Edge Architecture

Technology stack

SID Global Solutions
Digital Transformation Experts

Google Partner

Agenda

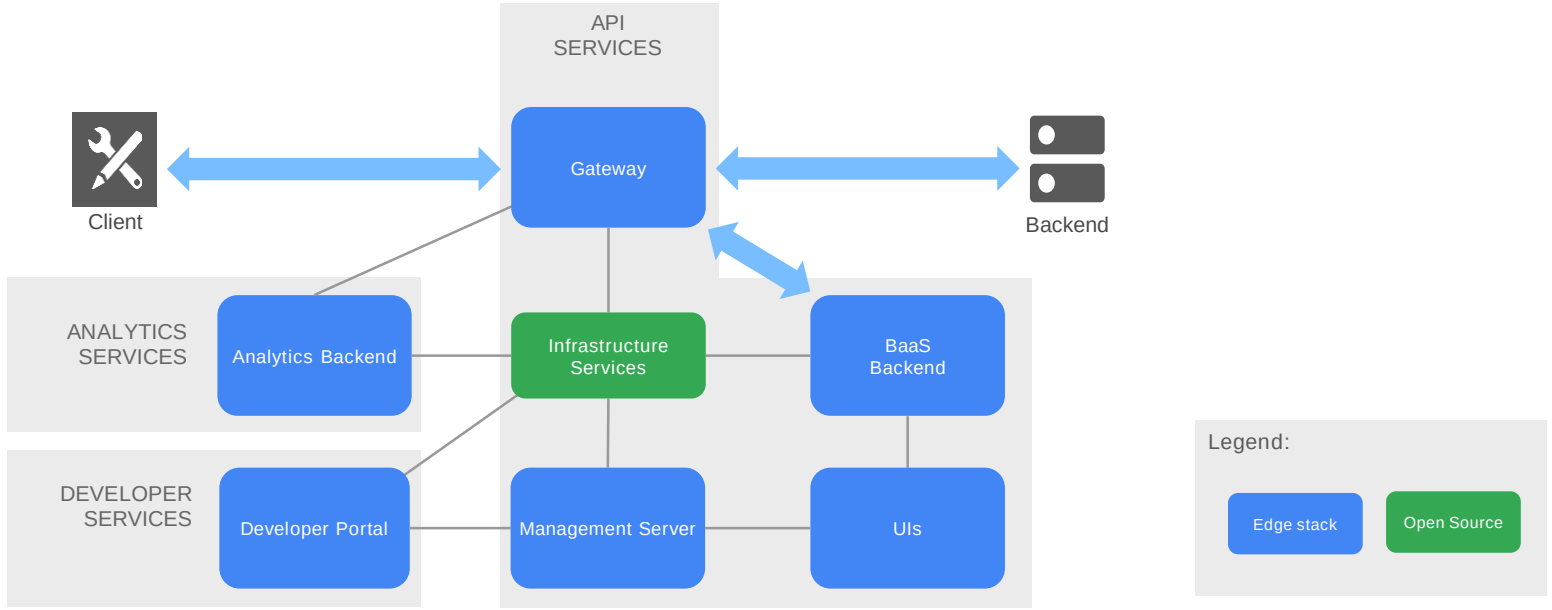
- Capabilities
- Components
- Technology stack
- Terminology and organizational structure

Edge Capabilities

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

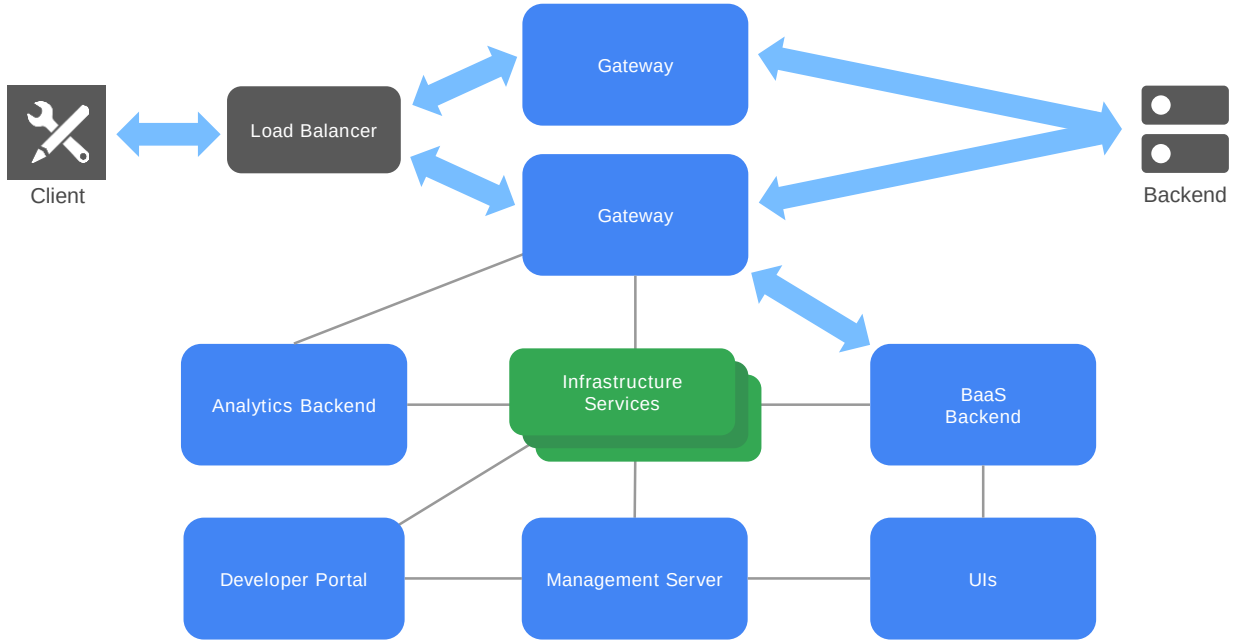
- Gateway: Routing and API calls processing.
- UIs: Enterprise UI, Developer Portal.
- Infrastructure Services: Persistence of runtime, analytics and management.
- Management Server: Provides REST API for all configuration and management tasks.

Note: Monetization, not showed in the diagram below, is part of Developer Services and leverages Gateway, Analytics Services and Management Server.



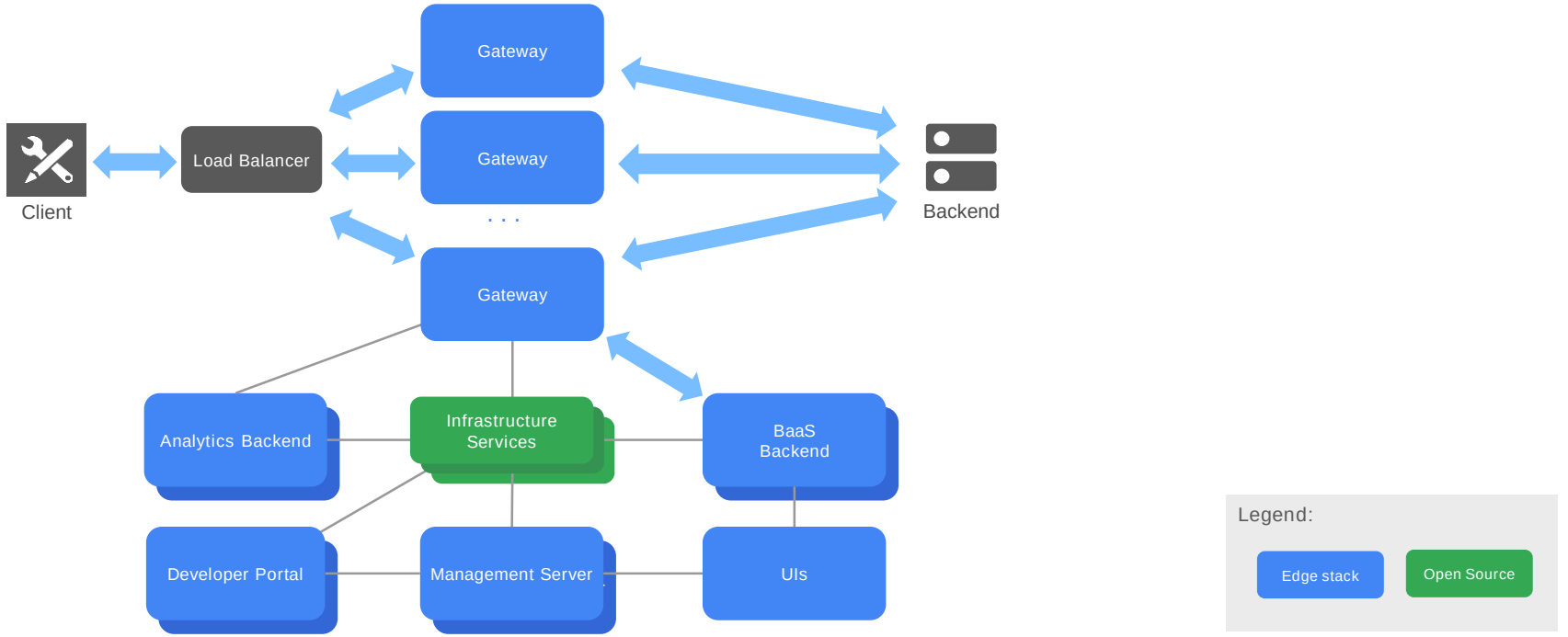
Edge Scalability

- Horizontally scalable.
- Additional Gateway components can be added to keep up with API volume, high availability and resiliency requirements
- As the number of Gateway increases, some of the supporting infrastructure services may need to scale out.



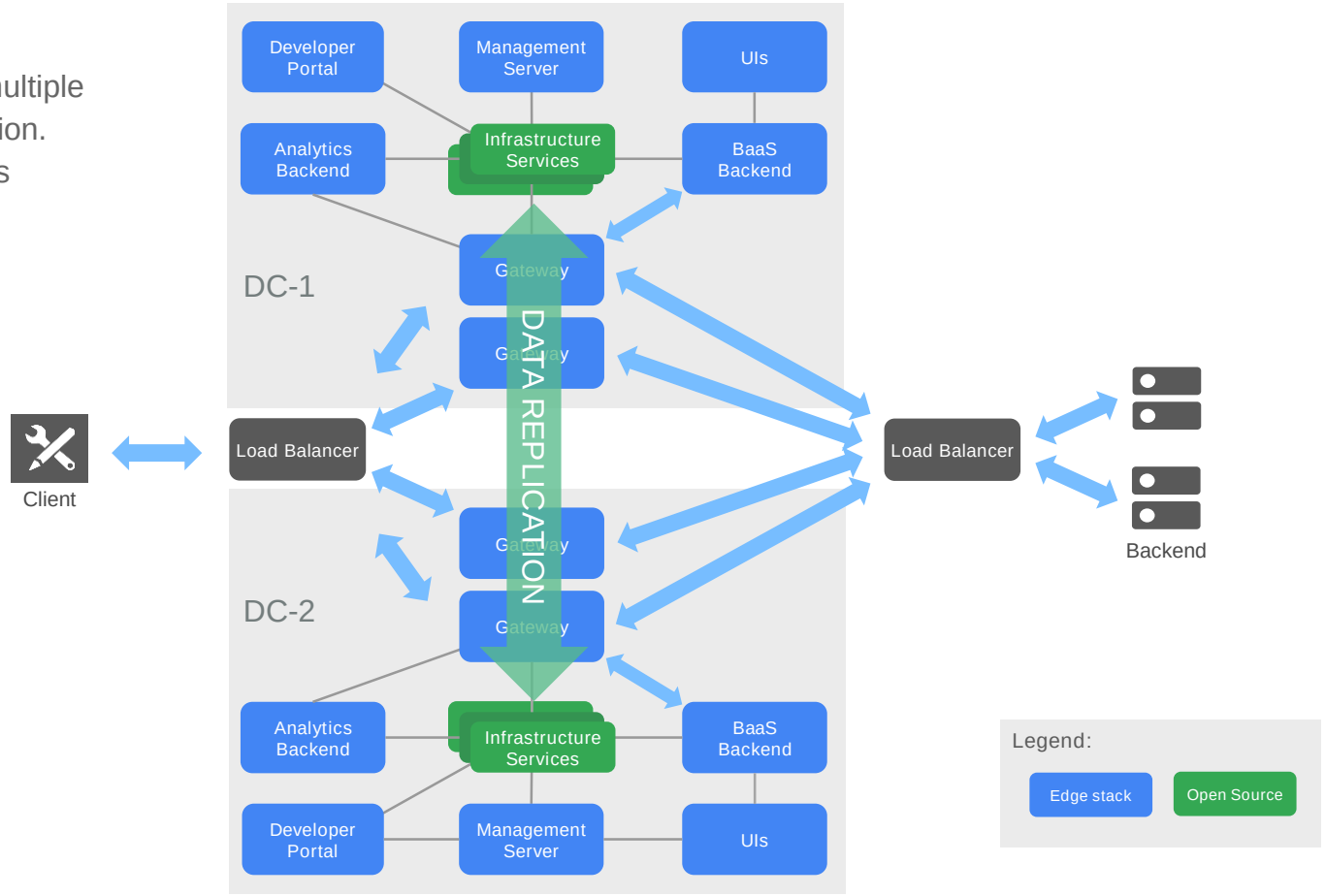
Edge Scalability

- Management Server, Analytics Backend, BaaS Backend and Developer Portal can also be set up in an HA way. Multiples instances of these capabilities within a single zone or across zones is possible.
- Gateway, Infrastructure services, analytics and other capabilities can scale-out independently from each other.



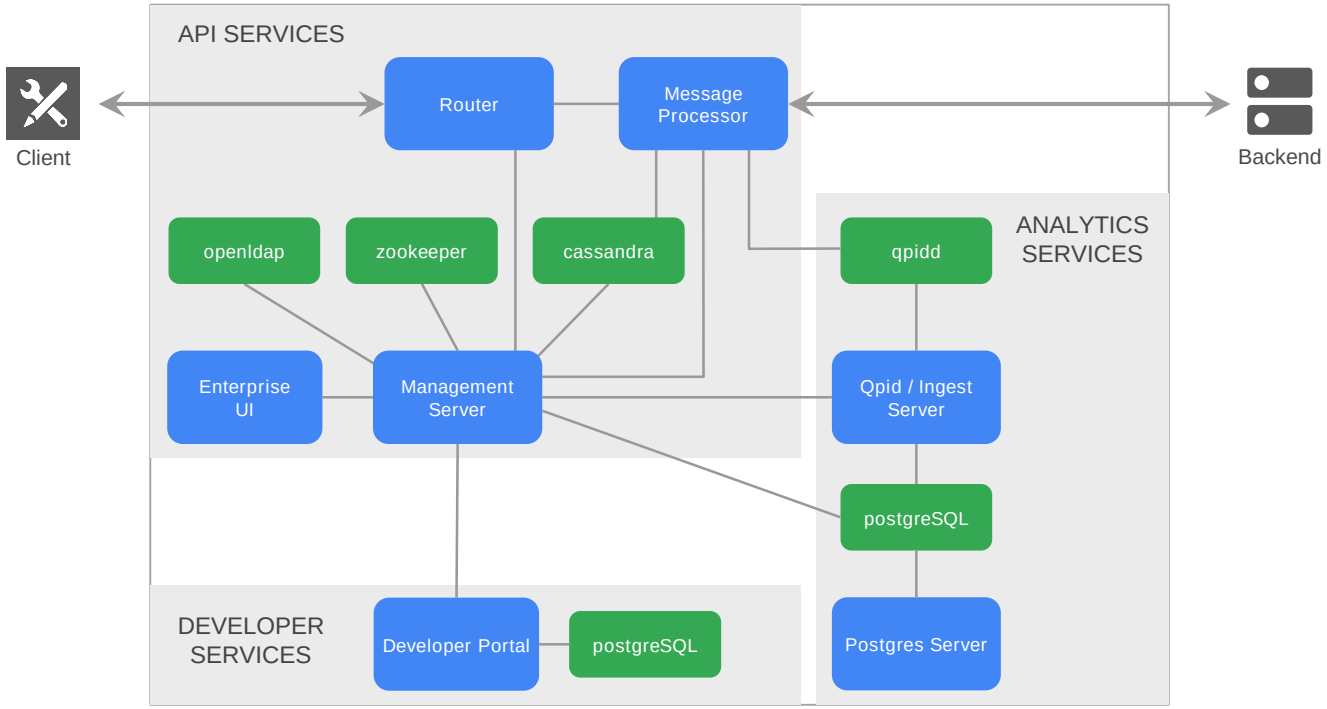
Edge Scalability

- Multi-DC and DR scalability.
- Edge is capable of scaling across multiple DCs/Regions in a active/active fashion.
- Active data replication between sites using eventual consistency.



Edge – Component View

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



- Router handles all incoming API traffic and dispatches it. The Router terminates the HTTP request, handles the SSL traffic.
- Message Processor handles API traffic for a specific organization and environment and which executes all policies.
- Management Server offers an API that is used by the Central Services server to communicate with the servers in each on-premises installation
- Cassandra stores application configurations, distributed quota counters, API keys, and OAuth tokens for applications running on the gateway.
- 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:

Edge stack | Open Source

Edge – Technology Stack

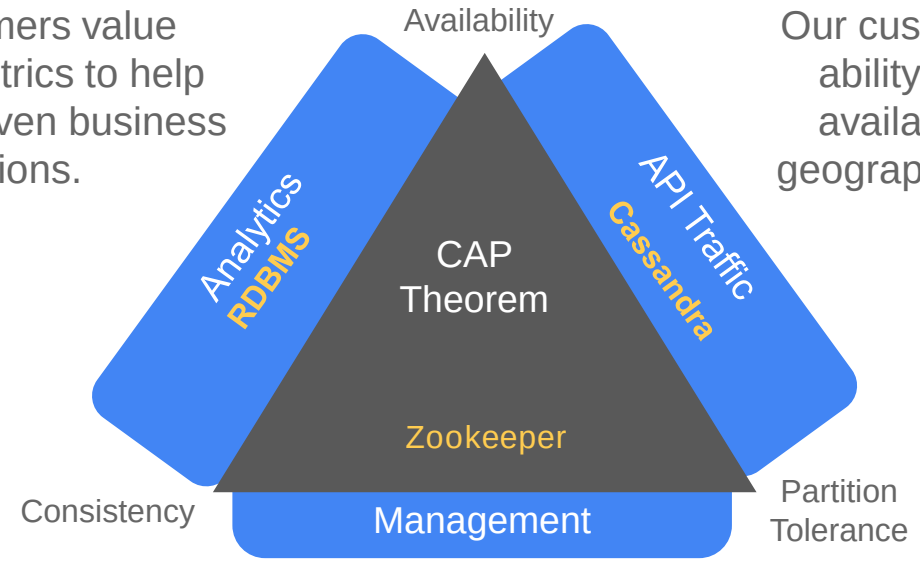
Edge components are, in general, Java based. Most components are based on a homegrown technology stack that leverages best in class open source technology under it. Below we highlight some of the underlying technologies used as building blocks.

Message Processor (MP)	Homegrown Java based
Enterprise UI (UI)	Play Framework
Management Server (MS)	Homegrown Java based
Postgres Server (PS)	Homegrown Java based
Qpid/Ingest Server (QS)	Homegrown Java based

Router (R)	Nginx (C++)
Zookeeper (ZK)	Apache Zookeeper
Cassandra (CS)	Apache Cassandra
LDAP (OL)	Open LDAP
Postgres (PG)	PostgreSQL
Qpid (QD)	Apache Qpid

Edge – Technology Stack Selection

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



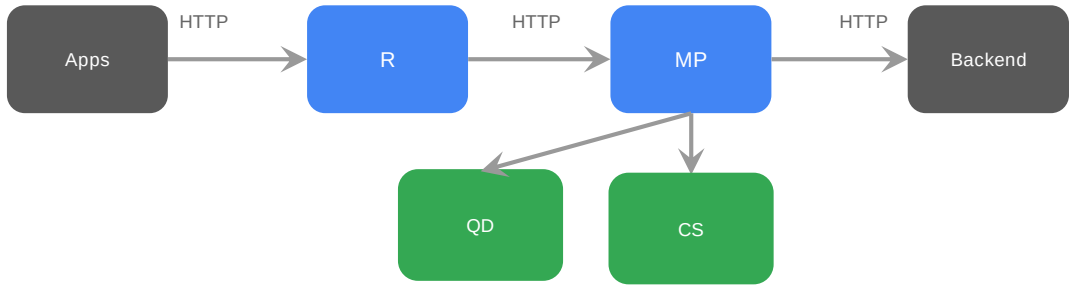
Our customers value the ability to have highly available API across geographically dispersed sites.

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

The CAP 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.

Router (R) and Message Processor (MP)



Router features

- Based on Nginx
- C++
- Multi-tenant aware (via virtual hosts)
- Receives inbound traffic on ports specified by virtual host
- Forwards requests to Message Processors in same Pod
- HTTP over port 8998
- Typical configuration: 1 Router for every 3-4 Message Processors

Message Processor features

- Java-based, Edge-specific component
- Multi-tenant aware (via HTTP Request Header Info)
- Receives inbound traffic on port 8998
- API Proxy bundles are deployed & executed here
- Admin Trace functionality is implemented here
- Analytics data originates here
- Evaluates policies defined in an API Proxy workflow
- Proxies legacy HTTP requests (REST, JSON, XML, SOAP, etc.)
- All components have management ports dedicated to management APIs (i.e. /servers/self/up)

Legend:

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

Cassandra (CS)

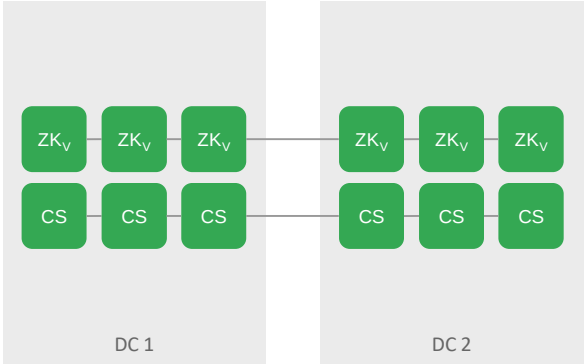
Apache Cassandra is an open source distributed database management system. It is designed to handle very large amounts of data spread out across many servers while providing a highly available service with no single point of failure.

Cassandra characteristics:

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

Cassandra is used by Edge 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.
- Cache
- Among others



http://cassandra.apache.org/

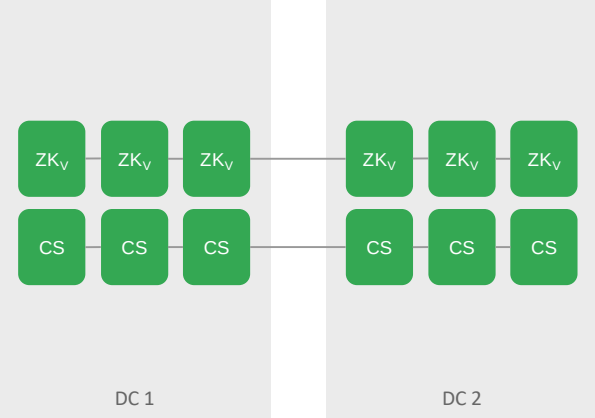
Legend:

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

Zookeeper (ZK)

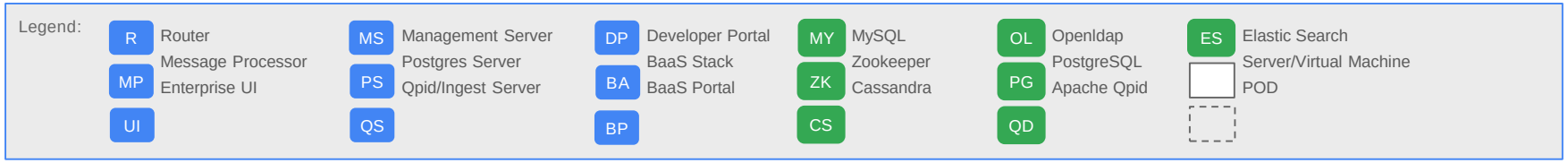
Apache ZooKeeper is a data store that provides distributed configuration service, synchronization service, and naming registry for large distributed systems.

Zookeeper is used by Edge 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/



Enterprise UI (UI), Management Server (MS) and Open LDAP (OL)

Enterprise UI

- Edge-specific component
- Java-based (Play framework)
- HTML user interface that makes use of Management API
- API Development, API Management Analytics, Trace APIs and other functions supported.
- Role-based access control.
- Horizontally scalable

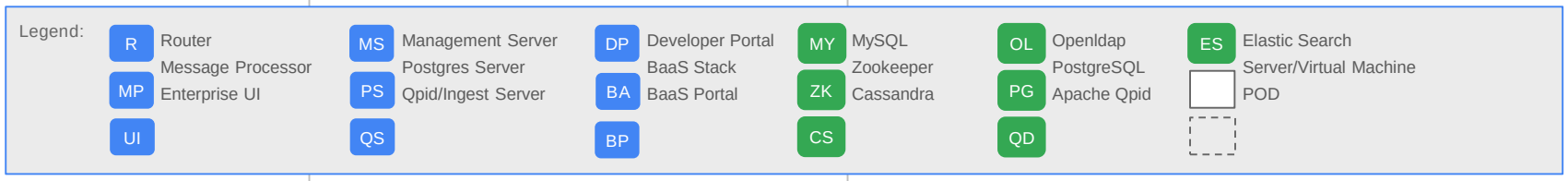
Management Server

- Edge-specific component
- Java-based
- Implements Edge Management API.
- Not intended to be used during runtime.
- Supports HTTP and HTTPs.
- Horizontally scalable

Open LDAP

- OpenLDAP is used by Edge to manage users, roles and permissions in multi-region planets
- Horizontally scalable
- Supports multi-master replication

http://www.openldap.org/

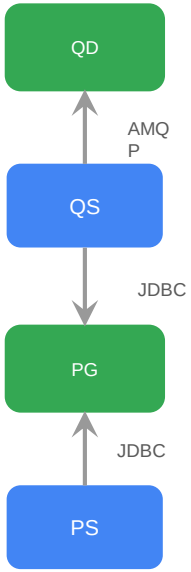


Qpid (QD), Qpid Server (QS), Postgres (PG) and Postgres Server (PS)

Qpid

Apache Qpid is queue broker based on AMPQ. It provides transaction management, queuing, distribution, security, management, clustering, federation and heterogeneous multi-platform support.

Qpid is used by Apigee Edge as messaging system for analytics and monetization data.
QS represents Edge ingest process. It is responsible for moving data between Qpid and PostgreSQL.

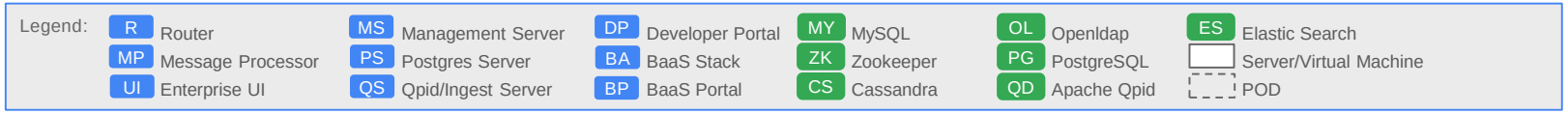


Postgres

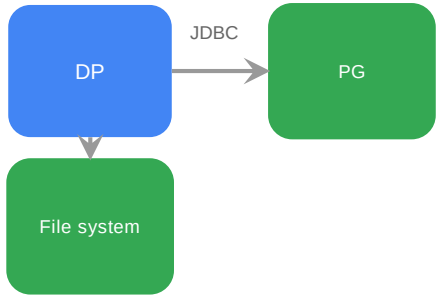
PostgreSQL is the analytics data repository.

- Open-source, relational database
- Client/server architecture
- Can interact with multiple, concurrent clients
- Supports active/slave configuration
- Streaming replication pushes all changes to slave database(s)
- Data is written to WAL (write ahead log files) files and then to the main DB
- WAL files are replicated to slave DB(s)
- All analytics data is stored here

PS perform raw data aggregation, it populates aggregated tables in PostgreSQL.



Developer Portal (DP) and Postgres (PG)



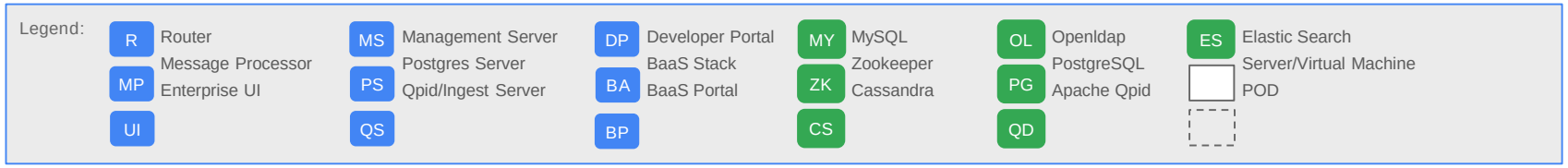
Developer Portal

- Customized version of open source CMS Drupal
- Acts as communication channel, API documentation publishing platform and application developers onboarding interface
- Out of the box, DevPortal display and allow access to data in a single organization
- Developer Portal can be customized to meet customers UI and onboarding needs
- Uses Nginx as HTTP server, Apache Solr for search and Postgres as database backend

Postgres

- Open-source relational database management system (RDBMS)
- Used to store information required by Drupal

https://www.drupal.org/



Contact us: Client Services Team || *clientservices@sidgs.com*

SID Global Solutions
Suite 500
407 W. Lincoln Highway
Exton, PA 19341
www.sidgs.com