BLOX FEST

Infoblox

Alan Conley, John Belamaric

Bloxfest - Containers

May 2016

Infoblox Microservices & Containers

- Container based microservices are an emerging approach for rapidly developing and delivering applications.
- We using containers in our own applications.
- We know how to manage addresses and names — what about the container landscape?
- We are looking at ways to enable developers to build and deploy applications more quickly.

What Are We Doing?

- Next Generation Platform (NGP)
  - Cloud Native
  - Microservices architecture. Container based.
- µDDI Research
  - R&D investigation into supporting customer container based applications
    - IPAM/libnetwork work
    - Service Registry & Discovery (SR&D), Load Balancing

Industry Participation

OCI

amazon web services, APCERA, apprenda, at&t, CISCO, ClusterHQ Container Data Management, CoreOS, DATERA, DELL, docker, EMC², facebook, FUJITSU, Goldman Sachs, Google, Hewlett Packard Enterprise, HUAWEI, IBM, Infoblox, intel, Joyent, KISMATIC, Kyup, MESOSPHERE, Microsoft, midokura, NUTANIX, ORACLE, Pivotal, POLYVERSE, portworx, RANCHER, redhat, resin.io, ROBIN, Scalock, SUSE, sysdig, Twistlock, UNIVA, Verizon Labs, Virtuozzo, vmware, weaveworks, Western Digital

CNCF

APCERA, APPFORMIX, apprenda, at&t, AVI Networks, Centrify, CISCO, cloudsoft, ClusterHQ The Container Data People, Container Solutions, CoreOS, CYCLECOMPUTING, Datawise.io, DEIS, docker, ebay, eldarion, FUJITSU, Goldman Sachs, Google, HUAWEI, IBM, iguaz.io, Infoblox, intel, ishi systems, Joyent, KISMATIC, MESOSPHERE, PROJECT CALICO, NCSOFT, NEC, NetApp, NGINX, portworx, RANCHER, redhat, RX-M, Scalock, SEL, SUPERNAP, sysdig, Twistlock, UNIVA, Virtuozzo, vmware, weaveworks, WILLIAMS GARCIA

Next Gen Platform (NGP)

SaaS, NGP & On-prem

Service Delivery Platform (SDP)

Customer Identity
Service Catalog
Licensing
Subscriptions
Provisioning/Config
Service Assurance
Cost Tracking
Problem & Incident
Metering & Billing

SaaS

ActiveTrust Dossier Threat Feeds

Threat Insight (Analytics)

DNS FW (Janus)

PaaS (NGP)

Control Plane

Application Infrastructure - persistence, messaging, logging, UI etc.

IaaS

AWS VMs, VPCs, Networking

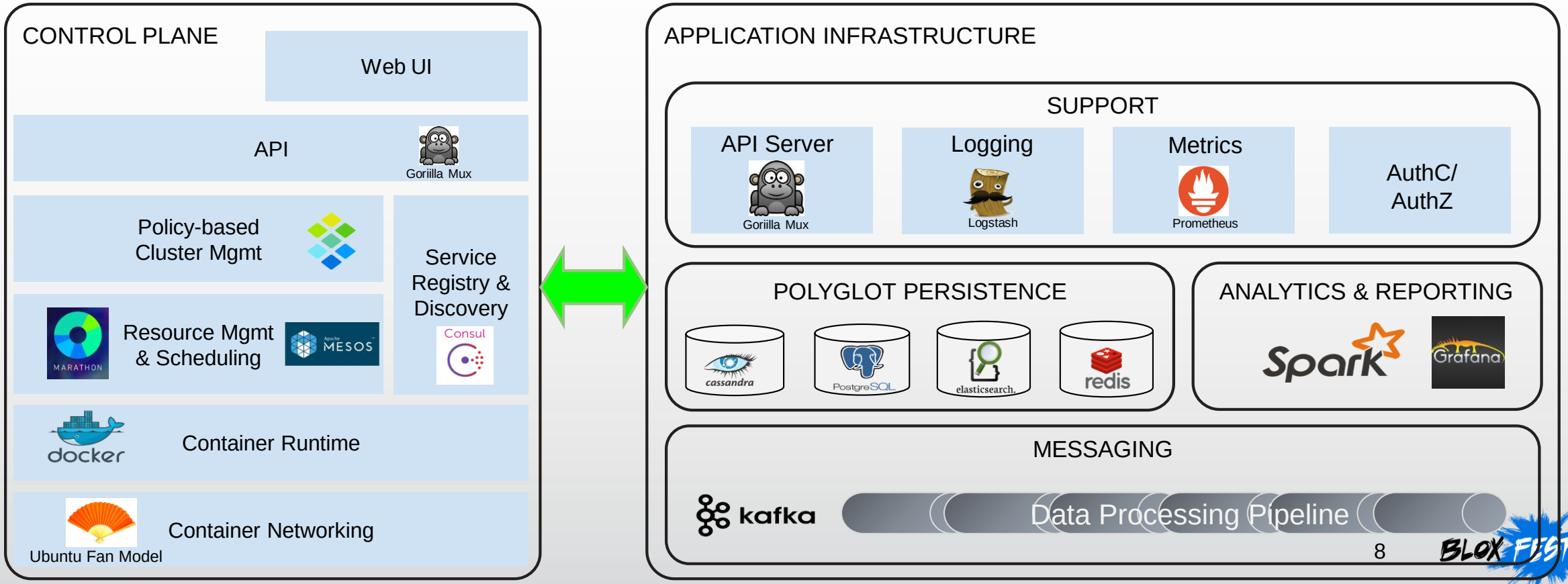
AWS Direct Connect

AWS Native Services (S3, ELB, EMR, etc.)

Architect for SaaS and on-prem

PaaS Layer (NGP)

Focus on SaaS applications and Control Plane (APIs and multi-tenancy)
For “on-prem” deployment model. Use our Control Plane, or your own?



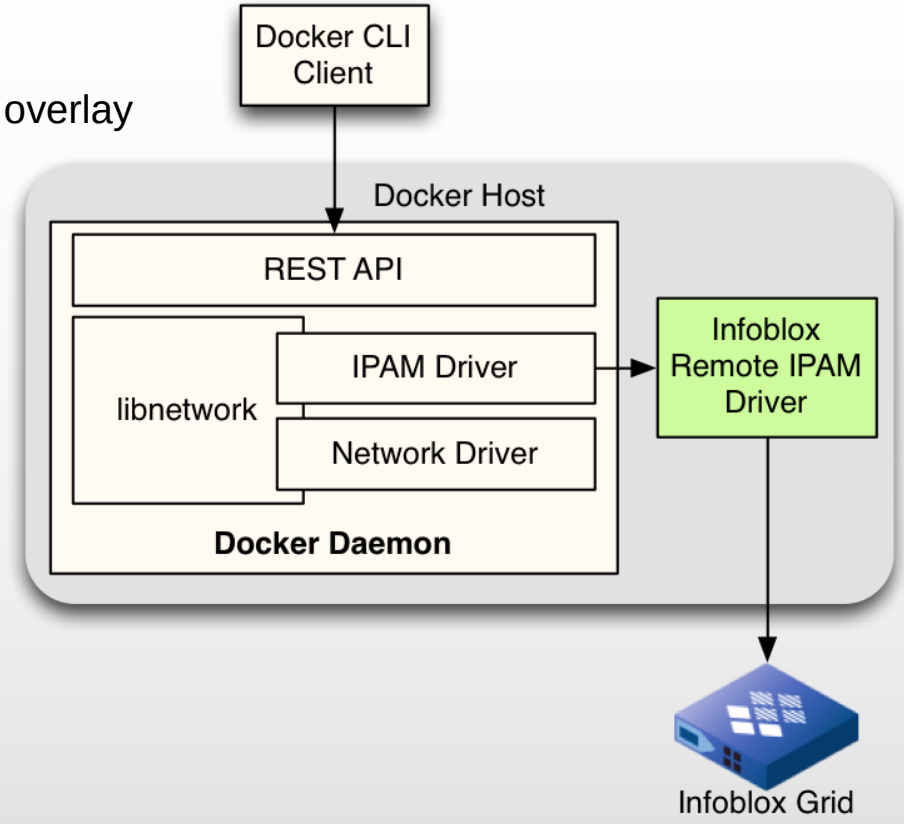
NGP Multi-tenant Elastic Scale Demo

- Clusters consisting of multiple hosts running in AWS
- Two Views
  - Control Plane Operator
  - Tenant Application Owner
- Operator sees Cluster usage (containers, CPU, Mem etc.)
- Tenant sees usage (QPS), Malware etc.

µDDI - IPAM for Containers

Infoblox Container IPAM Today

- Infoblox IPAM driver
  - For arbitrary network driver (bridge, overlay, etc.)
  - Flexible configuration with separate or combined address space per host or overlay
  - v1.0 available on Docker Hub
- App developer need not care about IPAM
- Centralized management of corporate IP space
  - IPAM offloaded from the app developer
  - Avoids IP conflicts and container routing issues
  - Enables more dynamic Ubuntu Fan-like IPAM strategies
  - Operator visibility into container space
- CNI/Rkt support under consideration

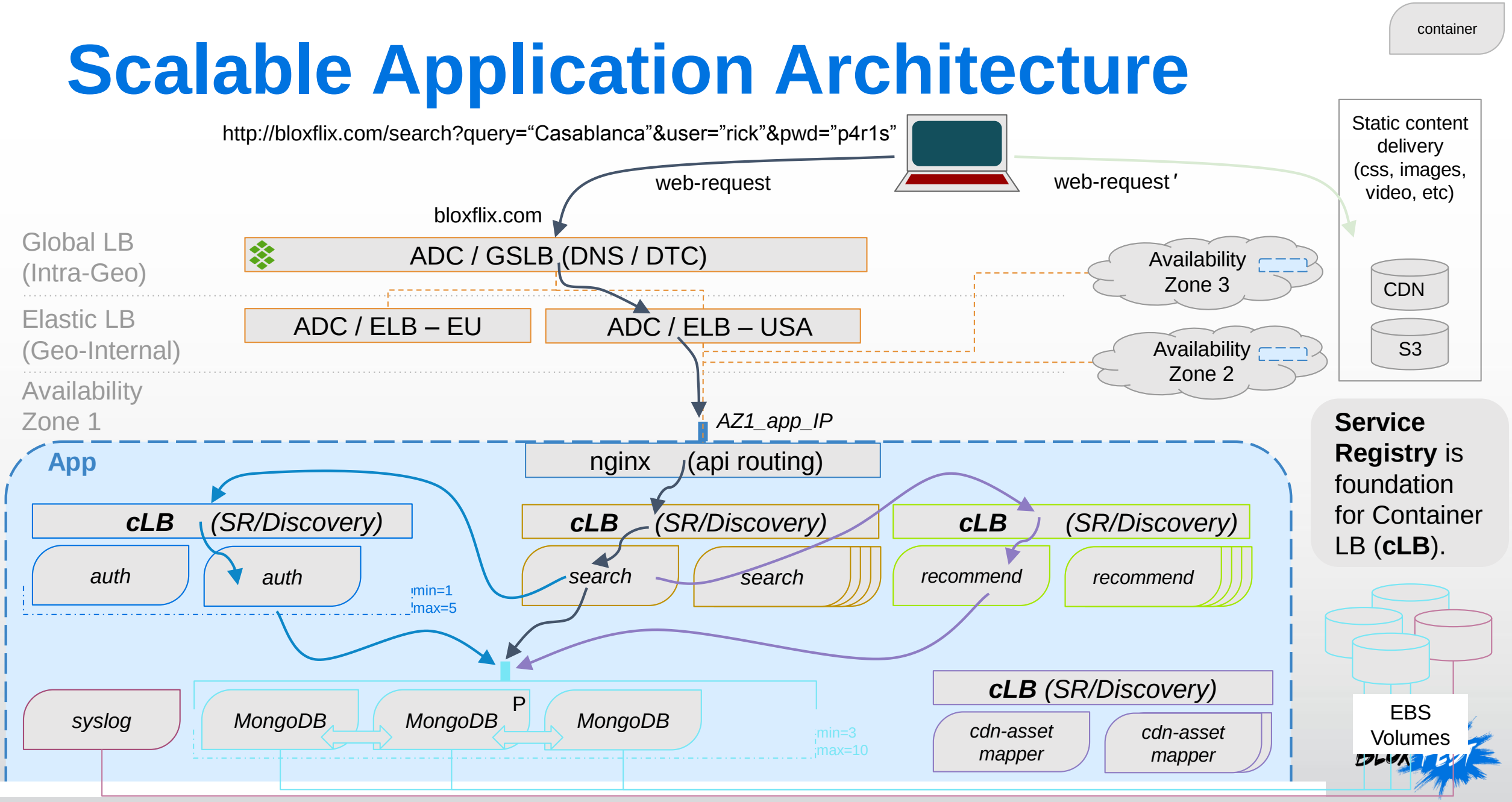


Infoblox Container IPAM Next Considerations

- Working with Docker to extend driver API
  - EA/container ID ⇨ To enable rich IPAM policy
- Automate DNS entries for externally facing containers
- Solve issues related to short-lived containers with changing IPs
  - Historical IP for log navigation/debugging
  - Historical IP tracking for forensics (with SR/D) and incident investigation
- Automate firewall configuration — inject IPs or CIDRs
- How are you managing IP addresses for your containers today?

µDDI - Service Registry / Discovery

Scalable Application Architecture



Service Registry is foundation for Container LB (**cLB**).

Containers and Service Discovery

- Container-based services are highly dynamic
  - Address may change often
  - How does one service find another?
  - Use DNS of course
- Current implementations are naïve
  - how to accommodate rapid change?
  - … hide failures?
  - … apply policy?
- Infoblox is investigating how to address service naming and discovery based on our DNS experience

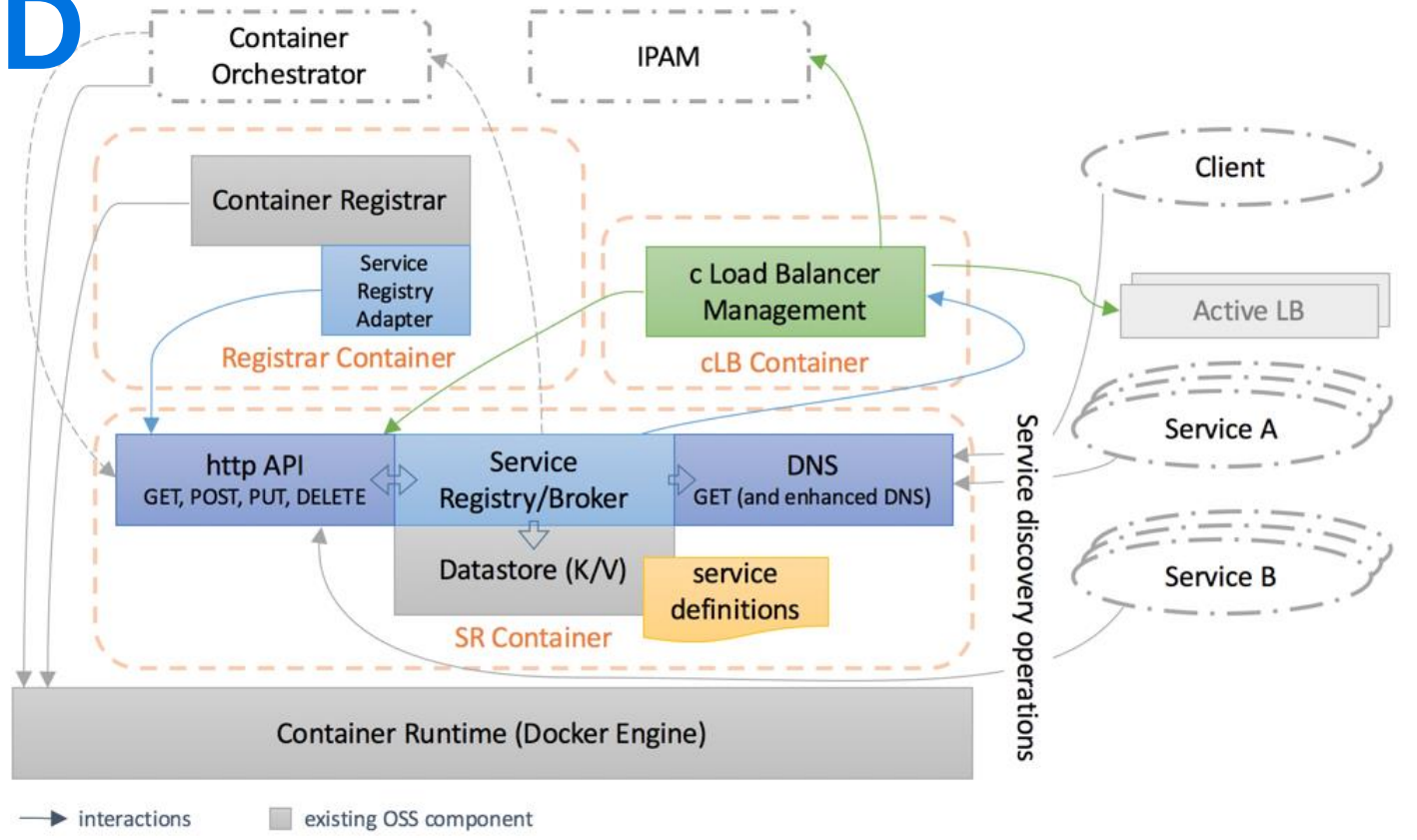
Current Landscape

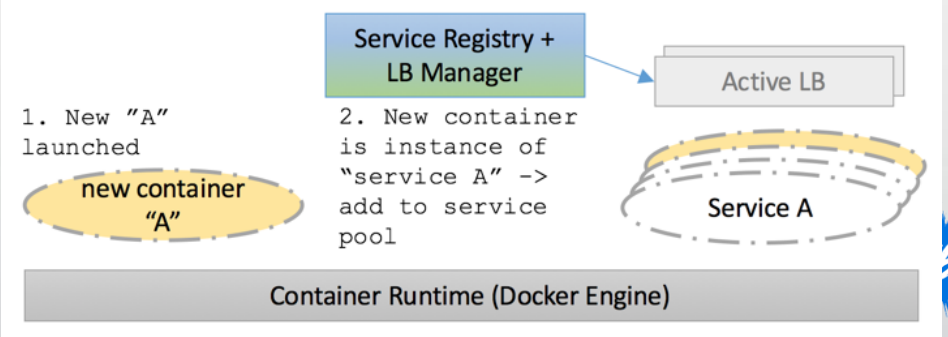
- What's out there?
  - Consul
  - WeaveDNS
  - Mesos-DNS
  - Rancher-DNS
  - SkyDNS
  - CoreDNS
  - etcd
- Consul and etcd are distributed key-value stores. Both use the Raft consensus algorithm.
- SkyDNS and CoreDNS sit on top of etcd
- Consul and the "DNS" services support DNS protocols, but are also augmented with other capabilities (i.e. http interfaces which are more suitable for app developers.)

- We believe improvements can be made to the SR/D function

Policy-based SR&D

- App developer wants to focus on app
- Availability, reliability, scalability are important and should be provided by infrastructure
- App developer doesn't need to
  - Select between container instances
  - See services for other tenants
  - See container froth
- Policy-based Service Discovery provides:
  - Easy control and change (DNS & LB)
  - Application-optimized LB
  - Application-optimized service publishing
- Additional goals
  - Reliable services (clients not exposed to container froth)
  - Transparent scaling (add more containers, infrastructure adapts based on policy)
  - Containers don't register themselves — handled by infrastructure





THANK YOU