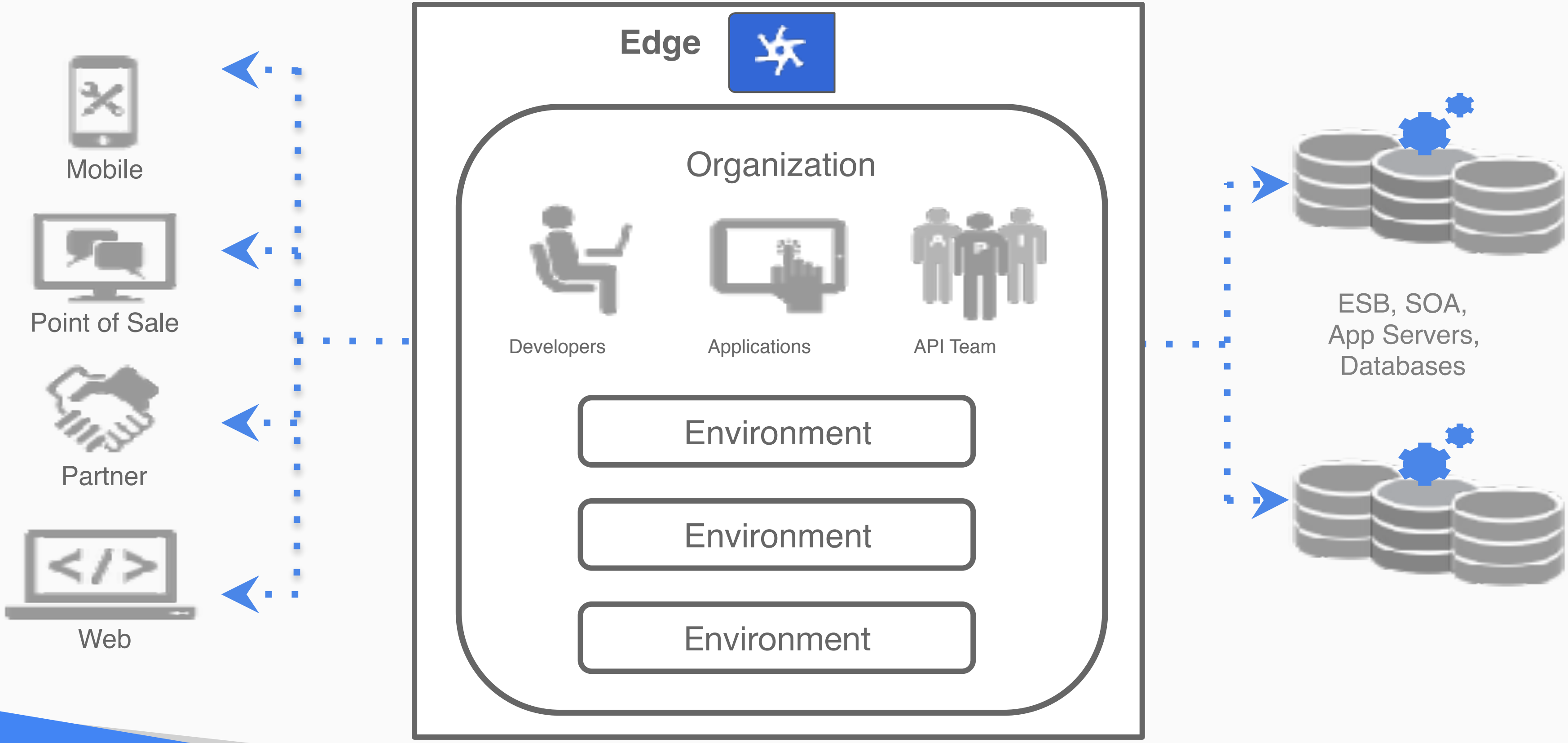
Google Cloud

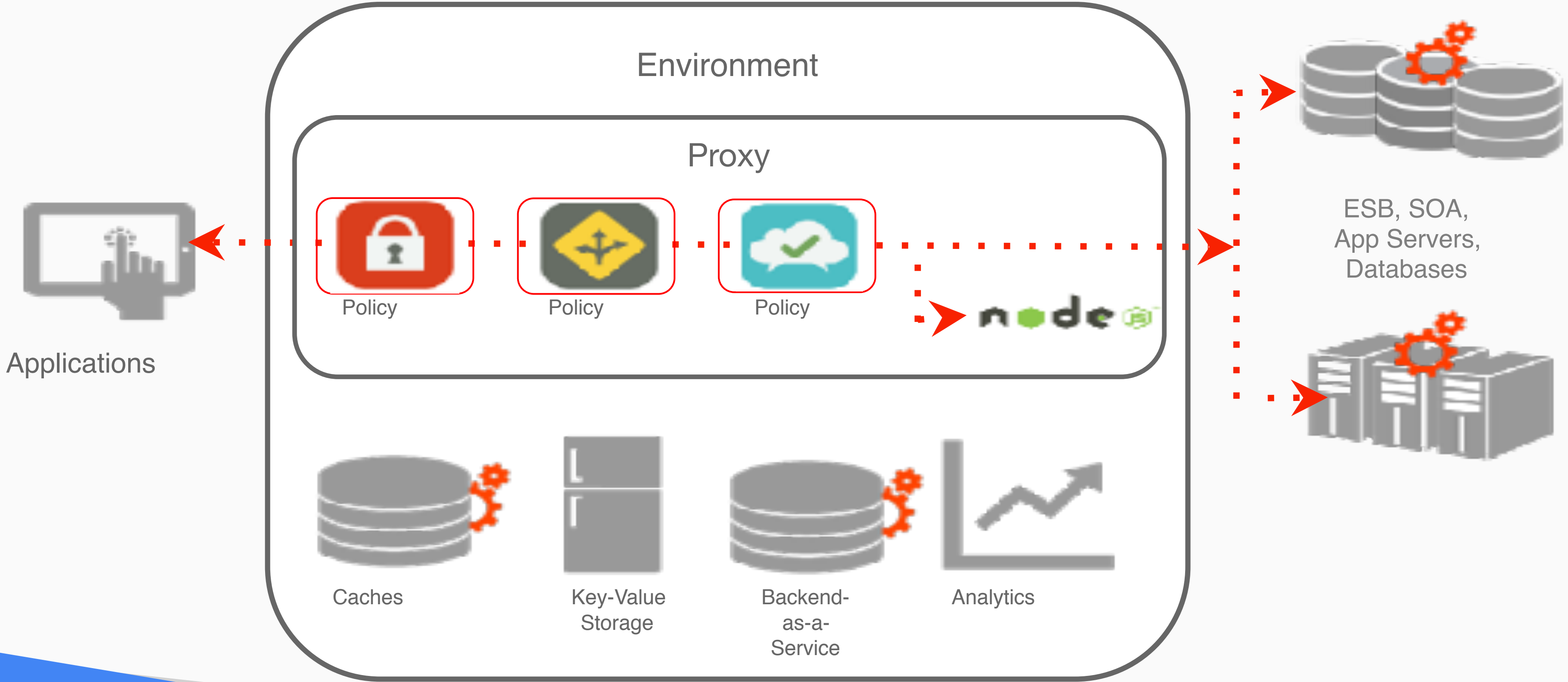
Edge Overview Series

Fundamental Concepts and Keywords

Concepts in a nutshell



API Proxies and Policies



Key Terms

- An **Organization** is a collection of users, APIs and other resources.

- An **Environment** is a subset of APIs in an organization that are in a given deployment state
  - Default environments: Test and Prod

- An **API Proxy** is a set of configurable logic that handles API requests.

- **Flows** *(sometimes called resources)* represent a specific request type within an API proxy - usually qualified by verb and path, but often by other request parameters as well.

- **Policies** (also known as Flow steps) are bits of logic that can be executed during the course of processing a request
  - Policies can be applied to all resources in a proxy or only to select resources
  - Policies can be conditionally executed

Developers and Apps

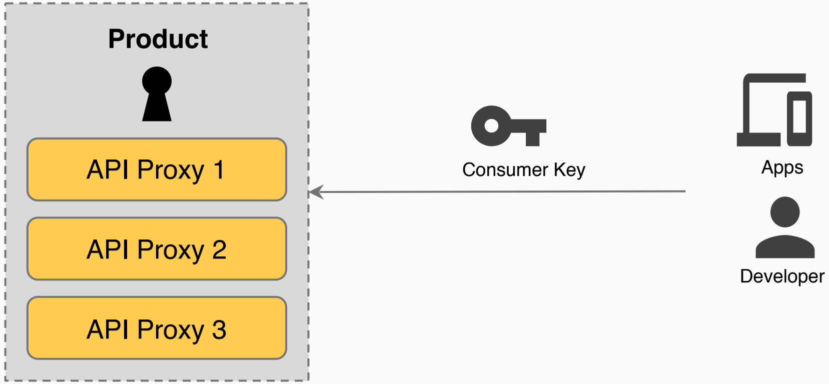
- **Developers** are the internal or external partners that create applications that use your API products
  - can be internal or external, and generally represent individuals
  - can be grouped into Companies

- Developers are associated with **Applications**, which are developer-written programs that use APIs
  - can also be included in Companies

- Applications (and Flows**)** can be grouped into **Products** for exposure to Application Developers
  - Application Developers are restricted by **Products**

NOTE: Companies, Company App Family, Company Developers can be configured using API. For configuring the above using UI, the org needs to enable monetization and some customization in the Developer Portal

API Security



Consumer Keys are:

- assigned to an application when it is created
- linked to a product when the product is associated with an application

API Products are:

- collections of API resources, combined with a service plan and presented to developers as a bundle
- the central mechanism for authorization and access control

NOTE: Consumer Key is also known as Client ID and API Key

Role based Access for Edge Users

- **Permissions** define create/read/update/delete access to resources
- **Roles** identify a collection of permissions that can be assigned to a user
- Predefined roles assign common permissions to key Edge resources

Organization Administrator

- Administrator of an organization
- Responsible for mainly user management but has super access to everything

Business User

- API Program Manager
- Responsible for success of API program and developer management, KPIs

Operations Administrator

- API Operation Manager
- Responsible for production and test deployment, troubleshooting

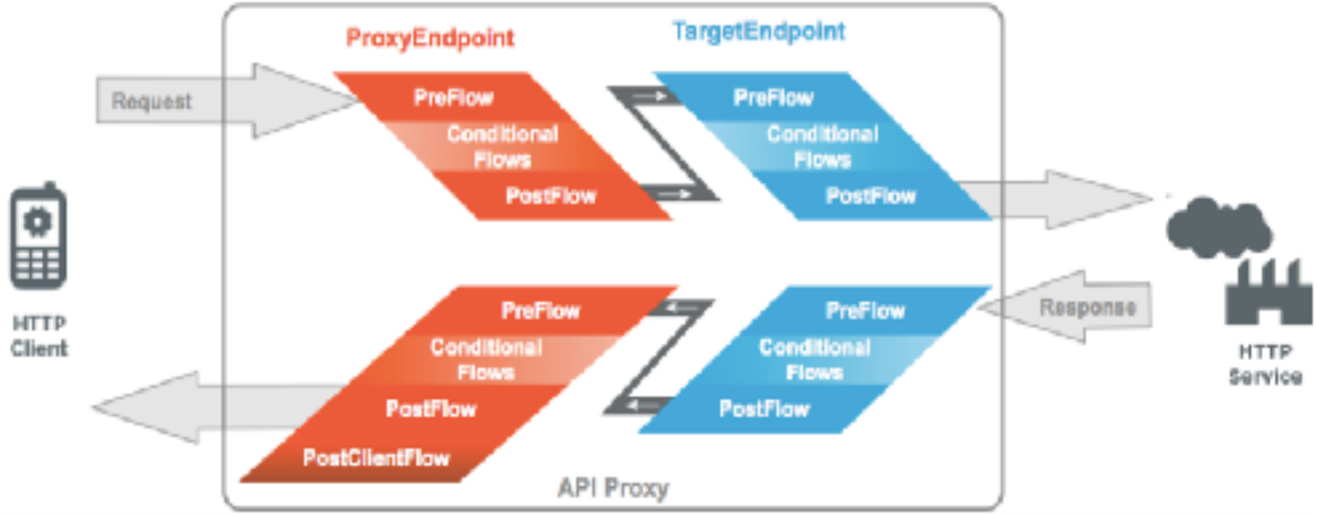
User

- API Developer
- Responsible for development of API proxy, policy management, troubleshooting etc.

API Proxy and Target Endpoints

- Define an API as a series of resources that access a given target system.
- Client-side interfaces can be accessed using either HTTP or HTTPS
- Targets can be accessed using either HTTP or HTTPS, using either one-way SSL or two-way SSL with mutual authentication
- REST and SOAP targets supported
- “First match” selection: define a set of resource criteria matching incoming requests, and the first match found controls request execution
- Resource matching on path nodes, query parameters, HTTP verbs and other types of conditions
- Determine target service for requests using either static or dynamic routing with route rules defined in the proxy endpoint

API Proxy Flows

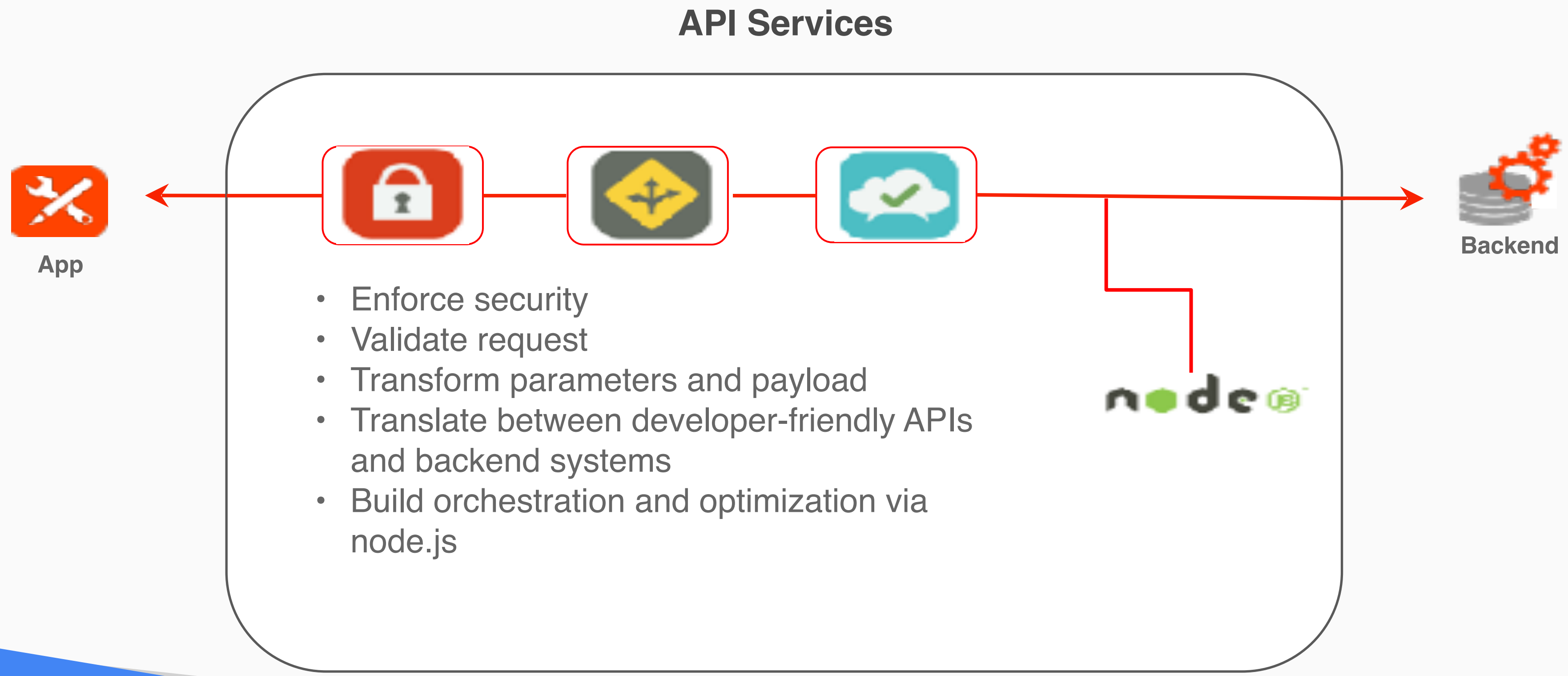


The Power of Policies

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 Operations policy ● Raise Fault policy ● SOAP Message Validation policy ● XML to JSON policy ● XSL Transform policy	● 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

* Cloud Enterprise only
† On-Premises installation only

Processing Pipeline



- Enforce security
- Validate request
- Transform parameters and payload
- Translate between developer-friendly APIs and backend systems
- Build orchestration and optimization via node.js

Variables and Conditions

- **Variables** allow you to store data for use during policy execution
  - Create using the Assign Message policy or from JavaScript/Java policies
  - Edge provides an extensive set of predefined variables covering areas such as:
    - System (date/time, hostname, etc.)
    - Configuration (organization/environment/application name, proxy base path, etc.)
    - Request and response (client IP address, query and form parameters, headers, request body, target hostname, timing data, etc.)
    - Policy (variables specific to the individual policy, such as rate limit info)
    - OAuth 1.0a and 2.0 (information related to access tokens, etc.)

- **Conditions** allow you to control when a policy gets executed and which of a number of resource definitions is selected for processing
  - Compare path nodes, HTTP verbs, headers, query parameters, form parameters or variables with each other

Controlling Edge using Management APIs

- Create, manage and delete just about anything:
  - API proxies
  - Policies
  - Developers and companies
  - Apps, app families and app keys
  - API products
  - Environments
- Export and import entities
- Manage users within org

Controlling Edge using Mgmt APIs

- Create, delete, approve and revoke OAuth authorization codes, access tokens and refresh tokens
- Retrieve statistics for environments, APIs, apps, developers, etc.
  - By default, you can retrieve stats on response times, response status codes, request/response sizes, request/response latency, and other dimensions
  - Custom analytics policies allow you to gather and report on data from request paths, query params, headers or payloads
- Start debug sessions and retrieve information

What does it mean to “develop with Edge Platform”?

https://community.apigee.com/articles/41051/what-does-it-mean-to-develop-with-apigee-edge.html

Getting Setup

Steps

- Using a web browser Navigate to https://login.apigee.com/login
- If you have an apigee account, login
- If you do not have an apigee account, select “sign up”
  - Fill out form and select create account
  - Activate Account by selecting link in email
    - Could take 5 – 10 minutes

- Under API Management select the “launch” or “activate” button
- Welcome to Edge!!

THANK YOU