Apache Stratos Roadmap and Strategy

About Me

- Committer, PMC Chair of Apache Stratos
- Director Cloud Architecture at WSO2 Inc
- 12+ years industry experience
- Working on cloud computing more that 6 years
- 2005, co-founded the thinkCube, the pioneers in developing the next generation of Collaborative Cloud Computing products
- lakmal@apache.org
- lakmal@wso2.com
- twitter:lakwarus

Agenda

- What is Apache Stratos?
- Mision
- Strategy
- Overview of the Apache Stratos Architecture
- Current features - few highlights
  - Multi-factored auto scaling
  - Dynamic Load Balancing
  - Smart Policies
  - Multi-tenancy
  - Cloud bursting
  - Logging, metering and monitoring
- Apache Stratos Roadmap
  - Docker support
  - Composite application support

Apache Stratos

- Apache Stratos is a highly-extensible Platform-as-a-Service (PaaS) framework that helps run Apache Tomcat, PHP, and MySQL applications and can be extended to support many more environments on all major cloud infrastructures

- Stratos initially develop by WSO2 and 2013 donated to Apache Software Foundation

- After successfully complete the incubating process Stratos now graduated as Top Level Project

Mision

Provide most comprehensive Platform as a Service for Enterprise need!

Strategy

Growing up with the community

- Current Committers/PMCs (39) are from
  - WSO2, CISCO, Indiana University, SUSE
  - Many individuals from UK, Vietnam, Sri Lanka, USA etc
  - Meritocracy matter
- Conducting public Google hangouts (weekly or so)
  - Educate community
  - More interactive
  - Code reviews
  - Recorded and publish on the website
  - Very helpful for newcomers
- Frequent developer previews (by weekly or so)
  - Educate developer
  - Provide continuous delivery
  - Easy to do integration testing with what you are working on
  - Getting early feedback without waiting for the release

Growing up with the community...

- Participating/conducting meetups
  - Building community
  - Conducted Barcamp in San Francisco, Sri Lanka
- Participating/conducting Hackathons
  - Adopting best practices
  - Building community
  - ApacheCon NA 2014
  - WSO2 Hackathon 2014
- Speaking at many conferences
  - Taking your massage to the wide ordinance, other communities
  - Getting feedback
  - Making new contributors
  - ApacheCon NA, ApacheCon Europe, CloudOpen NA, CloudOpen Europe, QConSF, JaxLondon, WSO2Con

Not to reinvent the wheel

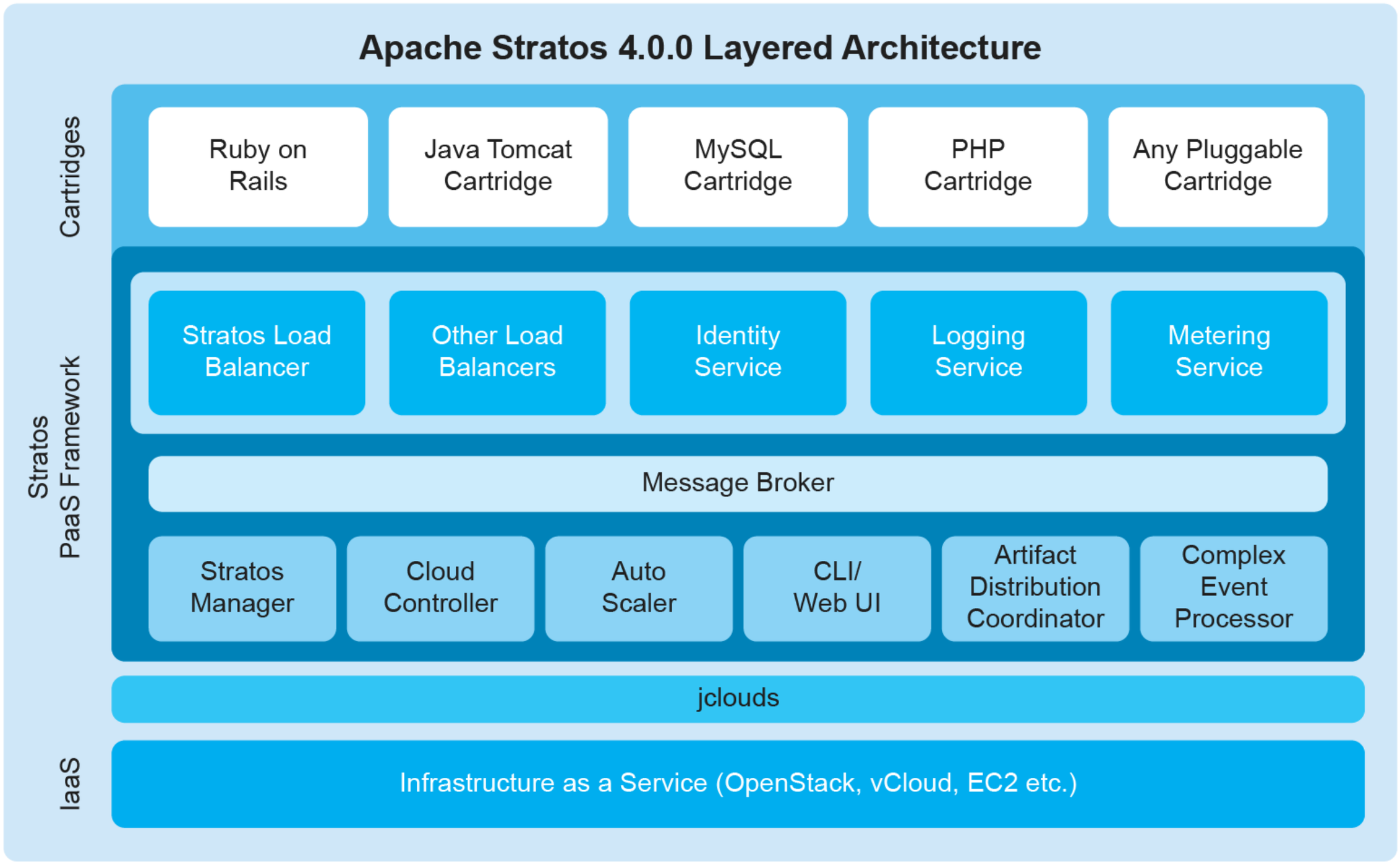
- Carefully evaluate existing open source software for suitable integration.
- After broad discussion @dev, choose best out of all
- Current third party project integrated
  - Apache Jclouds
  - Openstacks
  - CloudStacks
  - Docker
  - CoreOS
  - Kubernetes
  - WSO2 Carbon/CEP
  - ActiveMQ
  - etcd

Collaborative development

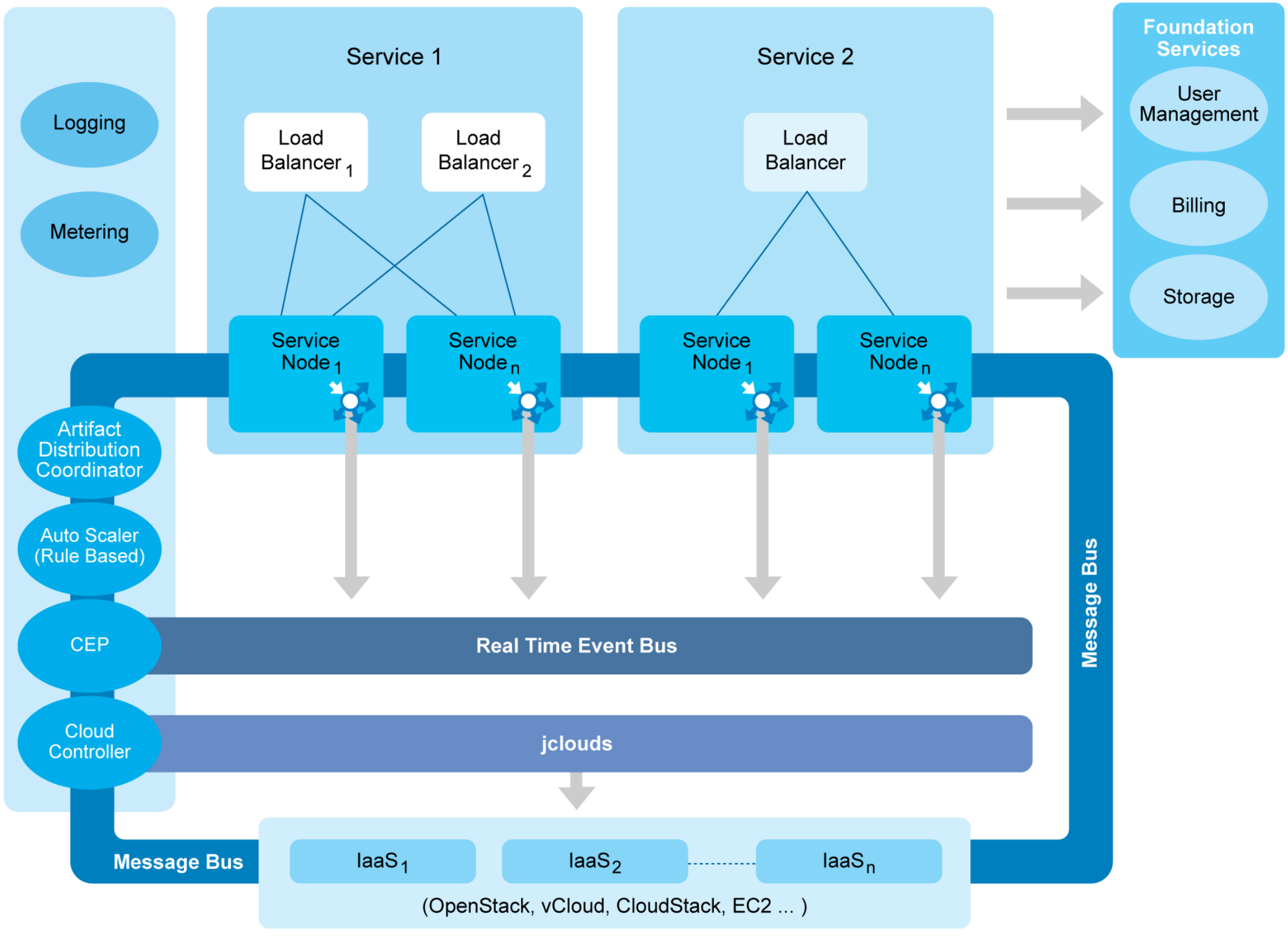
- Communicate with third party software community
- Meetup, hangout with them
- Giving feedbacks
- Contribute back/ sending upstream fixes/improvements
- Current work closely with
  - Apache Jclouds
  - Apache CloudStacks
  - CoreOS
  - Kubernetes
  - WSO2 Carbon/CEP projects

Overview of the Apache Stratos Architecture

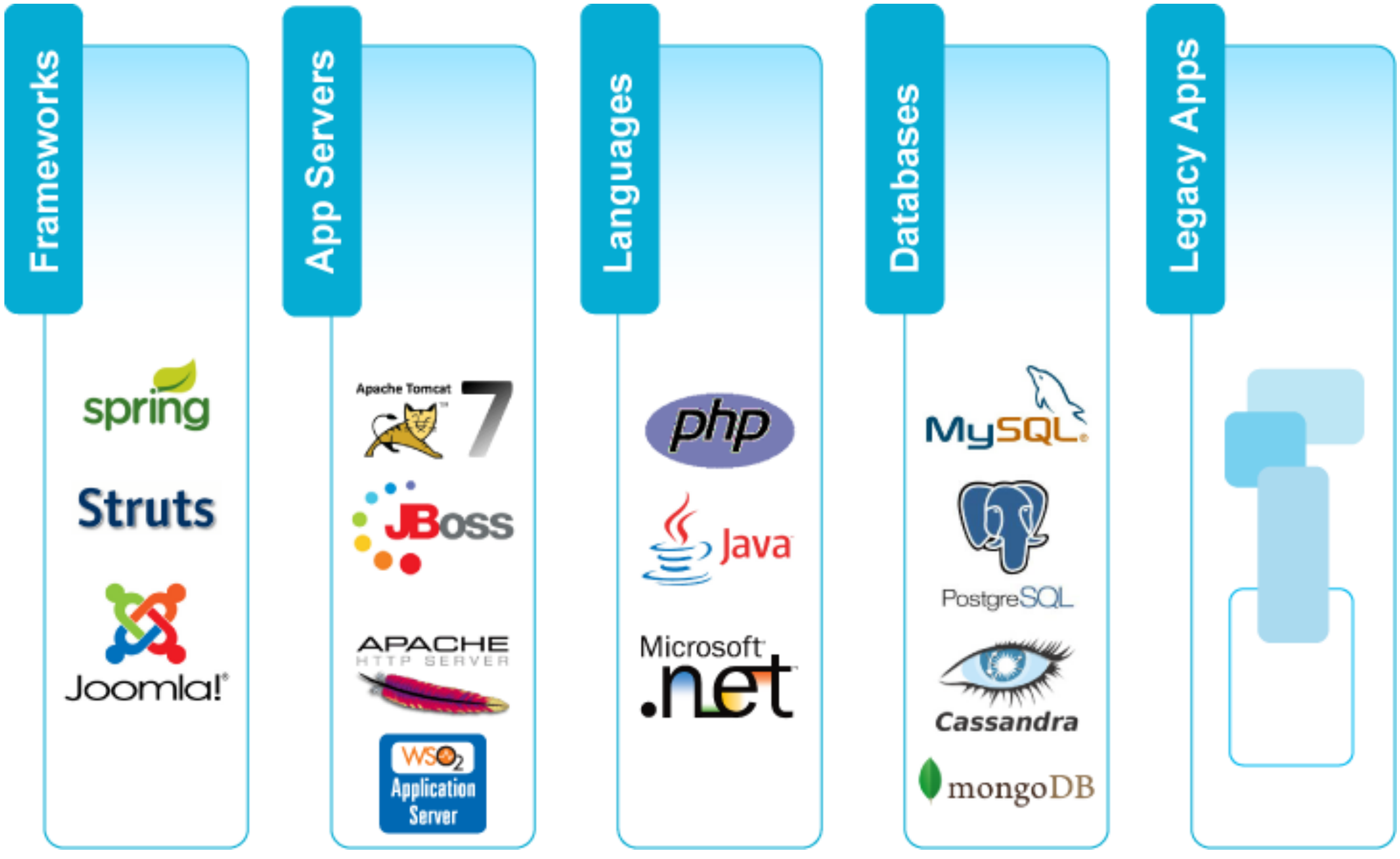
Apache Stratos Layered Architecture



Apache Stratos L1 Architecture for VM based Cartridges



Apache Stratos Cartridges

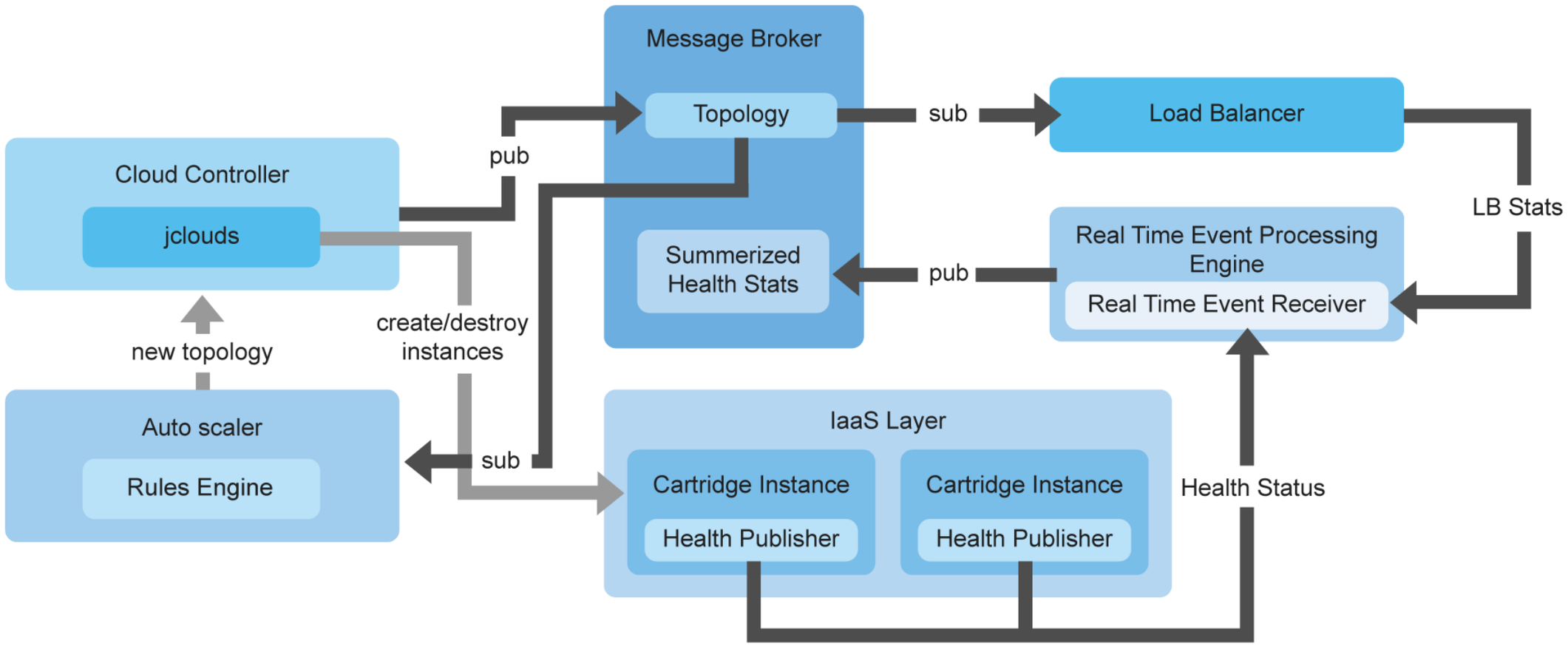


Current Features

Multi-factored Auto Scaling

What is it?

- Scaling algorithm can use multiple factors. such as
  - Load average of the instance
  - Memory consumption of the instance
  - In-flight request count in LB



Multi-factored Auto Scaling...

- Capable of predicting future load
  - Real time analysis of current load status using CEP integration
  - Predict immediate future load based on CEP resulting streams
  - Predicting equation $s=ut + ½ at^2$
  - s=predicted load, u=first derivative of current average load, t= time interval , a=second derivative of current load

Why should one care?

- Maximise resource utilization
- Easy to do capacity planning
- Dynamic load based resource provisioning
- Optimizing across multiple clouds

Scalable and Dynamic Load Balancing

How Scalable it is?

- In theory infinite
  - horizontal scaling
  - limited by resource (instance capacity) availability

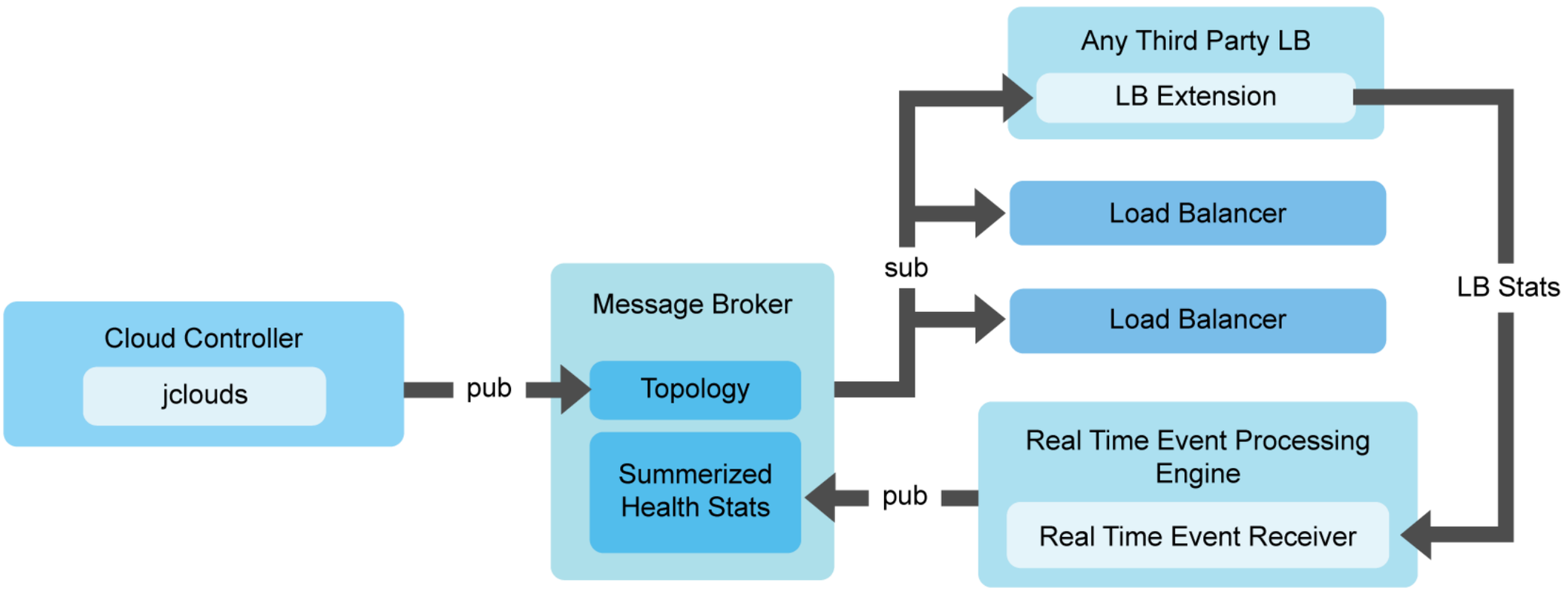
How Dynamic it is?

- Load Balancers are spawned dynamically
  - LB too is a cartridge
- In case of multi-cloud, multi-region, LB can scale per cloud/region
- Per service cluster LB

Scalable and Dynamic Load Balancing..

What is unique about Stratos

- Cartridge based LB model
- Can bring any third-party LB
  - HAProxy, nginx, AWS ELB
  - As easy as plugging into LB extension API



Smart Policies

What are the smart policies?

- Auto scaling
- Deployment

Auto scaling policy

- Define thresholds values pertaining scale up/down decision
- Auto Scaler refer this policy
- Defined by DevOps

Deployment policy

- Defined how and where to spawn cartridge instances
- Defined min and max instances in a selected service cluster
- Defined by DevOps based on deployment patterns

Smart Policies

Why should one care?

- Can provide cloud SLA

What are the advantages?

- Make DevOps life easy
  - help keep to SLA
- Make SaaS app delivery life easy
  - do not have to worry about availability in application layer

Multi-tenancy

What MT model does it support?

- Container MT
  - virtual Machine, LXC, Docker
- In-container MT
  - within VM/LXC/Docker tenancy

What is unique?

- Can have high tenant density

What are the advantage of this model?

- Optimizing resource utilization
  - by sharing resource such as CPU, memory across tenants
  - low footprint, based on utilization/usage of the tenants app
- No need dedicated resource allocation for tenants

Cloud Bursting

What is it?

- Expanding/provisioning application into another cloud to handle peak load.

Why Should one care?

- Resource peak time can be off-loaded to third party clouds/resources

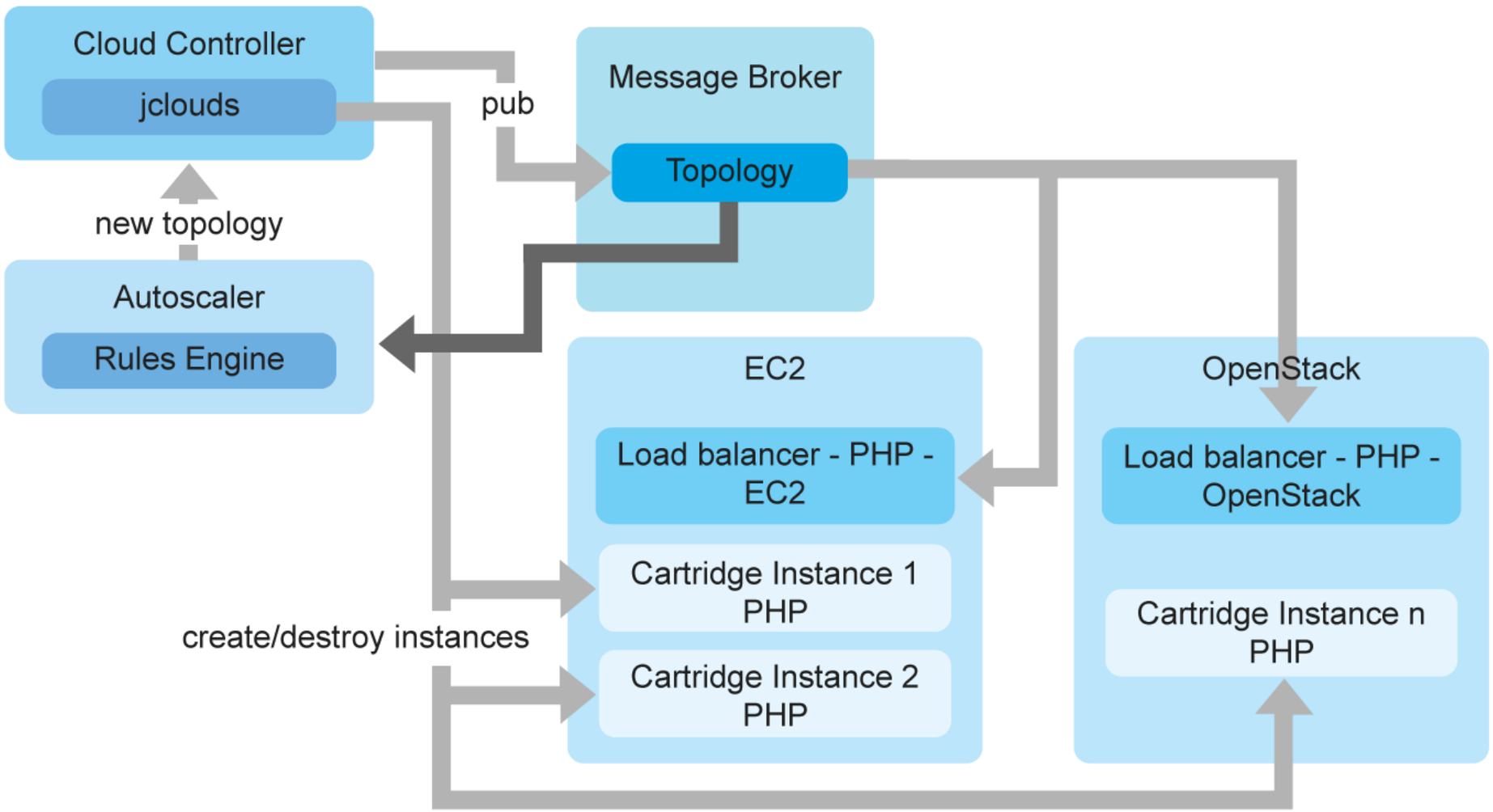
What is unique about it?

- Can off-load to any cloud
  - Private, Public and Hybrid
- Easy to managed with the model of LB per busting cloud

Cloud Bursting...

What are the advantages?

- Make DevOps life easy
- Low TCO, and higher utilization existing dedicated resources



Logging, Metering and Monitoring

What details are?

- Instance up/down time
- Each and every instances health status
  - application health, load average, memory consumption
- Application logs

Why should one care?

- Centralize view for all logging, metering and monitoring

What are the advantages?

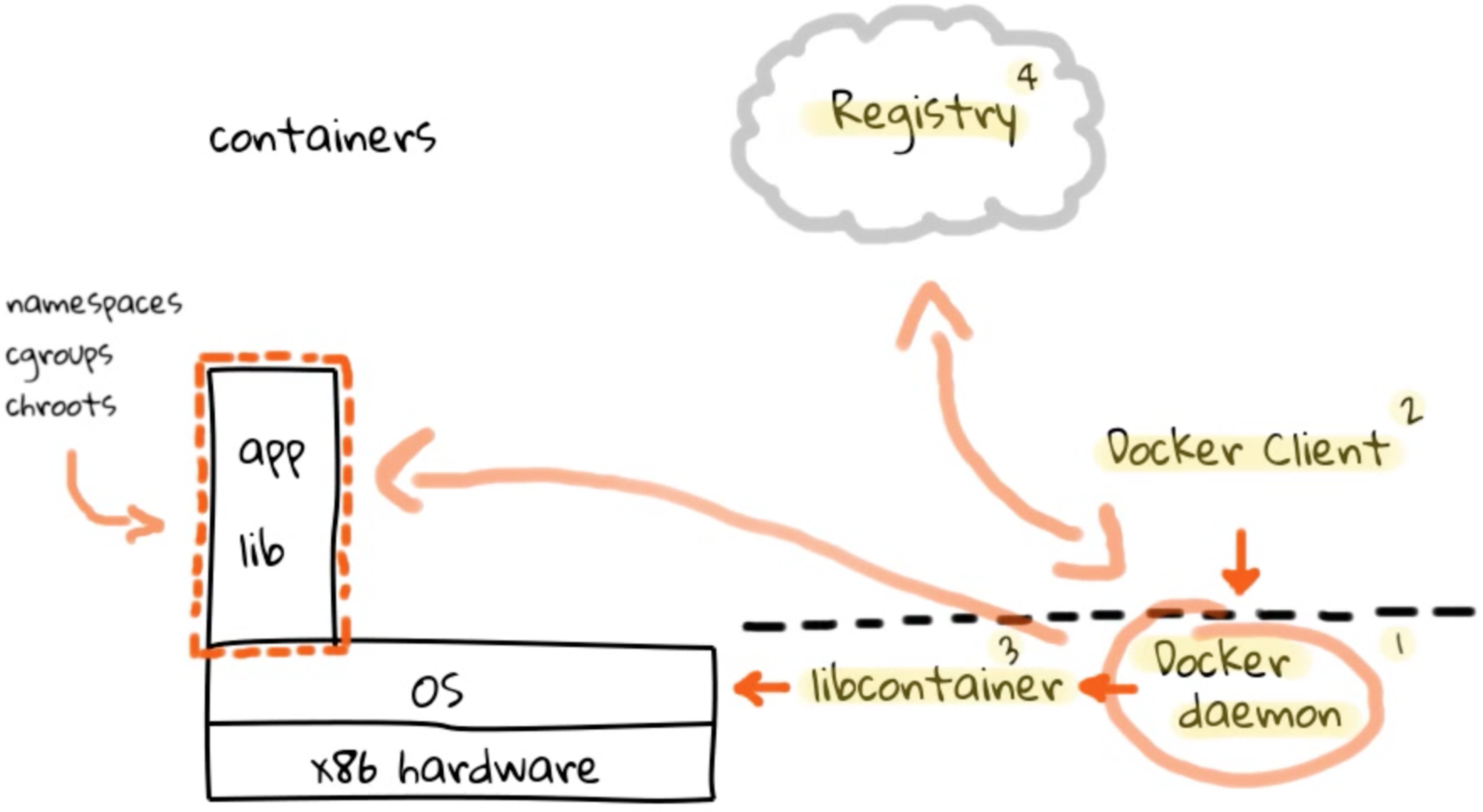
- DevOps life easy
  - centralize log viewer
  - centralize dashboard
- Easy to throttling

Apache Stratos Roadmap

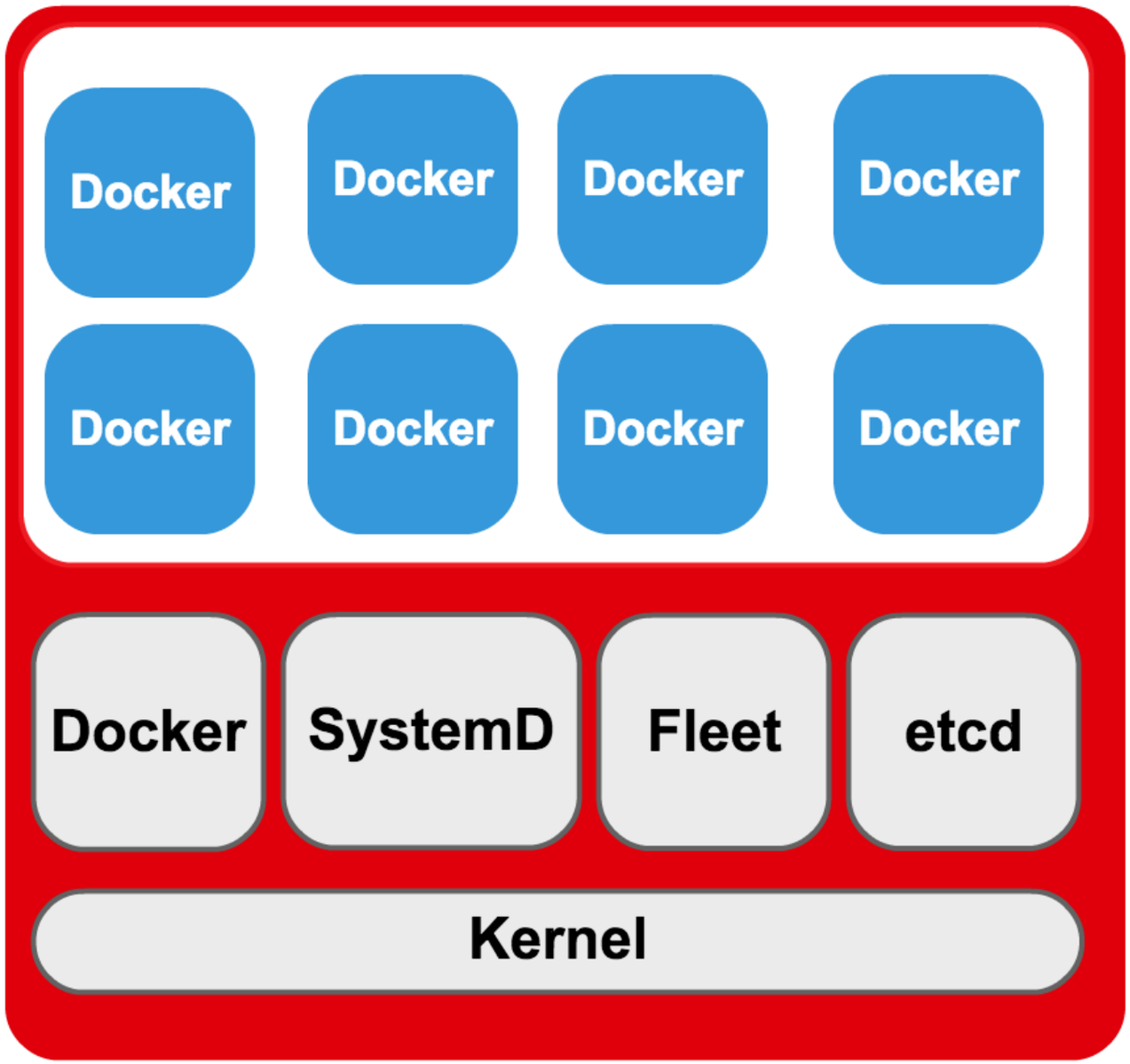
Docker Support

- Why just not with only Docker?
- Apache Stratos next release is mainly into
  - Docker based cartridge support
  - integration with CoreOS
  - integration with Kubernetes
  - integration with flannel
  - integration with discovery service
  - build in docker registry

What is Docker?

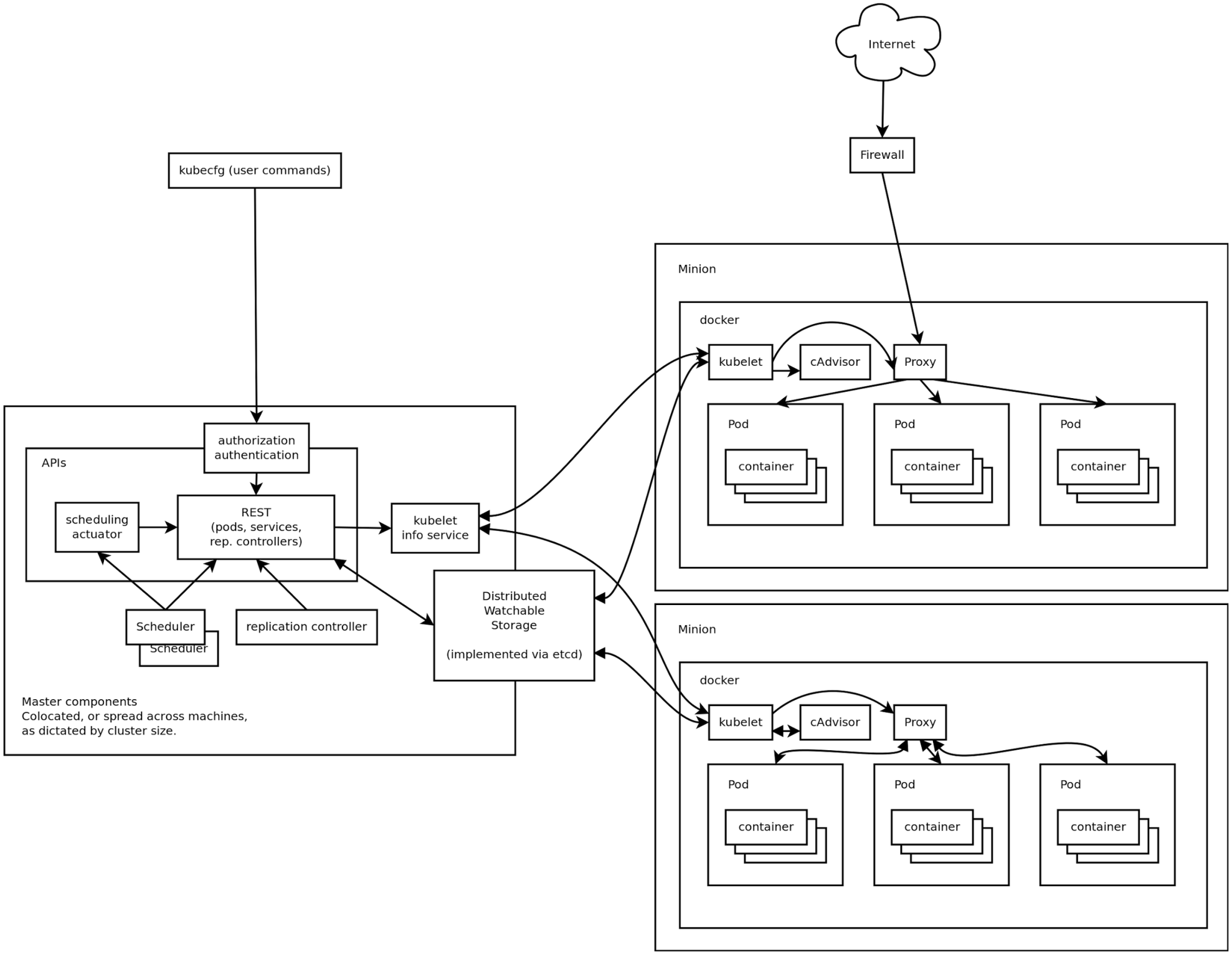


What is CoreOS?

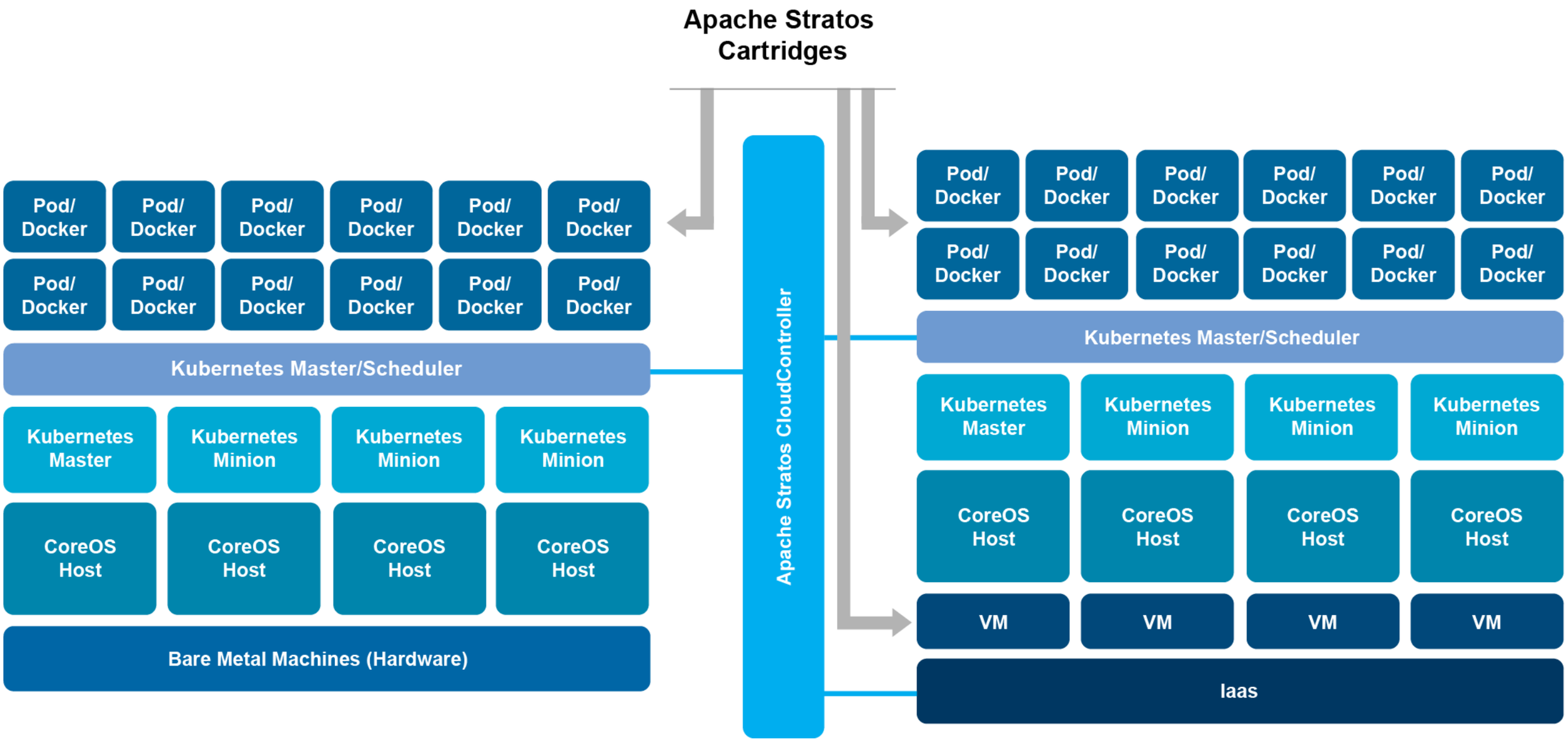


CoreOS Host

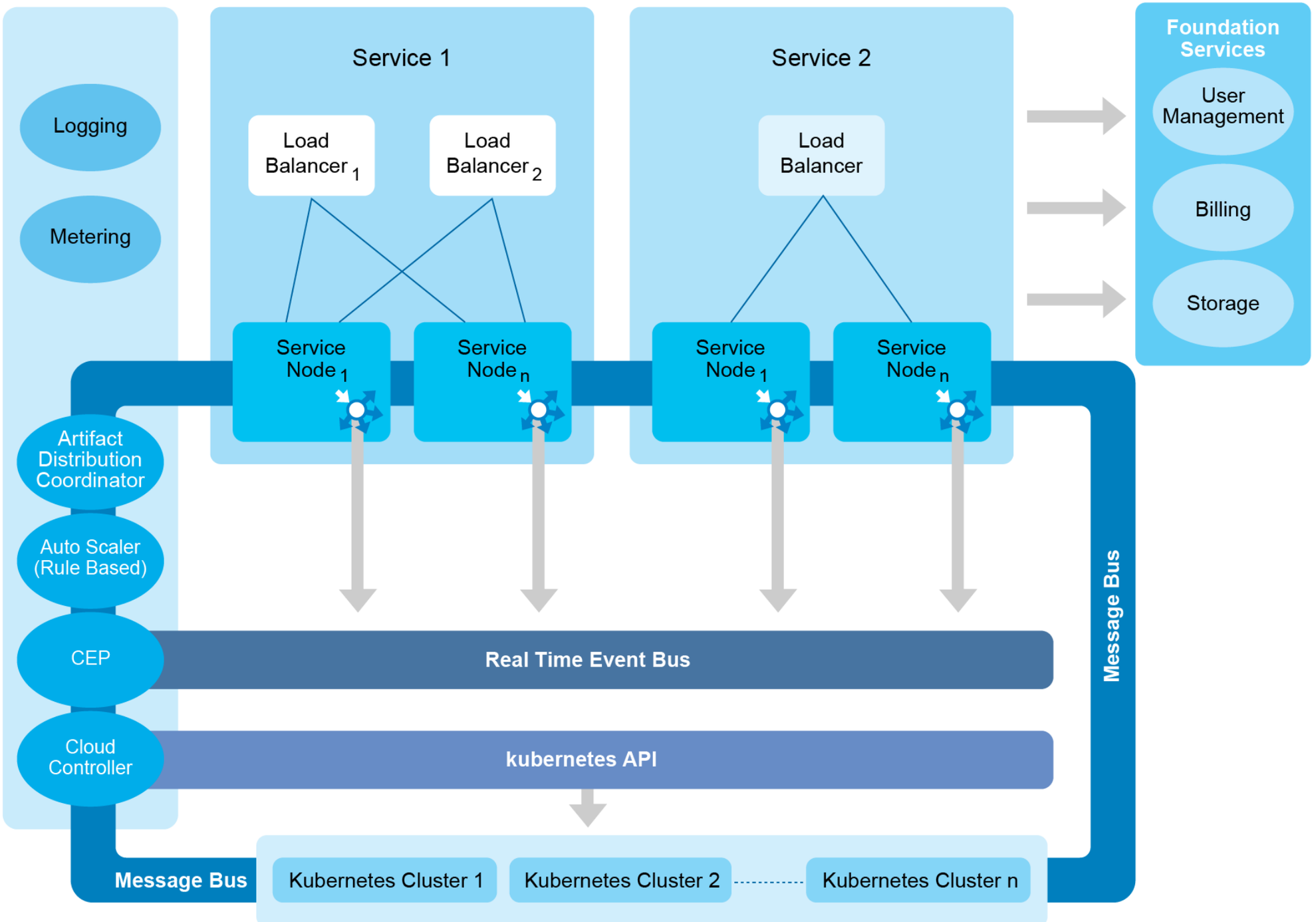
What is Kubernetes?



Two level of scalability



Apache Stratos L1 Architecture for Docker based Cartridges



Grouping and composite application support

- Json based complex application definitions
- Cartridge grouping
- Manage dependencies
  - startup order
  - termination order
  - maintaining ratio
- Manage dependency metadata
- Group wise scaling

More Information !

- http://stratos.apache.org
- http://lakmalsview.blogspot.com/2013/12/sneak-peek-into-apache-stratos.html
- https://cwiki.apache.org/confluence/display/STRATOS
- https://github.com/coreos/etcd
- https://coreos.com
- https://wso2.com
- https://www.docker.com/
- https://www.youtube.com/watch?v=tsk0pWf4ipw
- https://sysadmincasts.com/episodes/31-introduction-to-docker