



# Centralizing Kubernetes and Container Operations

Oleg Chunikhin | CTO, Kublr



# Introductions



**Oleg Chunikhin**  
**CTO, Kublr**

- ✓ **20 years** in software architecture & development
- ✓ Working w/ Kubernetes **since its release** in 2015
- ✓ **Software architect behind Kublr**—an enterprise ready container management platform
- ✓ Twitter **@olgch**



# History

- Custom software development company
- Dozens of projects per year
- **Varying target environments**: clouds, on-prem, hybrid
- Recurring need for unified application **delivery and ops platform** w/ monitoring, logs, security, multiple env, ...



# Docker and Kubernetes to the Rescue

- **Docker** is great, but local
- **Kubernetes** is great... when it is up and running
- Who **sets up** and **operates** K8S clusters?
- Who takes care of **operational aspects at scale**?
- How do you provide **governance** and ensure **compliance**?



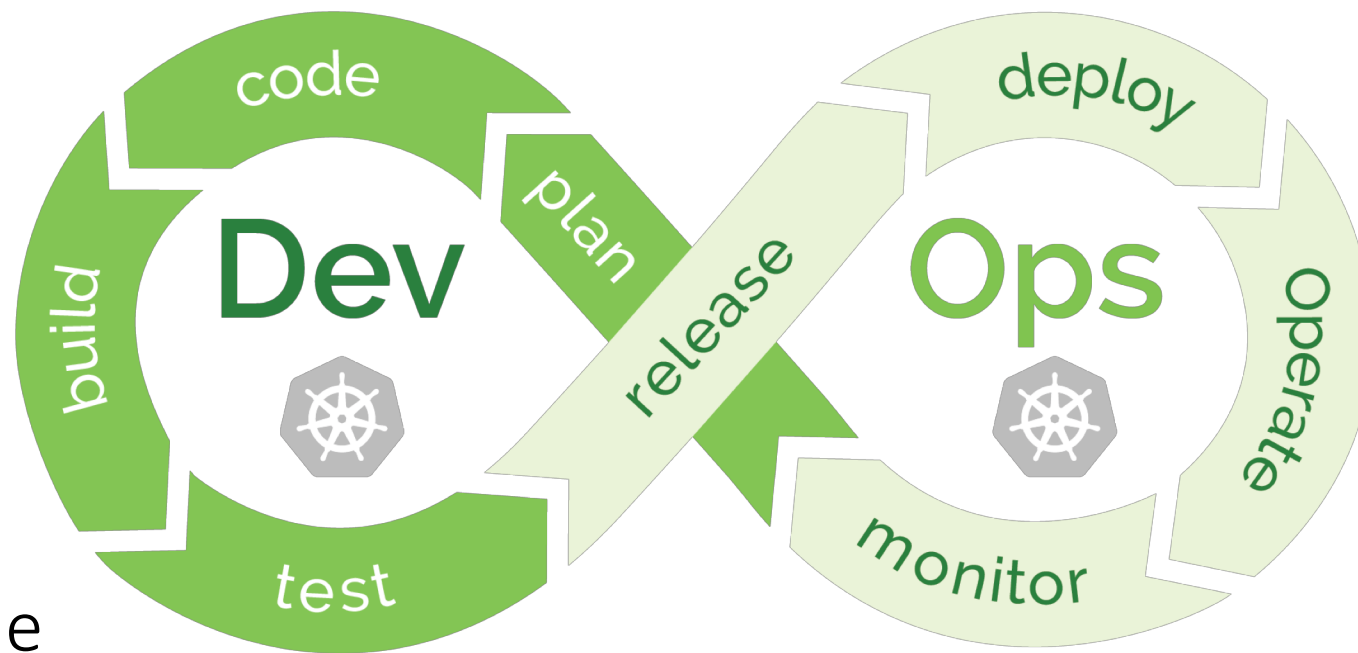




# Enterprise Kubernetes Needs

## Developers

- Self-service
- Compatible
- Conformant
- Configurable
- Open & Flexible



- Security
- Reliability
- Performance
- Portability

## SRE/Ops/DevOps/SecOps

- Org multi-tenancy
- Single pane of glass
- Operations
- Monitoring
- Log collection
- Image management
- Identity management

@olgch; @kublr



# Kubernetes Management Platform Wanted

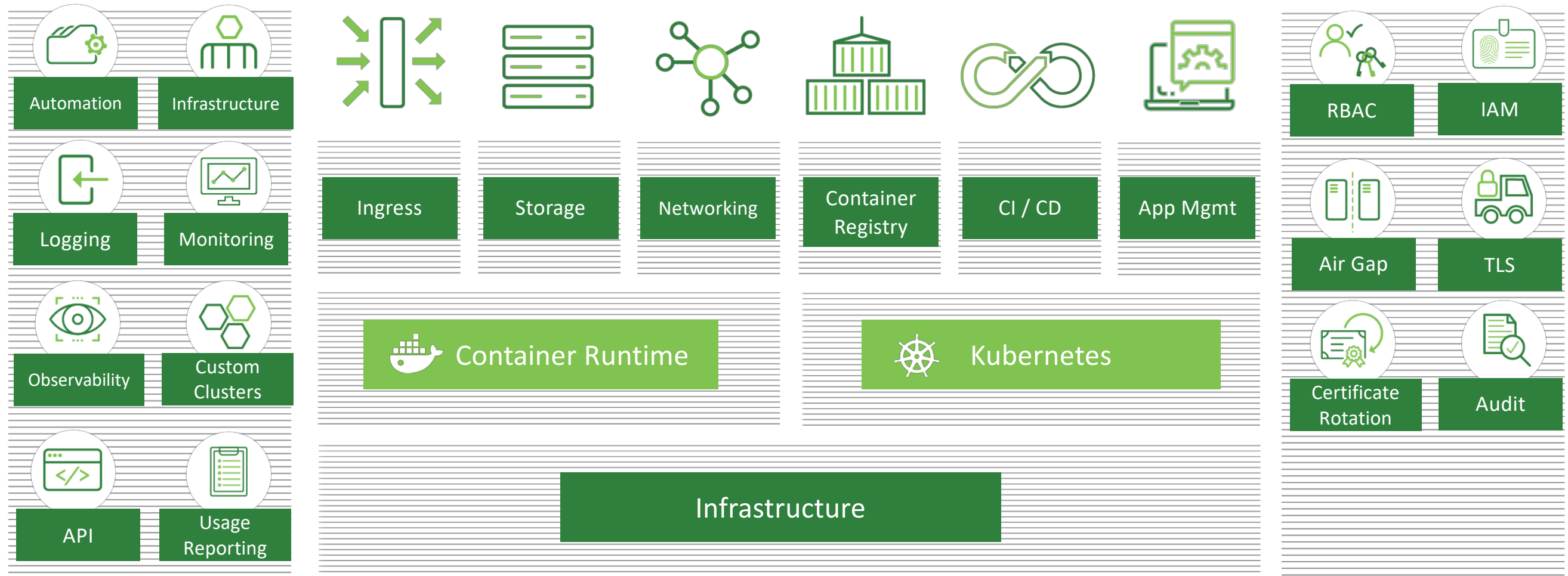
- **Portability** – clouds, on-prem, hybrid, air-gapped, different OS'
- **Centralized multi-cluster** operations saves resources – many environments (dev, prod, QA, ...), teams, applications
- **Self-service** and **governance** for Kubernetes operations
- **Reliability** – cluster self-healing, self-reliance
- **Limited management profile** – cloud and K8S API
- **Architecture** – flexible, open, pluggable, compatible
- **Sturdy** – secure, scalable, modular, HA, DR etc.

@olgch; @kublr



## OPERATIONS

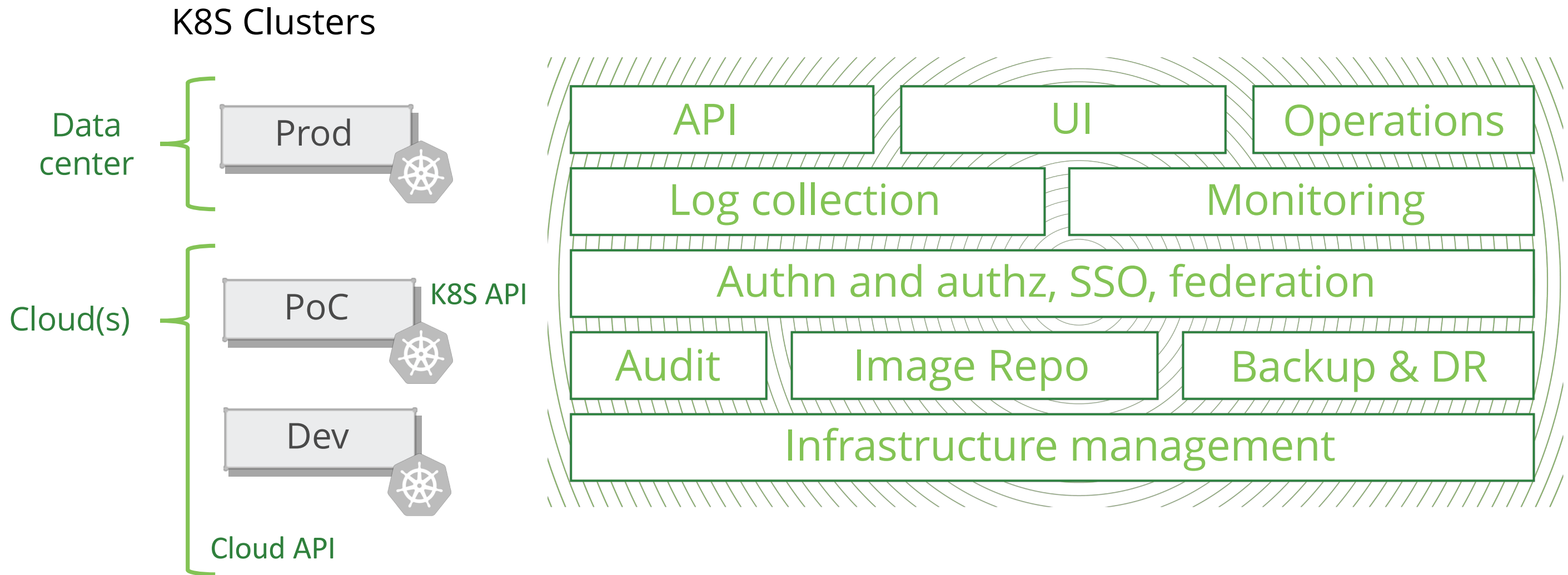
## SECURITY & GOVERNANCE



@olgch; @kublr




# Central Control Plane: Operations



@olgch; @kublr



# Central Control Plane: Operations

 **kublr**  
Version 1.9.1-ga2

BLOGDOCUMENTATIONadmin


Clusters


Clusters


Type Search


☒ Hide removed clusters


ADD CLUSTER


  
Credentials


  
Back Up

  
Documentation



  
Sys. Cluster



  
Centralized Monitoring


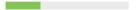
kublr-demo-01


 us-east-1


1 Master - 1 Work

 CPU  6% 0.25/4 cores



 Memory  51% 3.98/7.72 Gb



 Disk Space  28% 8.29/28.96 Gb



kublr-demo-02


 us-east-1


1 Master - 1 Work

 CPU  6% 0.26/4 cores



 Memory  50% 3.91/7.72 Gb



 Disk Space  29% 8.41/28.96 Gb


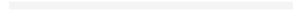
cluster-1526500060


 DOKU-METAL

1 Master - 2 Work

 CPU  -- --

 Memory  -- --

 Disk Space  -- --

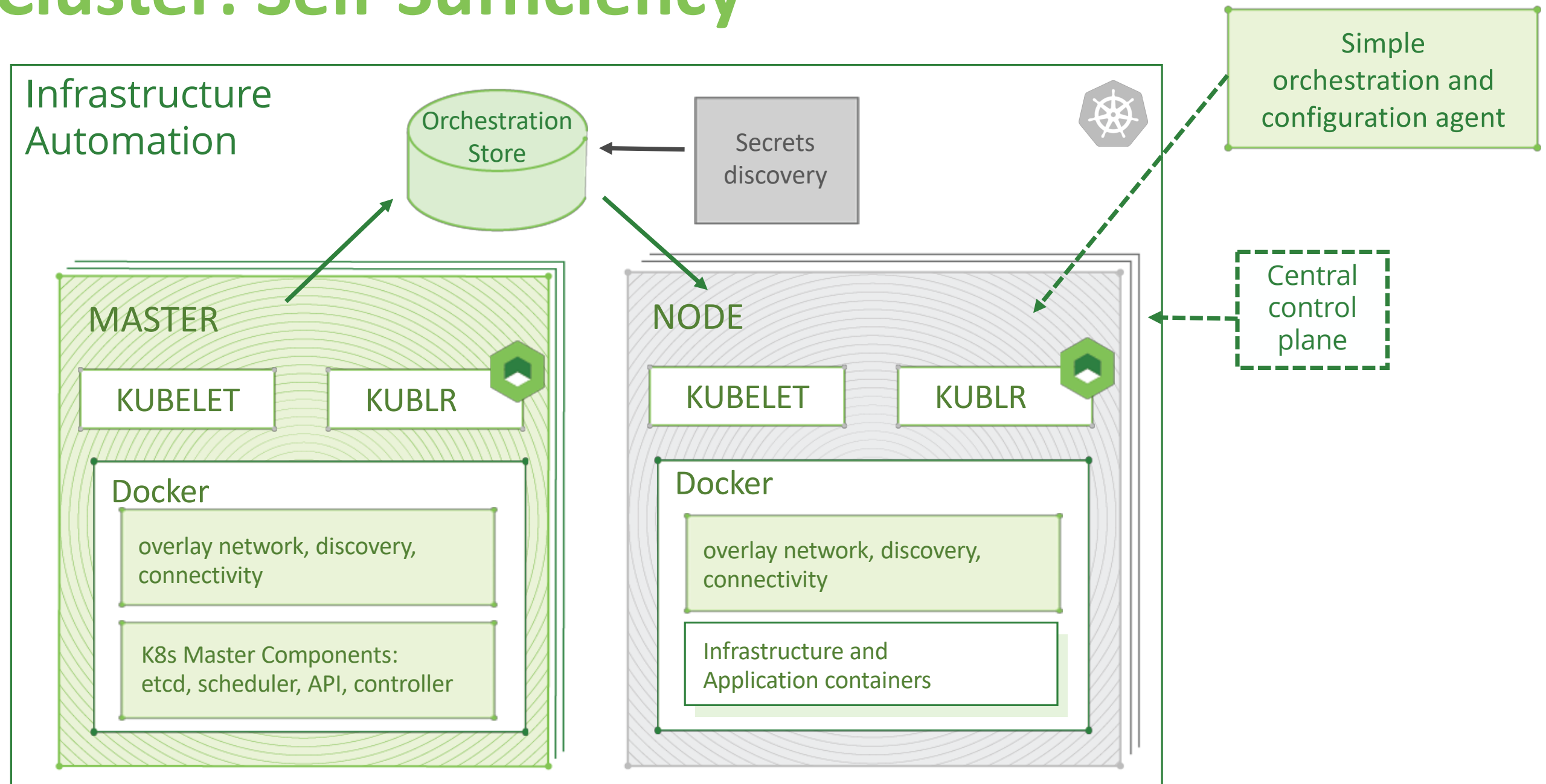


@olgch; @kublr





# Cluster: Self-Sufficiency

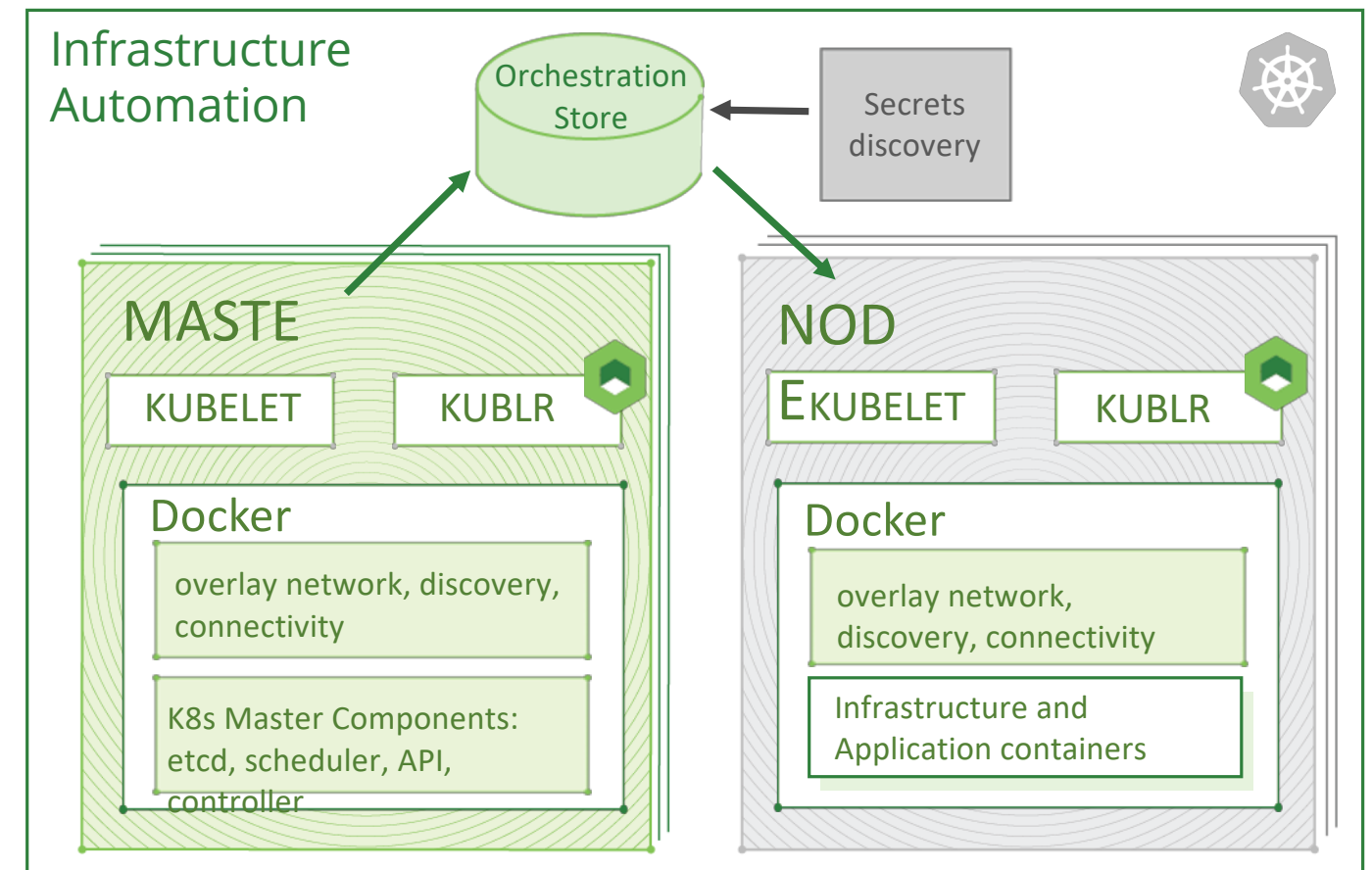


@olgch; @kublr



# Cluster: Portability

- (Almost) everything runs in **containers**
- Simple (single-binary) **management agent**
- Minimal **store** requirements
  - Shared, eventually consistent
  - Secure: RW files for masters, RO for nodes
  - Thus the store can be anything: S3, SA, NFS, rsynced dir, provided files, ...
- Minimal **infra automation** requirements
  - Configure and run configuration agent
  - Enable access to the store
  - Can be AWS CF, Azure ARM, BOSH, Ansible, ...
- **Load balancer is not required** for multi-master; each agent can independently fail over to a healthy master

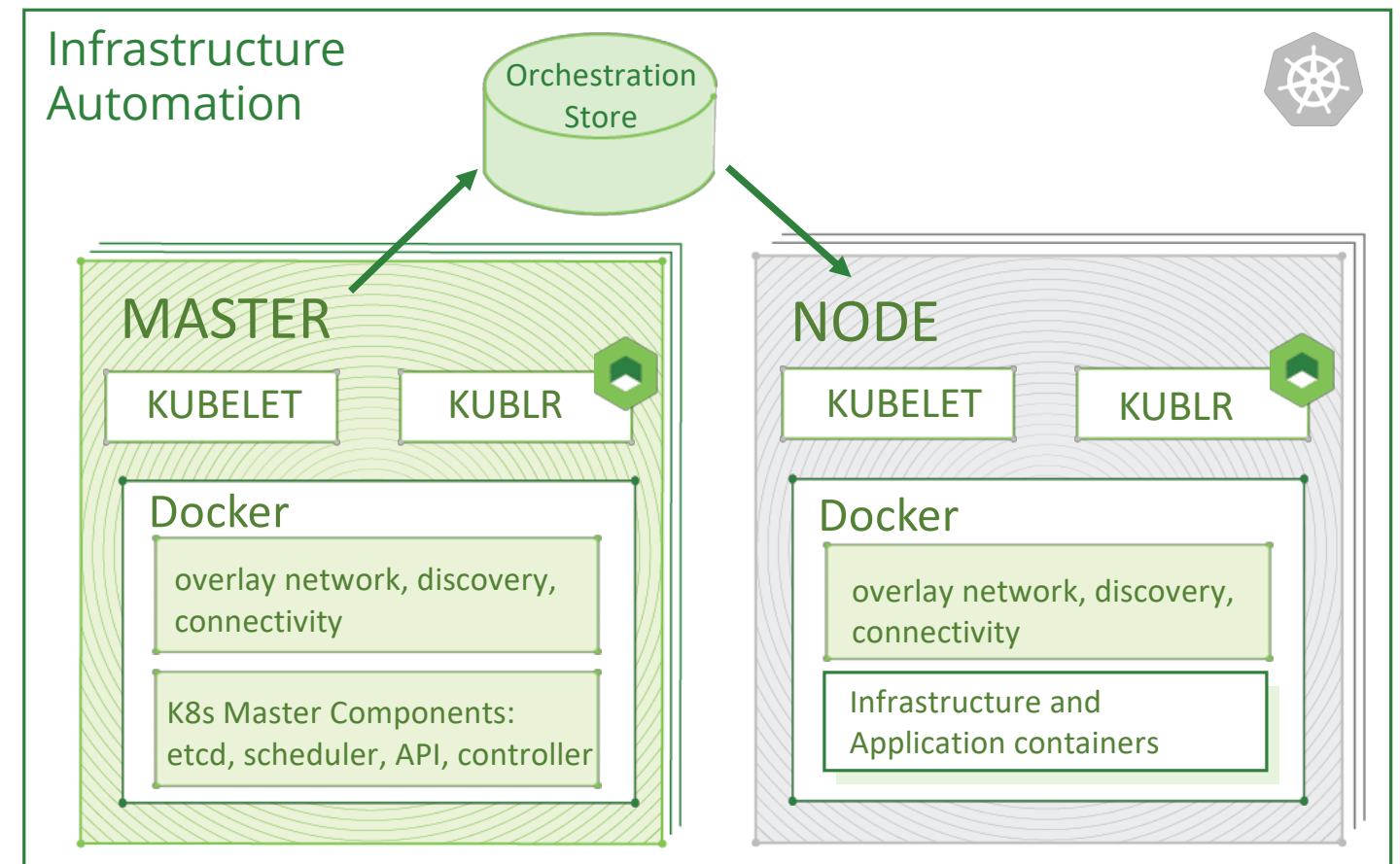


@olgch; @kublr



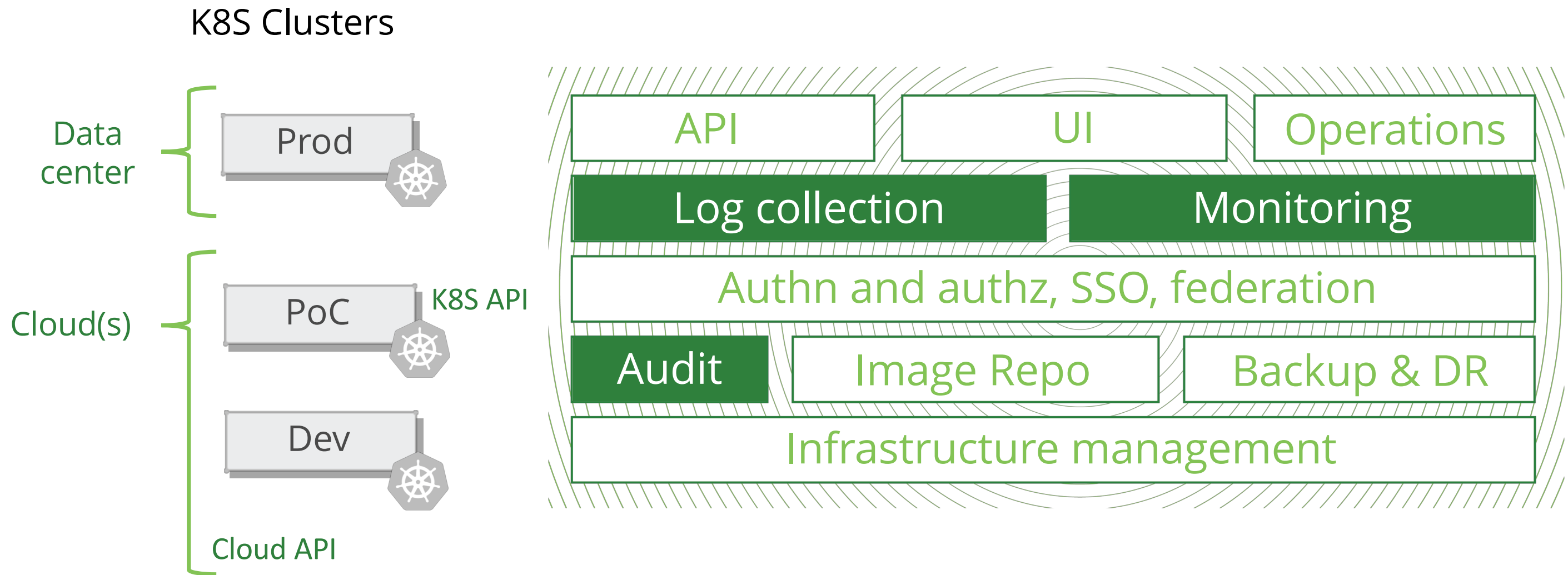
# Cluster: Reliability

- Rely on underlying platform as much as possible
  - ASG on AWS
  - IAM on AWS for store access
  - SA on Azure, S3 on AWS
  - ARM on Azure, CF on AWS
- Minimal infrastructure SLA tolerate temporary failures
- Multi-master API failover on nodes
- Resource management, memory requests and limits for OS and k8s components





# Central Control Plane: Logs and Metrics





# Centralized Monitoring and Log Collection. Why Bother?

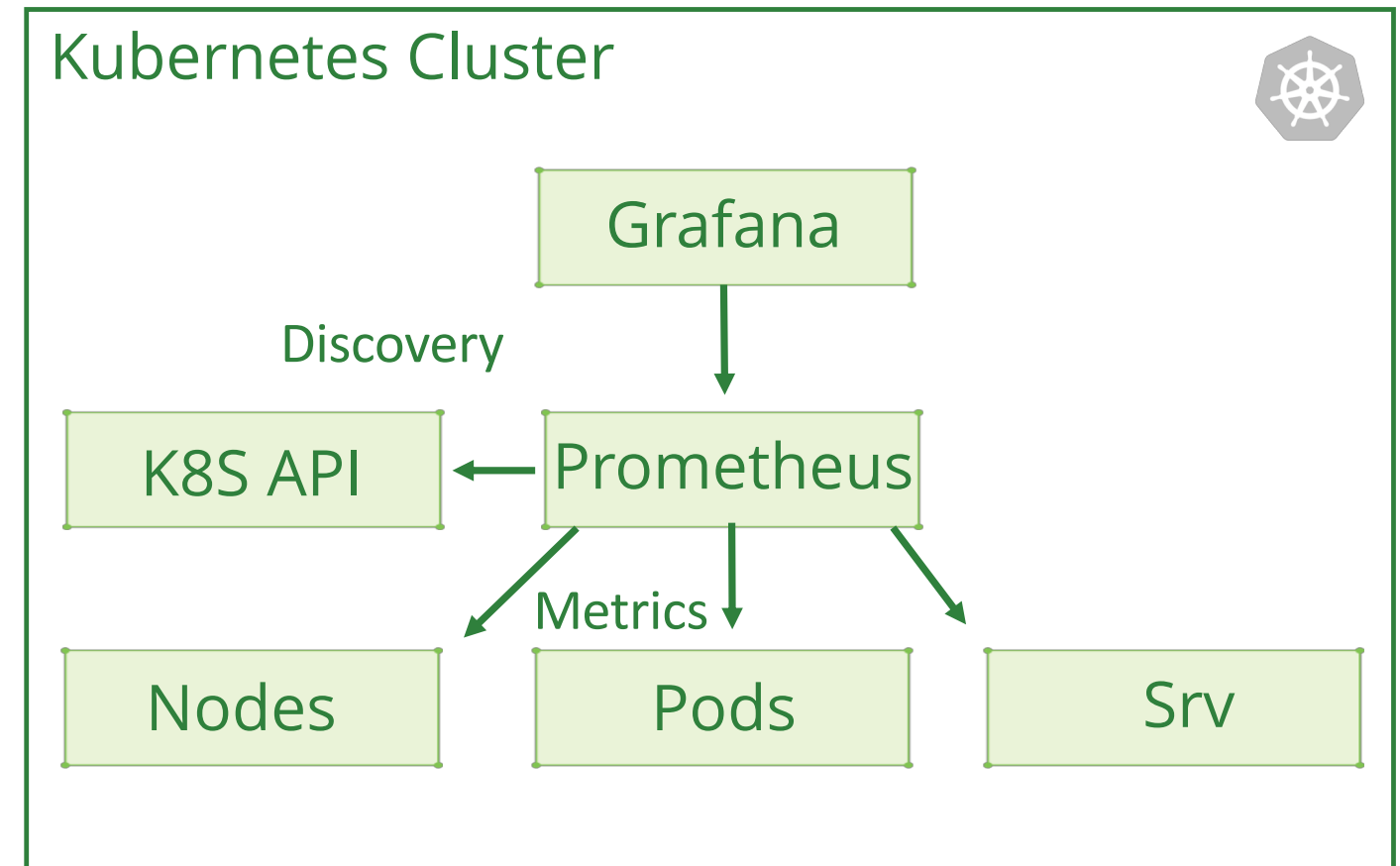
- Prometheus and ELK are heavy and not easy to operate; need attention and at least 4-8 Gb RAM... each, per cluster
- Cloud/SaaS monitoring is not always permitted or available
- Existing monitoring is often not container-aware
- No aggregated view and analysis
- No alerting governance





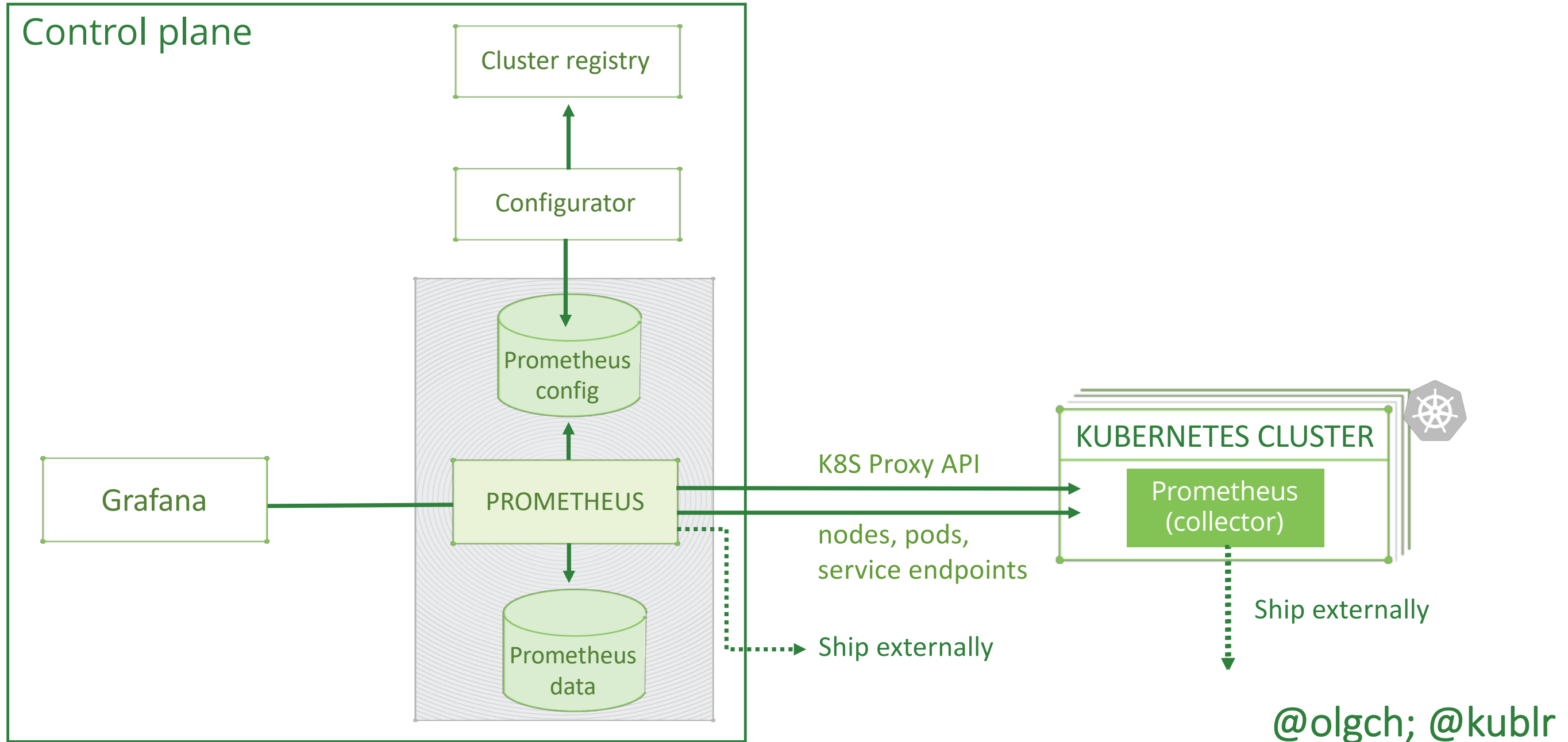
# K8S Monitoring with Prometheus

- Discover nodes, services, pods via K8S API
- Query metrics from discovered endpoints
- Endpoint are accessed directly via internal cluster addresses





# Centralized Monitoring



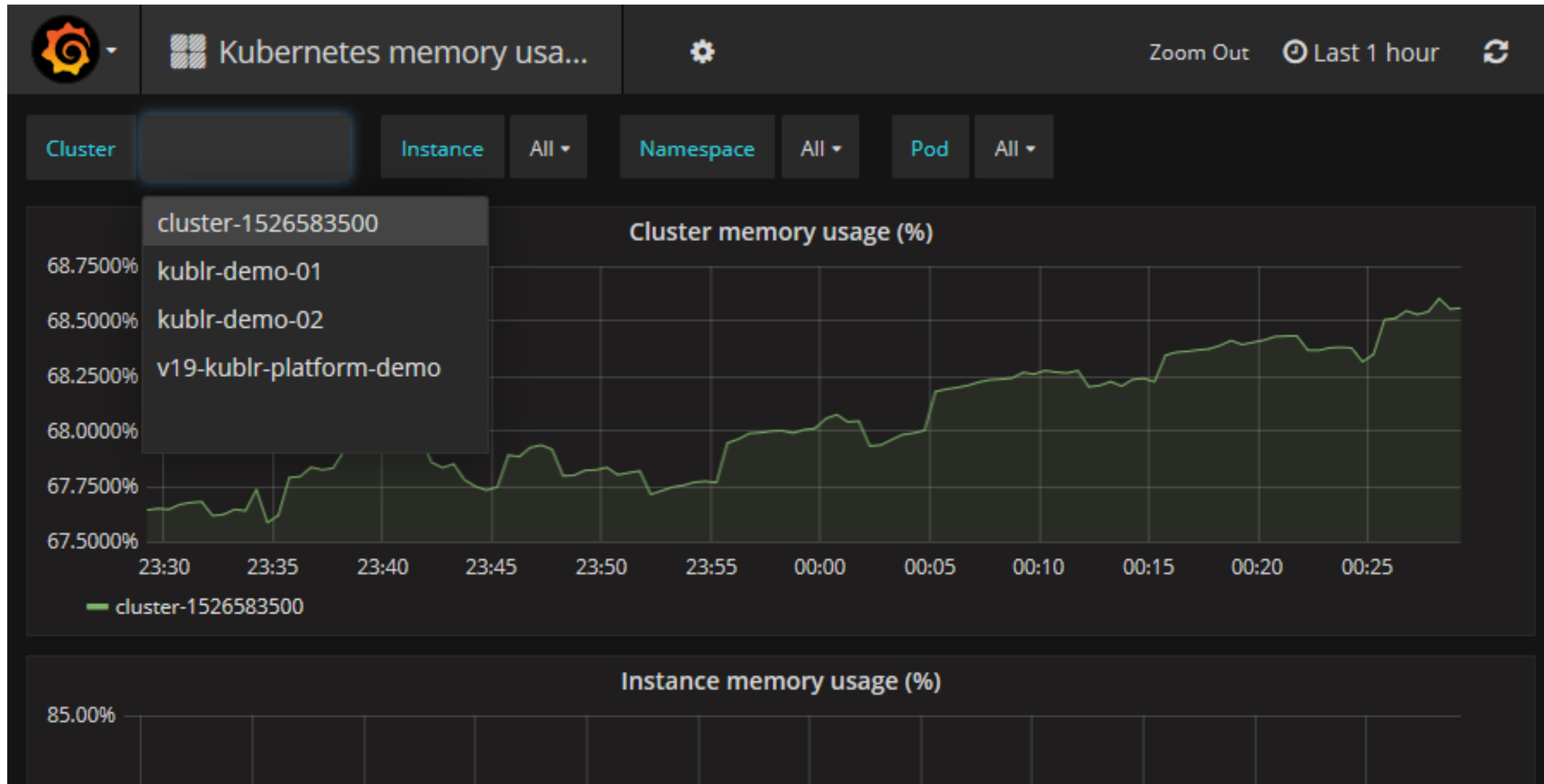


# Centralized Monitoring: Considerations

- Prometheus resource usage tuning
- Long-term storage (m3)
- Configuration file growth with many clusters
- Metrics labeling
- Additional load on API server



# Centralized Monitoring

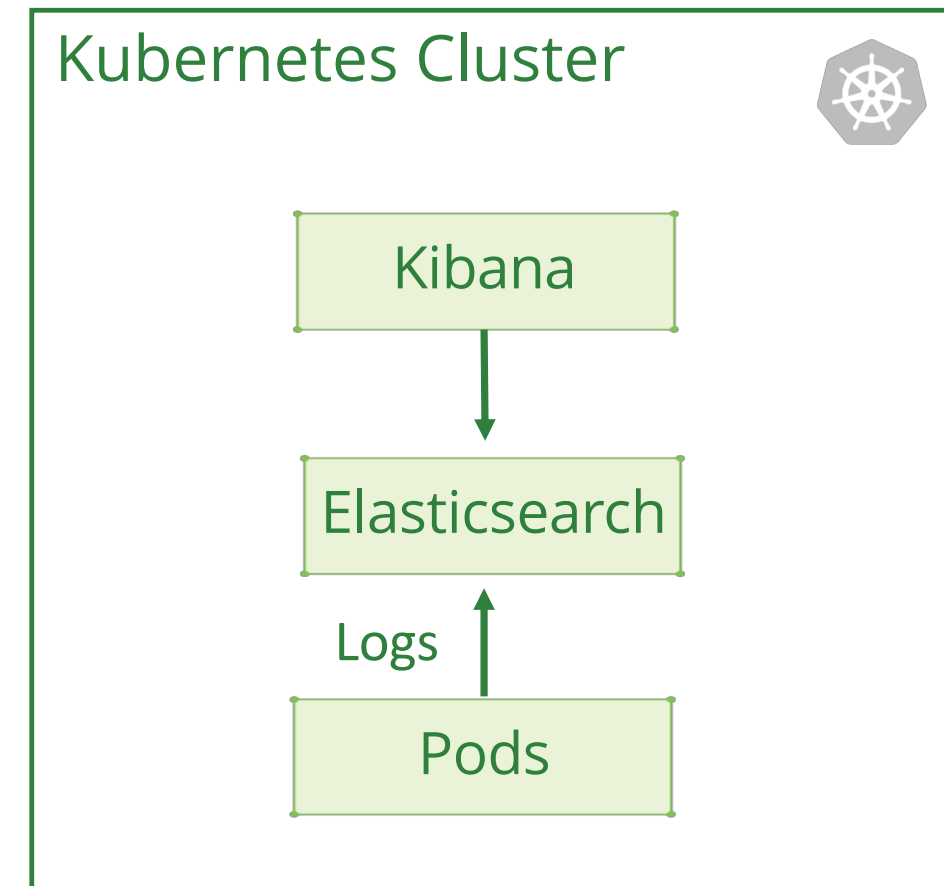


@olgch; @kublr



# K8S Logging with Elasticsearch

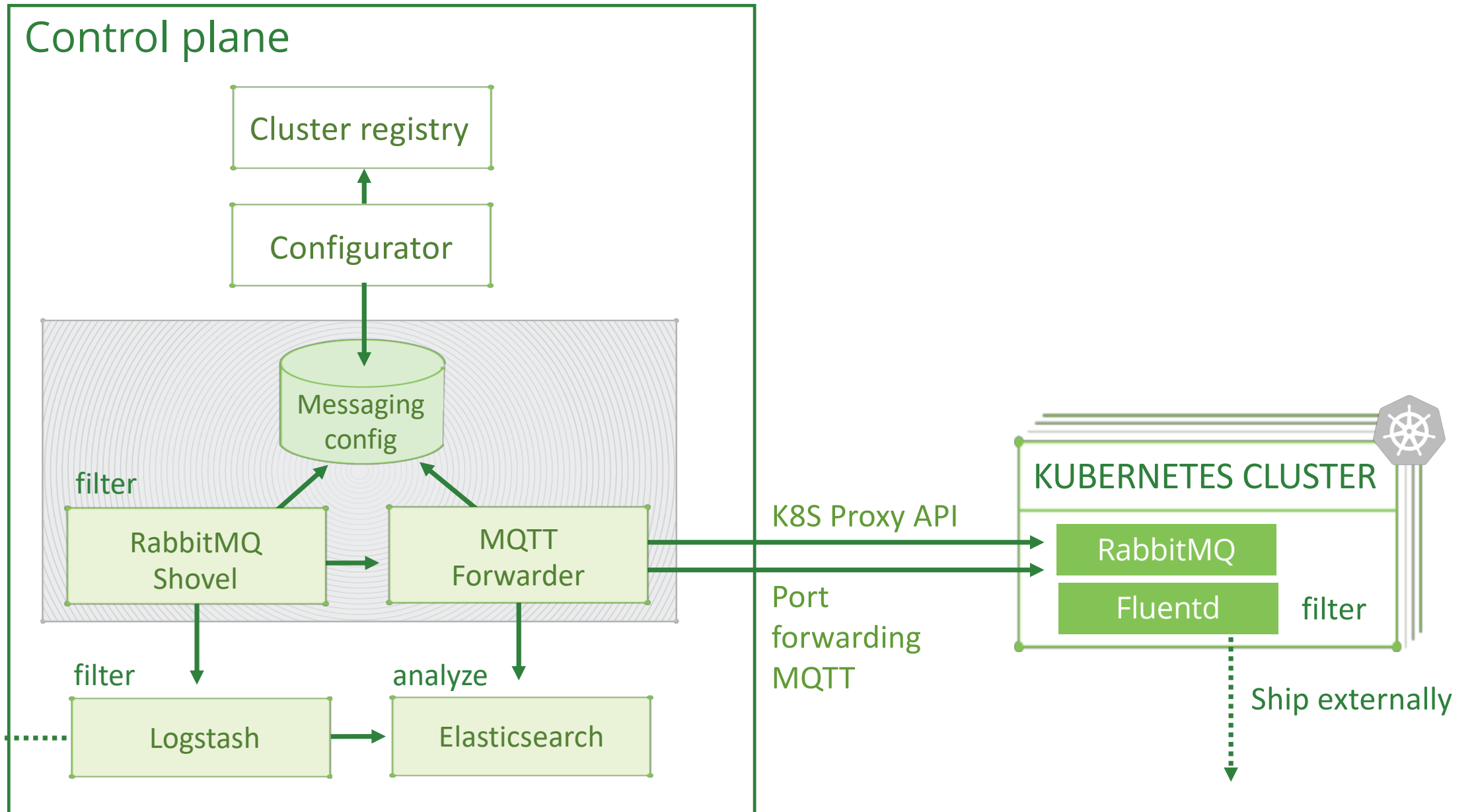
- Fluentd runs on nodes
- OS, K8S, and container logs collected and shipped to Elasticsearch
- Kibana for visualization







# Centralized Log Collection



@olgch; @kublr



# Centralized Log Collection: Considerations

- Tune Elasticsearch resource usage
- Take into account additional load on API server
- Log index structure normalization

```
{
  "data": {
    "elasticsearch": {
      "version": "6.x"
    }
  }
}
```

```
{
  "flatData": [
    {
      "key": "elasticsearch.version",
      "type": "string",
      "key_type": "elasticsearch.version.string",
      "value_string": "6.x"
    },
    ...
  ]
}
```



# The Rest: Considerations

- **Identity management**  
Use Identity Broker (e.g. KeyCloak): Users, Authn, Autzn, SSO, RBAC, Federation, ...
- **Backup and disaster recovery**  
K8s metadata + app data/volumes: full cluster recovery or copy
- **Docker image management**  
Docker image registry (e.g. Nexus, Artifactory, Docker Hub);  
image scanning;  
air-gapped or isolated environment: image registries proxying and caching,  
“system” images



# Q&A

Take Kublr for a test drive!  
[kublr.com/deploy](https://kublr.com/deploy)

Free non-production license.

@olgch; @kublr





**Oleg Chunikhin**

Chief Technology Officer

oleg@kublr.com

@olgch

**Stay in touch! Signup for our  
newsletter at [kublr.com](https://kublr.com)**

Kublr | [kublr.com](https://kublr.com)

@kublr