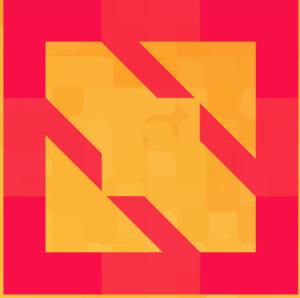




KubeCon



CloudNativeCon

---

North America 2019

---





KubeCon



CloudNativeCon

North America 2019

# eBay Search On K8s

*Yashwanth Vempati (K8s platform)*

*Mohnish Kodnani (Search)*



# Motivation

Run a large scale, latency sensitive application like ebay's Search Engine on K8s and the design choices we made to achieve this feat.

# eBay Search Background



KubeCon

CloudNativeCon

North America 2019

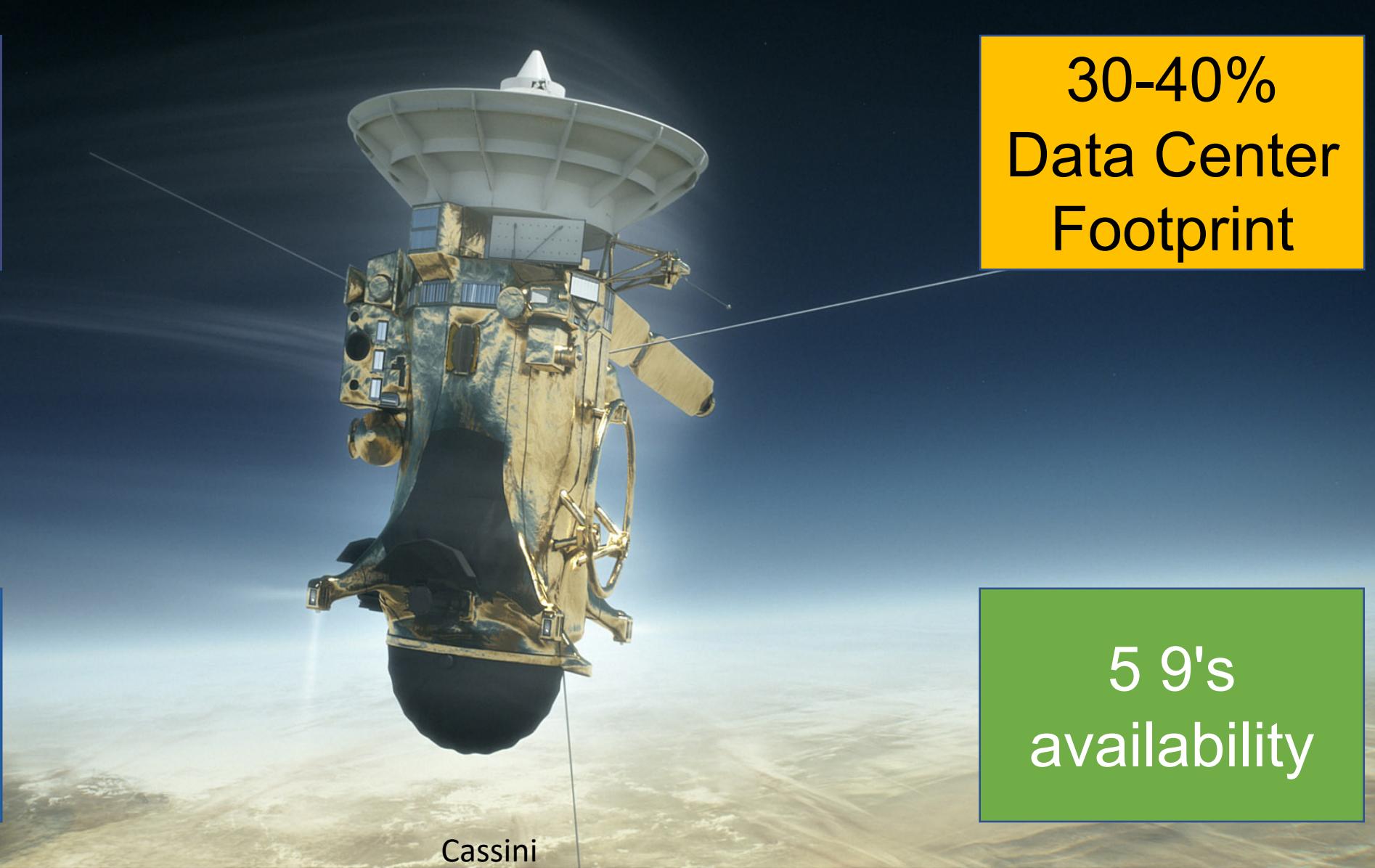
1.4 Billion  
Active  
Listings

300K QPS

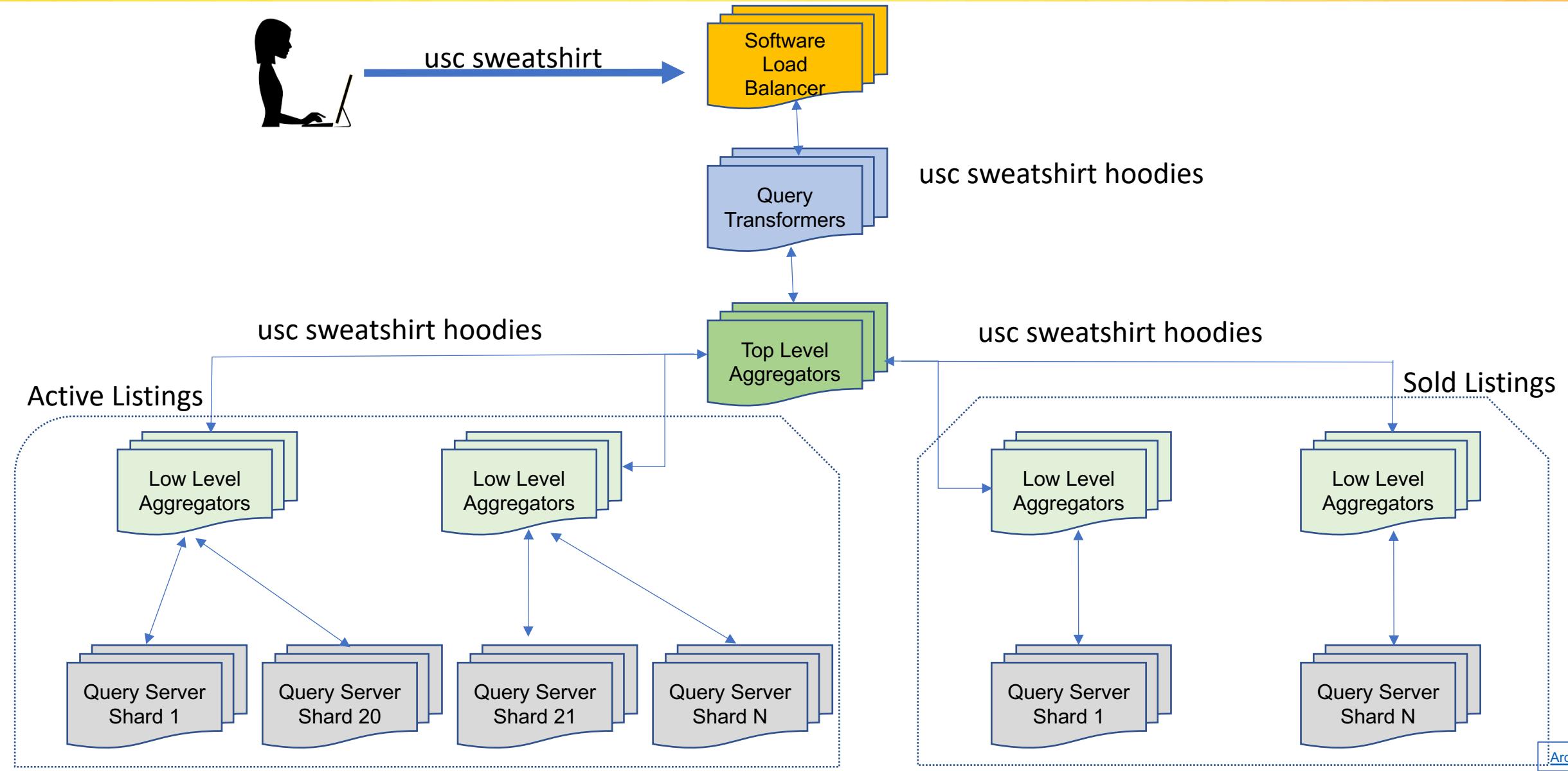
30-40%  
Data Center  
Footprint

100%

5 9's  
availability



# Architecture



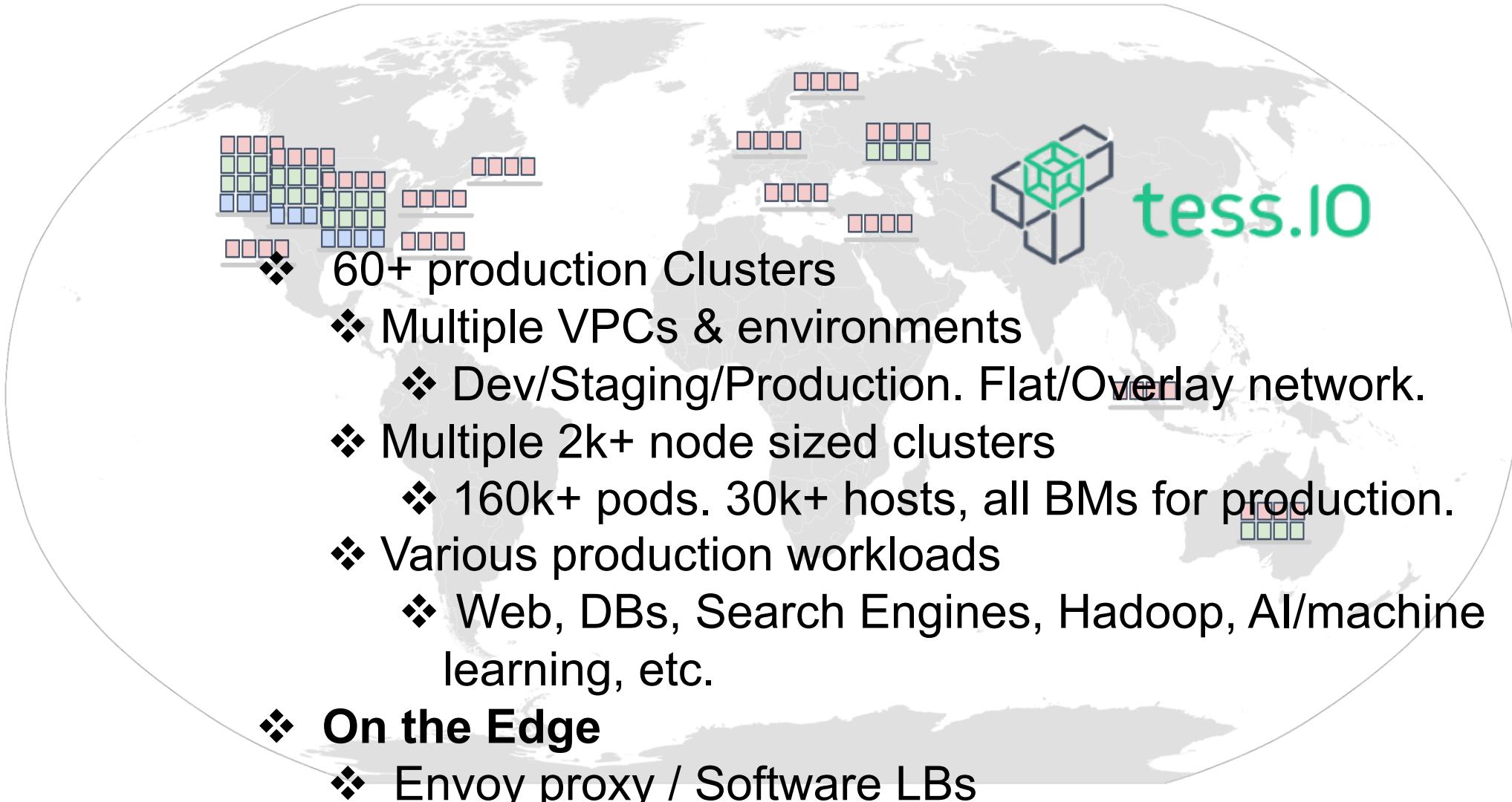
# eBay K8s Footprint



KubeCon

CloudNativeCon

North America 2019



ebay

# The Why ?

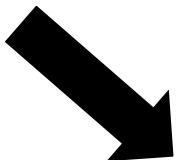


KubeCon

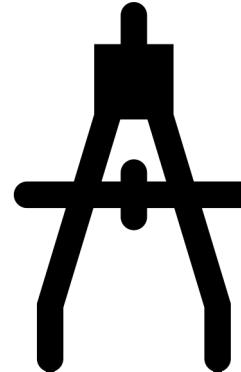
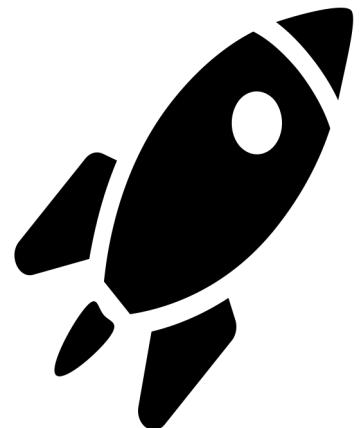


CloudNativeCon

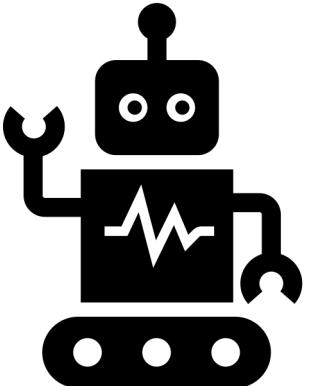
North America 2019



Speed



Scale

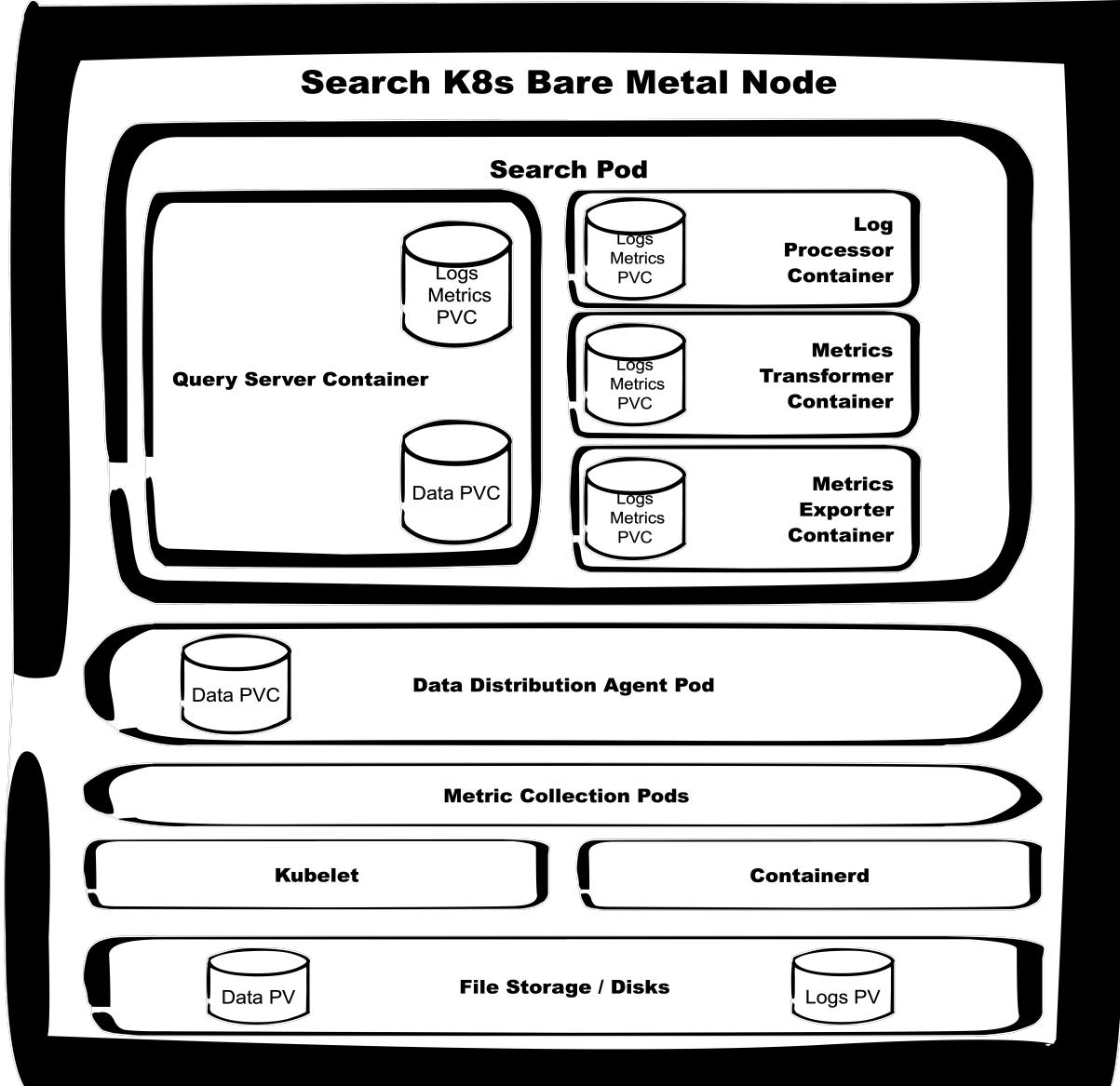


Automate

Flexible

ebay

# Search Node View on K8s



- Query Serving Pod.
  - Main query server container.
  - Log exporter.
  - Metric exporter.
- Data Distribution Agent Pod.
- Metric Collection Pods.
- Local disk persistent volumes (PVs).

# Search Grid Deployment

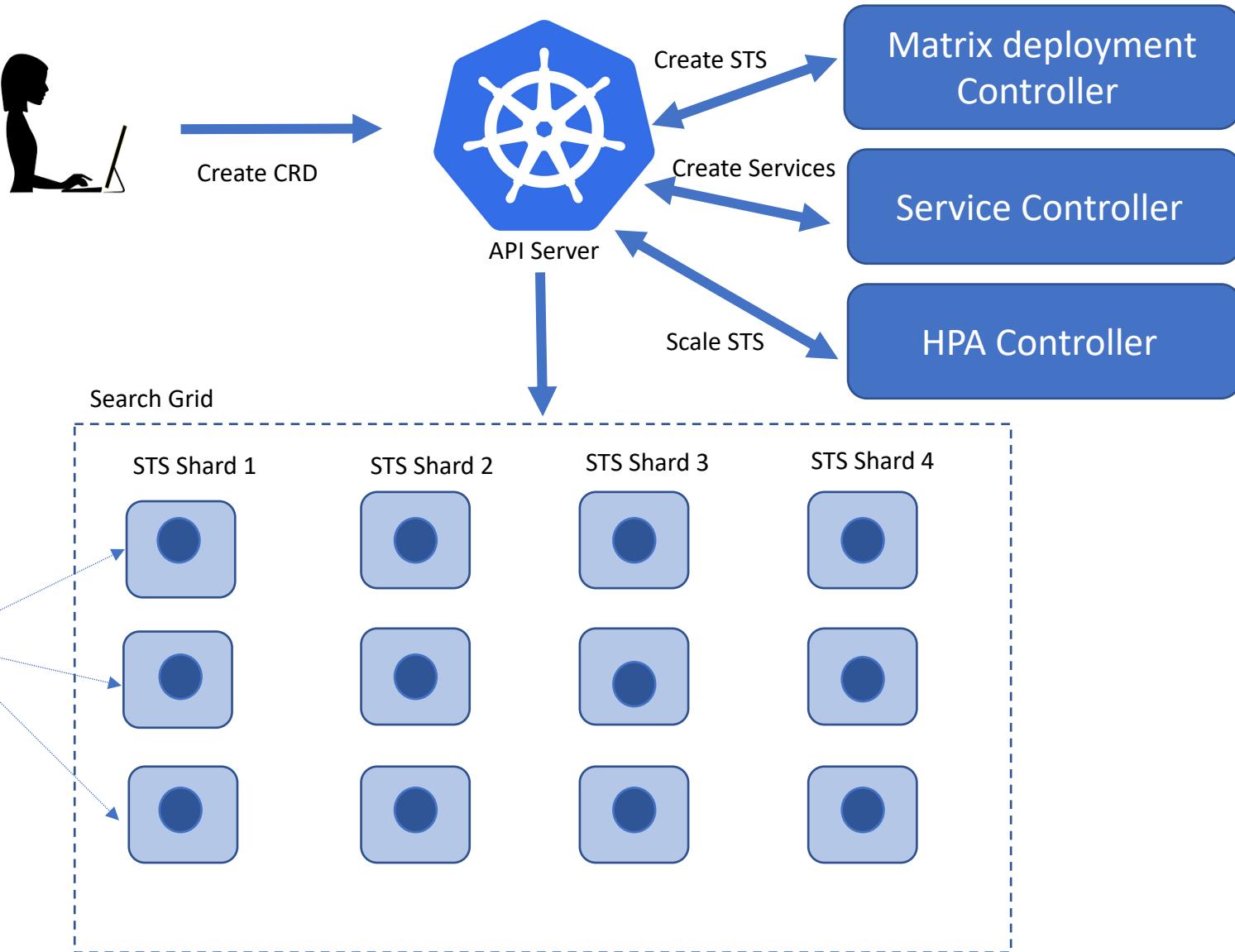


KubeCon



CloudNativeCon

North America 2019



```
apiVersion: deployment.com.ebay.cassini.tess.controllers/v1alpha1
kind: MatrixDeployment
metadata:
  name: preprod-completed-1-qry
  namespace: cassini
spec:
  columns: 4
  rows: 3
  component: qry
  usecase: completed
  containerVersions:
    logpusher: 3.1.4.7
    monitor-exporter: v1123
    query_server: 10.15.1997
  dataPackages:
    - buildTime: "201910100700"
      source: file:///inverted_index_folder
      type: inverted_index
      version: 9.3.1995
    - buildTime: "201910081418"
      sourceColo: file:///models_folder
      type: models
      version: 9.5.1977
      realm: preprod
      usecase: completed
  status:
    columns: 4
    rows: 2
    containerVersions:
      logpusher: 3.1.4.7
      monitor-exporter: v1123
      query_server: 10.15.1997
    dataPackageVersions:
      models: 9.5.1977
      inverted_index: 9.3.1995
```



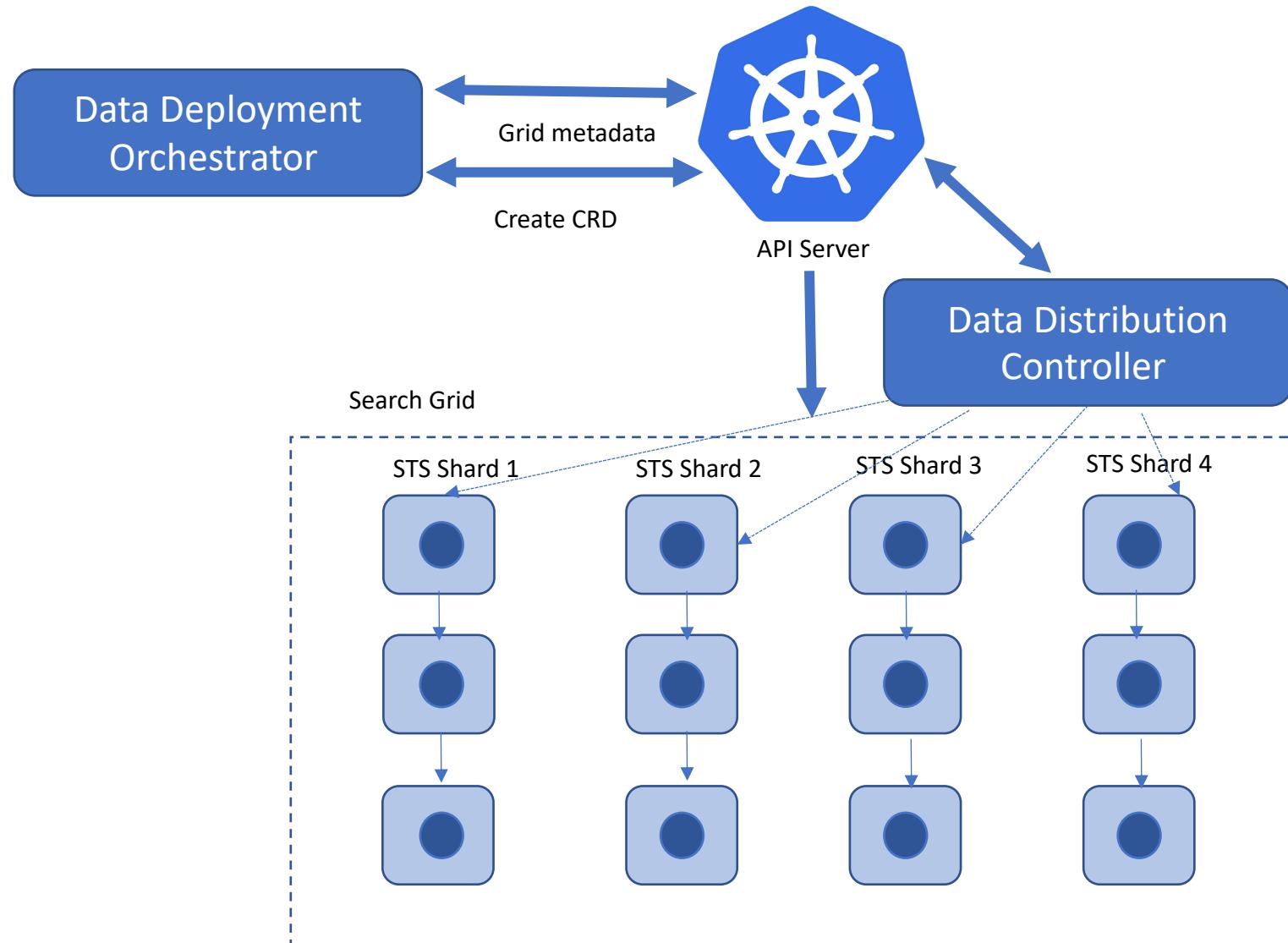
KubeCon



CloudNativeCon

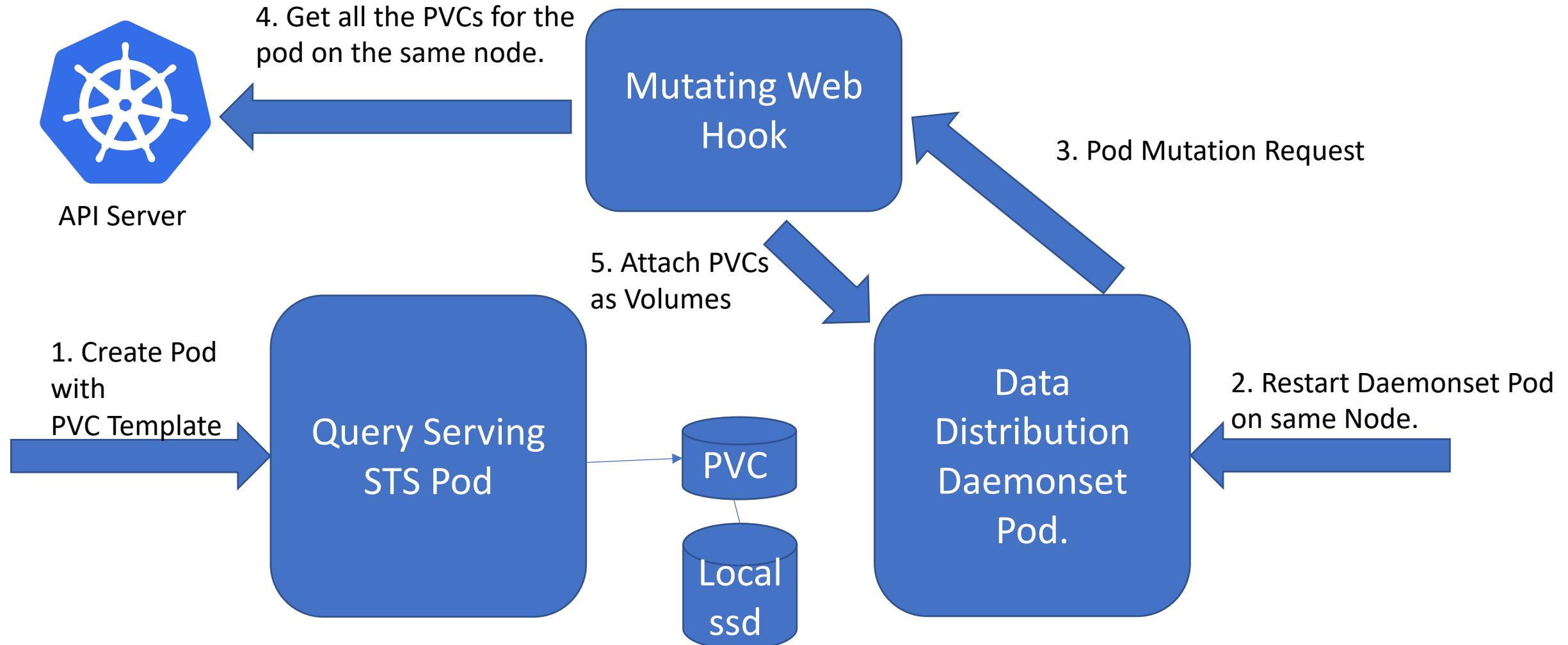
North America 2019

# Data Distribution



```
apiVersion: id.com.ebay.cassini.tess.controllers/v1alpha1
kind: FileDistribution
metadata:
  name: fd-col-2-models-2019032600
  namespace: cassini
spec:
  buildTime: "2019032600"
  indexType: models
  maxRetries: 10
  pool: preprod-completed-1-qry
  releaseVersion: 9.5.1977
  request:
    "1":
      filePath: file:///models_folder
      id: 1
      name: models_1.data
    "2":
      filePath: file:///models_folder
      id: 2
      name: models_2.data
  targetFilepath: /tmp/
  useCase: completed
status:
  state: COMPLETED
  downloadStatuses:
    "1":
      columnStatus:
        - name: 2019032600-1-0
          nodeURL: http://1.1.1.1:8000/537d13ff-d56a-4e57-a05a-5e21b2a71db1
          nodes:
            1.1.1.1:
              inRateMbps: 354
              outRateMbps: 342
              status: COMPLETED
              untarStatus:
                msg: Package models_1.data successfully downloaded.
                opId: 55fdea9d-816f-43a0-8012-c60e20c2af69
            1.1.1.2:
              inRateMbps: 342
              outRateMbps: 356
              status: COMPLETED
              untarStatus:
                msg: Package models_1.data successfully downloaded.
                opId: e7d9539b-a6f7-4145-ab25-82f504628cb6
        progress: 100
```

# Data sharing between Pods



# Out of the Box Performance



KubeCon



CloudNativeCon

North America 2019

K8s Pod



- At 18-20% CPU – 3.2K QPS

Bare Metal



- At 18-20% CPU – 3.6K QPS

**ebay**

# What moved the curve?



- **Kernel**
  - Latest kernel on K8s nodes.
- **CPU & Power**
  - Tuned p-state and c-state to leverage turbo boost.
- **Networking**
  - **Ipvlan**
    - Ipvlan for high performance.

# Performance Optimizations



KubeCon

CloudNativeCon

North America 2019

K8s Pod



- At 78-80% CPU – 9.5K QPS

Bare Metal



- At 78-80% CPU – 9.5K QPS

ebay

# Lessons Learned

- Breaking a monolithic application into independent micro services is difficult.
- Keeping operational migration minimal at this stage is more important.
- Design choice of having data distribution pod run as a Daemonset instead of a side-car posed challenges that could have been avoided.
- Node Remediation with Local PVC not yet fully ironed out.
- Performance optimizations for low latency applications.

# Future Work

- Move to max unavailable update strategy for STS.
- Volume Cloning.
- Node Remediation with Local PVCs.
- Multi cluster support.
- Leverage pod priority and preemption.

# Conclusion

Run a latency sensitive, large scale stateful application on K8s along with agility, flexibility and automation using K8s framework with minimal performance impact.