Zero to Thousands TPS: Private Cloud Operations Workshop

Maudrit Martinez
Apigee

Paul Mibus
Apigee

apigee

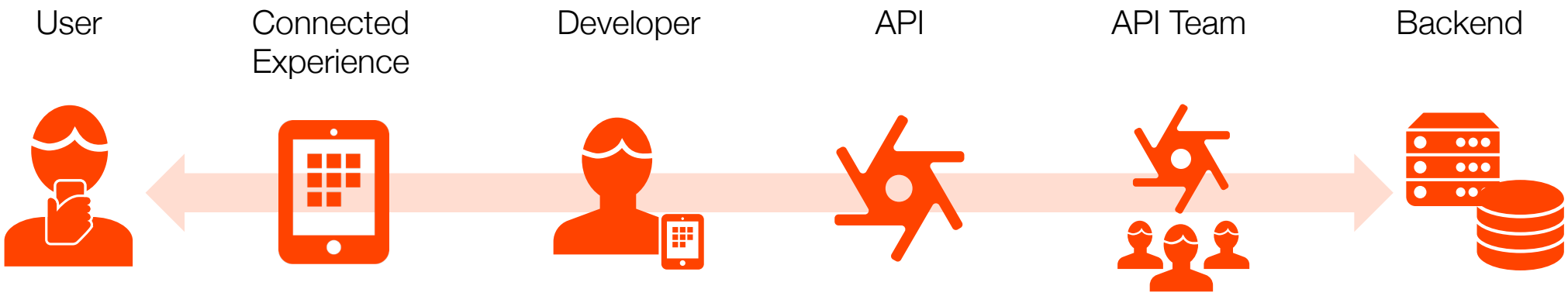
Agenda

9:30 – 10:00	**Apigee Overview**
10:00 – 11:00	Apigee Private Cloud Architecture
BREAK (15 MINS)	
11:15 – 12:00	Apigee Private Cloud Architecture (cont)
BREAK – LUNCH (60 MINS)	
13:00 – 15:45	Apigee Edge Install
BREAK (15 MINS)	
16:00 – 17:30	Platform Operations

apigee

Apigee Edge Overview

Intelligent API platform enables the digital value chain



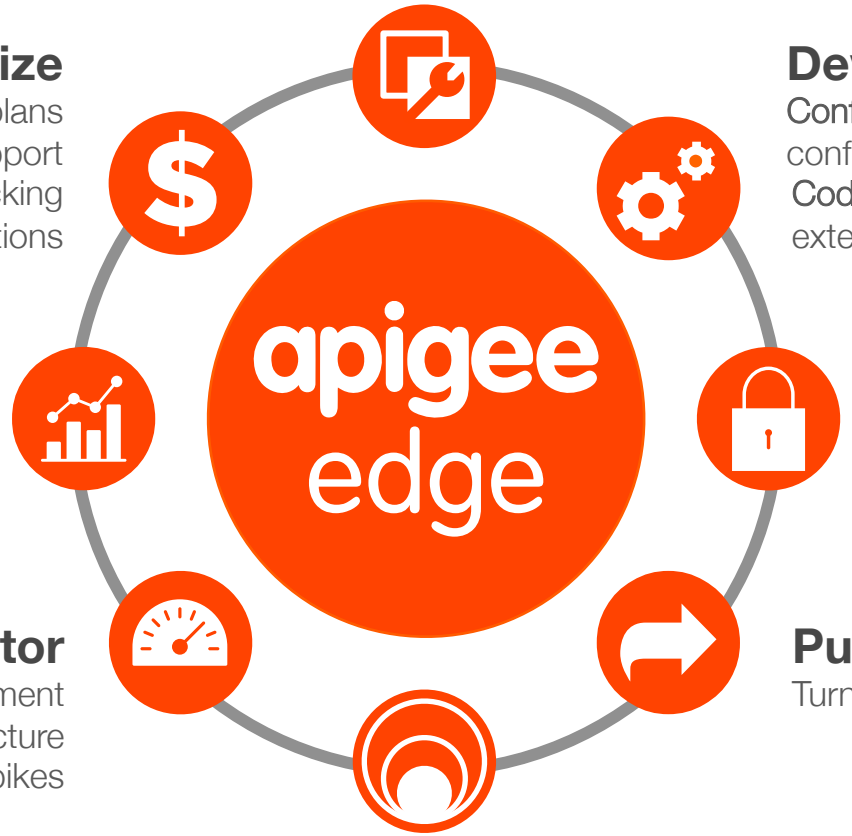
Intelligent API Platform

Apigee Edge

Design
Design first. Document Smart.
Full support for Swagger 2.0

Monetize
Flexible rate plans
Internationalization support
Usage tracking
Limits and notifications

Develop
Configuration: Over 30 ready-to-use and configurable policies
Code: Built-in support for Node, JavaScript and Java extensibility BaaS



Analyze
Complete visibility– from app end to backend
Automatically and continuously collect all API-traffic data out of the box

Secure
End-to-end security
Threat protection
Access control
Simple OAuth implementation for your APIs
PCI and HIPAA compliance

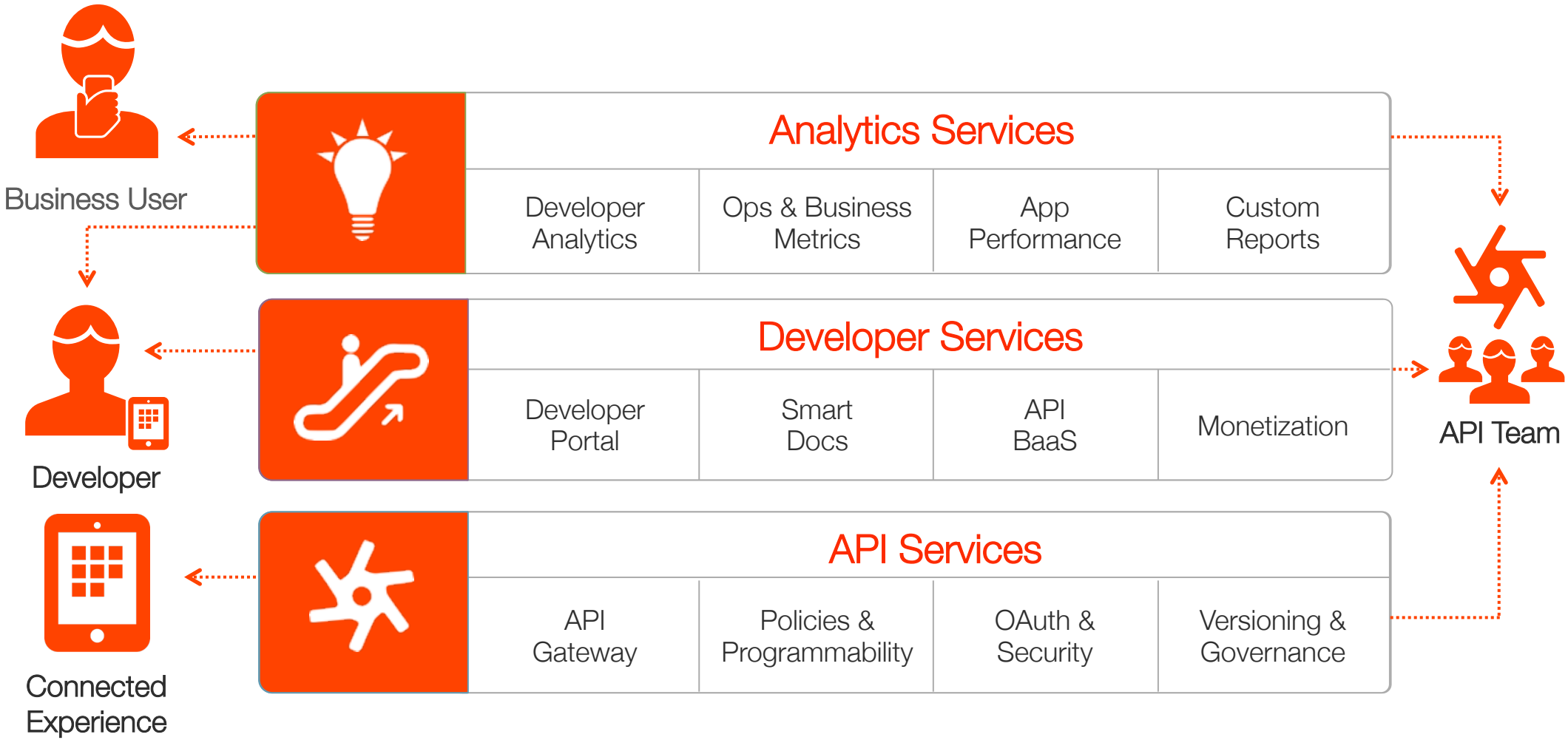
Monitor
Centralized control, decentralized development
Multi-tenant architecture
Billions of API calls, including large spikes

Publish
Turnkey developer portal

Scale
Self-service
State @ scale
Flexible deployment

apigee

Apigee Edge – API management



Apigee Edge policies – Build APIs faster

Manage interactions with API consumers and optimize performance

Transform, translate and reformat data for easy consumption

Secure APIs and protect back-end systems from attack

Extend with programming when you need it

Config - Over 30 ready-to-use and configurable policies

Code - Built-in support for Node, JavaScript, Java and Python extensibility

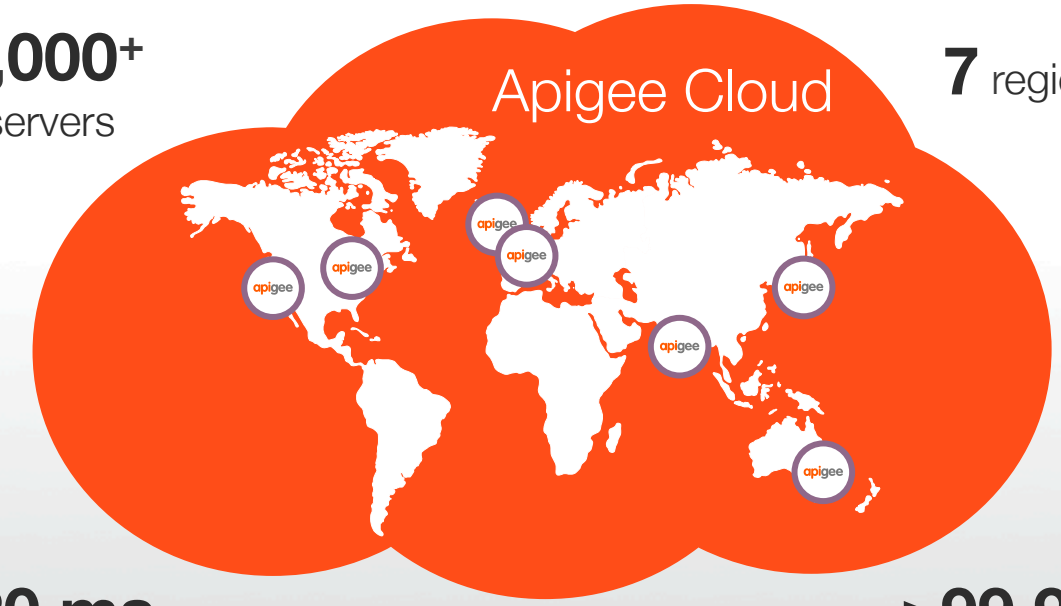
apigee

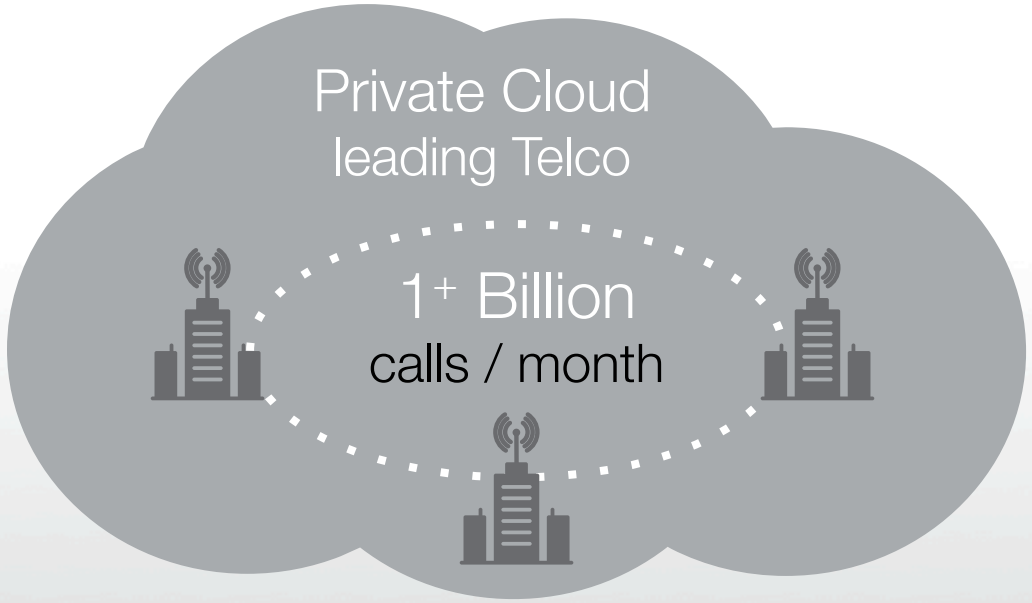
Flexible deployment options

19 availability zones

3,000+ servers

7 regions





Multi-Datacenter Deployment

~20 ms
average latency

~1/2 billion
transactions / day

>99.9%
API availability

Public Cloud = Private Cloud

apigee

Agenda

09:30 – 10:00	Apigee Overview
10:00 – 11:00	**Apigee Private Cloud Architecture**
BREAK (15 MINS)	
11:15 – 12:00	Apigee Private Cloud Architecture (cont)
BREAK – LUNCH (60 MINS)	
13:00 – 15:45	Apigee Edge Install
BREAK (15 MINS)	
16:00 – 17:30	Platform Operations

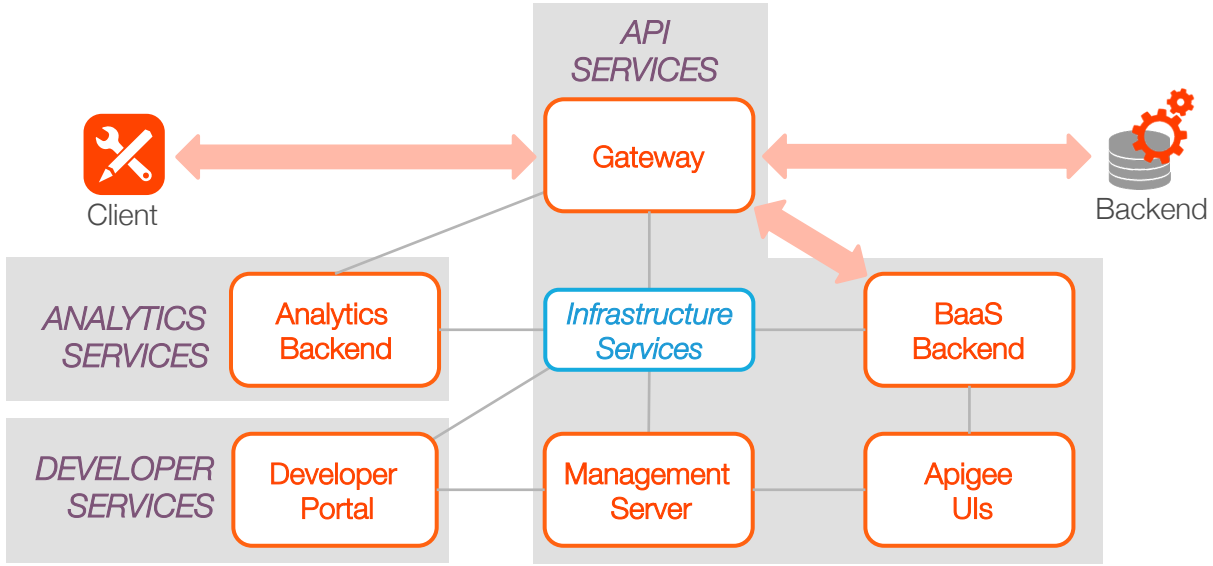
Apigee Private Cloud Architecture

Apigee Edge architecture – High level

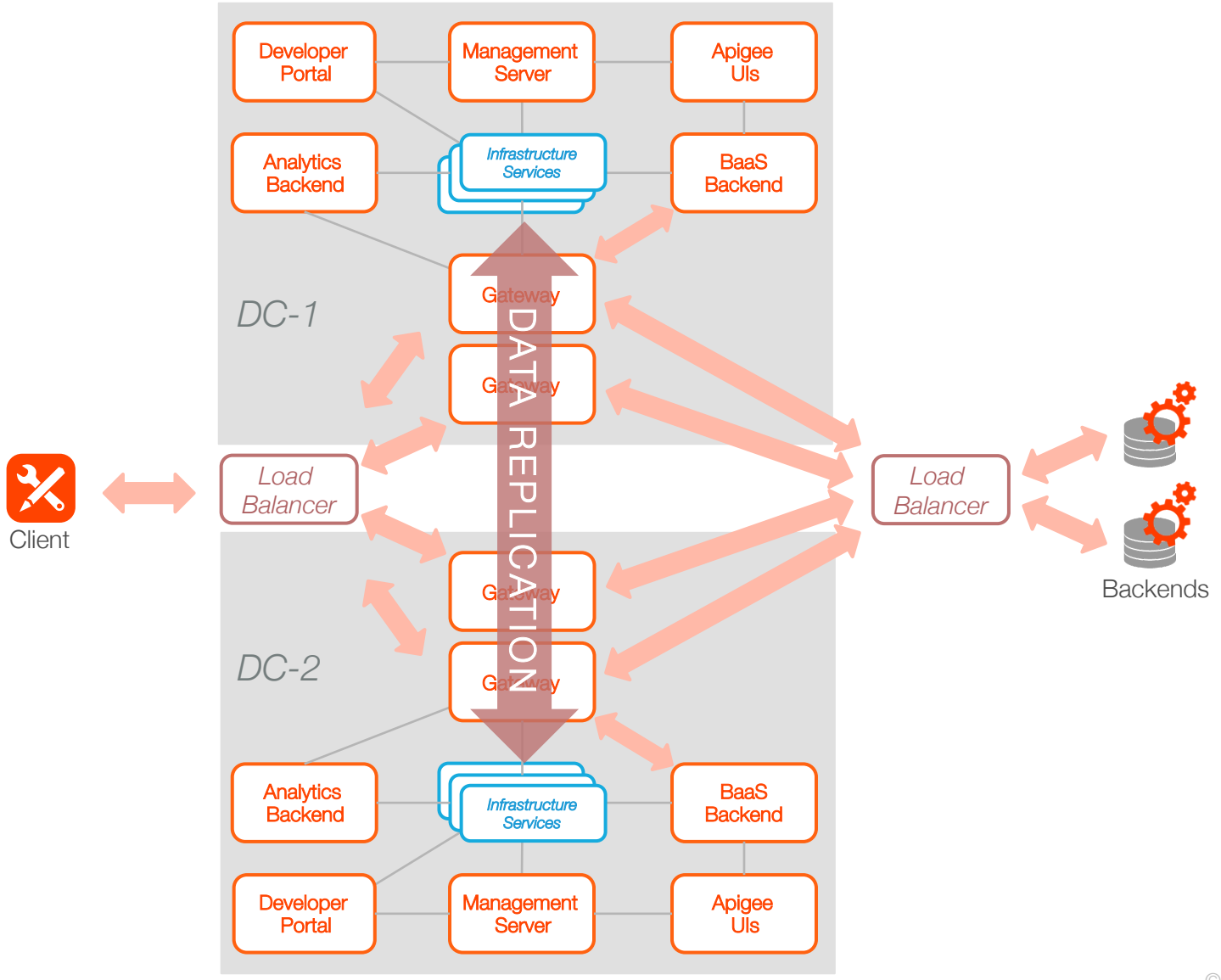
Apigee Edge is comprised of several stateless components that use infrastructure services to persist data

- Gateway: Routing and processing API calls
- Apigee UIs: Enterprise UI, Developer Portal
- Infrastructure Services: Persistence and queuing of run time data
- Management Server: Provider of REST APIs for all configuration tasks

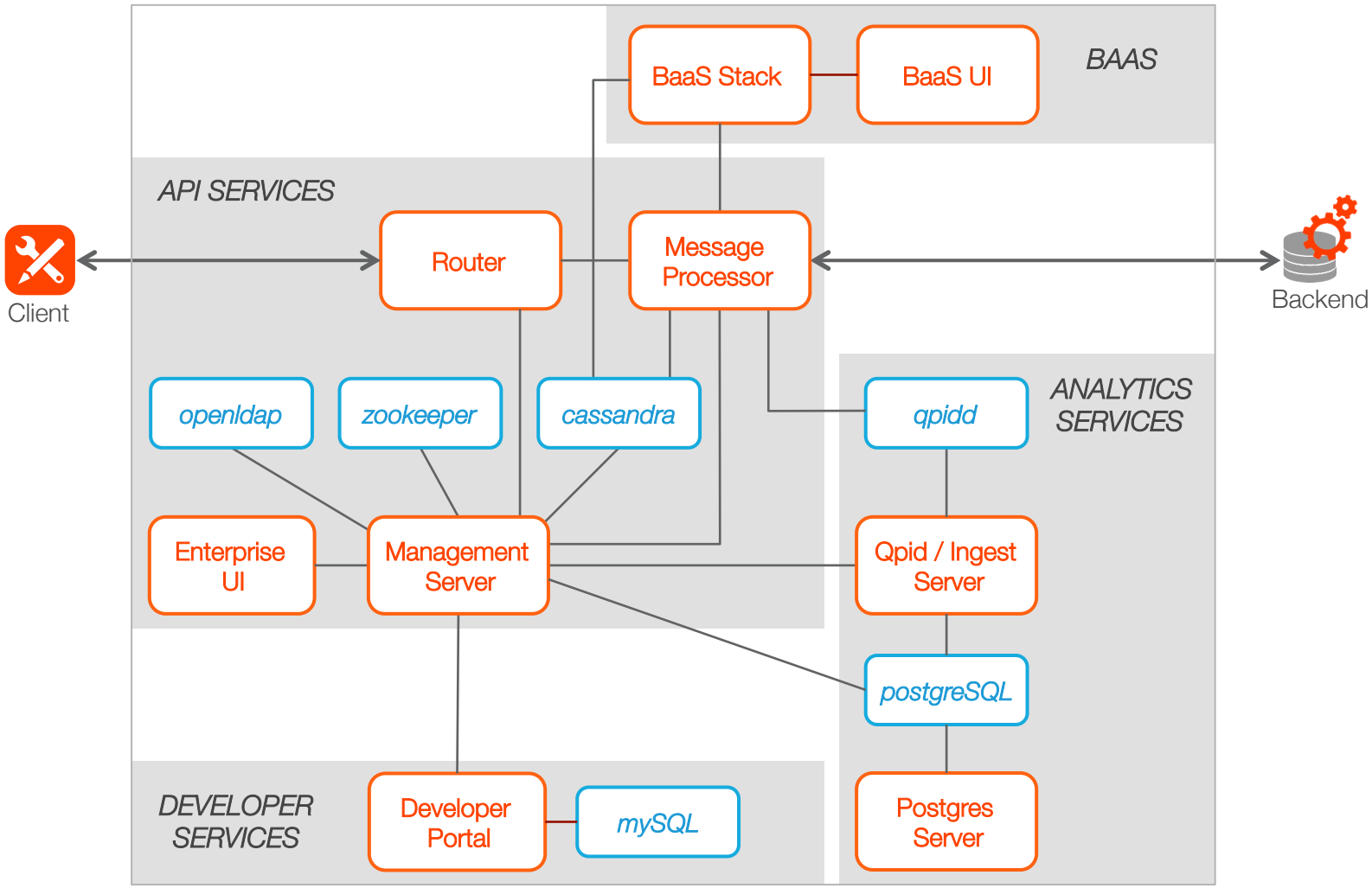
Note: Monetization is part of Developer Services and leverages Gateway, Analytics Services and Management Server



Apigee Edge architecture – High level



Apigee Edge architecture – Component view



- Router handles all incoming API traffic and dispatches it. The Router terminates the HTTP request, handles the SSL traffic, and uses the virtual host name, port, and URI to steer requests to the appropriate node.
- Message Processor handles API traffic for a specific organization and environment and which executes all policies.
- Apache Cassandra stores application configurations, distributed quota counters, API keys, and OAuth tokens for applications running on the gateway.
- Apache ZooKeeper contains configuration data about all the services of the zone and which notifies the different servers of configuration changes.
- OpenLDAP contains organization user and roles.
- Management Server offers an API that is used by the Central Services server to communicate with the servers in each on-premises installation.
- QPID Server manages queuing system for analytics data.
- Postgres Server manages analytics database.

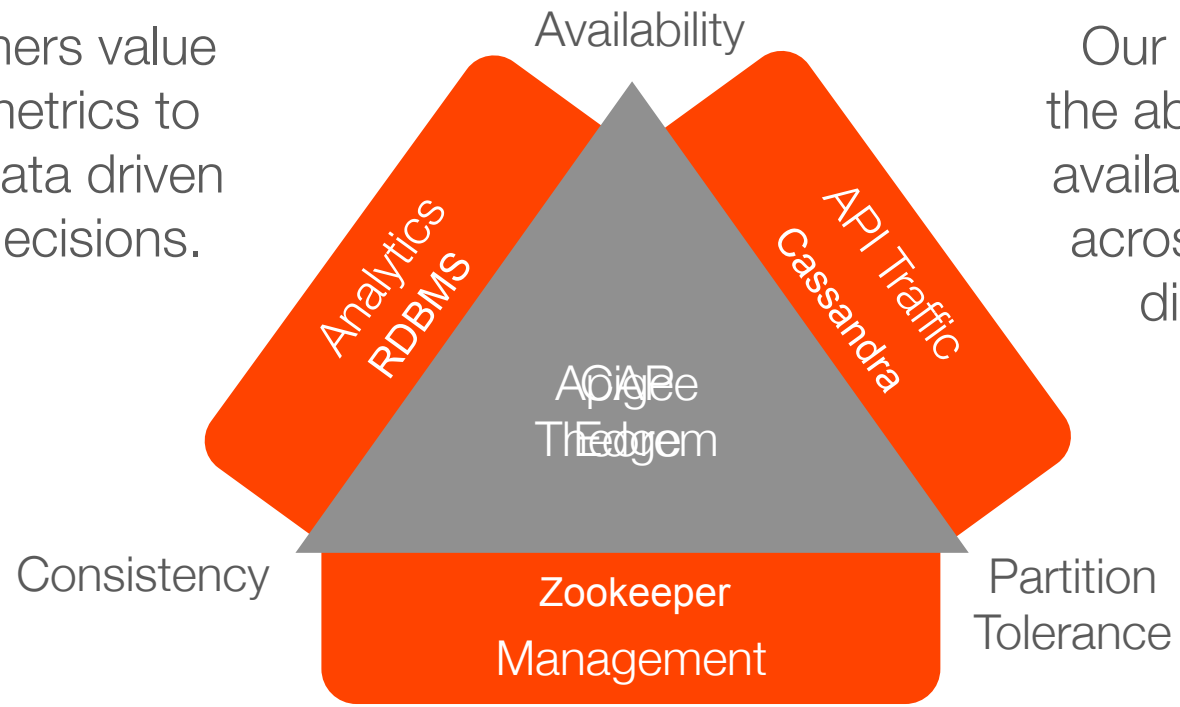
Legend:

Apigee | Open Source

apigee

Apigee Edge architecture – Technology Stack

Our customers value accurate metrics to help drive data driven business decisions.



Our customers value the ability to have highly available API expanded across geographically dispersed sites.

Our customers value the ability to centralized management of distributed components.

In theoretical computer science, the **CAP theorem**, also known as Brewer's theorem, states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees:

- **Consistency** - all nodes see the same data at the same time.
- **Availability** - a guarantee that every request receives a response about whether it succeeded or failed.
- **Partition tolerance** - the system continues to operate despite arbitrary message loss or failure of part of the system.

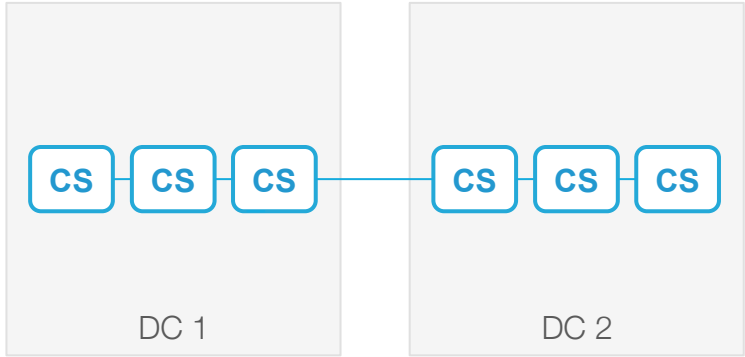
Text Credits: Wikipedia

Apache Cassandra

"Apache Cassandra is an open source distributed database management system. It is an Apache Software Foundation top-level project designed to handle very large amounts of data spread out across many commodity servers while providing a highly available service with no single point of failure." -- Wikipedia

Cassandra characteristics:

- All nodes are equal. Not master/slave or primary/secondary.
- An application can read/write data from any node.
- Data replication. Apigee Edge uses replication factor 3.
- Consistency managed by application. Apigee Edge uses one and local quorum.



Cassandra is used by Apigee for a variety of purposes, including:

- Storage of developer, application and API Product data
- Storage of access and refresh tokens
- Storage of key-value map data
- Audit logs
- Custom analytics report models

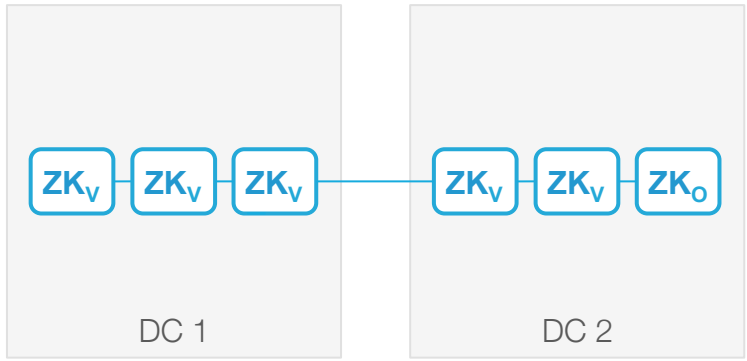
http://cassandra.apache.org/

Apache Zookeeper

"Apache ZooKeeper is a software project of the Apache Software Foundation, providing an open source distributed configuration service, synchronization service, and naming registry for large distributed systems."

– Wikipedia

Zookeeper is used by Apigee as a distributed configuration registry, tracking component location, configuration and status data. With some exceptions, it is NOT required to process API requests.



- **Leader:** The node that controls coordination of writes across distributed Zookeeper nodes
- **Voters:** Nodes that can vote on change proposal made by the Leader
- **Observers:** Do not vote on change proposals and must forward all writes to the Leader

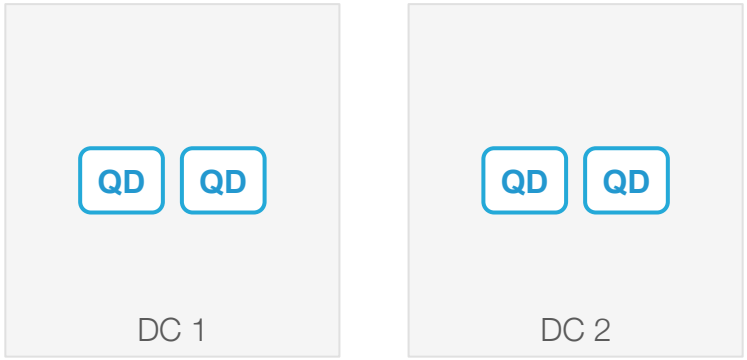
http://zookeeper.apache.org/

Apache Qpid

“Apache Qpid, an open-source (Apache 2.0 licensed) messaging system, implements the Advanced Message Queuing Protocol. It provides transaction management, queuing, distribution, security, management, clustering, federation and heterogeneous multi-platform support.”

– Wikipedia

Qpid is used by Apigee Edge as messaging system for analytics and monetization data.

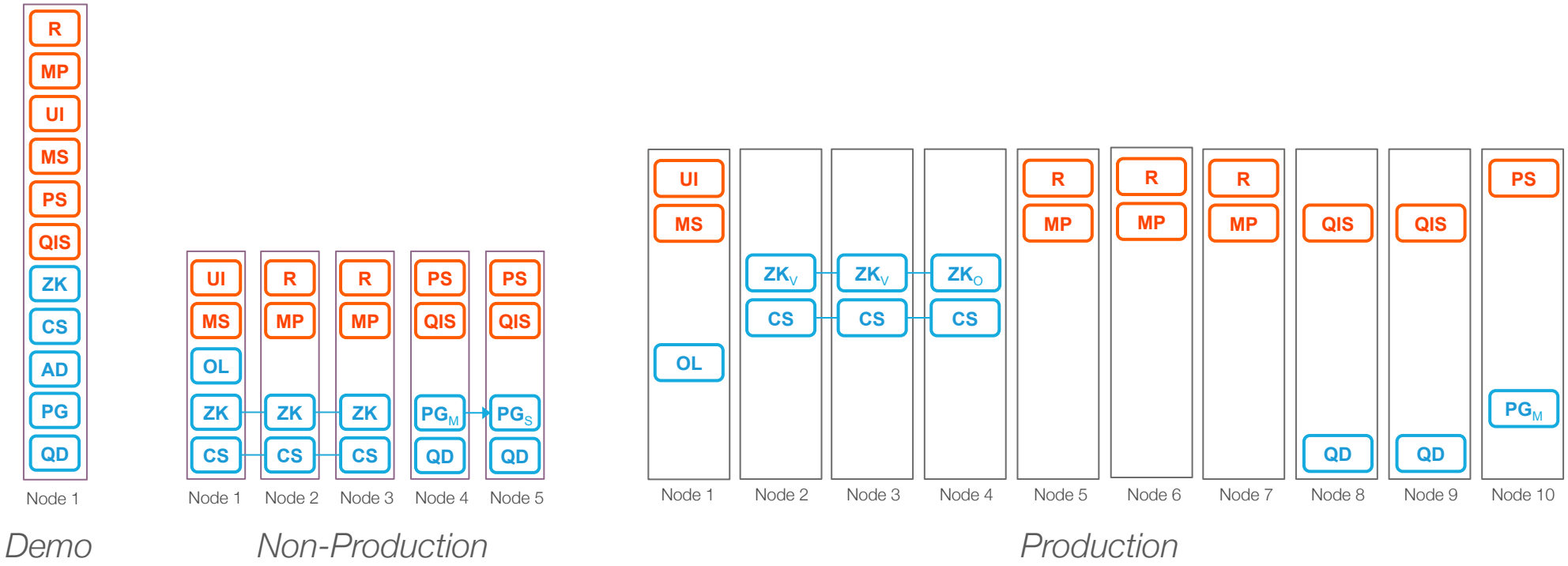


http://qpid.apache.org/

Legend:
R Router
MP Message Processor
UI Enterprise UI
MS Management Server
PS Postgres Server
QIS Qpid/Ingest Server
DP Developer Portal
BA BaaS Server
MY MySQL
ZK Zookeeper
CS Cassandra
OL Openldap
PG PostgreSQL
QD Apache Qpid
Server/Virtual Machine
POD

Apigee Edge architecture – Deployment options

- Apigee Edge architecture offers great flexibility when it comes to deployment options.
- Edge can be deployed from a single VM to a multi-datacenter active/active configuration.
- Edge has been designed from the ground up to be a true cloud scale solution, capable of running on both virtualized (including AWS) and physical hardware.

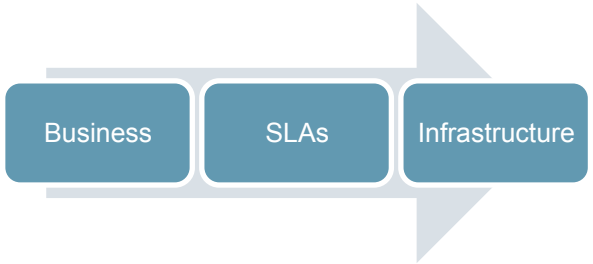


Legend:

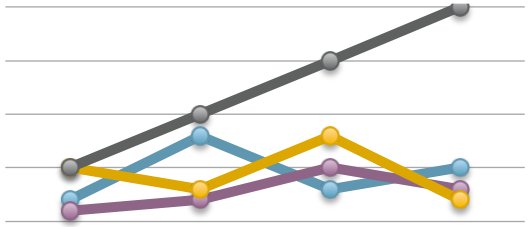
R Router	MS Management Server	DP Developer Portal	MY MySQL	OL Openldap	Server/Virtual Machine
MP Message Processor	PS Postgres Server	BA BaaS Server	ZK Zookeeper	PG PostgreSQL	POD
UI Enterprise UI	QIS Qpid/Ingest Server		CS Cassandra	QD Apache Qpid	

Topology selection – Key considerations

1. Know your business

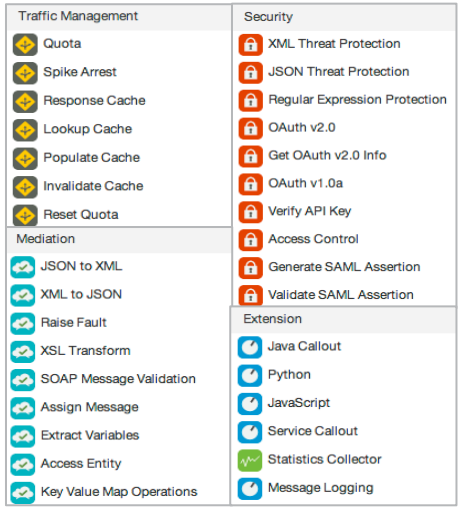


2. Understand traffic patterns

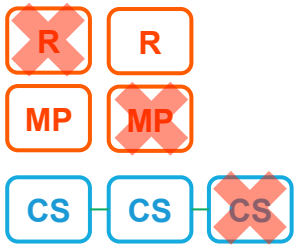


3. Plan for growth (2X, 5X, 10X?)

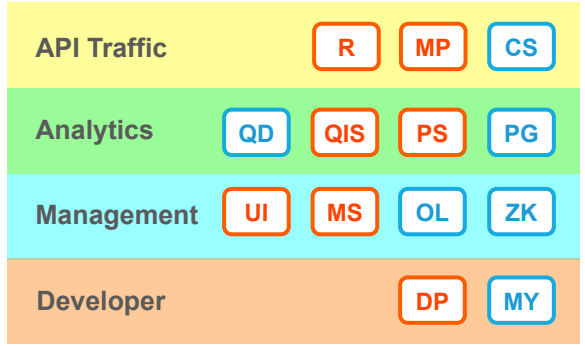
4. Know your API proxies



5. Remember everything fails



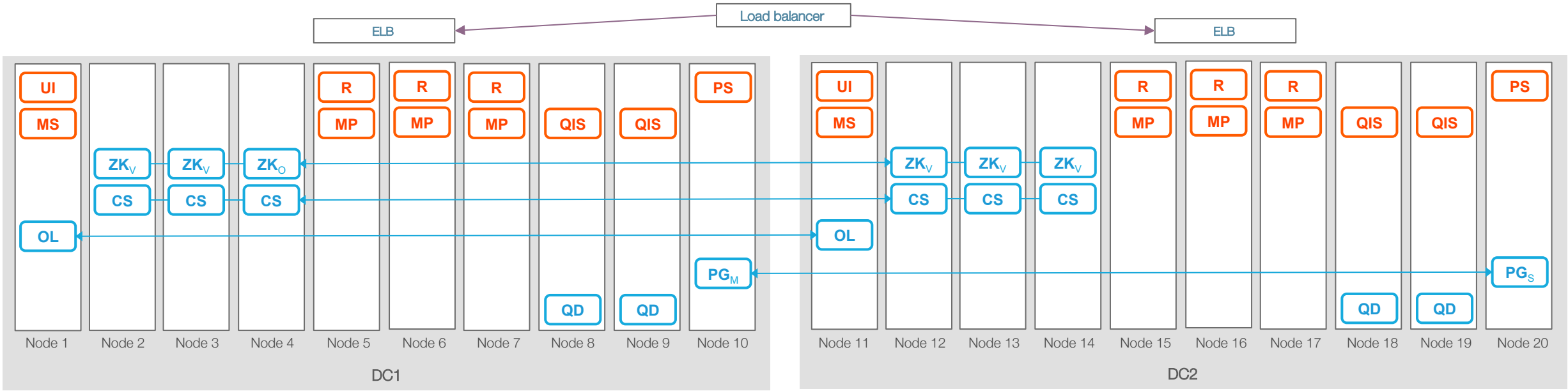
6. API Traffic, Analytics and Management, Developer Service components can and should scale independently.



Legend:

- R Router
- MP Message Processor
- UI Enterprise UI
- MS Management Server
- PS Postgres Server
- QIS Qpid/Ingest Server
- DP Developer Portal
- BA BaaS Server
- MY MySQL
- ZK Zookeeper
- CS Cassandra
- OL Openldap
- PG PostgreSQL
- QD Apache Qpid
- Server/Virtual Machine
- POD

Apigee Edge architecture – Architecture characteristics

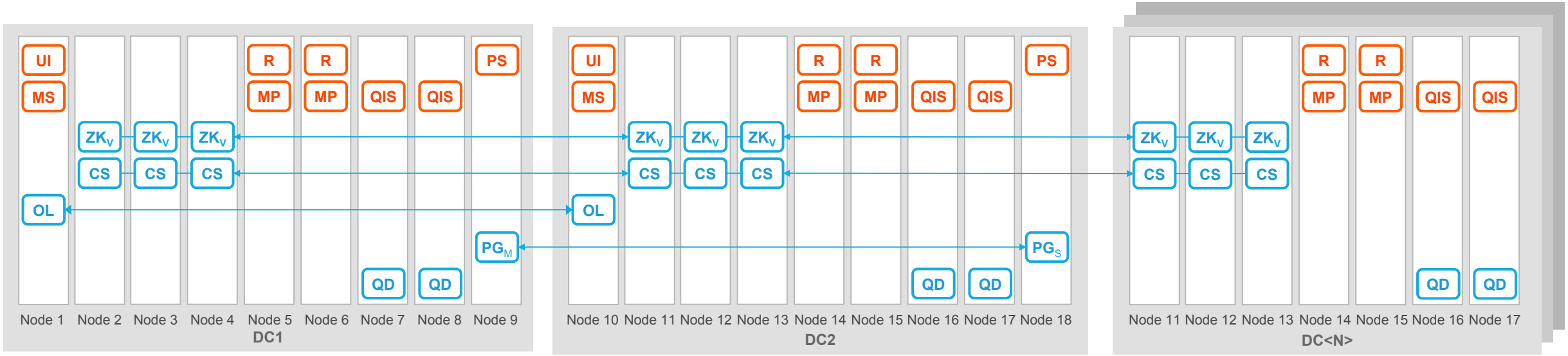


- Horizontally scalable. Components can be installed and configured to run from a single node (all in one setup) all the way to a multi DC centers, active/active, globally distributed setup.
- Ability to stack components.
- Distributed data replicated using eventual consistency.
- Asynchronous analytics data capture and processing.
- Centralized configuration for distributed components.
- Multitenant by design.
- Management via APIs and UI console.

Legend:

R Router	MS Management Server	DP Developer Portal	MY MySQL	OL Openldap	Server/Virtual Machine
MP Message Processor	PS Postgres Server	BA BaaS Server	ZK Zookeeper	PG PostgreSQL	POD
UI Enterprise UI	QIS Qpid/Ingest Server		CS Cassandra	QD Apache Qpid	

Apigee Edge architecture – Horizontal scalability

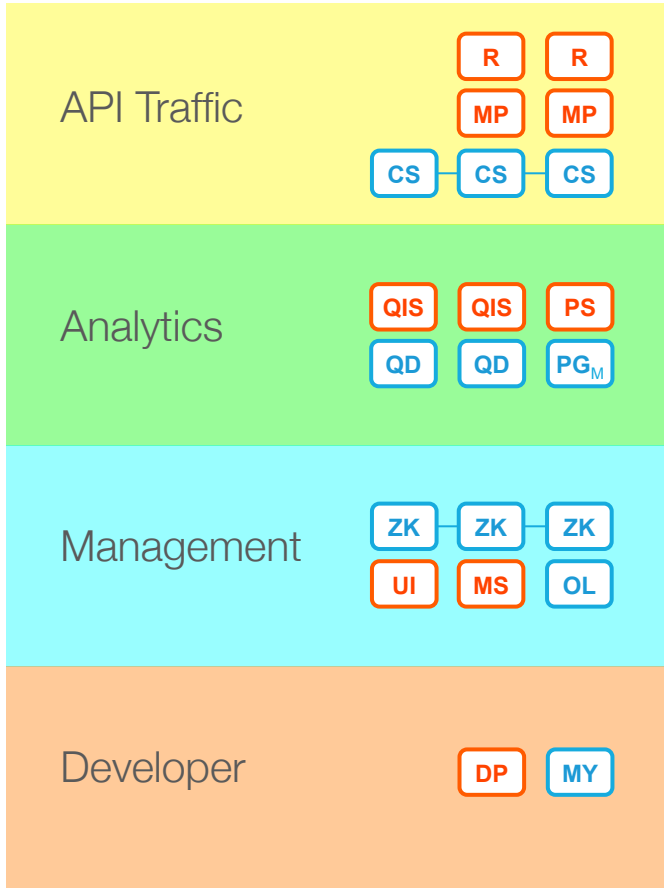


- Planet footprint is driven by customer requirements, transaction volumes, availability and reliability among others drive component stacking and number of nodes.
- R, MP and CS are critical components to handle live API traffic.
- Management and analytics components are not required on every DC.
- For high availability, ZK and QD should be available in all DCs.
- Analytics data replication via Master/Slave or Publish/Subscribe.

Legend:

R Router	MS Management Server	DP Developer Portal	MY MySQL	OL Openldap		Server/Virtual Machine
MP Message Processor	PS Postgres Server	BA BaaS Server	ZK Zookeeper	PG PostgreSQL		POD
UI Enterprise UI	QIS Qpid/Ingest Server		CS Cassandra	QD Apache Qpid		

Apigee Edge architecture – Scaling by capability



- Given the responsibility and capabilities offered by each component, scalability requirements and how they are implemented may vary.
- In most scenarios, scaling to accommodate higher number of TPS or API calls may impact only components serving live API traffic.
- Analytics and management components may grow in number mostly driven by demanding high availability requirements for those capabilities provided by analytics and management components.

Legend:
R Router
MP Message Processor
UI Enterprise UI
MS Management Server
PS Postgres Server
QIS Qpid/Ingest Server
DP Developer Portal
BA BaaS Server
MY MySQL
ZK Zookeeper
CS Cassandra
OL Openldap
PG PostgreSQL
QD Apache Qpid
Server/Virtual Machine
POD

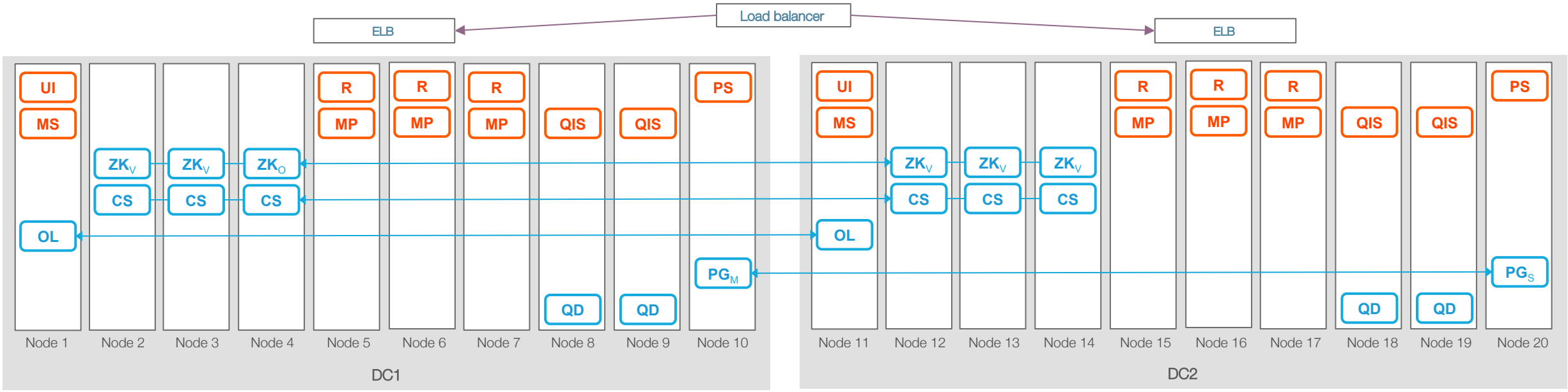
15 mins break

Agenda

09:30 – 10:00	Apigee Overview
10:00 – 11:00	Apigee Private Cloud Architecture
BREAK (15 MINS)	
11:15 – 12:00	**Apigee Private Cloud Architecture (cont)**
BREAK – LUNCH (60 MINS)	
13:00 – 15:45	Apigee Edge Install
BREAK (15 MINS)	
16:00 – 17:30	Platform Operations

apigee

Apigee Edge architecture – Multitenant by design



Legend:

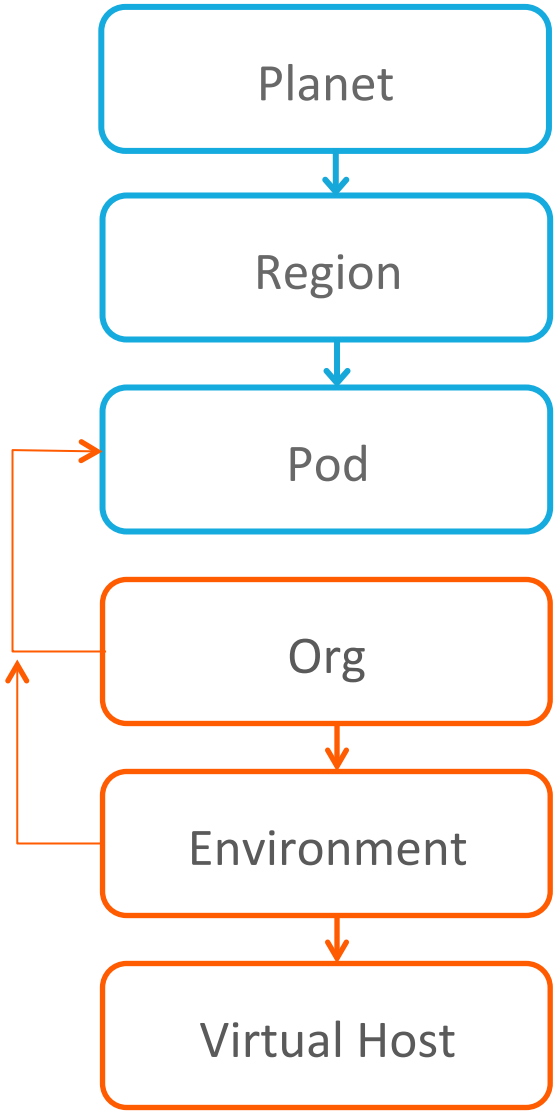
- R Router
- MP Message Processor
- UI Enterprise UI
- MS Management Server
- PS Postgres Server
- QIS Qpid/Ingest Server
- DP Developer Portal
- BA BaaS Server
- MY MySQL
- ZK Zookeeper
- CS Cassandra
- OL Openldap
- PG PostgreSQL
- QD Apache Qpid
- Server/Virtual Machine
- POD

Multitenancy

"Multitenancy is a reference to the mode of operation of software where multiple independent instances of one or multiple applications operate in a shared environment. The instances (tenants) are logically isolated, but physically integrated. The degree of logical isolation must be complete, but the degree of physical integration will vary. The more physical integration, the harder it is to preserve the logical isolation. The tenants (application instances) can be representations of organizations that obtained access to the multitenant application (this is the scenario of an ISV offering services of an application to multiple customer organizations). The tenants may also be multiple applications competing for shared underlying resources (this is the scenario of a private or public cloud where multiple applications are offered in a common cloud environment)." - Gartner

Multitenancy = Software capabilities to support shared infrastructure while providing tenants with data and processing partitioning.

Apigee Edge Architecture – Multitenancy



Planets represent an entire physical environment and 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

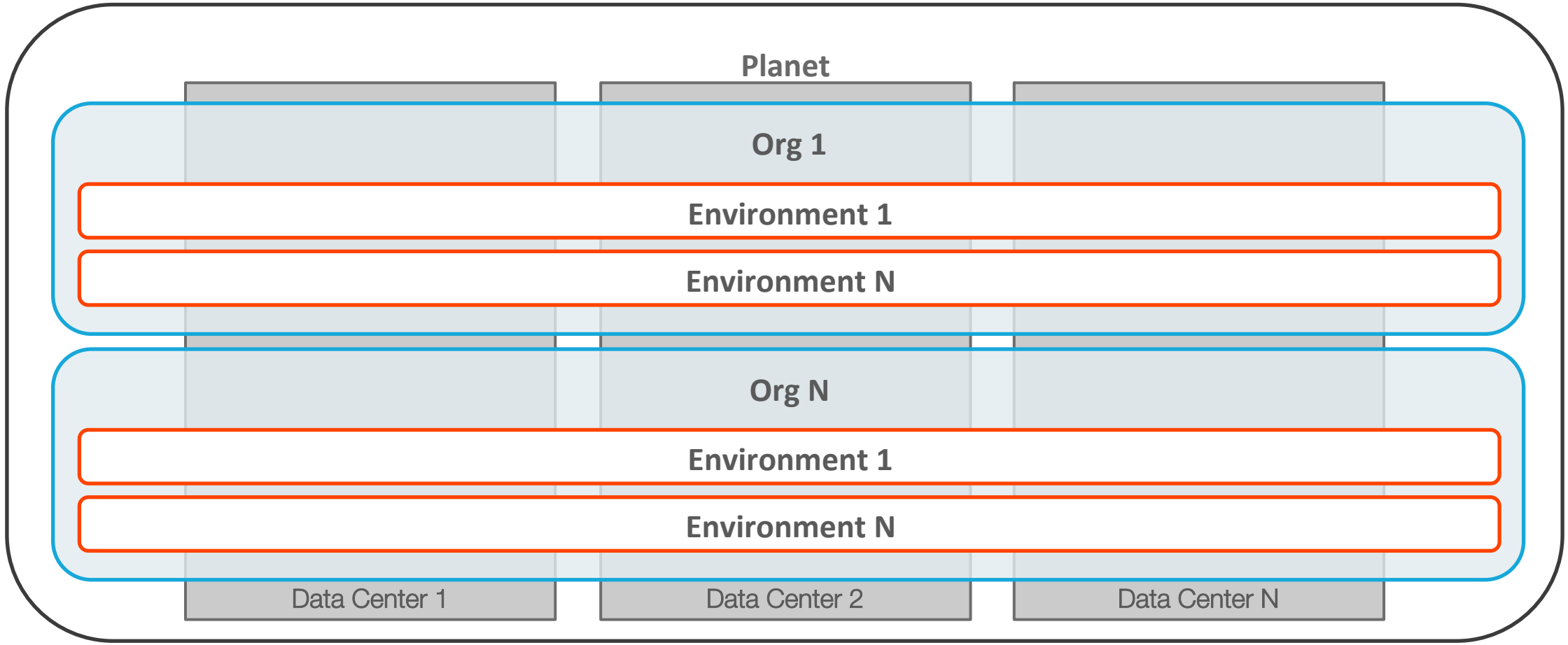
Environments are virtual routes that allow API bundles to be deployed and tested within and Org. Environments can be associated with pods for servers independently of the Org.

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

apigee

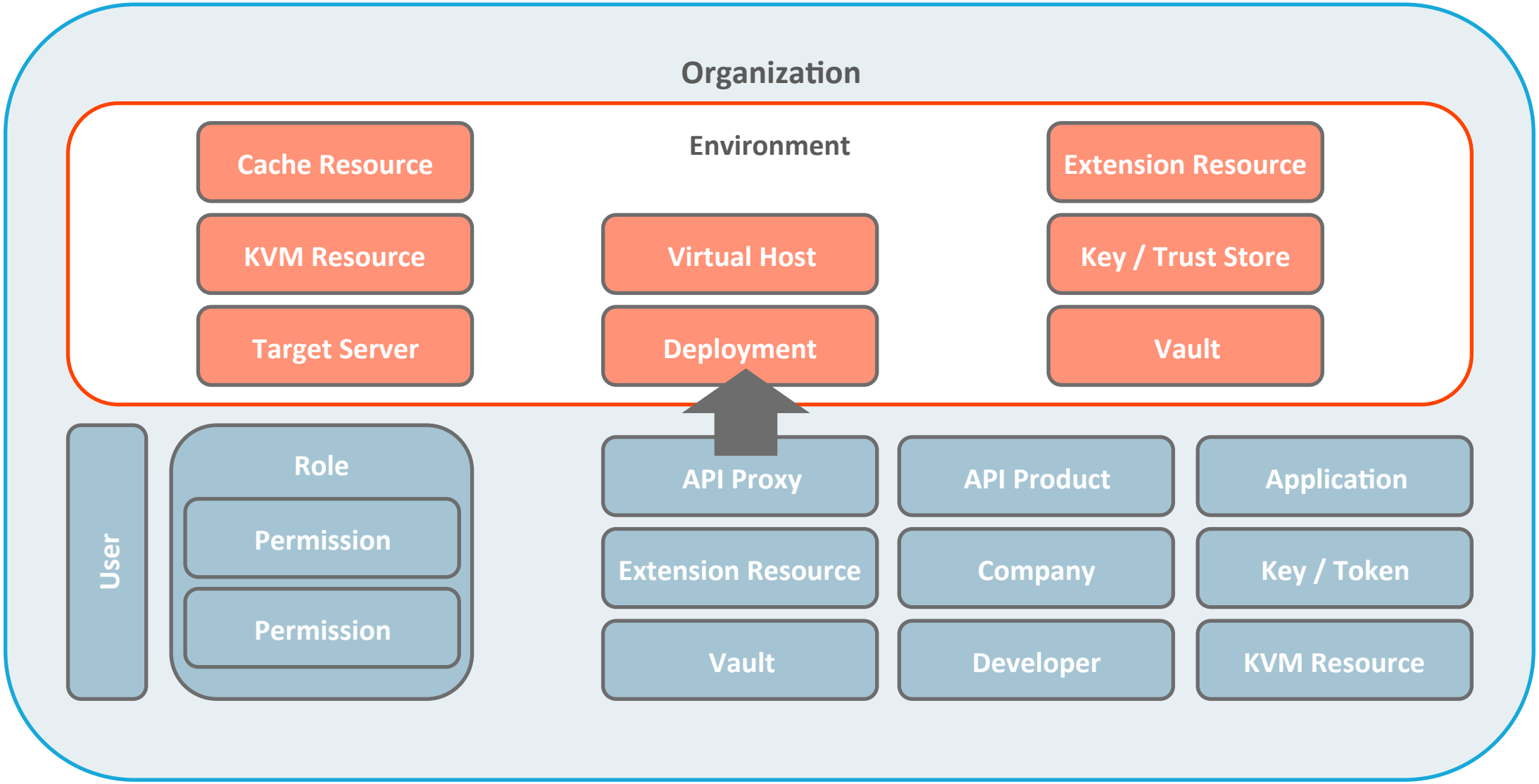
Apigee Edge Architecture – Multitenancy

- A Planet can expand multiple DCs. Organization and Environment can expand across the planet.
- Data partitioning by Organization and Environment. Processing partitioning by Org+Env can be configured.

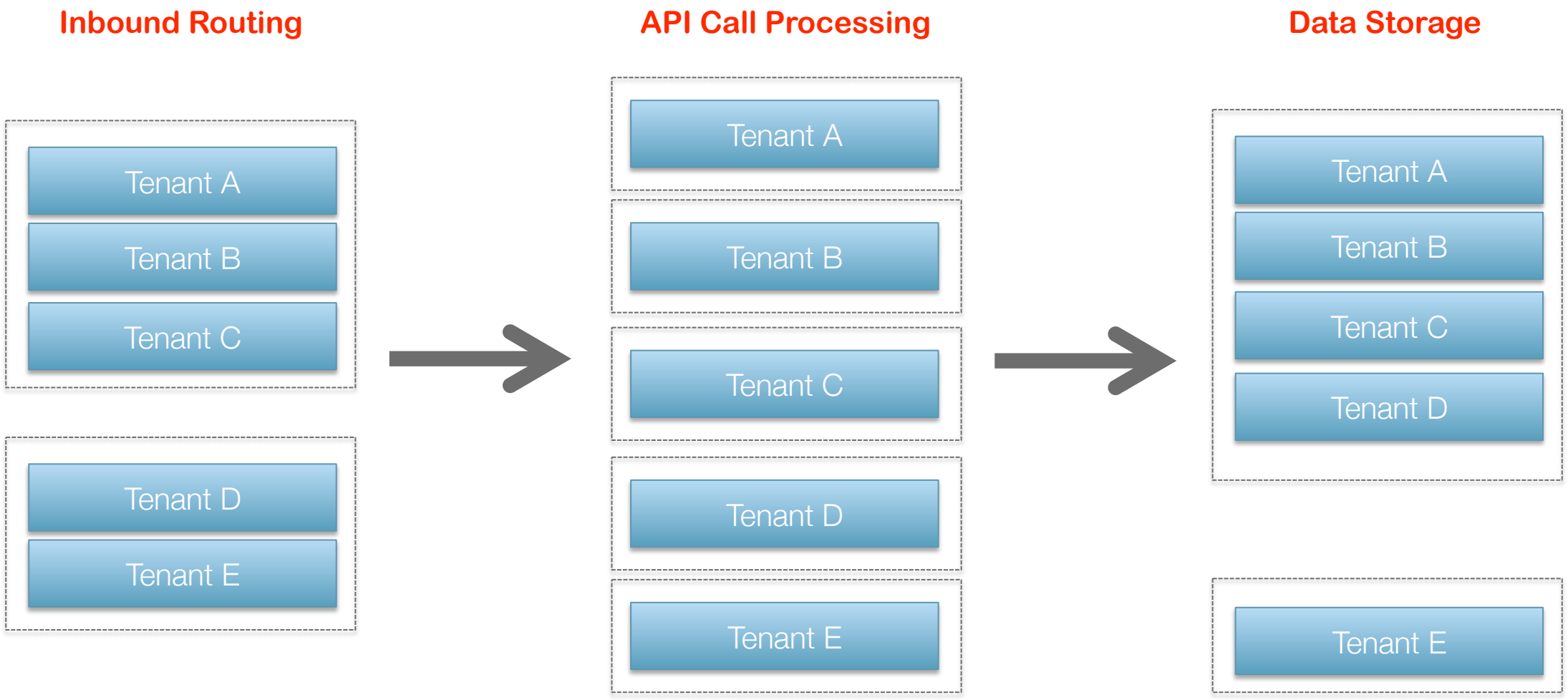


apigee

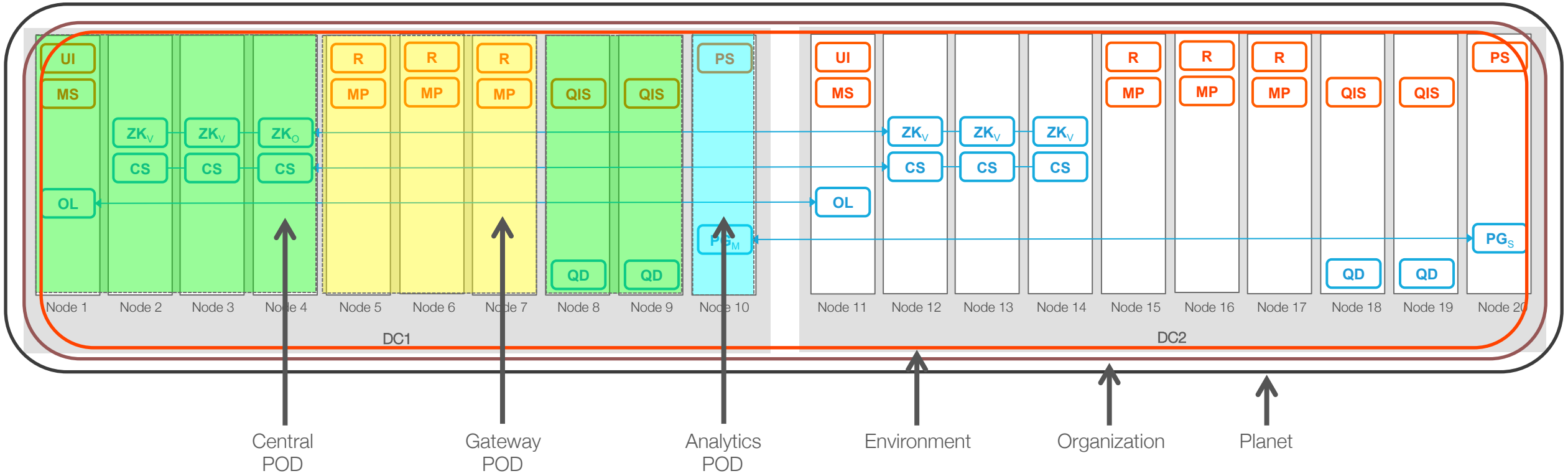
Apigee Edge Architecture – Organizational structure



Apigee Edge Architecture – Physical partitioning



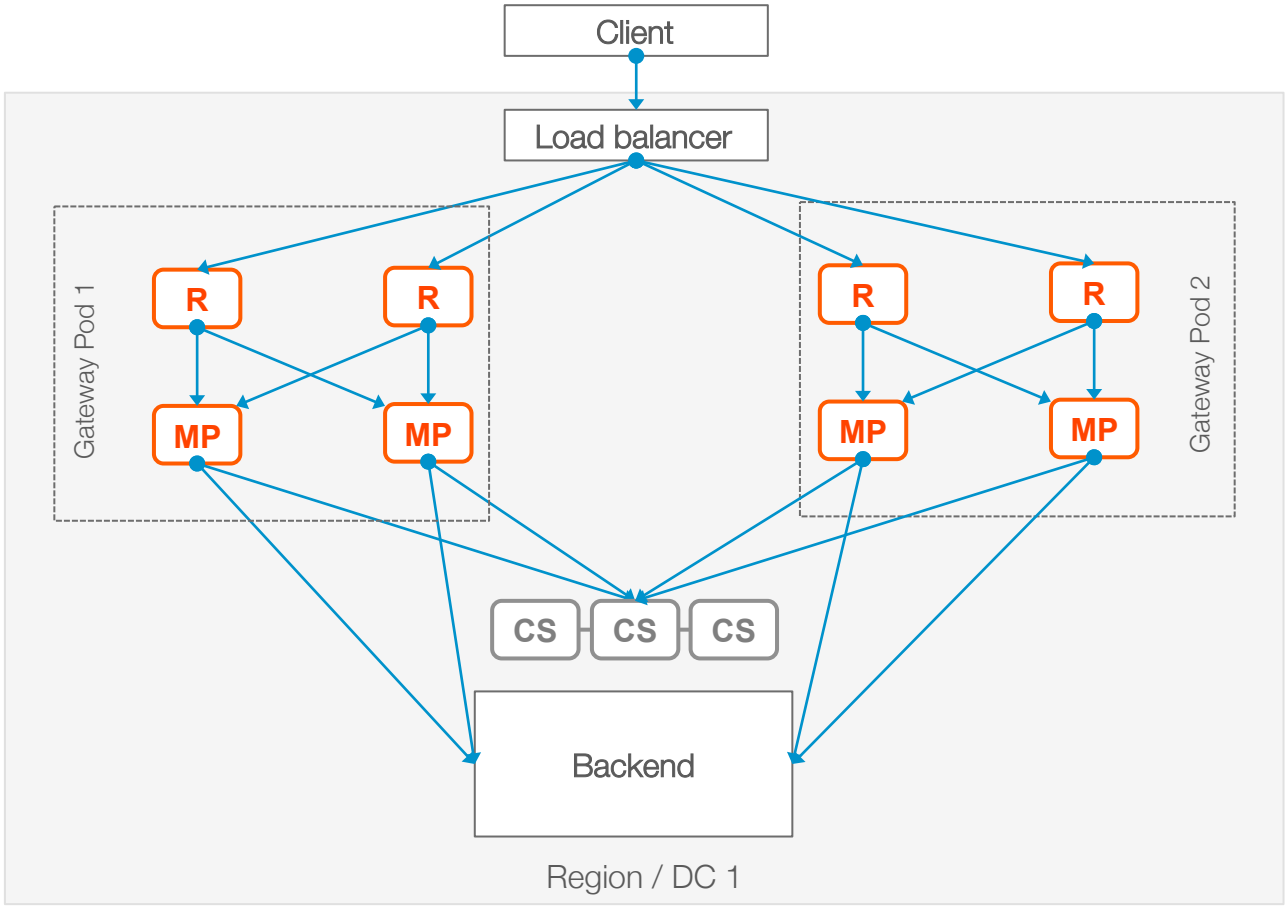
Apigee Edge Architecture – Organizational structure



Legend:

- R Router
- MP Message Processor
- UI Enterprise UI
- MS Management Server
- PS Postgres Server
- QIS Qpid/Ingest Server
- DP Developer Portal
- BA BaaS Server
- MY MySQL
- ZK Zookeeper
- CS Cassandra
- OL Openldap
- PG PostgreSQL
- QD Apache Qpid
- Server/Virtual Machine
- POD

Apigee Edge Architecture – API traffic data flow



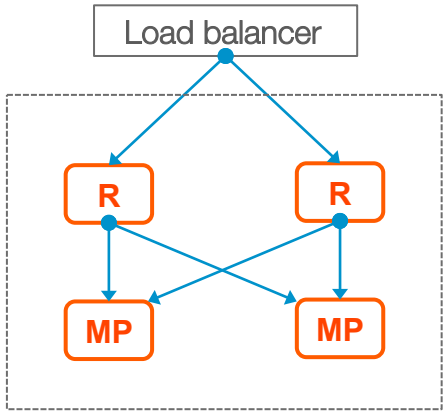
- Routers send requests to Message Processors in their Gateway pod.
- If there are two or more gateway pods in a region, then routers will ignore message processors in the other gateway pods.
- Message Processors respect the region as their scope.
- For two data center the same rules apply as for one datacenter.
- All Apigee components are configured to only use the Cassandra nodes in their region / data center.

Legend:
R Router
MP Message Processor
UI Enterprise UI
MS Management Server
PS Postgres Server
QIS Qpid/Ingest Server
DP Developer Portal
BA BaaS Server
MY MySQL
ZK Zookeeper
CS Cassandra
OL Openldap
PG PostgreSQL
QD Apache Qpid
Server/Virtual Machine
POD
API call flow
Analytics flow

Apigee Edge Architecture – API traffic data flow

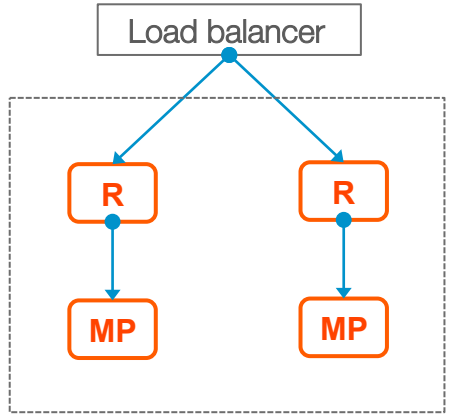
R+MP Configuration options

Tenant Aware Routing



- Rs direct traffic to appropriate MPs, load balancing between them.
- Default behavior.
- Health check heartbeat allows R to automatically take MP out of/into rotation if unresponsiveness is detected.

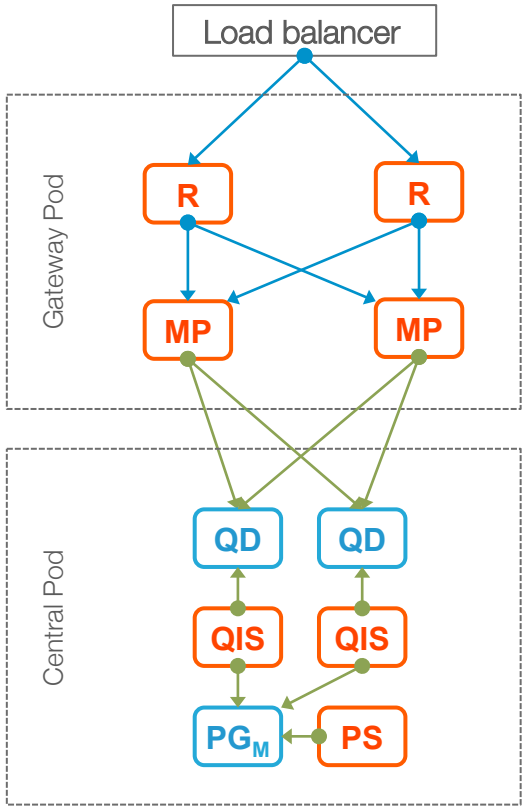
Server Affinity



- R can also be configured to connect to a dedicated MP. Using server affinity, all traffic handled by a R is exclusively sent to its corresponding MP.
- This configuration option offers customer the ability to isolate R/MP for dedicated use cases without impacting API traffic flowing across other R/MP within the same pod.

Legend:
R Router | MP Message Processor | UI Enterprise UI
MS Management Server | PS Postgres Server | QIS Qpid/Ingest Server
DP Developer Portal | BA BaaS Server
MY MySQL | ZK Zookeeper | CS Cassandra
OL Openldap | PG PostgreSQL | QD Apache Qpid
Server/Virtual Machine | POD
API call flow | Analytics flow

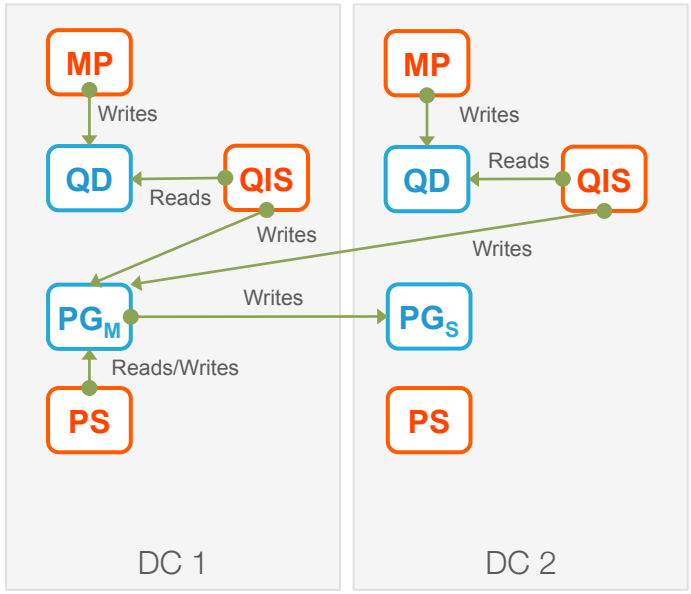
Apigee Edge Architecture – Analytics data flow

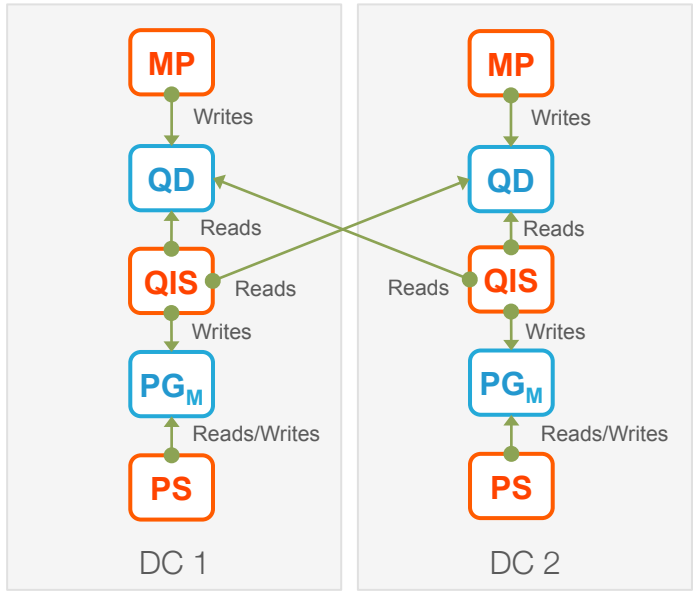


- Ingest services in Qpid Ingest Server will collect Analytics data from all queues and store in PostgreSQL.
- Postgres Server aggregates analytics data asynchronously.
- Message Processors respect the region as their scope and will offload analytics data to all Apache Qpid queues in their region / data center.

Legend:
R Router
MP Message Processor
UI Enterprise UI
MS Management Server
PS Postgres Server
QIS Qpid/Ingest Server
DP Developer Portal
BA BaaS Server
MY MySQL
ZK Zookeeper
CS Cassandra
OL Openldap
PG PostgreSQL
QD Apache Qpid
Server/Virtual Machine
POD
API call flow
Analytics flow

Apigee Edge Architecture – Analytics data flow





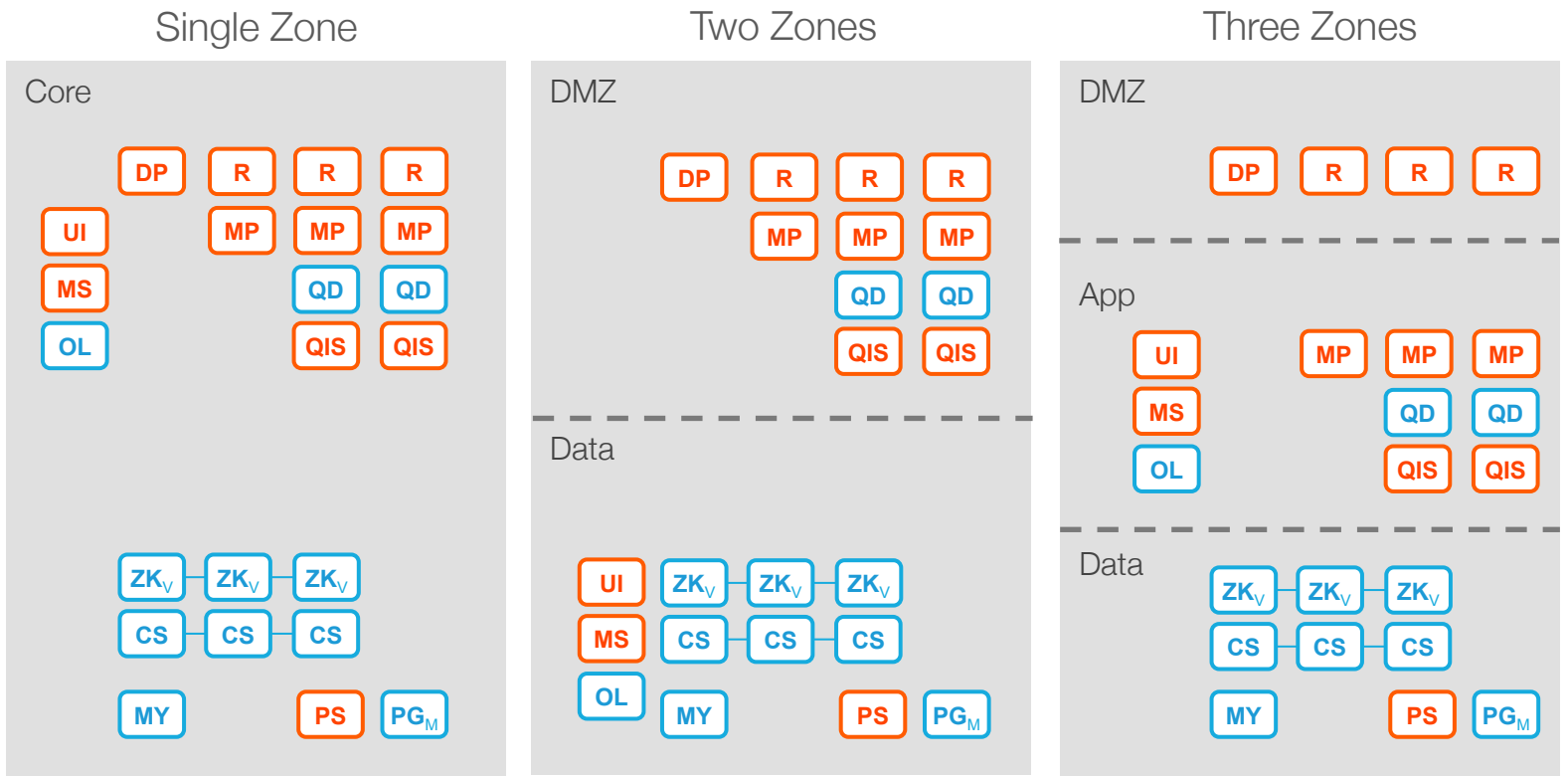
- Analytics data is partitioned per Environment.
- Analytics data size generated by MPs per transaction is about 1kb.
- PostgreSQL contains Raw Data Tables and Aggregated Data Tables. Raw Data Tables grow as analytics data is collected. Appropriate data retention policy and purge processes are required.
- Custom reports allow customers to define user-defined queries which run against raw data.

Legend:

R	Router	MS	Management Server	DP	Developer Portal	MY	MySQL	OL	Openldap
MP	Message Processor	PS	Postgres Server	BA	BaaS Server	ZK	Zookeeper	PG	PostgreSQL
UI	Enterprise UI	QIS	Qpid/Ingest Server			CS	Cassandra	QD	Apache Qpid

Server/Virtual Machine
POD
API call flow
Analytics flow

Apigee Edge Architecture – Network Zoning



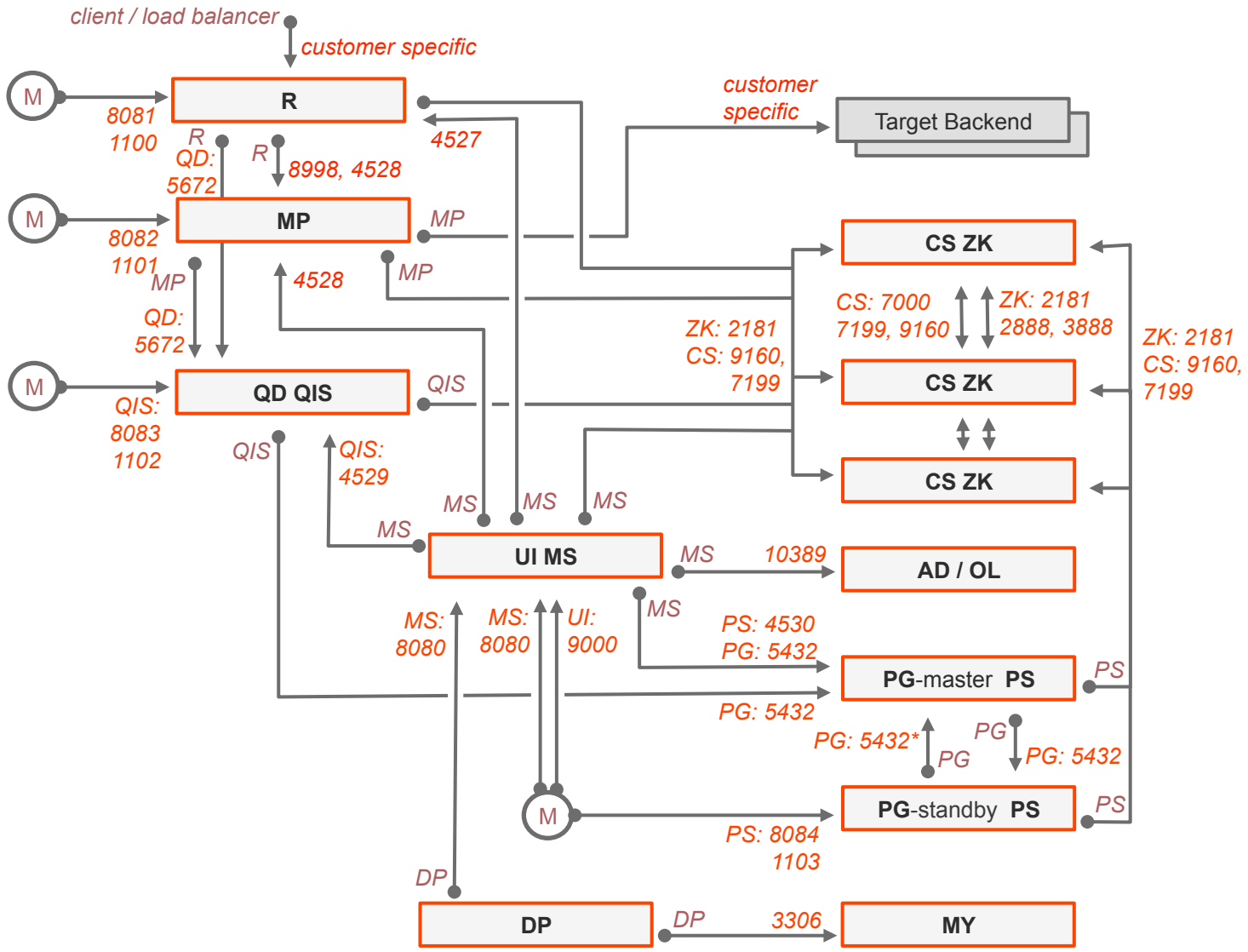
- Edge architecture does not impose network zone requirements. Network zoning will be driven by customer operation and security requirements.
- Firewalls and security appliances between zones should accommodate the connectivity and traffic characteristics of Apigee components without adding latency overhead.
- Keep API traffic as direct as possible. Keep MP dependencies close.
- Pay attention to cross-zone connections.

Legend:

R	Router	MS	Management Server	DP	Developer Portal	MY	MySQL	OL	Openldap		Server/Virtual Machine	→	API call flow
MP	Message Processor	PS	Postgres Server	BA	BaaS Server	ZK	Zookeeper	PG	PostgreSQL		POD	→	Analytics flow
UI	Enterprise UI	QIS	Qpid/Ingest Server			CS	Cassandra	QD	Apache Qpid				

Apigee Edge Architecture – Components Connectivity

Single DC View



Legend:

API traffic	R	Router
	MP	Message Processor
	CS	Cassandra
Configuration	MS	Management Server
	UI	Management UI
	OL	OpenLDAP
	ZK	Zookeeper
Analytics	QD	Apache Qpid
	QS	Qpid Server
	PG	PostgreSQL
	PS	Postgres Server
Dev	DP	Developer Portal
	MY	MySQL

from → *to* 1-way

svc:port ↔ 2-way

M → management calls HTTP or JMX)

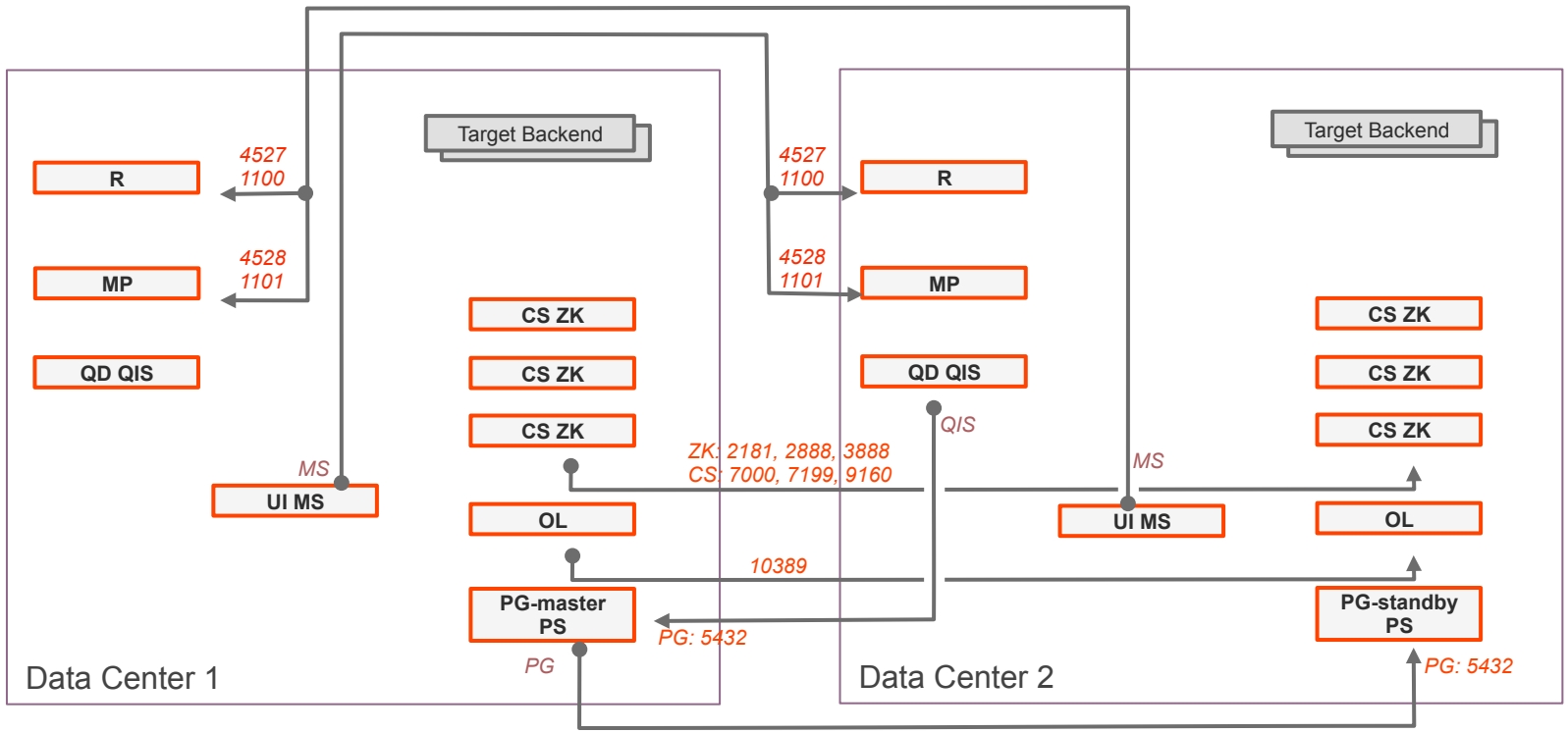
Notes:

- For CS all nodes talk to all other CS nodes
- For ZK all nodes talk to all other ZK nodes
- * both directions to enable PostgreSQL failover

apigee

Apigee Edge Architecture – Components Connectivity

Multi DC View



Legend:

API traffic	R	Router
	MP	Message Processor
	CS	Cassandra
Configuration	MS	Management Server
	UI	Management UI
	OL	OpenLDAP
	ZK	Zookeeper
Analytics	QD	Apache Qpid
	QS	Qpid Server
	PG	PostgreSQL
	PS	Postgres Server
Dev	DP	Developer Portal
	MY	MySQL

from *to* — 1-way

svc:port — 2-way

M — management calls HTTP or JMX)

Notes:

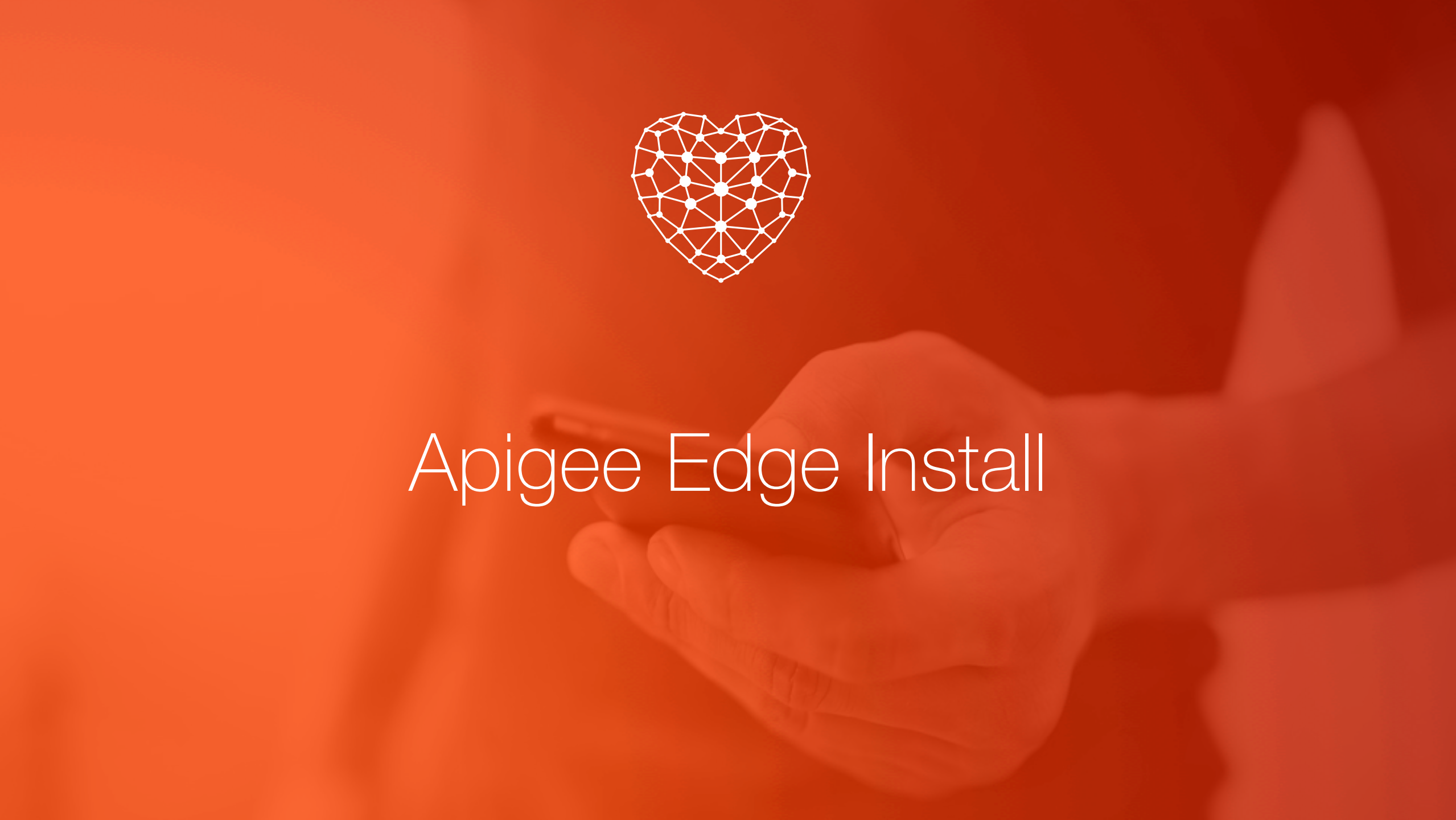
- For CS all nodes talk to all other CS nodes
- For ZK all nodes talk to all other ZK nodes
- * both directions to enable PostgreSQL failover

60 mins break

Agenda

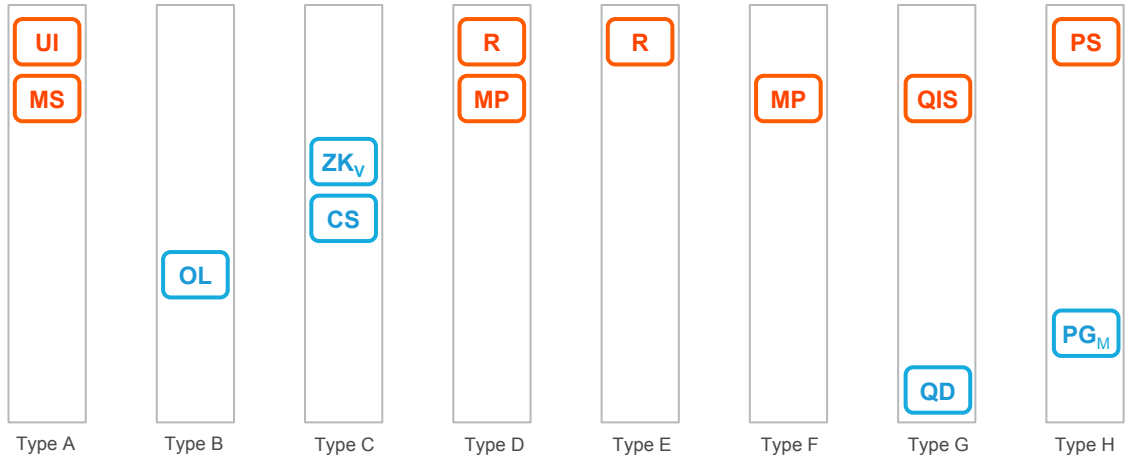
09:30 – 10:00	Apigee Overview
10:00 – 11:00	Apigee Private Cloud Architecture
BREAK (15 MINS)	
11:15 – 12:00	Apigee Private Cloud Architecture (cont)
BREAK – LUNCH (60 MINS)	
13:00 – 15:45	**Apigee Edge Install**
BREAK (15 MINS)	
16:00 – 17:30	Platform Operations

apigee



Apigee Edge Install

System Requirements – Hardware Specification



- Cassandra
  - Local storage with SSD or fast HDD supporting 2000 IOPS
- PostgreSQL
  - Local storage with SSD or fast HDD supporting 1000 – 8000 IOPS
  - CPU, Memory and Size the storage according to the data retention requirements. Implement archive / purge process and keep less data for best performance.
  - Operations Guide provides formula to estimate storage needs.

Type	Components	CPU	Memory	Disk
A	UI, MS	2 core	4 GB	60 GB
B	OL	2 core	4 GB	60 GB
C	ZK, CS	8 core	16 GB	250 GB local storage with SSD or fast HDD supporting 2000 IOPS
D	R, MP	4/8 core	8/16 GB	100 GB
E	R	4 core	8 GB	60 GB
F	MP	8 core	16 GB	100 GB
G	QIS, QD	8 core	16 GB	500 GB local storage with SSD or fast HDD supporting 1000 IOPS
H	PS, PG	8 core	16 GB	500 GB to1 TB local storage with SSD or fast HDD supporting 4000-8000 IOPS

Legend: R Router; MP Message Processor; UI Enterprise UI; MS Management Server; PS Postgres Server; QIS Qpid/Ingest Server; DP Developer Portal; BA BaaS Server; MY MySQL; ZK Zookeeper; CS Cassandra; OL Openldap; PG PostgreSQL; QD Apache Qpid; Server/Virtual Machine; POD

System Requirements – Software Dependencies

Operating System
- Red Hat Enterprise Linux (64-bit):
  - 6.3, 6.4, 6.5, 6.6, 7.0
- CentOS (64-bit):
  - 6.3, 6.4, 6.5, 6.6, 7.0
- Oracle Linux (64-bit):
  - 6.5

JDK
- Oracle JDK 1.7
- OpenJDK 7

SSL/TLS
- 1.0
- 1.2

Other software
- Cassandra 2.0.15
- Zookeeper 3.4.5
- QPID 0.14
- PostgreSQL 9.3
- Play (UI) 2.3.4
- OpenLDAP 2.4

https://apigee.com/docs/api-services/reference/supported-software

apigee

Private Cloud Installer – Software Dependencies

- awk
- basename
- bash
- chkconfig
- curl
- date
- dirname
- echo
- expr
- grep
- hostname
- id
- ls
- perl
- pgrep (from procps)
- ps
- pwd
- python
- rpm
- rpm2cpio
- sed
- sudo
- tar
- tr
- uname
- unzip
- useradd
- wc
- yum

Complete list of prerequisites can be found in Apigee Edge Install and Configuration Guide, Page 19.

In addition to the tools above, some nodes require the installation of additional software components such as:

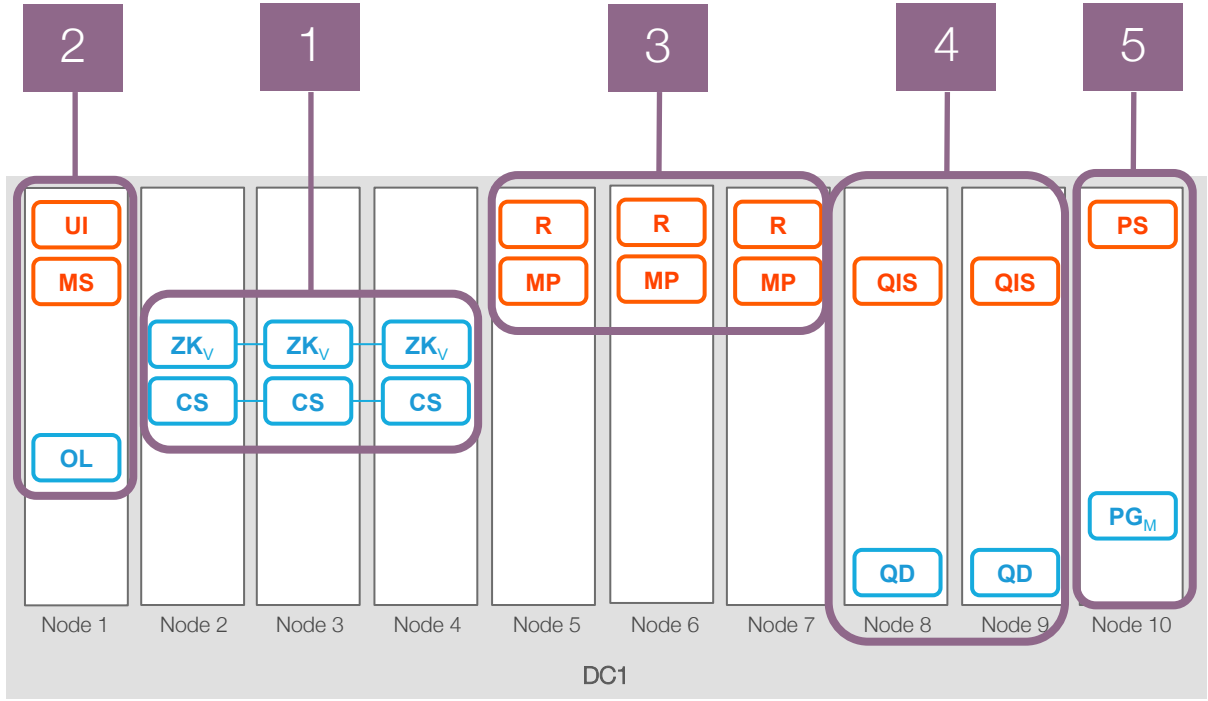
- ntp (all nodes)
- rsync (PostgreSQL nodes)
- openldap-clients openldap-servers (Management Server node)

Apigee Edge Installation Process

1. Install data store hosts
2. Install management hosts
   - Install LDAP first if using standalone LDAP hosts
3. Install router, message processor
4. Qpid hosts
5. PostgreSQL hosts
6. Configure PostgreSQL replication, if needed
7. Create organization(s) and environment(s)

If applicable, install:

7. Developer portal
8. Install monetization
9. App services (BaaS)



Legend:

R	Router	MS	Management Server	DP	Developer Portal	MY	MySQL	OL	Openldap		Server/Virtual Machine
MP	Message Processor	PS	Postgres Server	BA	BaaS Server	ZK	Zookeeper	PG	PostgreSQL		POD
UI	Enterprise UI	QIS	Qpid/Ingest Server			CS	Cassandra	QD	Apache Qpid		

Private Cloud – Software Install

Apigee Private Cloud (apigee-edge-4.15.07.00.zip) has everything required to install and configure Apigee Edge and BaaS components except for the Developer Portal. The Developer Portal is distributed using a different package (DeveloperServices-4.15.07.00.tar).

Apigee Edge installation steps:

1. Acquire a license key and copy it to all nodes.
2. Download the Apigee Private Cloud from ftp.apigee.com and copy it to all nodes.
3. Installation downloads and installs required system software via Yum. It requires Internet connection or local repository.
4. Unzip apigee-edge-4.<YY>.<MM>.<V>.zip
5. Run the primary installation script:

```
/<unzip-location>/apigee-edge-4.<YY>.<MM>.<V>/apigee-install.sh -j /usr/java/default
-r <inst-root> -d <data-root>
```

Private Cloud – Software Setup

1. Once Apigee binaries are installed, run:

```
/<inst-root>/apigee4/share/installer/apigee-setup.sh
```

2. Setup script requires a profile type and information about the system, provided interactively or via response file (silent install).
3. Create an organization, environment(s) and organization administrator, run:

```
/<inst-root>/apigee4/bin/setup-org.sh
```

Private Cloud – Silent Install

- Silent install provides a way for specifying, in advance, all necessary values used by apigee-setup.sh
- Silent install provides a way for specifying, in advance, all necessary values used by apigee-setup.sh

```
/<inst-root>/apigee4/share/installer/
apigee-setup.sh -p ds -f <response-
file-name>
```

- The response files contain a number of variables definitions to be used by apigee-setup.sh. A response file per DC/Region will be needed since some values are unique per DC/Region.

Single machine setups:
- aio = All In One (Gateway and Analytics Standalone)

Cluster node setup for ZooKeeper and Cassandra (min 3 nodes):
- ds = Datastore Cluster Node

LDAP setup for OpenLDAP:
- ld = LDAP Node

Separate components setup:
- ld = LDAP Node
- ms = Gateway Management Server
- r = Gateway Router
- mp = Gateway Message Processor
- rmp = Gateway Router and Message Processor
- qs = Analytics Qpid Server
- ps = Analytics Postgres Server
- mo = Monetization Server

Note: The list above doesn't show all the available options. Refers to Apigee Edge Install and Configuration Guide, section Basic Host Installation, page 35 for details.

Private Cloud – Silent Response File

- IP1=
- IP2=
- IP3=
- IP4=
- IP5=

- HOSTIP=$(hostname -i)
- MSIP=$IP1
- ADMIN_EMAIL=
- APIGEE_ADMINPW=
- LICENSE_FILE=/root/opdk/license.txt
- USE_LDAP_REMOTE_HOST=n
- LDAP_TYPE=1
- APIGEE_LDAPPW=
- ENABLE_AX=y
- MP_POD=gateway
- REGION=dc-1

- USE_ZK_CLUSTER=y
- ZK_HOSTS="$IP1 $IP2 $IP3"
- ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
- USE_CASS_CLUSTER=y
- CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1"
- CASS_USERNAME=
- CASS_PASSWORD=
- SKIP_SMTP=y
- SMTPHOST=
- SMTPPORT=25
- SMTPUSER=
- SMTPPASSWORD=
- SMTPSSL=n
- BIND_ON_ALL_INTERFACES=y

Apigee Edge
Install Lab

Apigee Edge Install Lab

Scope

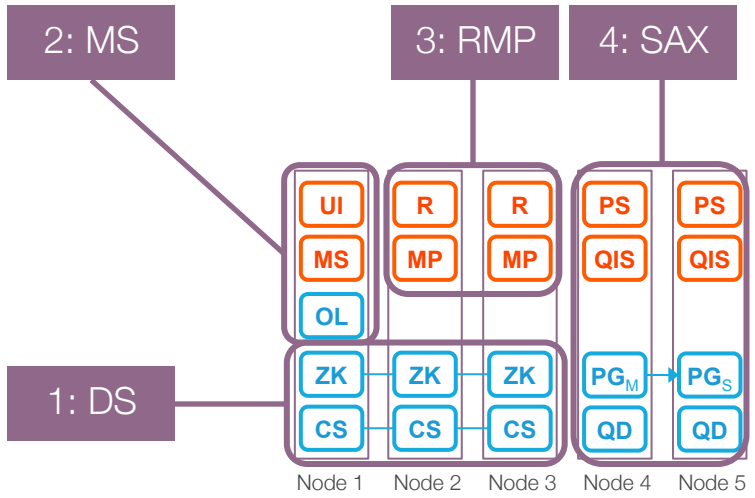
- 5 nodes Apigee Edge Install
- VMs instances running in Amazon AWS
- Installation and setup using silent install process.
- Setup order DS, MS, RMP, SAX

AWS Instances IP

- http://iloveapis.com/workshops/ops/

Documentation

- ApigeeEdgePrivateCloud-Install-Config-Guide.pdf
  - Page 35 - Basic Host Installation
  - Page 47 – Onboarding
  - Page 120 - Appendix A: Silent Installation
  - Page 121 - 5-host Clustered Installation



Apigee Edge Install Lab - Prerequisites

Preparation

A. Software, sample install file and JDK are located in /root/opdk/

B. Install prerequisites
   - In all nodes:
     - Install JDK (rmp -ivh <rmp>)
     - `yum -y install ntp`
   - In node 1:
     - `yum -y install openldap-clients openldap-servers`
   - In node 4 and 5:
     - `yum -y install rsync`

C. In all nodes, `unzip apigee-edge-4.15.07.00.zip`

D. Update silent install file with private IP addresses corresponding to your AWS instances. Copy silent install files to all nodes.

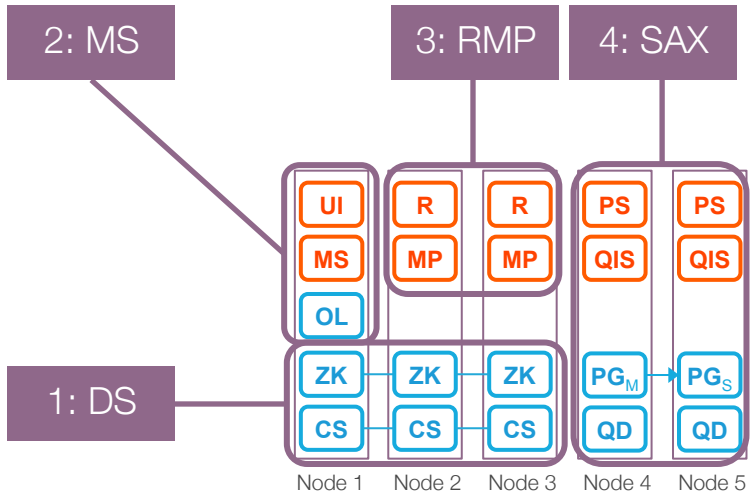
Additional considerations:
- For the purpose of the lab, we took care of some prerequisites. When planning for real world installation, prerequisites described in Apigee Edge Install and Configuration Guide, Page 19-35 should be completed.

Apigee Edge Install Lab – Installation and Setup

Install Steps

A. Unzip apigee-edge-4.15.07.00.zip

B. `./apigee-install.sh -j /usr/java/default -r /opt -d /opt`



Setup Steps

1. `/opt/apigee4/share/installer/apigee-setup.sh -p ds -f /root/opdk/response.txt`
2. `/opt/apigee4/share/installer/apigee-setup.sh -p ms -f /root/opdk/response.txt`
3. `/opt/apigee4/share/installer/apigee-setup.sh -p rmp -f /root/opdk/response.txt`
4. `/opt/apigee4/share/installer/apigee-setup.sh -p sax -f /root/opdk/response.txt`

Apigee Edge Install Lab – Installation and Setup

Setup Master-Standby Replication for PostgreSQL

- Apigee Edge Install and Configuration Guide, Page 59

Onboarding

- Apigee Edge Install and Configuration Guide, Page 47
- /opt/apigee4/bin/setup-org.sh

Access Apigee Edge

- http://<management-server-public-ip>:9000

Platform Operations

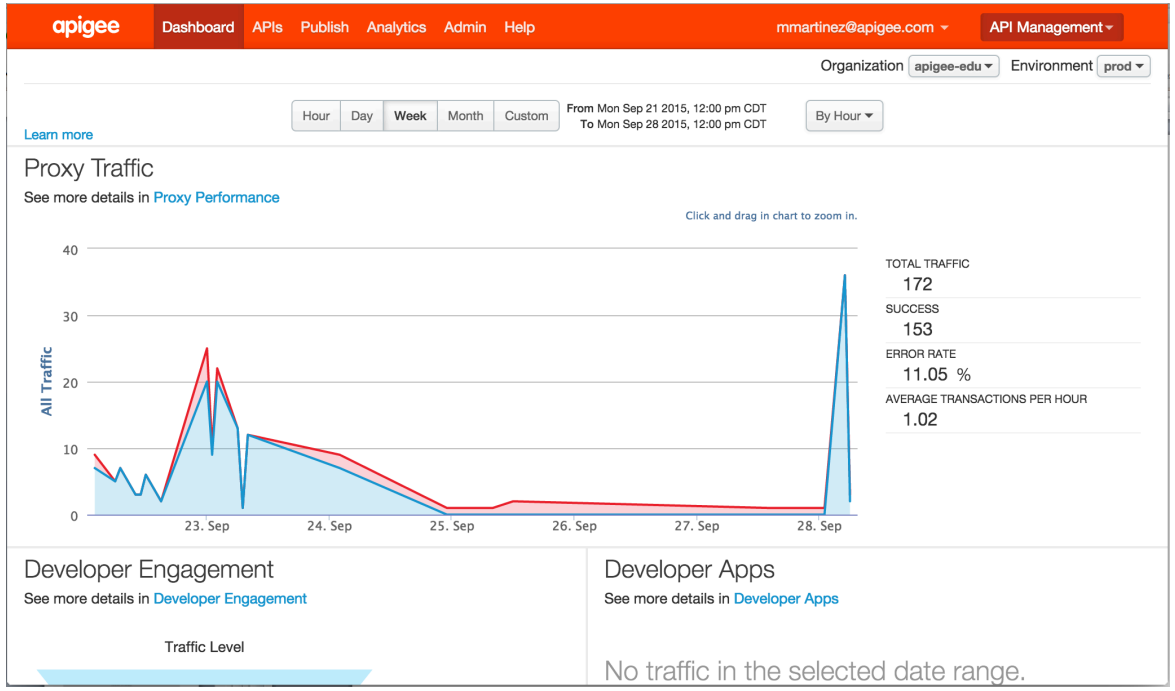
Agenda

09:30 – 10:00	Apigee Overview
10:00 – 11:00	Apigee Private Cloud Architecture
BREAK (15 MINS)	
11:15 – 12:00	Apigee Private Cloud Architecture (cont)
BREAK – LUNCH (60 MINS)	
13:00 – 15:45	Apigee Edge Install Lab
BREAK (15 MINS)	
16:00 – 17:30	**Platform Operations**

apigee

Platform Operations - Walkthrough

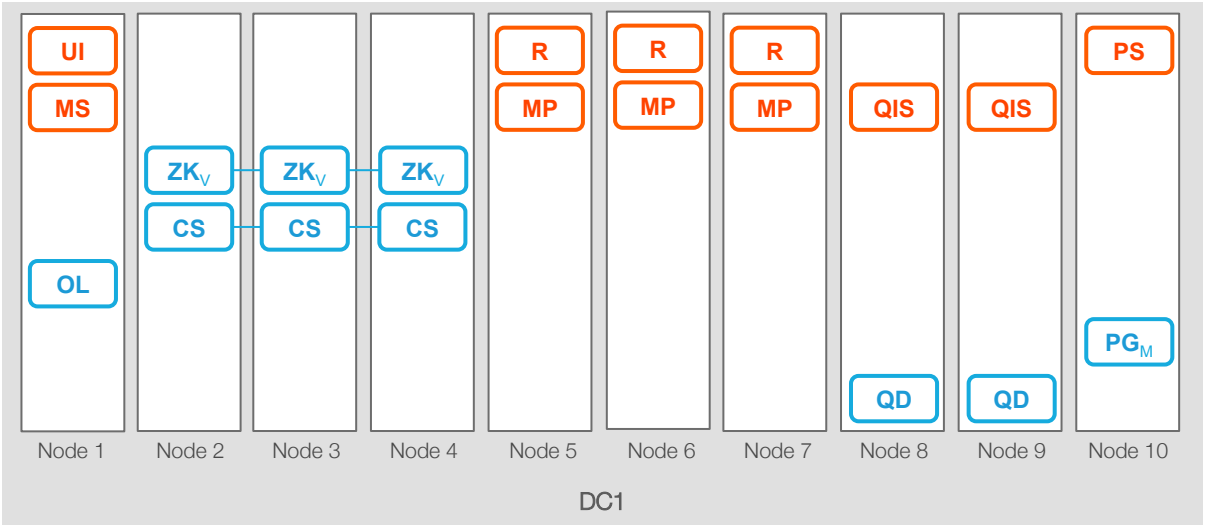
- Directory Structure
  - `/<inst-root>/apigee4`
- Start / Stop / Status
  - `/<inst-root>/apigee4/bin/all-start.sh`
  - `/<inst-root>/apigee4/bin/all-status.sh`
  - `/<inst-root>/apigee4/bin/all-stop.sh`
- Logs file
  - `/<inst-root>/apigee4/var/log`
- Management UI
  - `http://<management-server-host>:9000`
- Management API
  - `http://apigee.com/docs/management/apis`
  - `http://<management-server-host>:8080`



Platform Operations – Backup and Restore

- Backup/restore
  - Edge includes backup scripts for:
    - LDAP
    - Cassandra
    - ZooKeeper
    - PostgreSQL
    - UI (custom reports)
    - Component UUIDs
  - <inst-root>/apigee4/bin/backup.sh

- High Availability and Disaster Recovery
  - Multiple copies of the data
  - Eventual consistency
  - Horizontal scalability and resiliency



Platform Operations – Recurring Maintenance

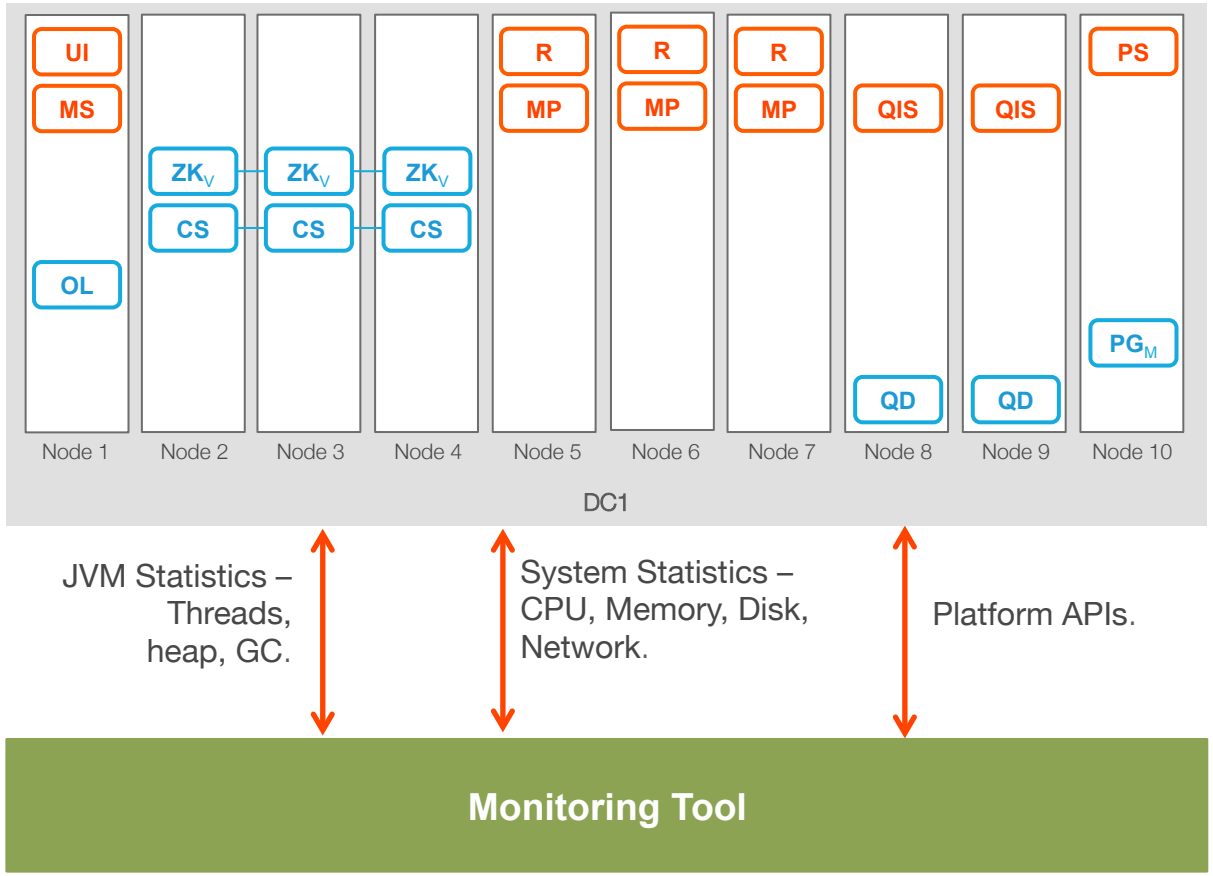
- Cassandra nodes repair

```
/opt/apigee4/share/apache-cassandra/bin/nodetool repair -pr
```

- Periodically prune detailed analytics data based on retention requirements

```
/opt/apigee4/bin/pg-data-purge.sh
```

Platform Operations – Component Level Monitoring

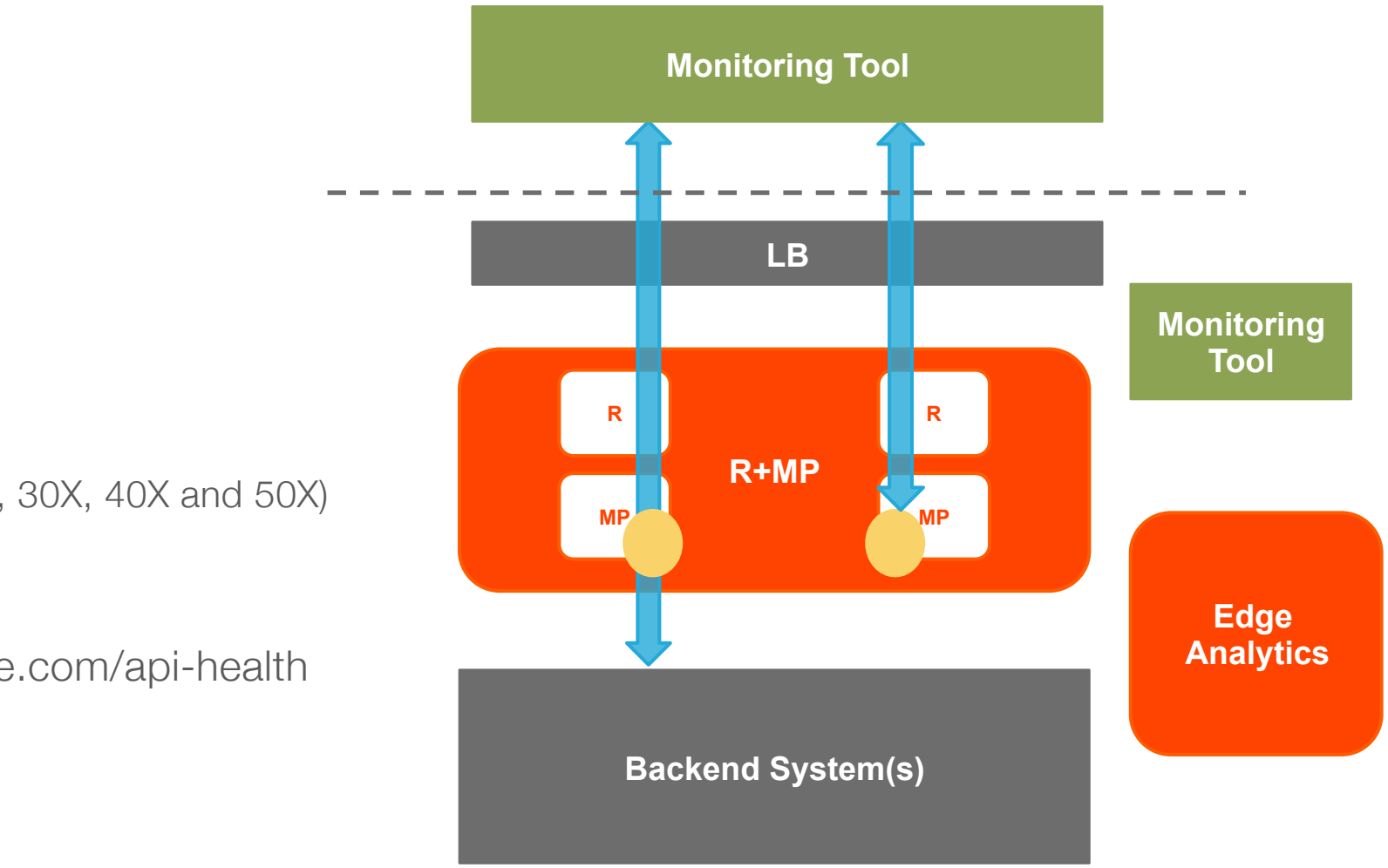


- System-Level Checks
  - CPU, Memory, Disk, Load, Network
- Process/Application Checks
  - Thread statistics, Memory utilization
- Components monitoring
  - JMX
  - Metrics
  - Management API (http://<host>:<port>/v1/servers/self/up)
  - Router health check of the Message Processer
  - Log monitoring
    - HeartBeat
    - Mark down/mark up events
    - Logging Policy

Legend: R Router; MP Message Processor; UI Enterprise UI; MS Management Server; PS Postgres Server; QIS Qpid/Ingest Server; DP Developer Portal; BA BaaS Server; MY MySQL; ZK Zookeeper; CS Cassandra; OL Openldap; PG PostgreSQL; QD Apache Qpid; Server/Virtual Machine; POD

Platform Operations – APIs Monitoring

- API-Level Checks
  - API calls 20X responses
  - API call error within Apigee
  - API call error on backend
  - JMX
  - Metrics
- Message Flow Checks
  - Number of active clients
  - Number of responses (10X, 20X, 30X, 40X and 50X)
  - Connect failures
- API Health - https://pages.apigee.com/api-health



Legend:

R	Router	MS	Management Server	DP	Developer Portal	MY	MySQL	OL	Openldap	Server/Virtual Machine
MP	Message Processor	PS	Postgres Server	BA	BaaS Server	ZK	Zookeeper	PG	PostgreSQL	POD
UI	Enterprise UI	QIS	Qpid/Ingest Server			CS	Cassandra	QD	Apache Qpid	

Takeaway

- Public Cloud = Private Cloud
- Architecture purpose built and proven
- Horizontally scalable, components can be scaled independently of each other
- Multi-tenant by design

Thank you!

apigee