



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 25-27 SEPTEMBER 2017

Internet Scale Content Management with Apache Oak on  
Kubernetes

Fernando Saito, Galo Gimenez, HP Inc



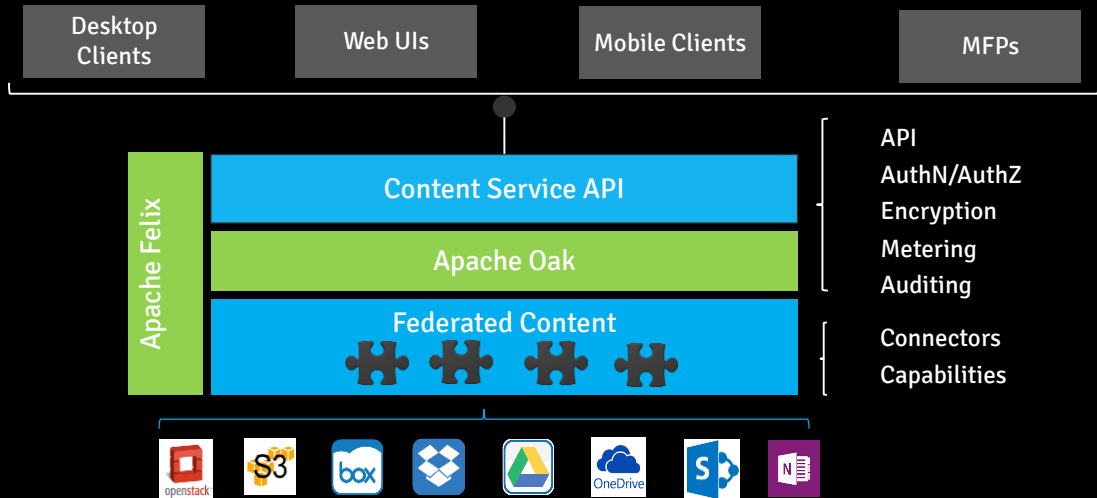
# Content at HP

- HP is in the business of transforming digital objects to physical objects, and physical objects to digital objects
- Digital objects come in multiple forms, 3D models, documents, intermediate rendered artifacts, print jobs, etc.
- Our secure document management platform allows HP devices and applications access and store content
- 65M devices connected, ~50K documents per hour

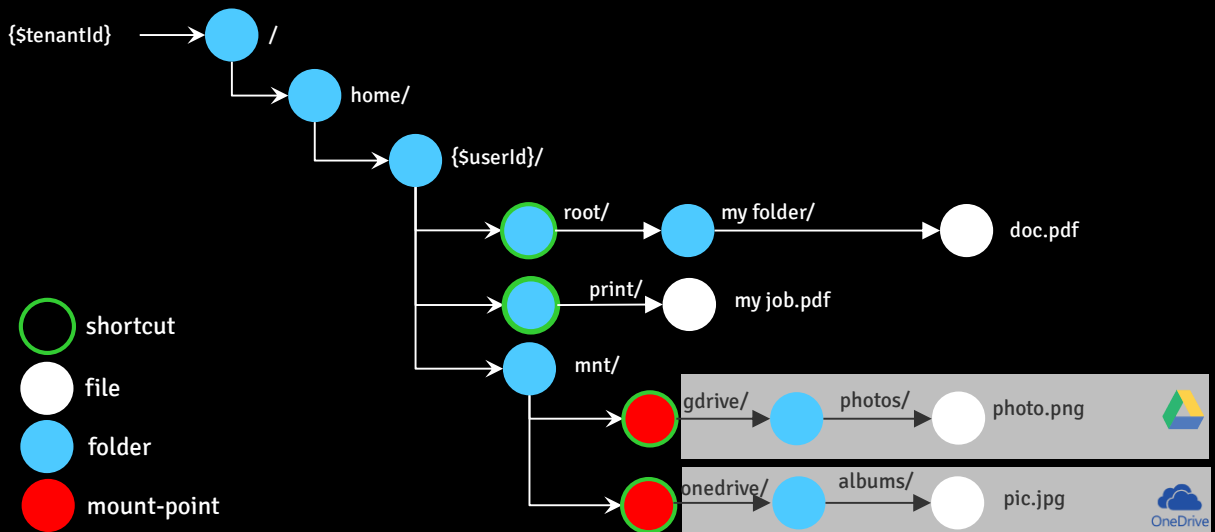


Content is a key enabler for many HP devices, be it multi-function printers, high speed commercial printers, 3D printers, or immersive computing platforms for creative solutions like HP Sprout. HP is in the business of transforming content from digital to physical and from physical to digital. Enabling this transformation keeps our devices constantly connected to cloud services.

# SDM Content Service Architecture



# Federated Content Structure

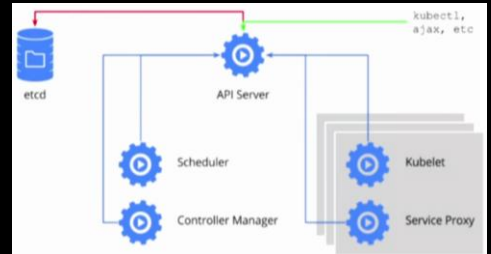


## Federated Content

- Register custom node types (connector/mount-point)
- Data access abstraction layer
  - Node interface implementation
  - Federated repository service API client implementation



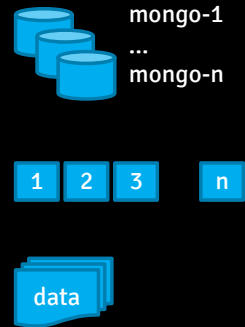
- A container scheduler system inspired in Google experience running containers
- Pods – scheduling containers in the same node
- Discovery – DNS based discovery allows legacy workloads to work
- Stateless and Statefull workloads



Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications

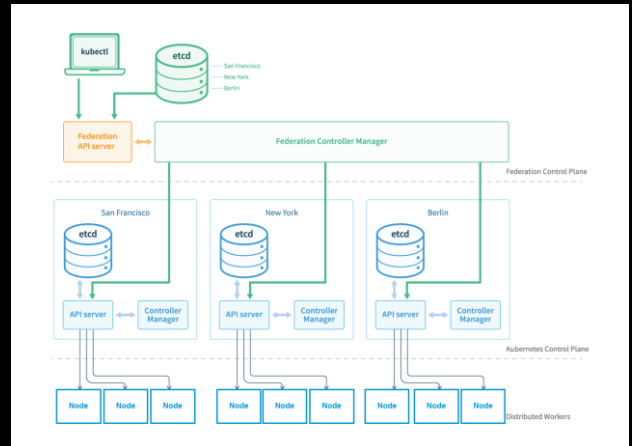
# Kubernetes StatefulSets

- Designed for state full apps – i.e. Oak, MongoDB
- Consistent Naming – (Journaling has the same node names, Mongo replicas)
- Ordered start – (Solves race conditions setting up MongoDB)
- Attached to permanent storage – (Can use local Lucene Indexes and H2 caches)



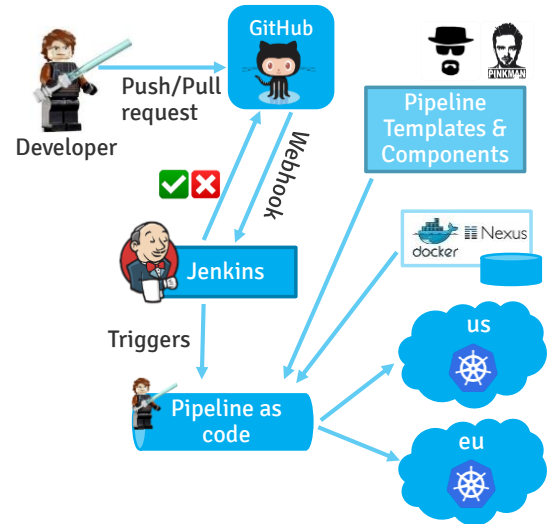
# Kubernetes Federation

- Single API to access multiple Kubernetes Clusters
- Cluster state reconciliation
- Federated Services



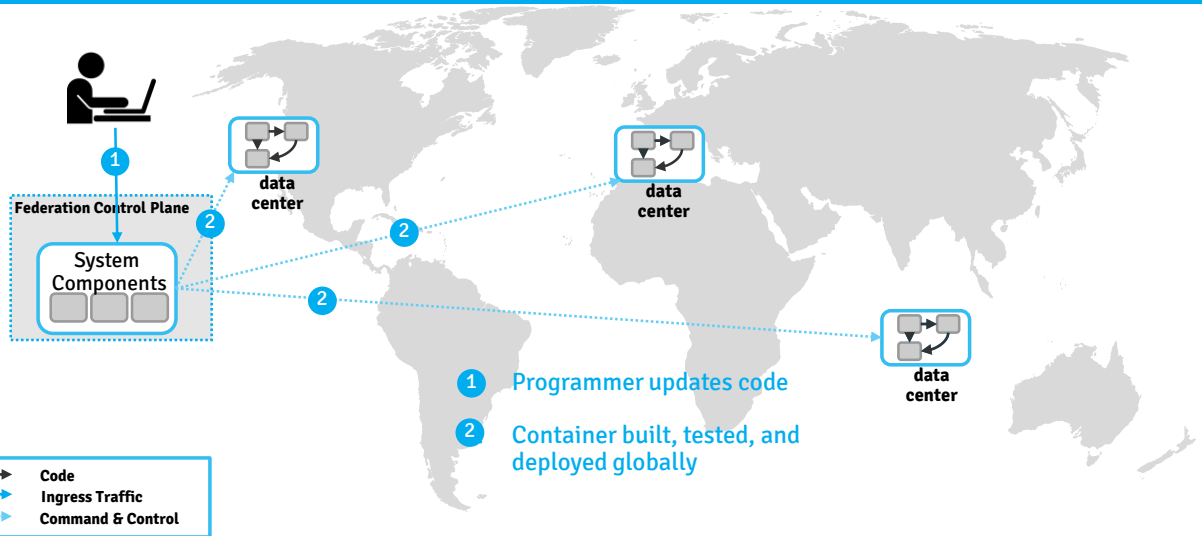
## But Federation still lags some features

- Custom deployment toolset
- Multiple kubernetes clusters contexts
- Annotation modifiers
  - Region and Environment specific configurations
  - Regional and Global DNS
  - Public and Private names
- Resource quota



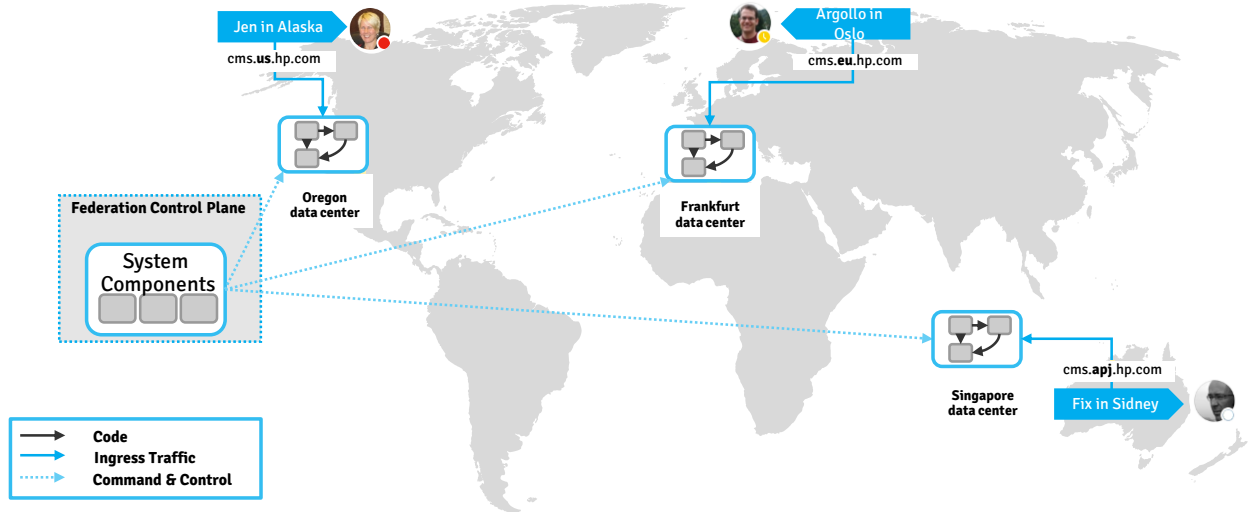


# Global Deployment



# Optimize Customer Experience

User data is stored in the data center appropriate for that user based on global data compliance policies

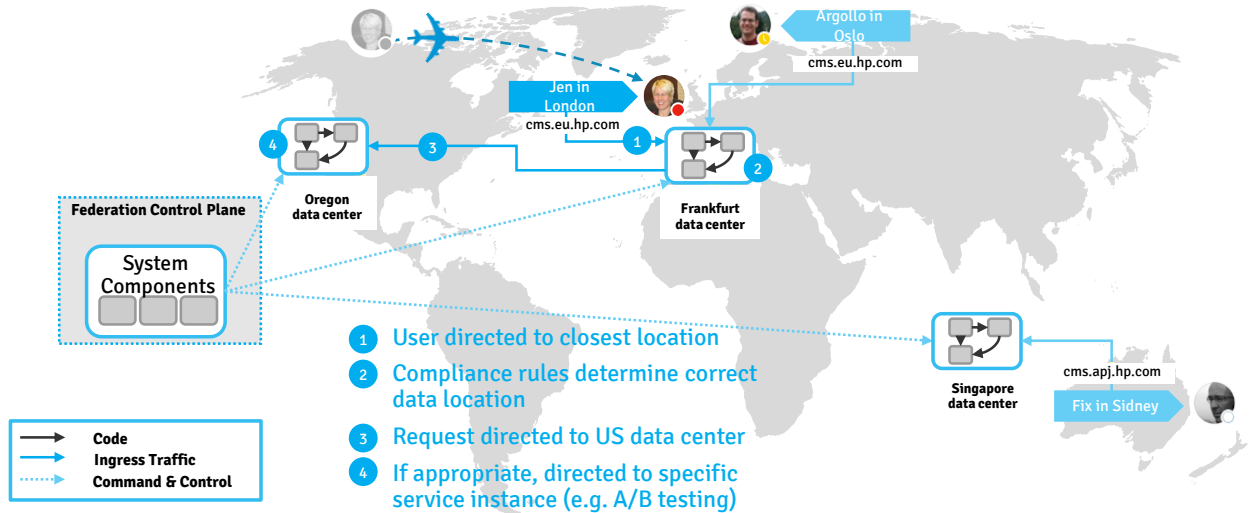


10

User signs up for an HP application or service, their data gets stored in the data center that is appropriate based on the global data compliance policies that are implemented as part of the service mesh or policy enforcement.

# Routing Customers to the Right Data Center

User initially served from closest datacenter based on location, but data still stored in appropriate data center



11

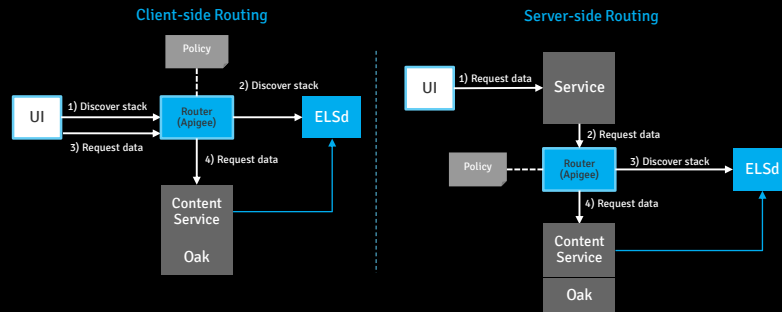
Business trip to London, I start to log in to my HP application...

1. My request is directed to closest location based my geolocation
2. At the service mesh my request detects that my data actually resides in the US data center, so my
3. request is directed to US data center
4. And optionally , if we want, we can direct the request to a specific service instance or version, for example if we wanted to do A/B testing or canary testing.

So briefly, and at a high level, that is what GDRS does from an application team or developer perspective, as well as an end customer perspective.

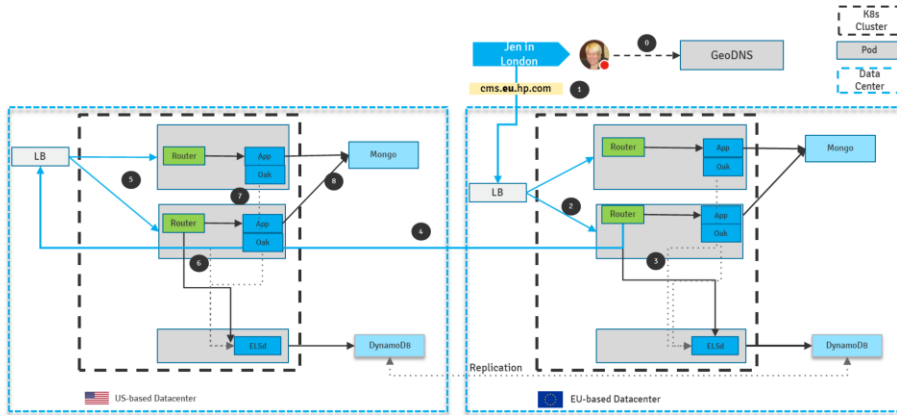
# Data Discovery – Router & Entity Locator

- The router extracts a routing key (on JWT or SAML token) to locate the service instance where the record is stored
- ELSd allows multiple services to store metadata about records they own
- Services update ELSd entity to service instance mapping. Consistency checks run periodically
- Client and server-side routing



12

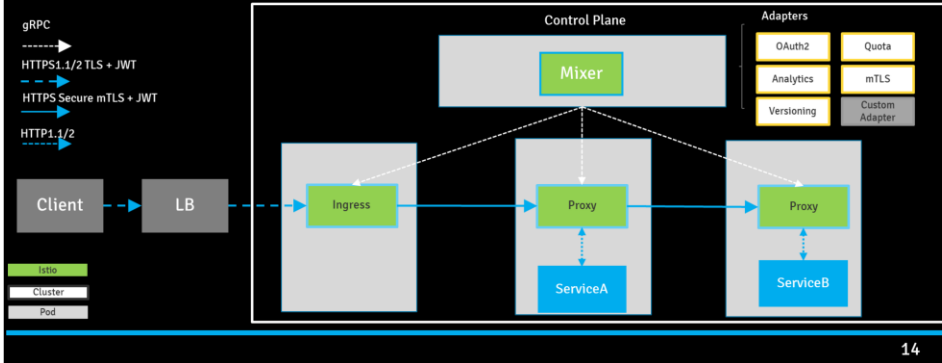
<http://docs.apigee.com/microgateway/latest/overview-edge-microgateway>  
<https://github.com/hpcwp/elsd>





## Next Steps - Istio Control Plane

Services are enriched by the Istio infrastructure on admission to the Kubernetes cluster  
Ingress applies policies to external traffic. Proxy applies policies to internal traffic  
Control plane for policy, telemetry, security,...



14

<https://istio.io/docs/concepts/what-is-istio/overview.html>

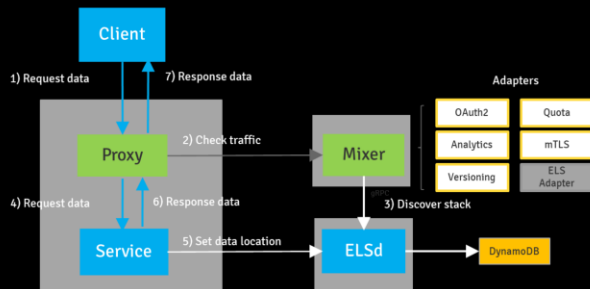
Istio is an open source project. It is a network traffic fabric, to support high level networking functions like quotas, authentication, canary test, monitoring.

Istio advantages over Apigee

- mTLS / Mutual TLS: guarantees the identity of the server to the client as well as identify of the client to the server

## Next Steps - Entity Locator Service

ELS allows multiple services to store metadata about records they own  
It provides an client- and server-side routing  
The service is globally replicated using AWS DynamoDB – every change is pushed to other instances immediately



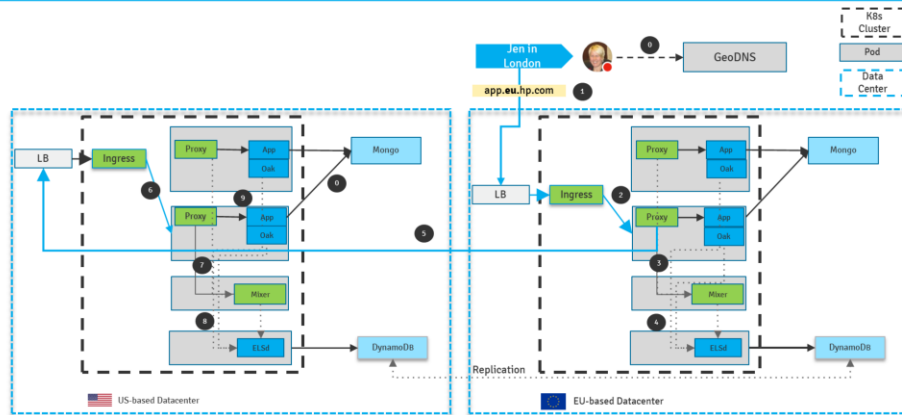
15

**Proxy:** Istio uses an extended version of the Envoy proxy, a high-performance proxy developed in C++, to mediate all inbound and outbound traffic for all services in the service mesh.

**Mixer:** Mixer is responsible for enforcing access control and usage policies across the service mesh and collecting telemetry data from the Envoy proxy and other services. The proxy extracts request level attributes, which are sent to Mixer for evaluation.

<https://grpc.io/>

# Next Steps - Data Localization





Thanks / Danke