



# Kubernetes I Deep Dive

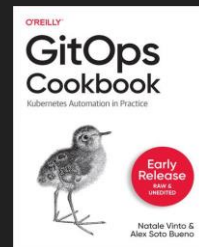
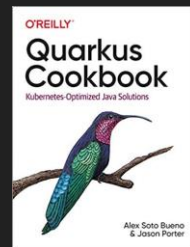
Homework [dn.dev/kube-tutorial](https://dn.dev/kube-tutorial)



# Alex Soto (lordofthejars.com)

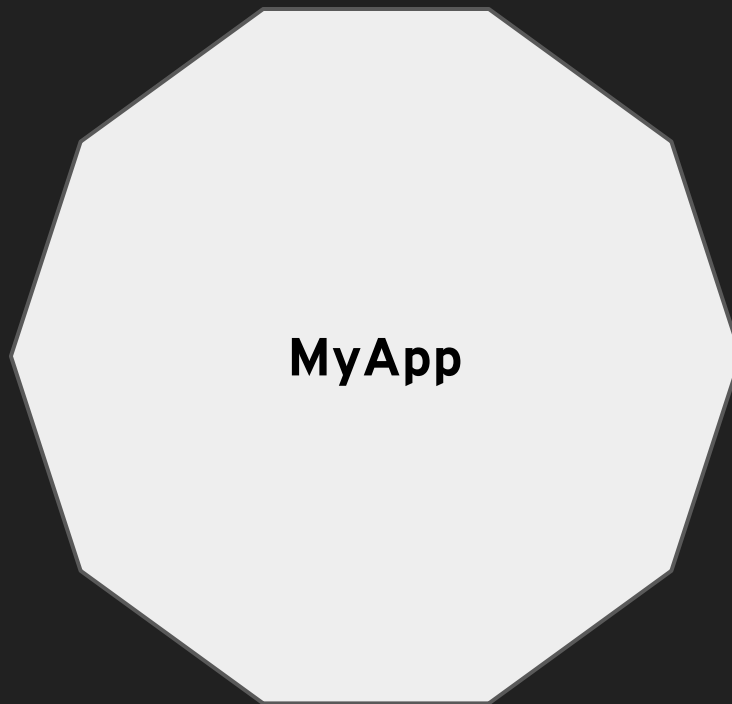


- @alexsotob
- asotobue@redhat.com
- Currently Red Hat's Director of Developer Experience
- Featured speaker at technology events around the globe
- A Java Champion since 2017
- Writer, University Professor, Radio collaborator
- A big fan of testing and continuous delivery in 21st century



# Why Kubernetes?

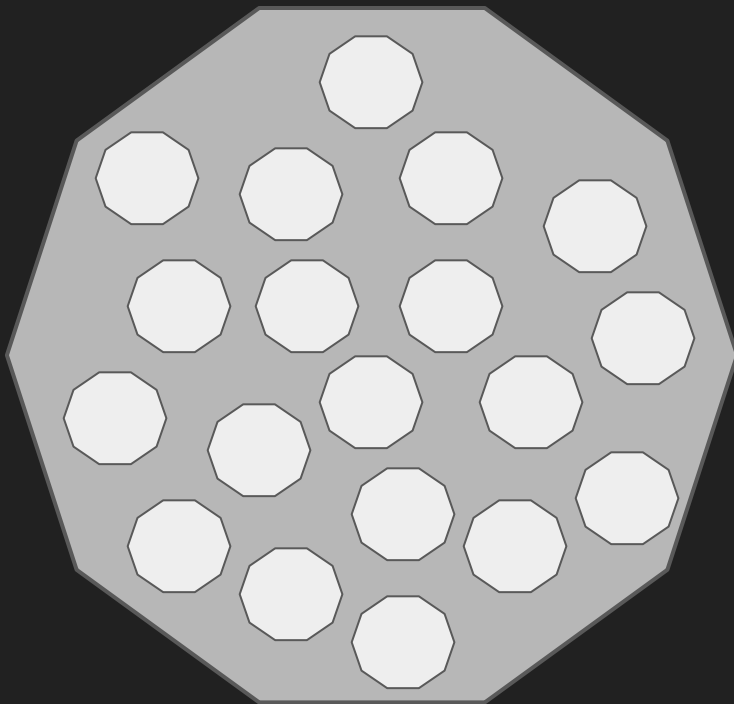
# Monolith



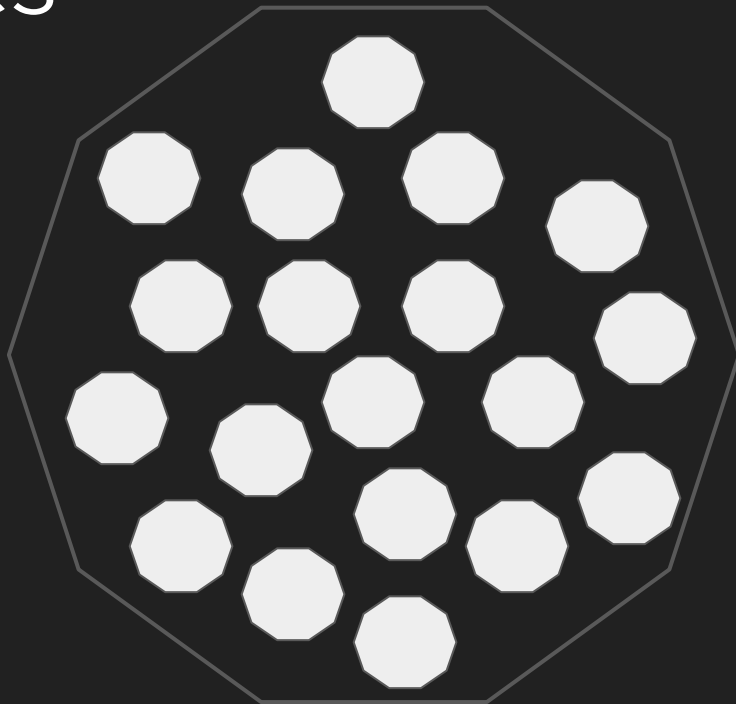
# Application Modules



Red Hat  
Developer

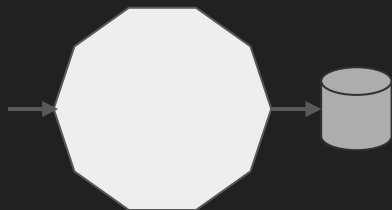


# Microservices

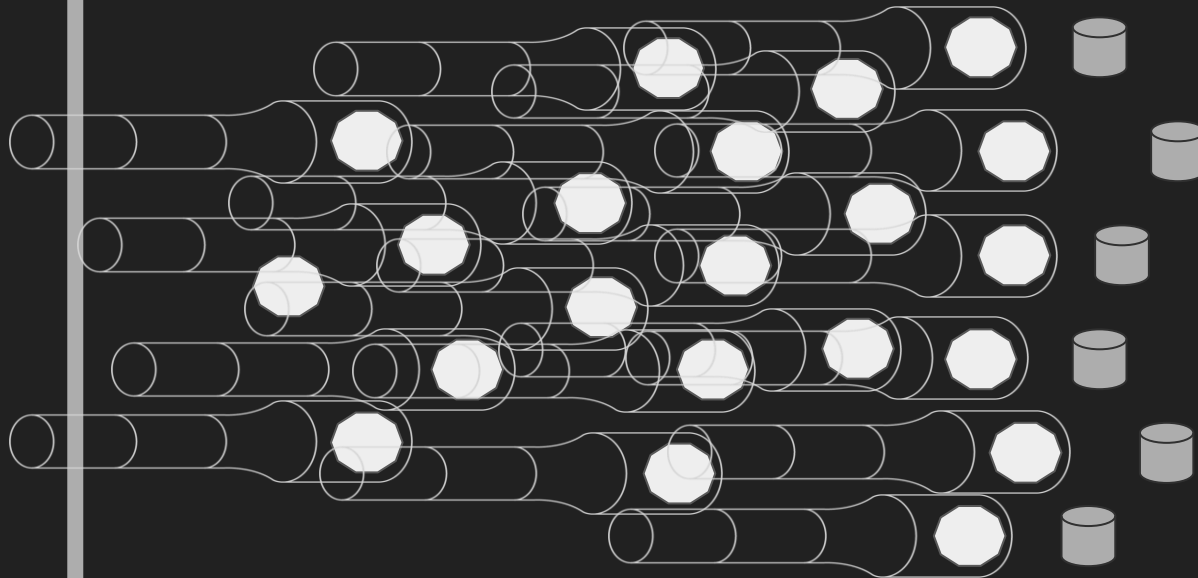


Every 3 months

Every 3 weeks/days/hours



Love Thy Mono



# Containers

## DEVELOPERS



- Cloud-native apps
- Simplified packaging
- Simplified testing

## IT OPERATIONS



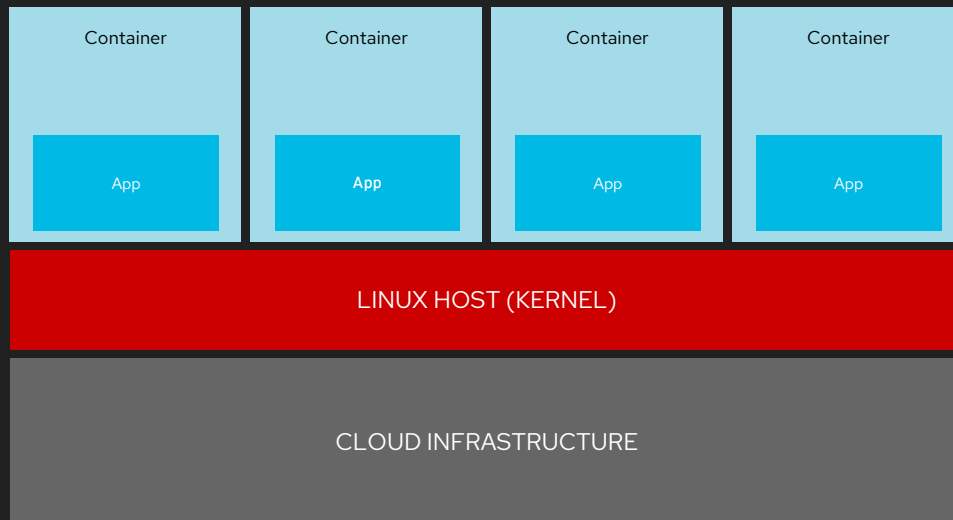
- Consistent app deploys
- Automated app deploys
- Improved app performance
- Multi-cloud consistency

## BUSINESS LEADER



- Enable devops culture
- Enable hybrid cloud
- Reduce vm licensing costs
- Accelerate app-dev cycles

## CONTAINERS



# DevOps Challenges for Multiple Containers

- How to scale?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
- How to keep them running?
- How to update them?
- Rebuild Container Images?



# Kubernetes

# Meet Kubernetes

- Greek for “Helmsman,” also the root of the word “Governor” (from latin: gubernator)
- Container orchestrator
- Supports multiple cloud and bare-metal environments
- Inspired by Google’s experience with containers
- Open source, written in Go
- Manage applications, not machines

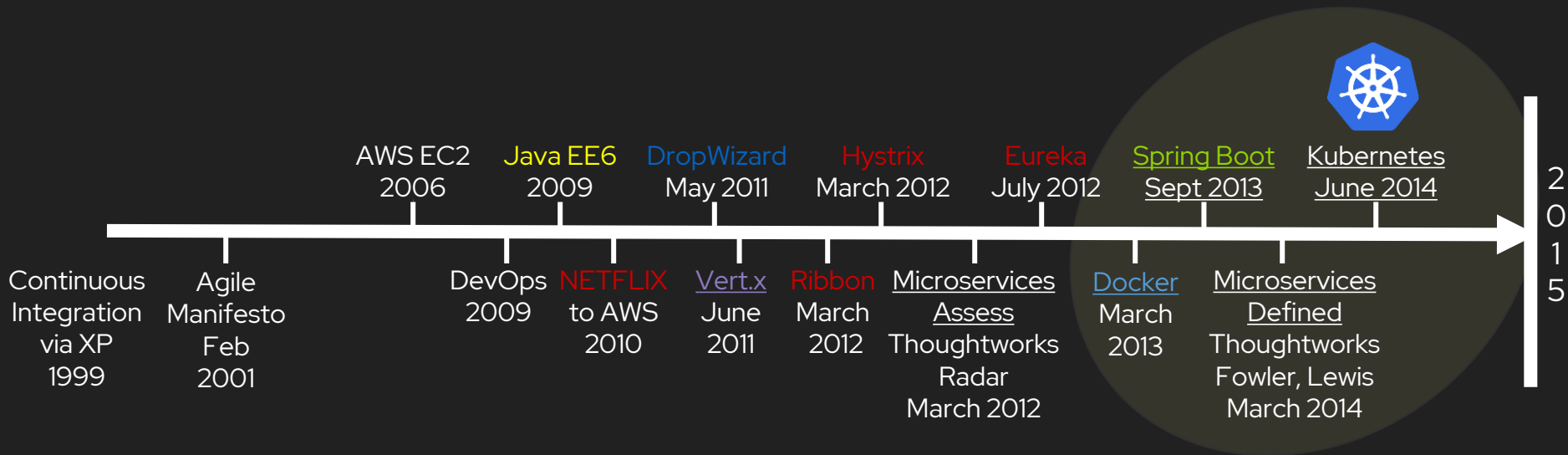


**OPENSIFT**

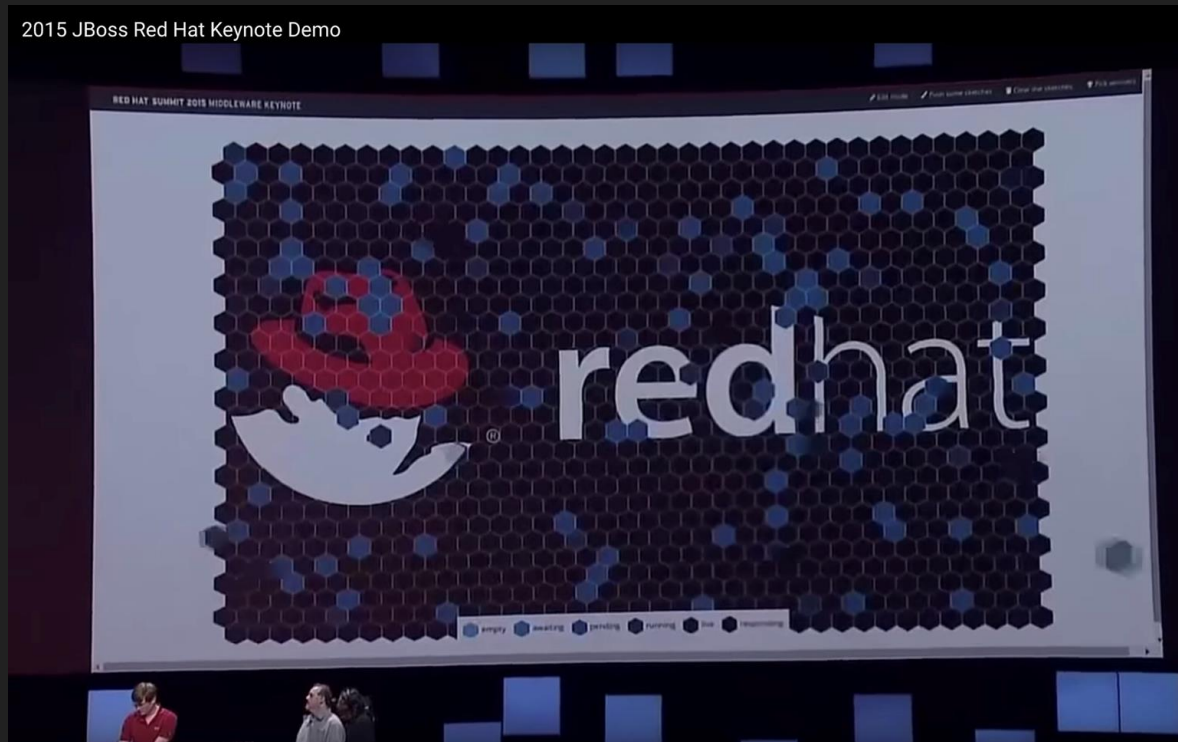


[Learn more](#)

# History of Microservices



2015 JBoss Red Hat Keynote Demo



# 2015

Launch 1000+  
Containers

Audience  
Claims a  
Container

<https://www.youtube.com/watch?v=GCtpncA0Ea0&feature=youtu.be&t=1031>

dn.dev/kubemaster1

# What is Kubernetes?

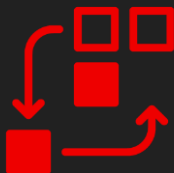


## Pod



- ✓ 1+ containers
- ✓ Shared IP
- ✓ Shared storage  
(ephemeral)
- ✓ Shared resources
- ✓ Shared life cycle

## ReplicaSet/ Deployment



- ✓ The desired state  
- replicas,  
pod template:  
health checks,  
resources,  
image

## Service



- ✓ Grouping of  
pods (acting as  
one) has stable  
virtual IP and  
DNS name

## Persistent Volume



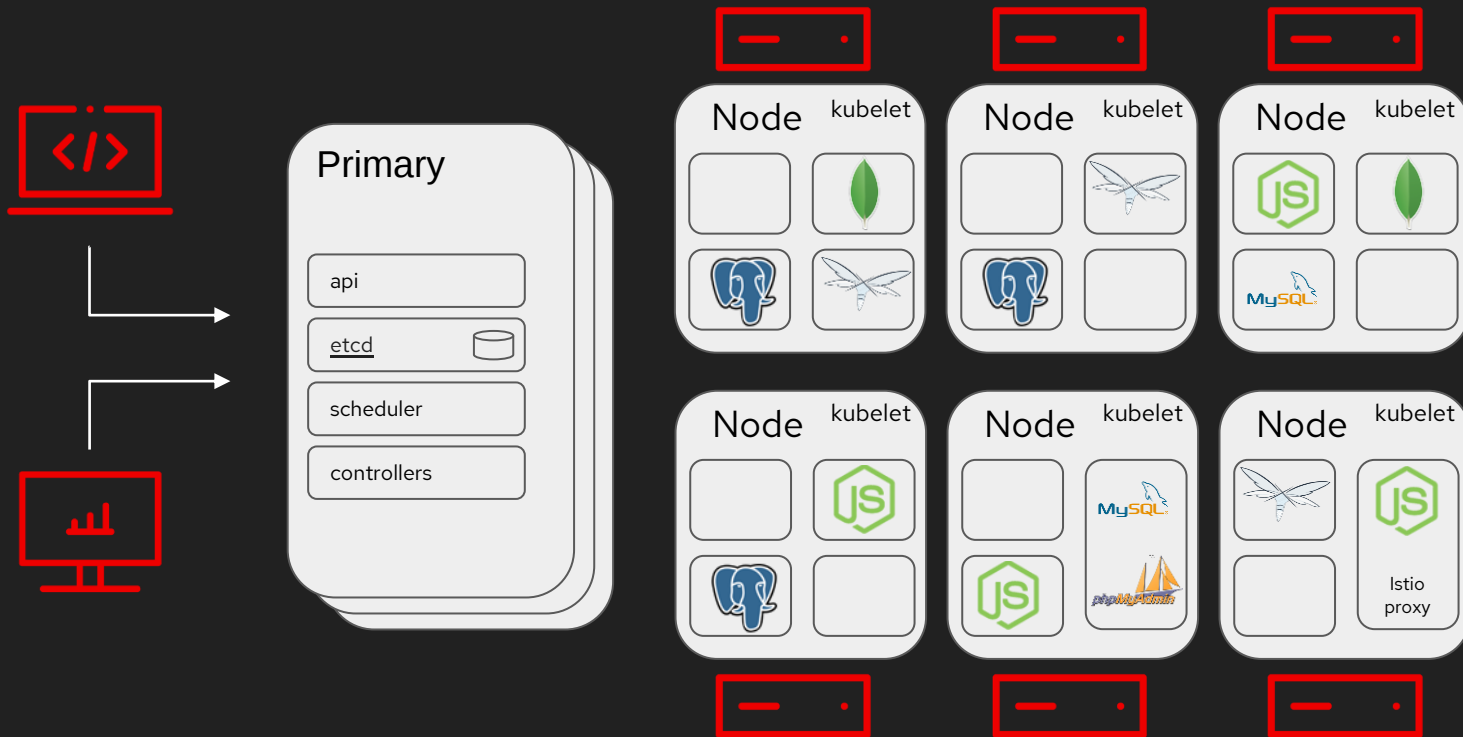
- ✓ Network  
available storage
- ✓ PVs and PVCs

## Label

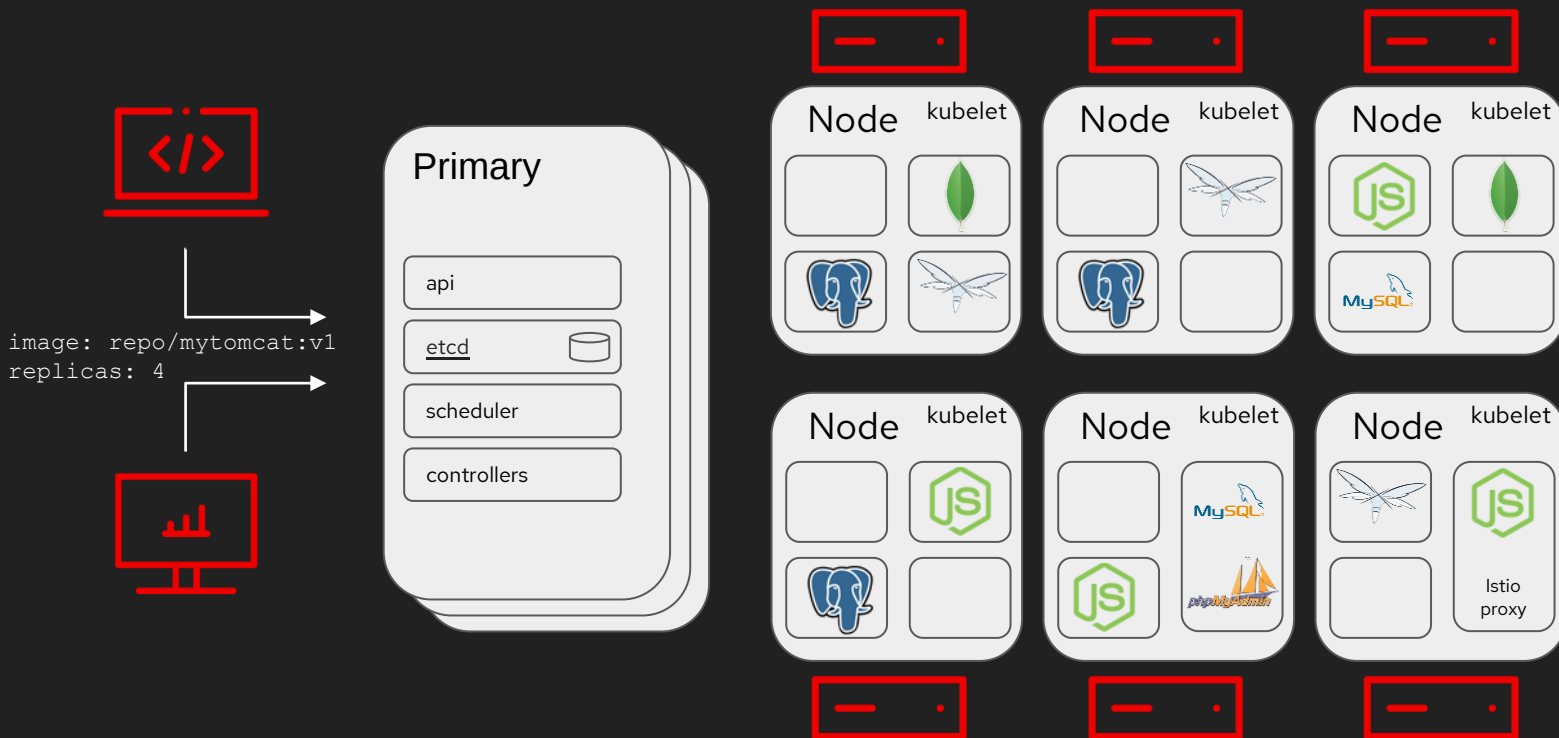


- ✓ Key/value pairs  
associated with  
Kubernetes objects  
(env=production)

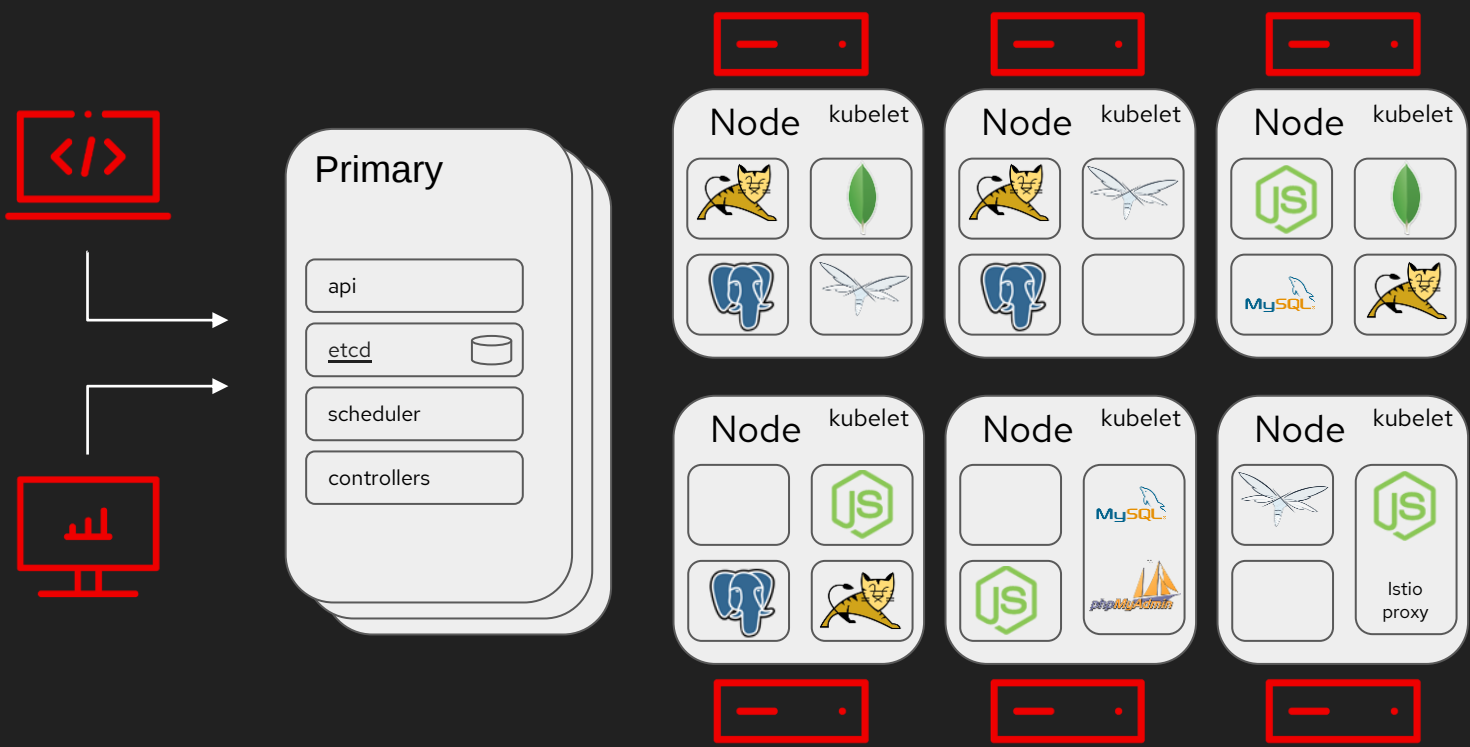
# Kubernetes Cluster - Nodes



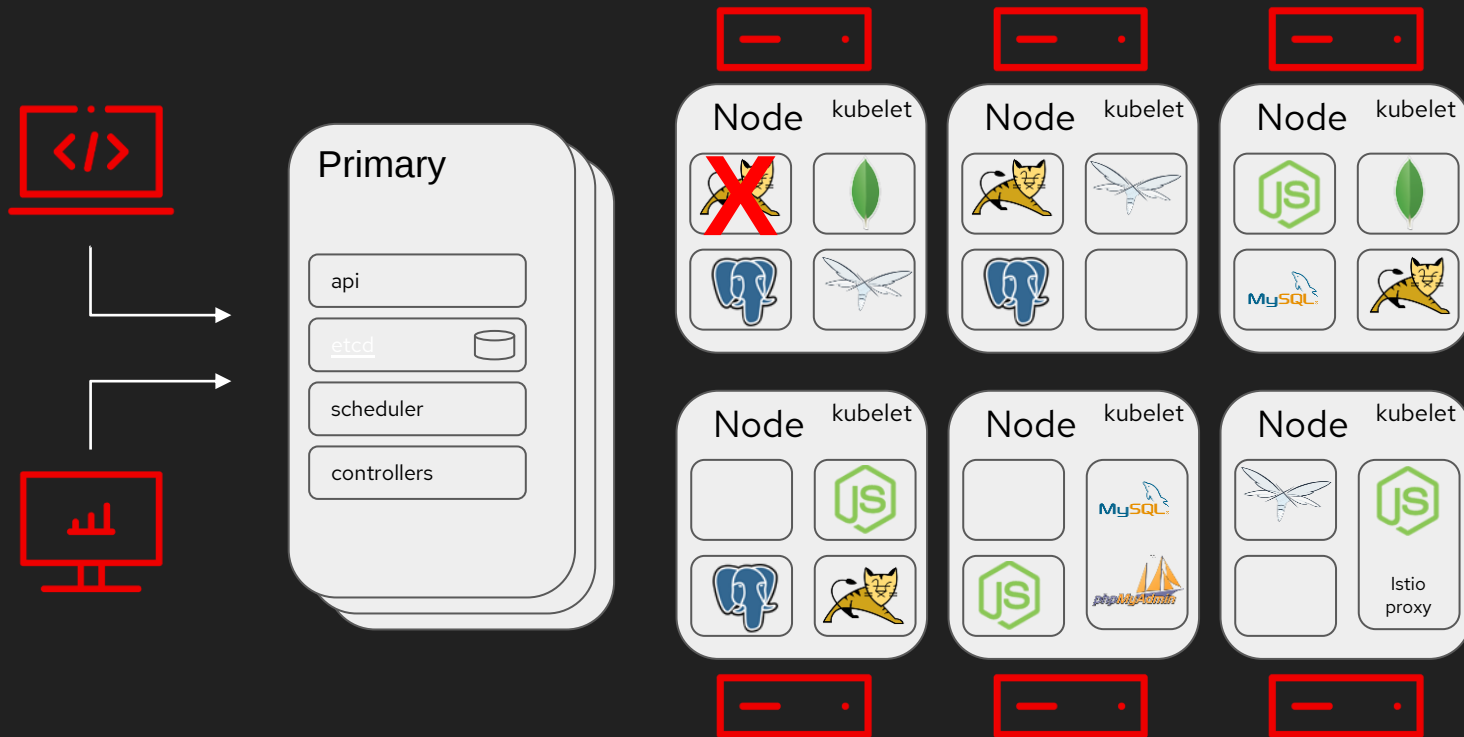
# Kubernetes Cluster - Declarative API



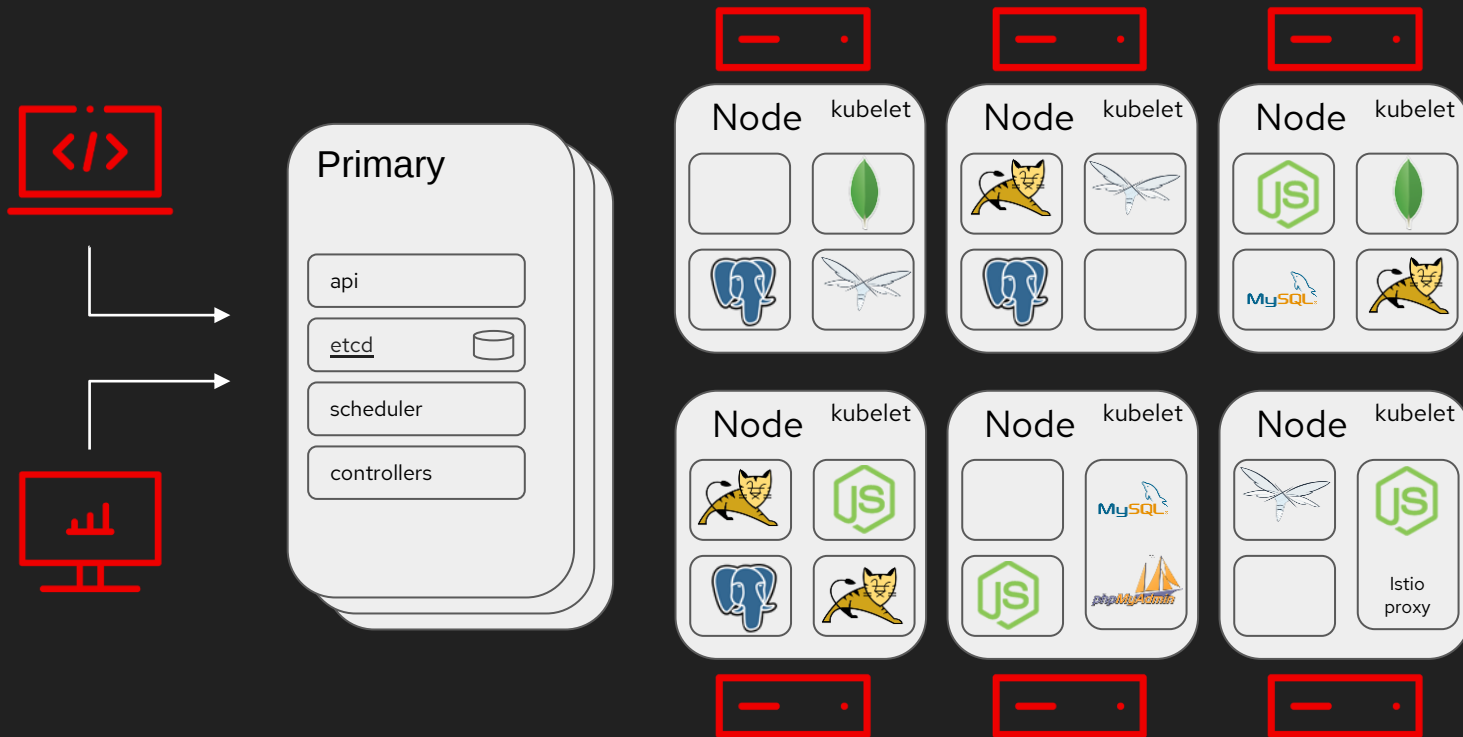
# Kubernetes Cluster - 4 Tomcat Instances



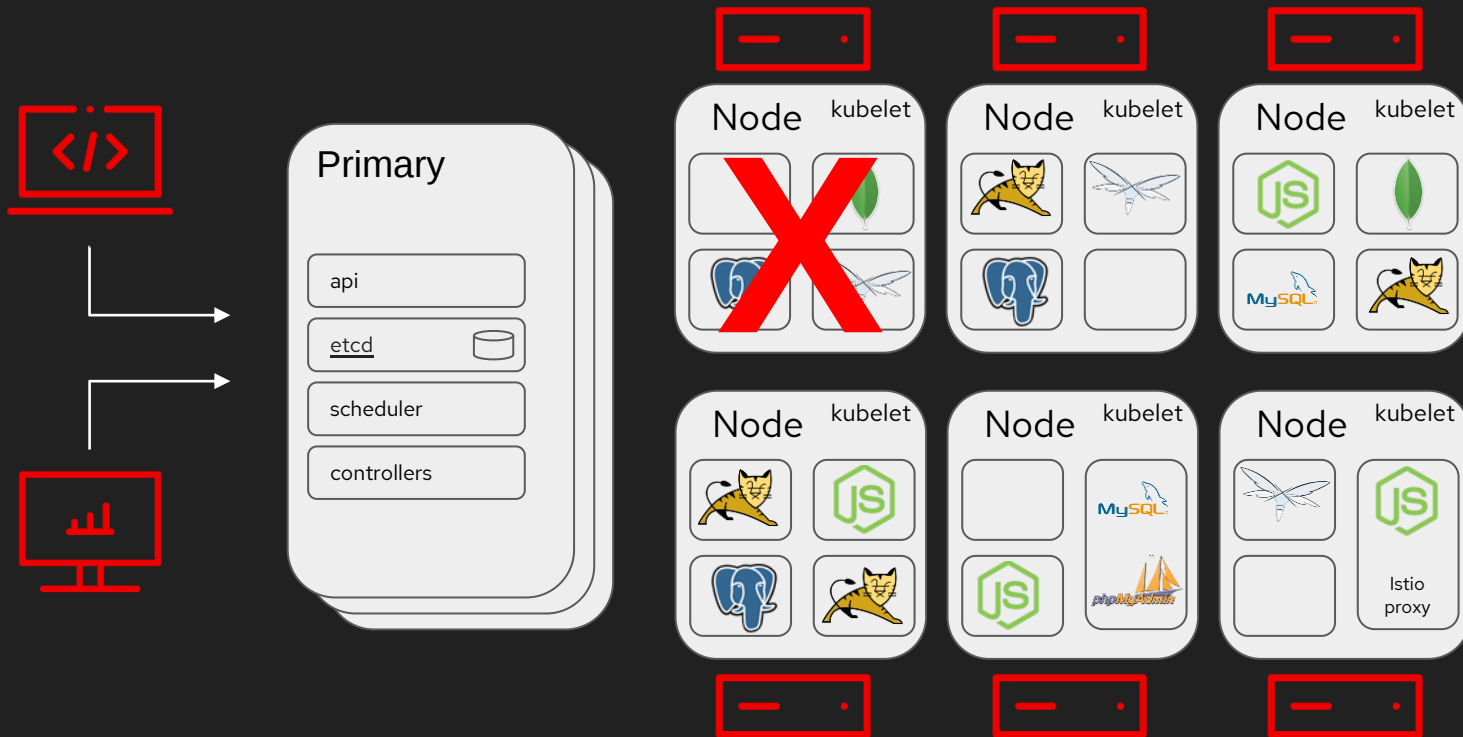
# Kubernetes Cluster – Pod Failure



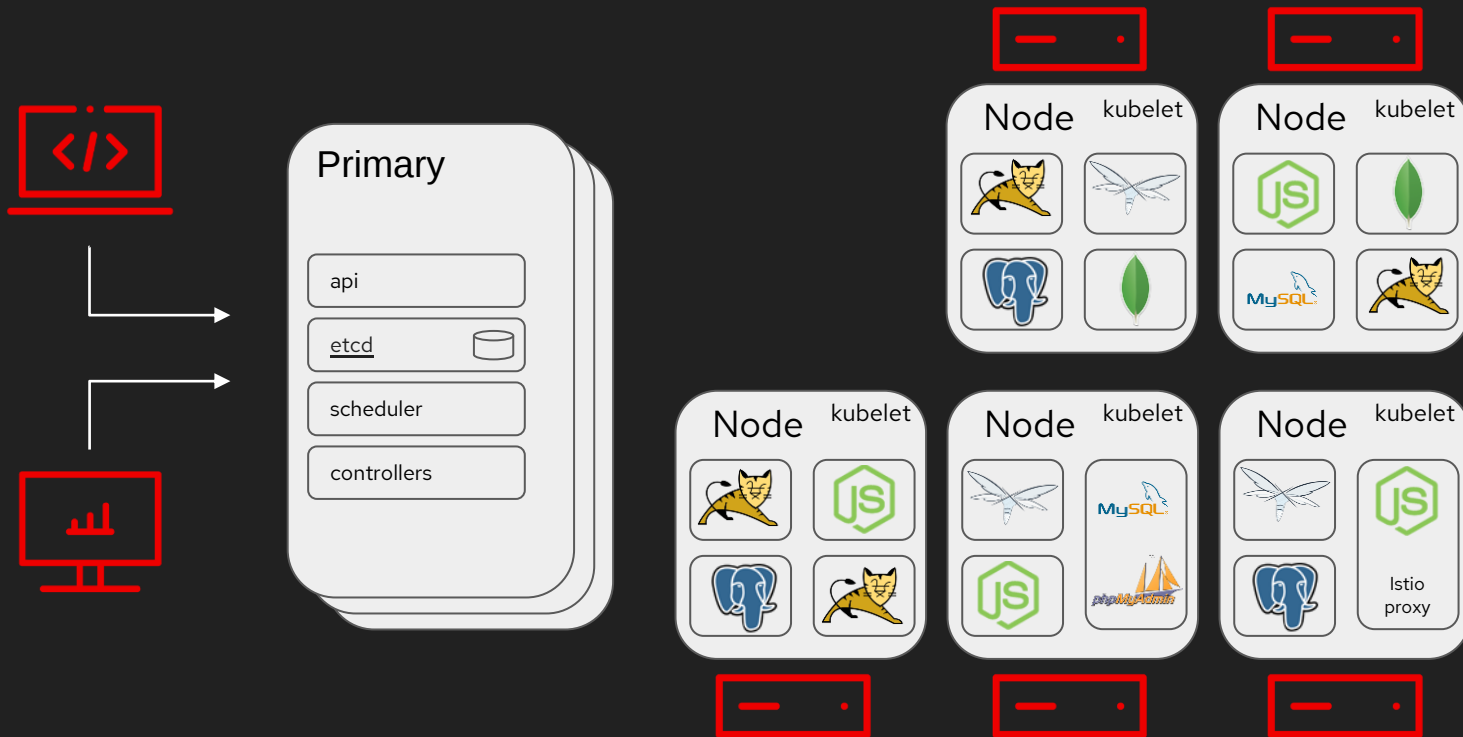
# Kubernetes Cluster – Recovery



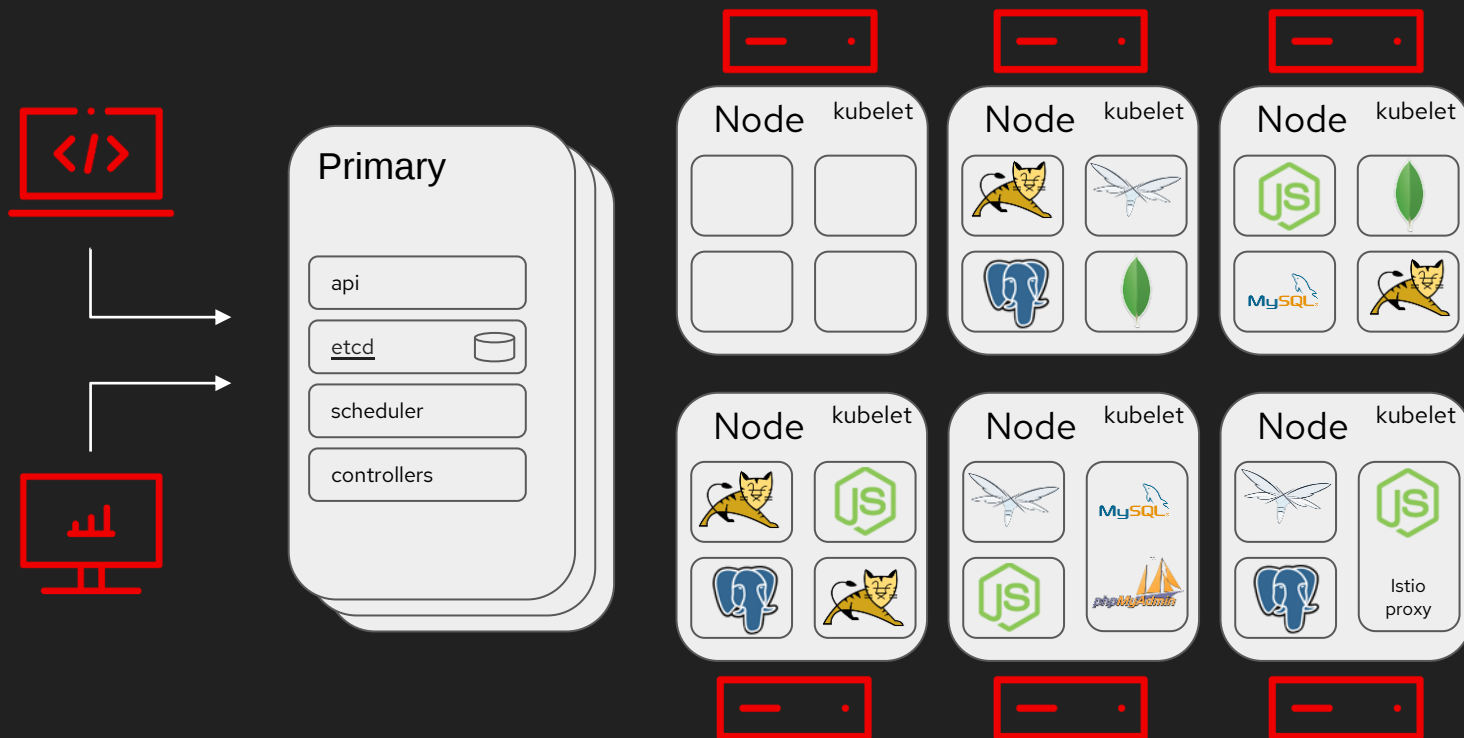
# Kubernetes Cluster – Node Failure



# Kubernetes Cluster – Pods Replaced



# Kubernetes Cluster – New Node



# Live coding

# Free Resources

O'REILLY®

Compliments of  
Red Hat  
Developer

# Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant  
Cloud Native Applications

Second  
Edition



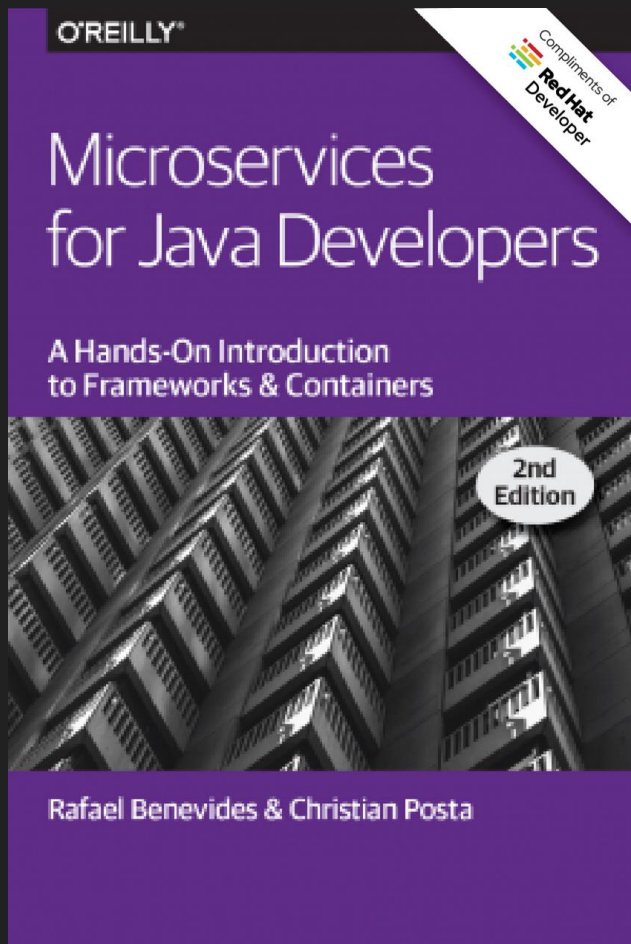
Burr Sutter & Christian Posta

 **Red Hat**  
Developer

 DevNation

[Download](#)

<https://dn.dev/istiobook>



[Download](https://dn.dev/javamsbook)

<https://dn.dev/javamsbook>

O'REILLY®

# Kubernetes Patterns

Reusable Elements for Designing  
Cloud-Native Applications



Bilgin Ibryam &  
Roland Huß



Red Hat  
Developer

DevNation

[Download](https://dn.dev/k8s-patterns)

<https://dn.dev/k8s-patterns>

O'REILLY®

# Knative Cookbook

Building Effective Serverless Applications  
with Kubernetes and OpenShift



Burr Sutter &  
Kamesh Sampath



Red Hat  
Developer

DevNation

[Download](#)

dn.dev/knative-cookbook

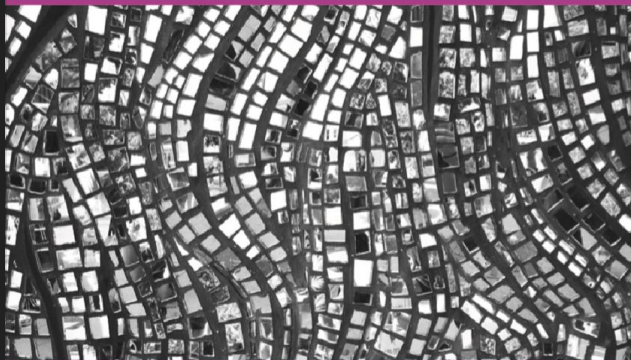
dn.dev/kubemaster1

O'REILLY®

Compliments of  
**Red Hat**  
Developer

# Migrating to Microservice Databases

From Relational Monolith  
to Distributed Data



Edson Yanaga

 **Red Hat**  
Developer

 DevNation

[Download](#)

[bit.ly/mono2microdb](https://bit.ly/mono2microdb)

# The End

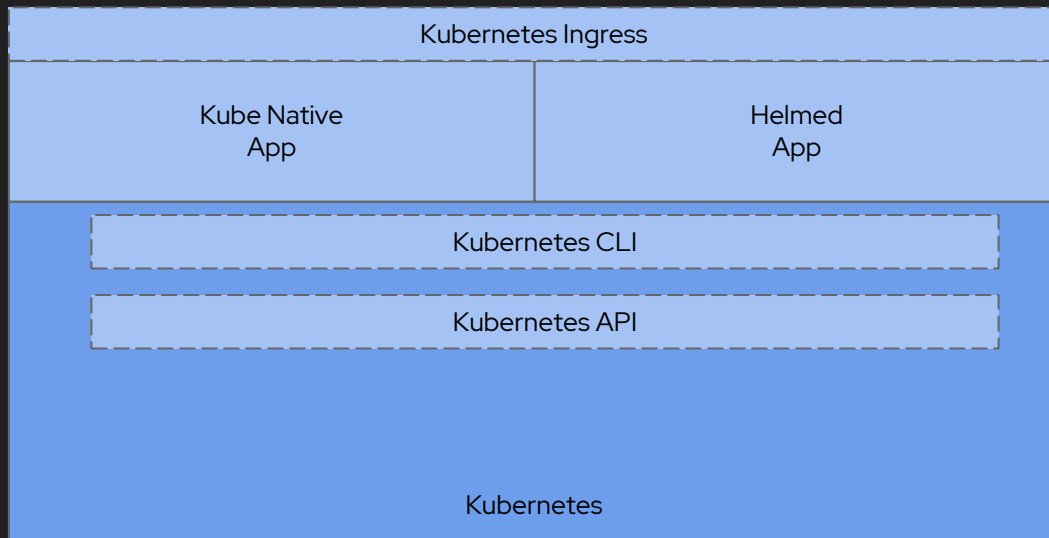
# Backup Content

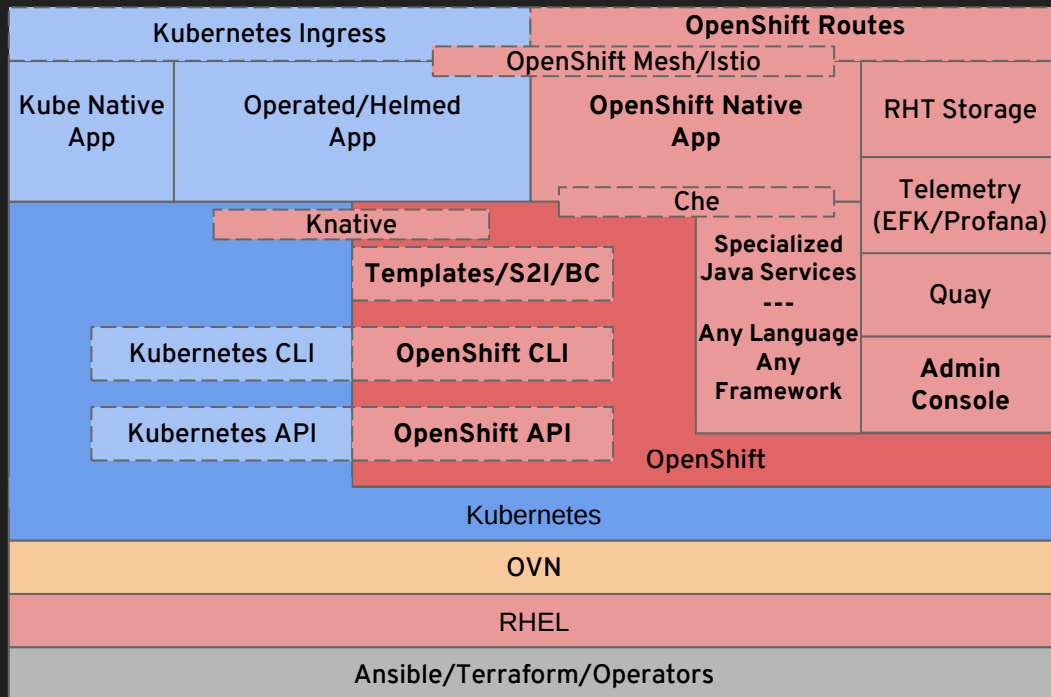


**Red Hat**  
Developer



|                                        | Kubernetes | OpenShift |
|----------------------------------------|------------|-----------|
| Multi-container, multi-host scheduling | ✓          | ✓         |
| Self-service provisioning              | ✓          | ✓         |
| Service discovery                      | ✓          | ✓         |
| Persistent Storage                     | ✓          | ✓         |
| Multi-tenancy                          |            | ✓         |
| Networking (SDN)                       |            | ✓         |
| Image Registry                         |            | ✓         |
| Image Build Tools                      |            | ✓         |
| Metrics                                |            | ✓         |
| Log Aggregation                        |            | ✓         |
| Ingress                                |            | ✓         |
| UX: Console, Service Catalog, "oc"     |            | ✓         |
| Secured by Default                     |            | ✓         |





# Strangler Strategy

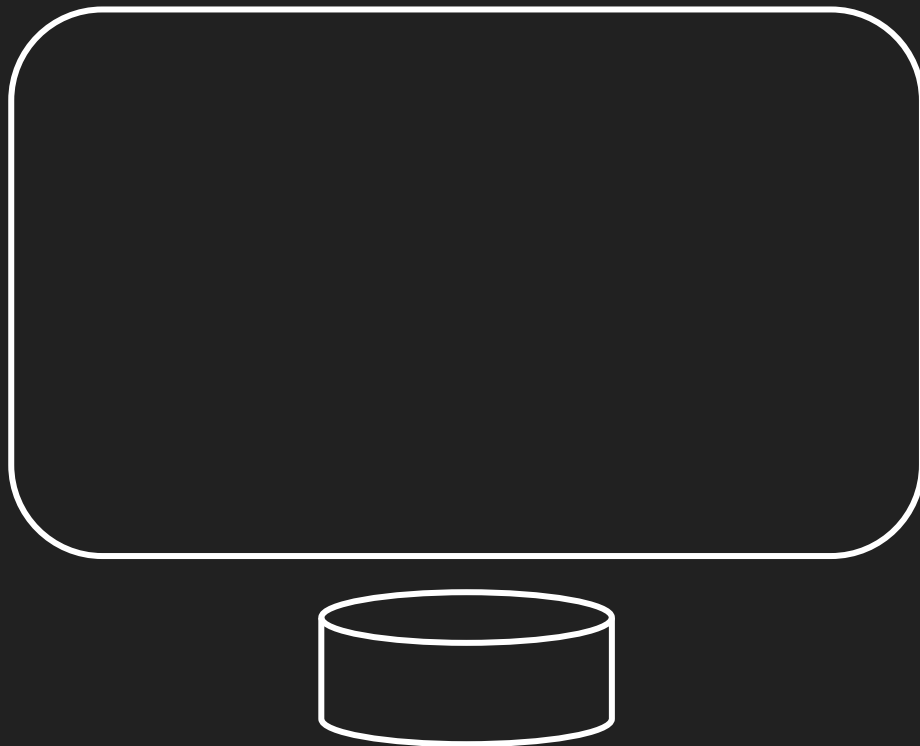




# Application on Whiteboard



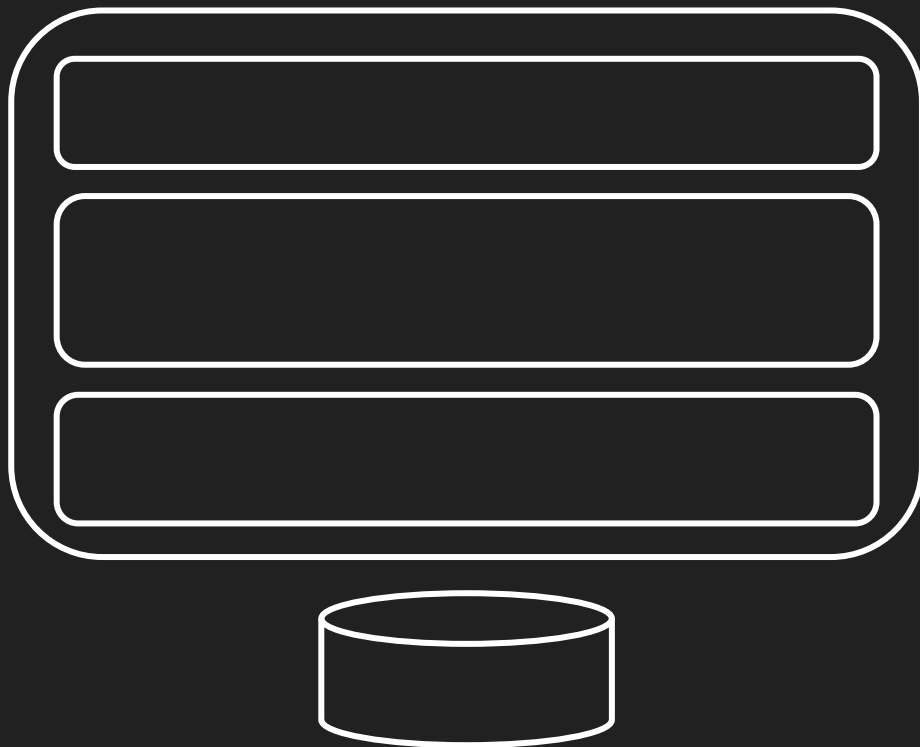
Red Hat  
Developer



# Whiteboard had 3 Tiers



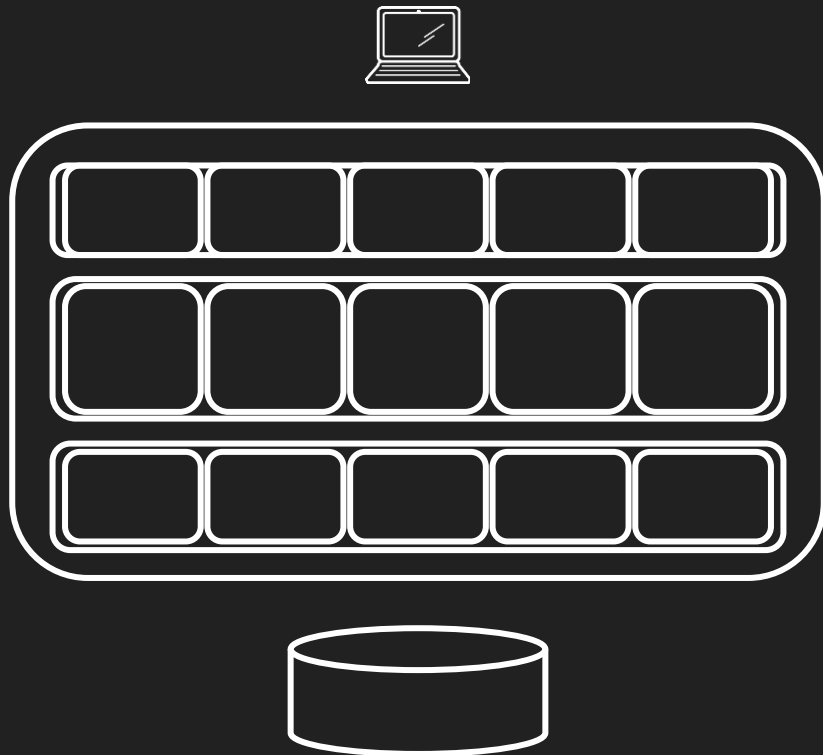
Red Hat  
Developer



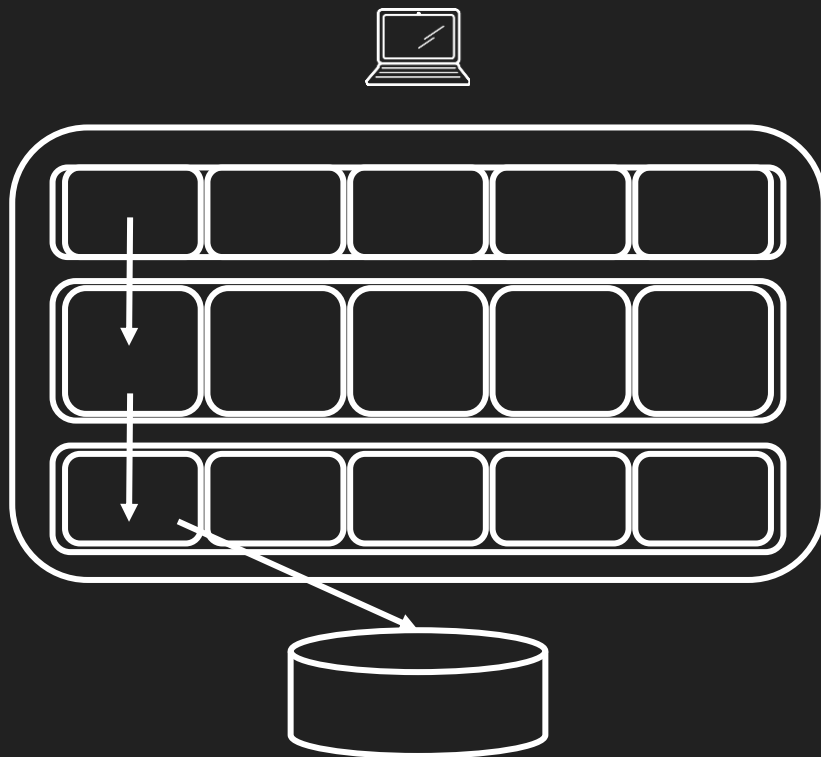
# UI, Logic, Data



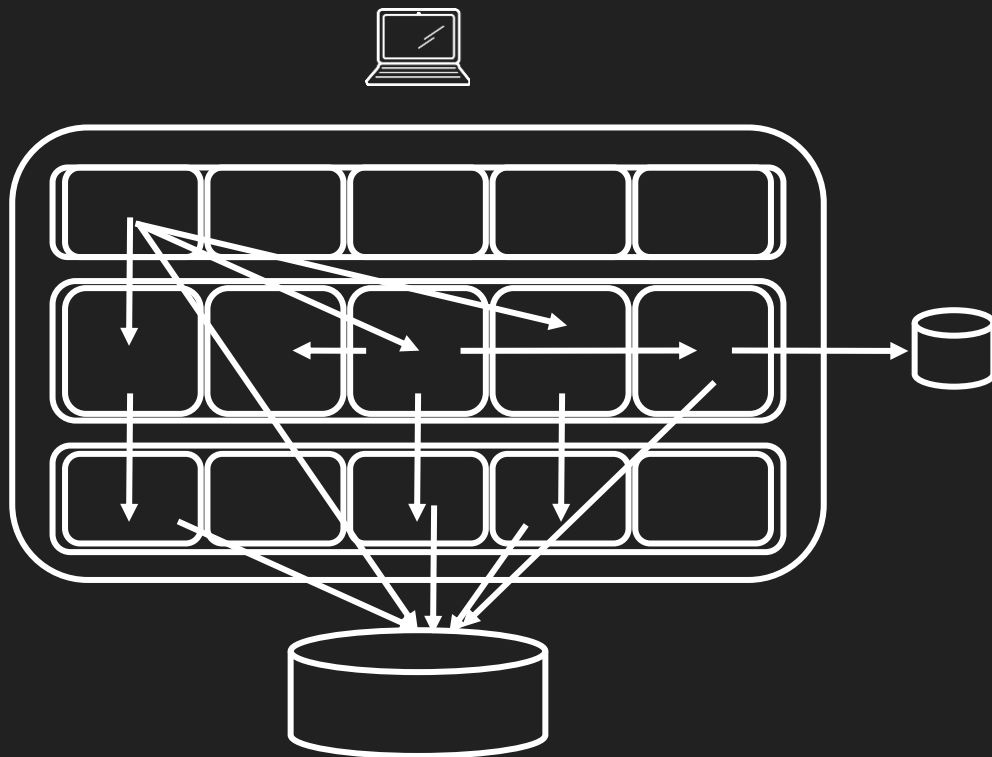
Red Hat  
Developer



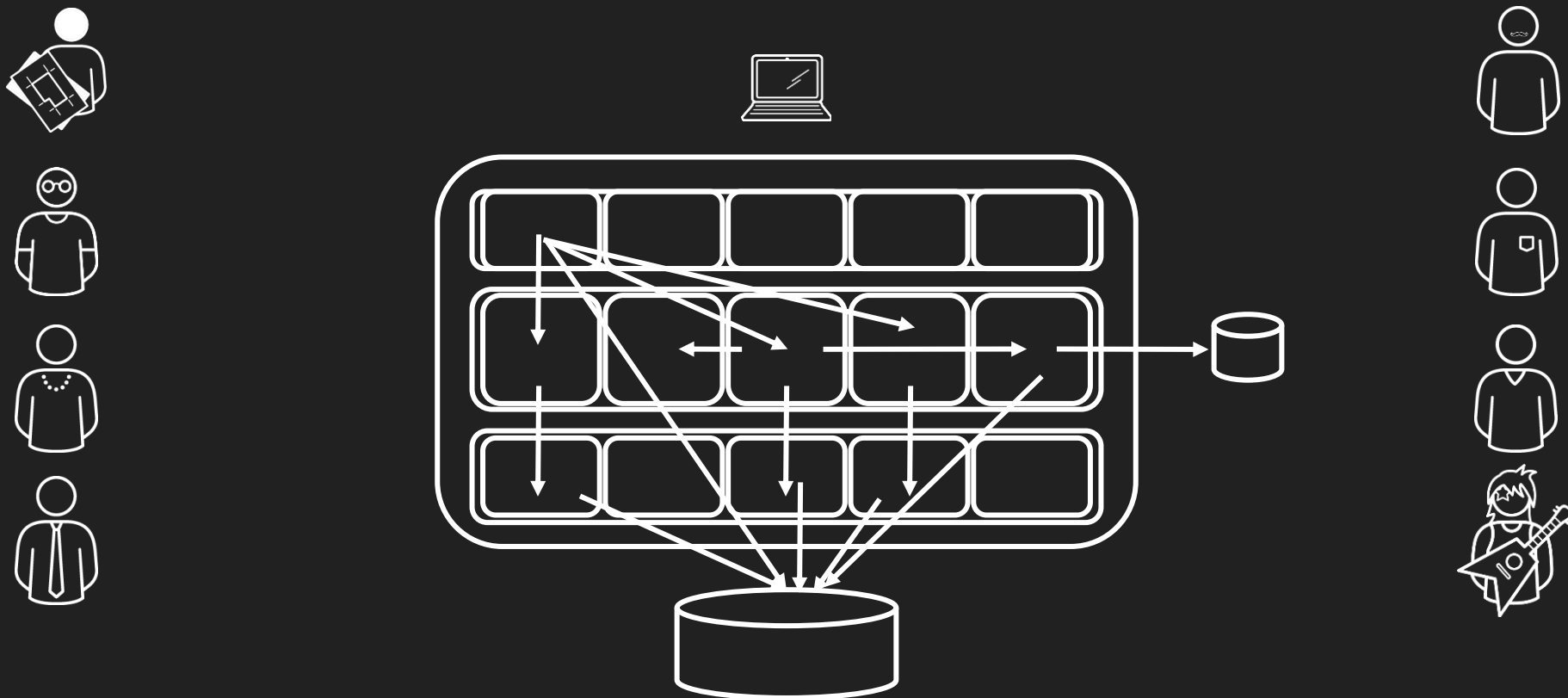
# Clean Architecture



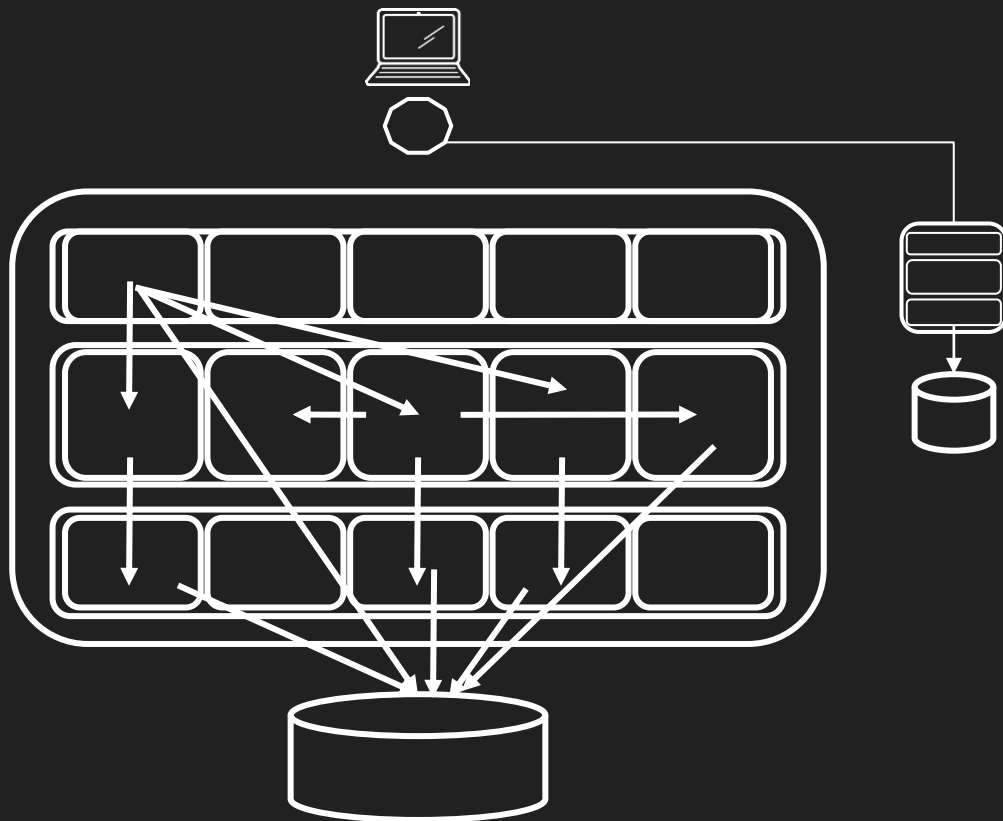
# Real Life: Non-Majestic Monolith



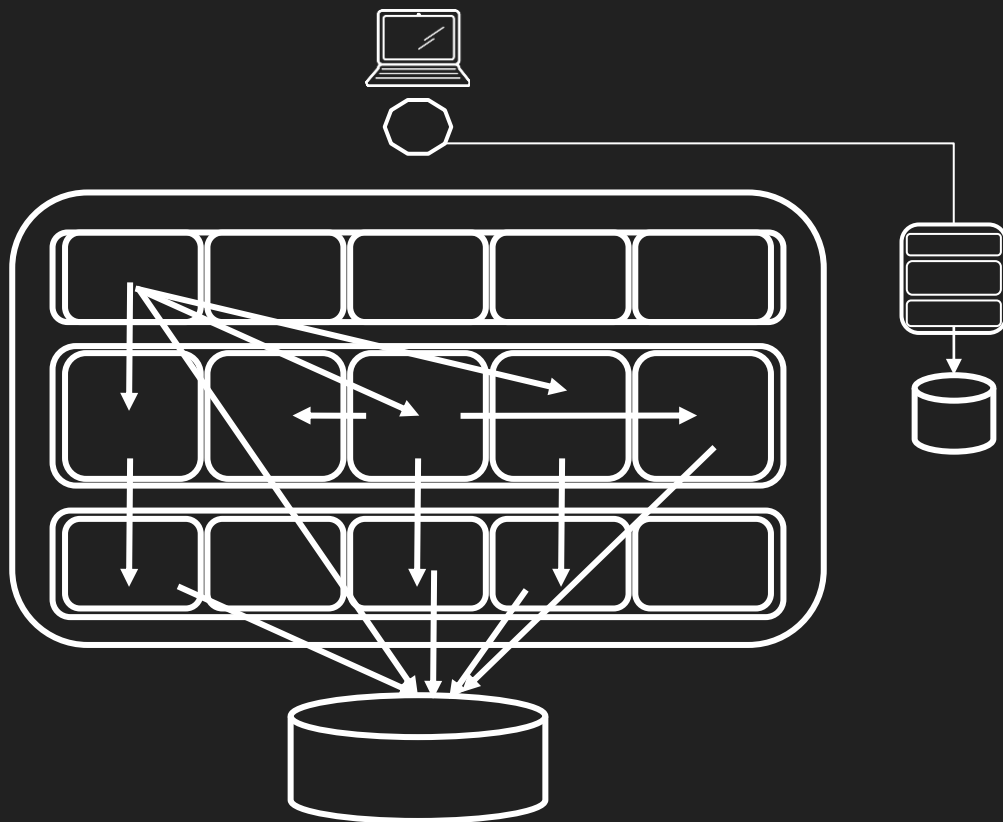
# Large Team



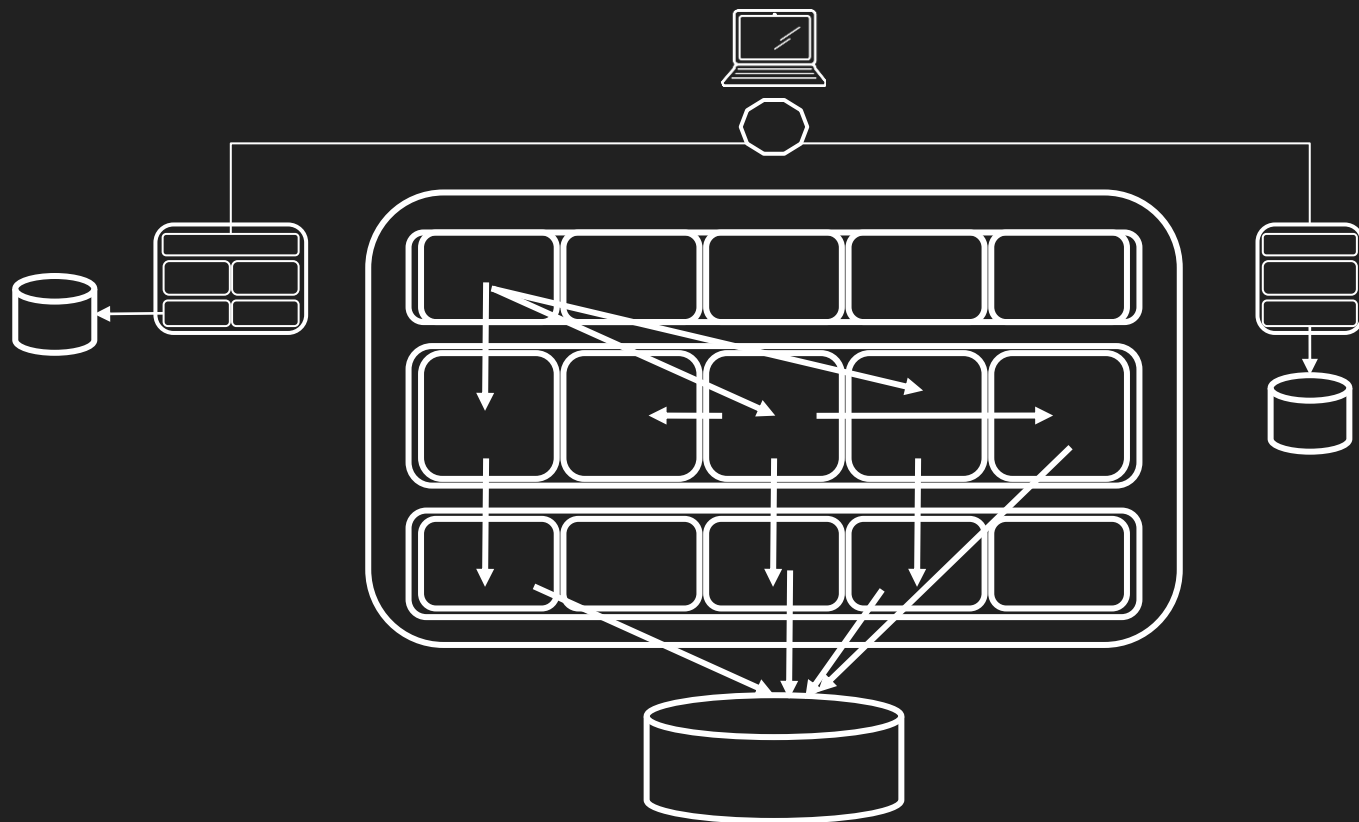
# Strangle



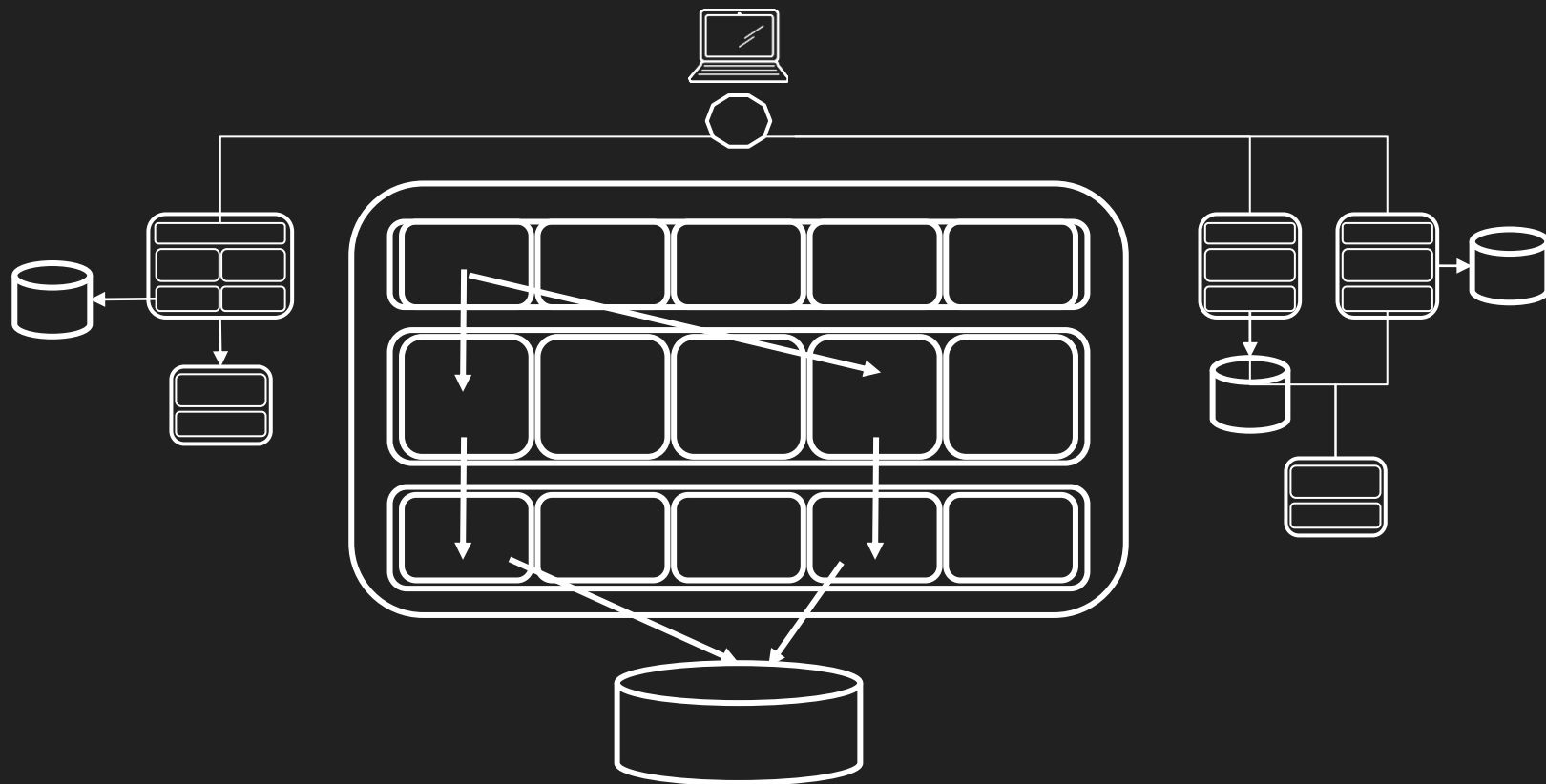
# Strangle Hug



# Friend Hug



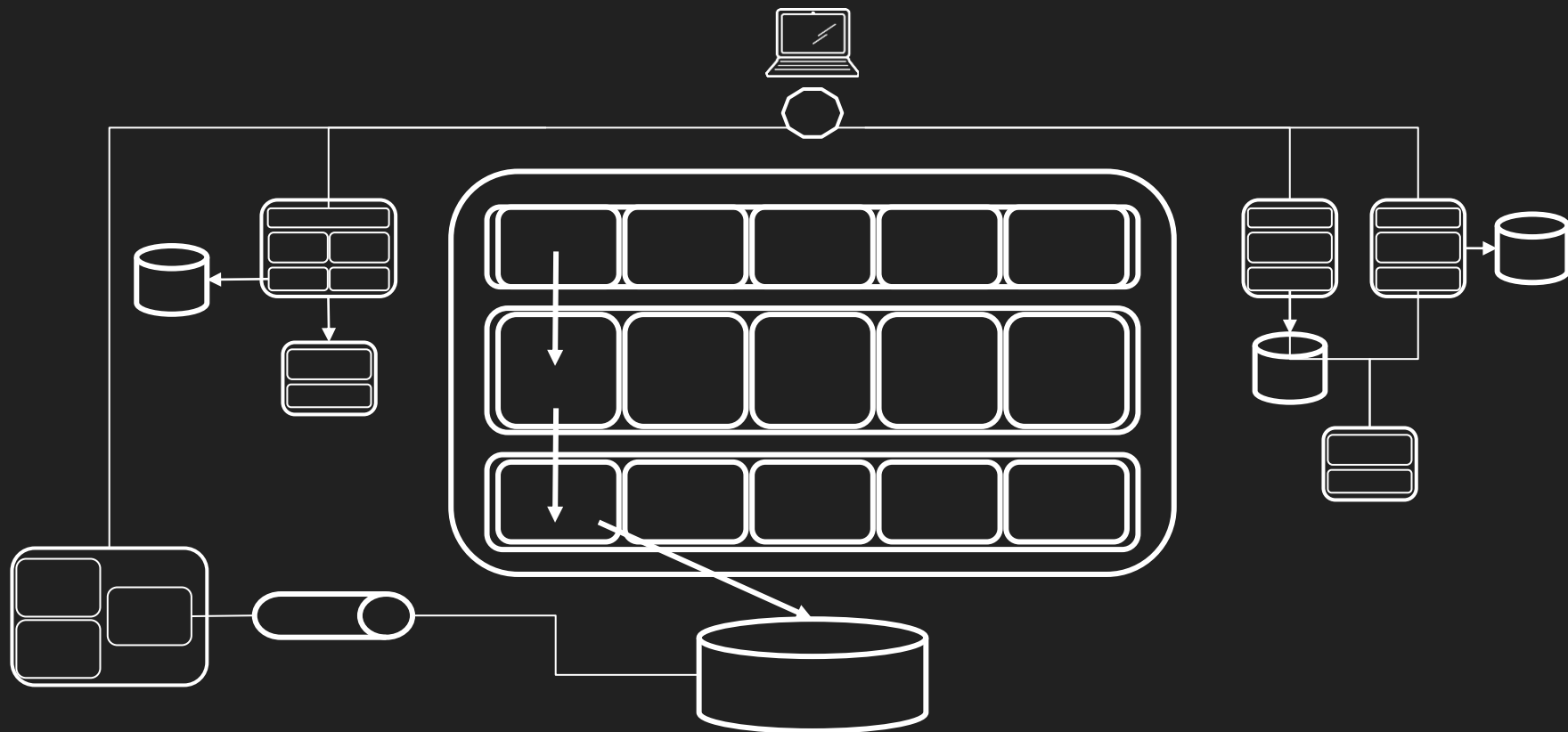
# Family Hug



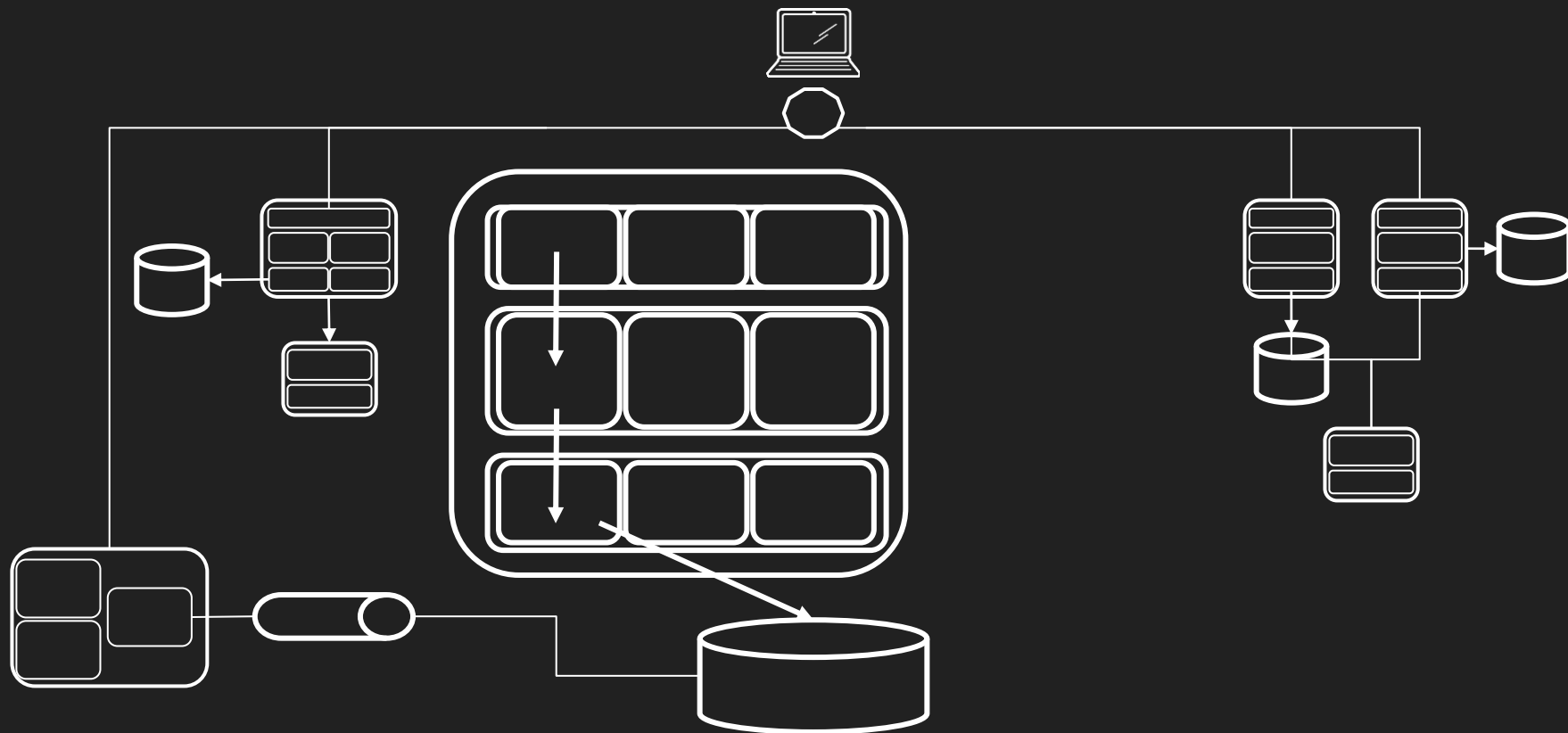
# Bear Hug



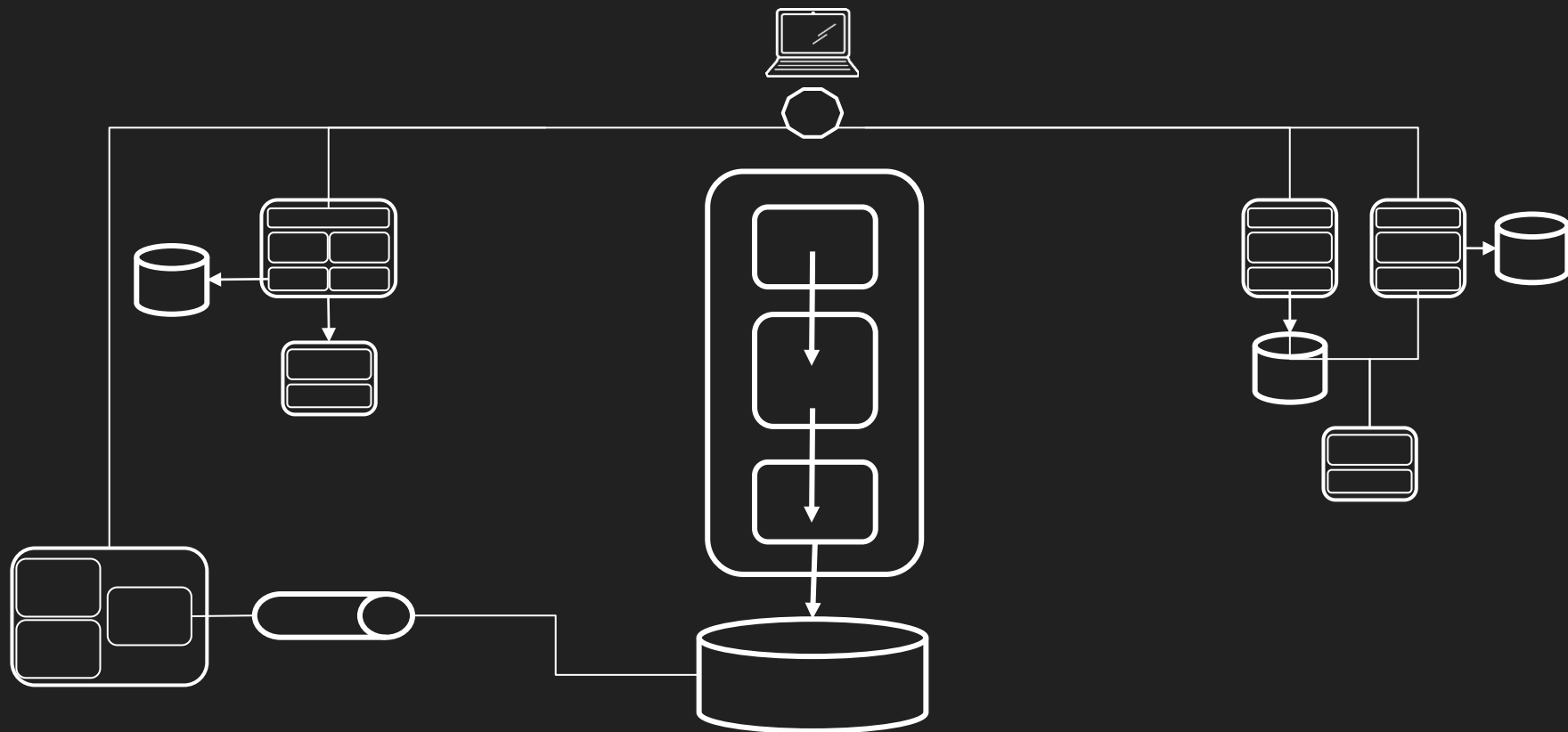
Red Hat  
Developer



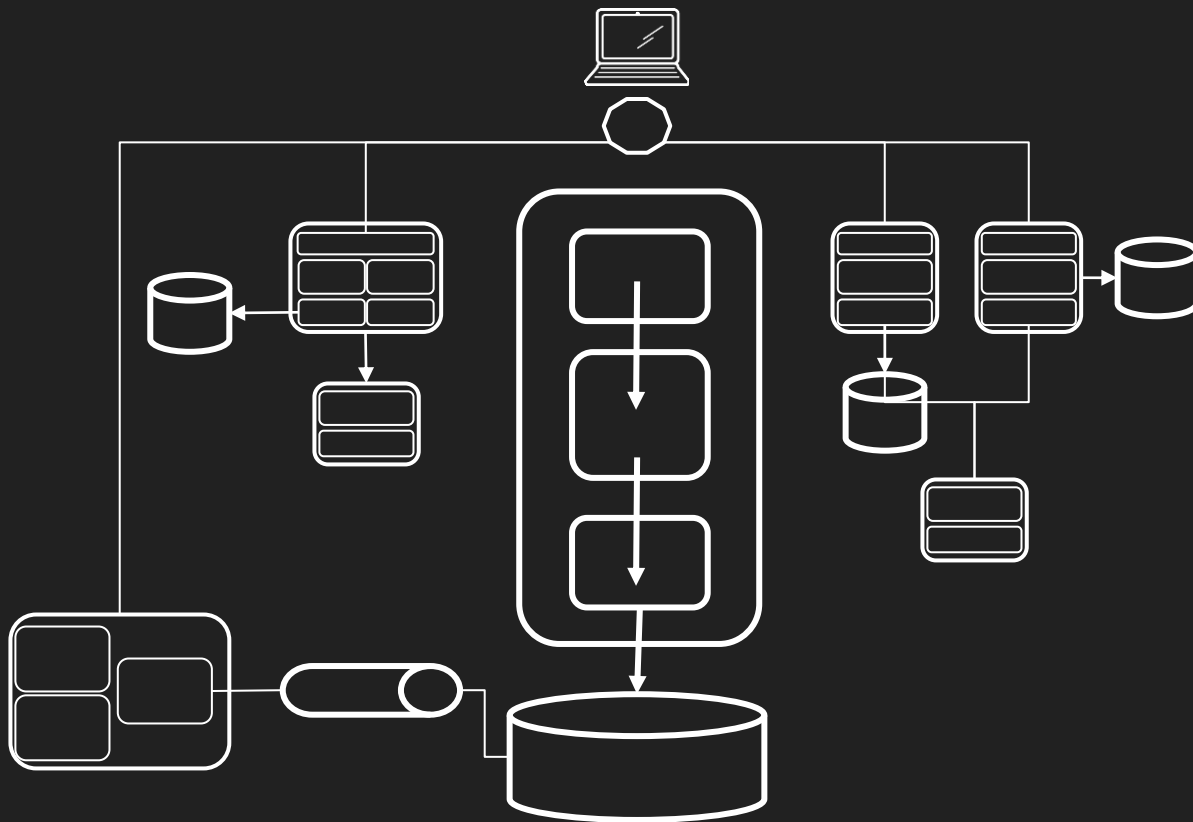
# "Refactor"



# Squeeze



# Streamline



# Embrace Others

