



DevOps in Automotive Industry

February 2021



Visteon®

Presenters

Visteon®

Zdravko Chiflishki
DevOps Engineer



Lubomir Tsirov
Software Leader



Todor Todorov
DevOps Leader



Zhivko Zapryanov
DevOps Manager



DEVOPS

➤ What is DevOps?

Specifics in the Automotive industry

DevOps in Visteon

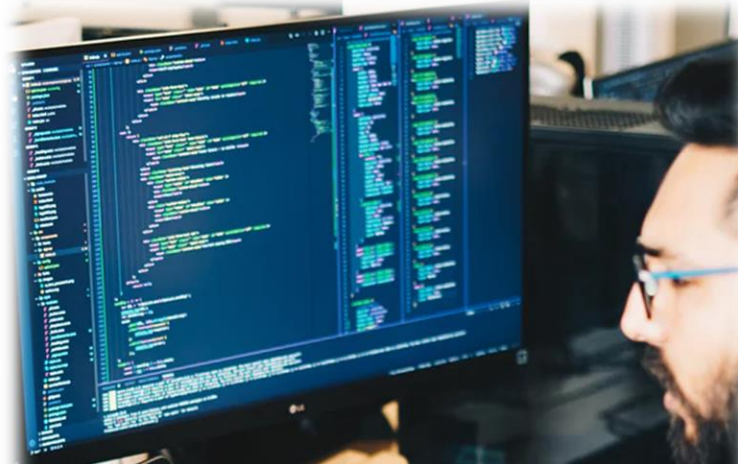
Infrastructure efficiency

CI/CD Onboarding

Monitoring

What is DevOps

DevOps is a methodology that aims to alter and improve the relationship between development...

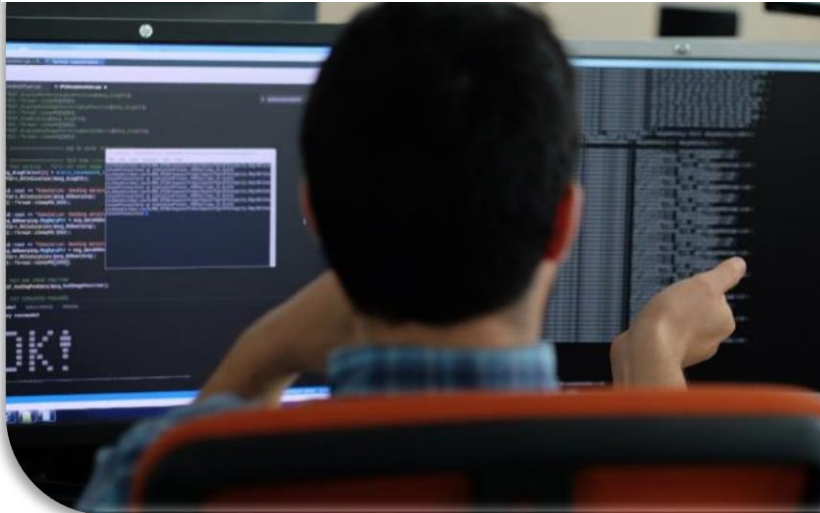


... and IT operations, advocating better communication and collaboration between the two units.

Software Development

DevOps

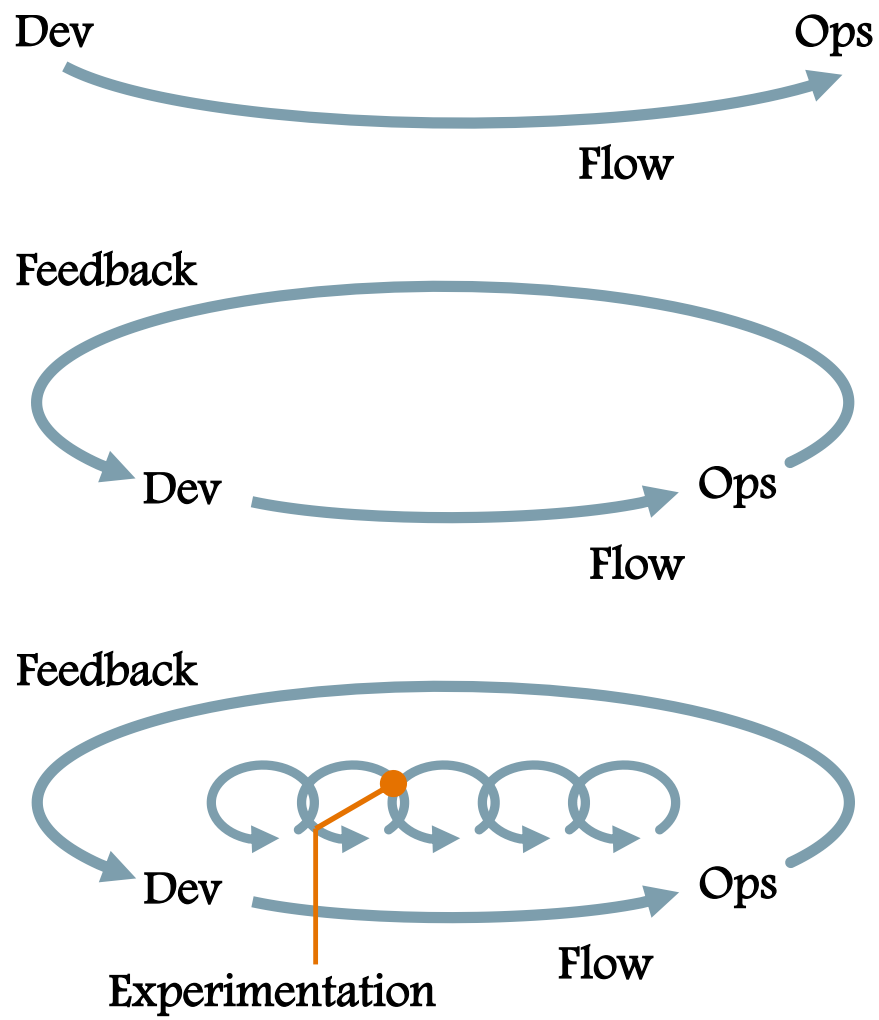
IT Operations



10+ Deploys per Day: Dev and Ops Cooperation at Flickr



DevOps Core Principles



DEVOPS

What is DevOps?

➤ **Specifics in the Automotive industry**

DevOps in Visteon

Infrastructure efficiency

CI/CD Onboarding

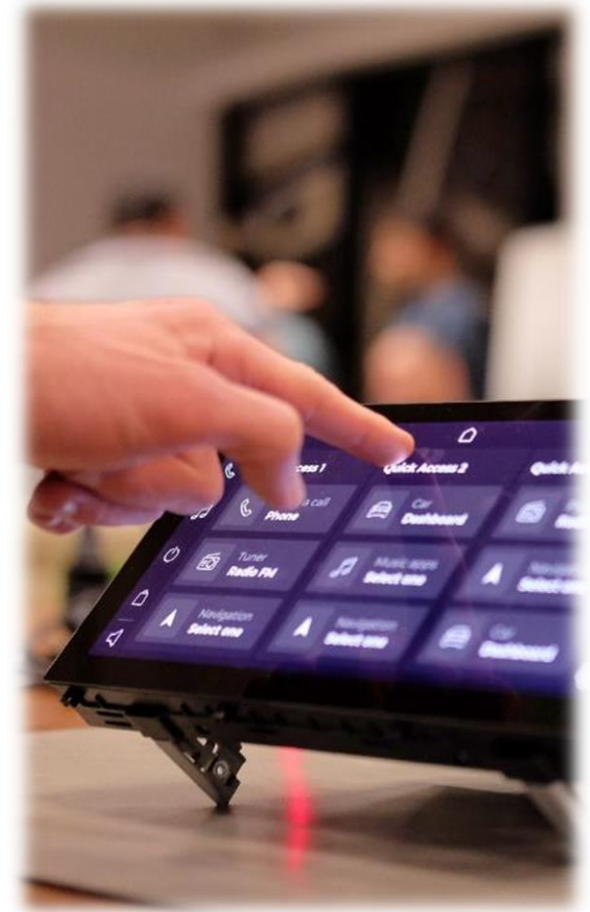
Monitoring

Evolution of the automotive electronics



Why DevOps is important in Automotive

- Increasing customer demands for short delivery cycles
- Standardization and fast onboarding
- Removing silos and improving cross skills collaboration
- Shifting left testing, reduce number of defects in production
- Consumers have “app” mentalities and expectations
- Fast integration and adaptation of cutting edge technologies



Challenges to implement DevOps in Automotive

Complex and diverse products:



- Millions LOC
- Multiple OS in one device
- Interaction with other devices

Working with embedded devices:



- You can't simple test on your PC
- Licenses and hardware

Extensive quality requirements:



- Safety
- Security
- Regulations
- Robustness

Complex Software Development Process:



- Multiple skills are involved
- Detailed quality checks on step
- Extensive design and documentation

DEVOPS

What is DevOps?

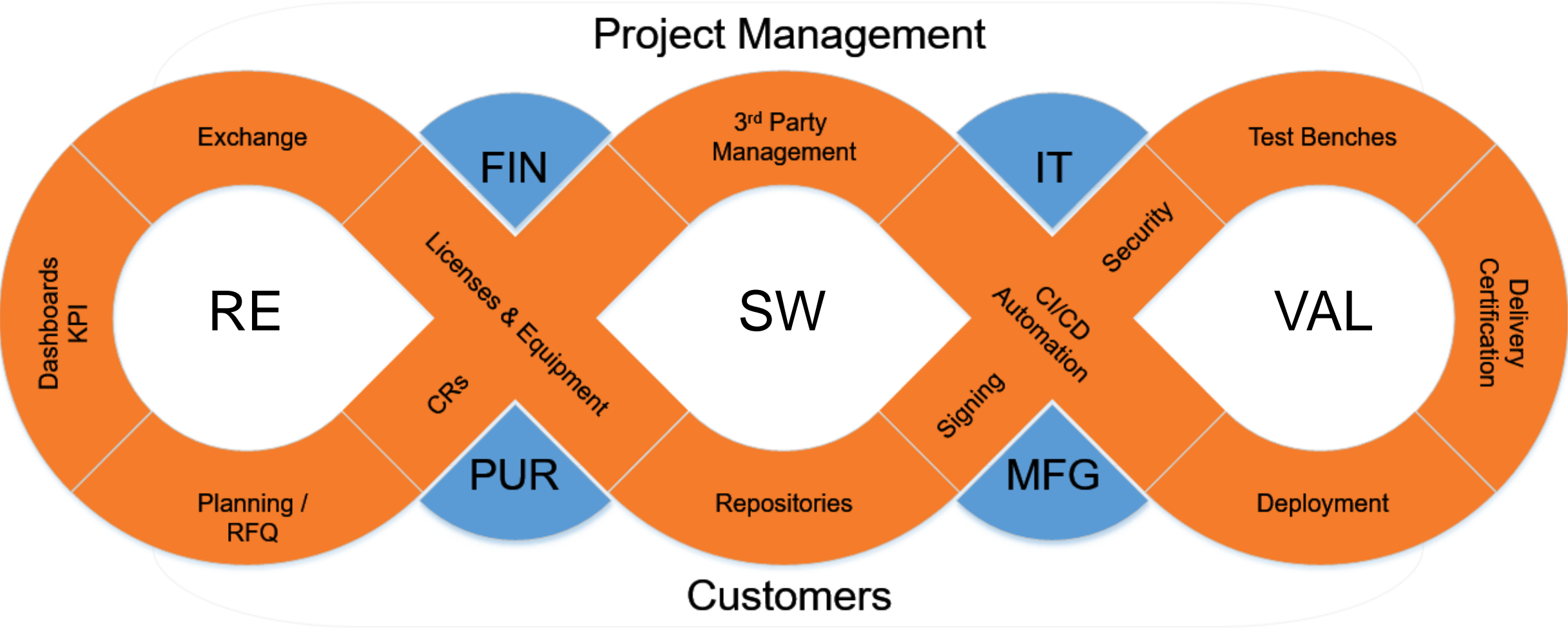
Specifics in the Automotive industry

➤ **DevOps in Visteon**

Infrastructure efficiency

CI/CD Onboarding

Monitoring



Our factors for successful DevOps transformation



Culture



Collaboration and innovations

Process



Understanding and optimization

Automation



Start with the bottlenecks



DEVOPS

What is DevOps?

Specifics in the Automotive industry

DevOps in Visteon

➤ **Infrastructure efficiency**

CI/CD Deployment

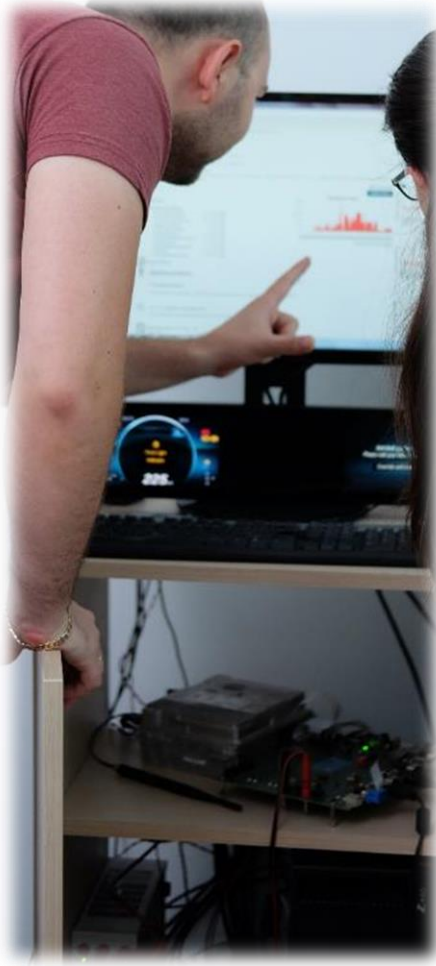
Monitoring

Q&A

14



Infrastructure challenges



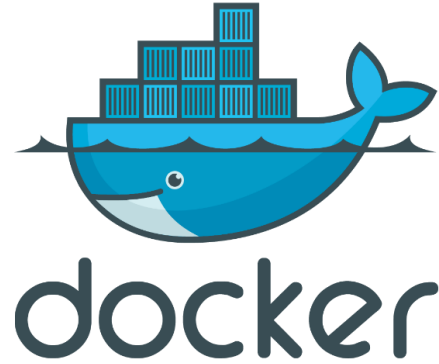
- HW resources sharing
- Utilization and efficiency
- Resilience and high availability
- Performance and quality
- Standard build environment
- Onboarding time



Introduction to Docker

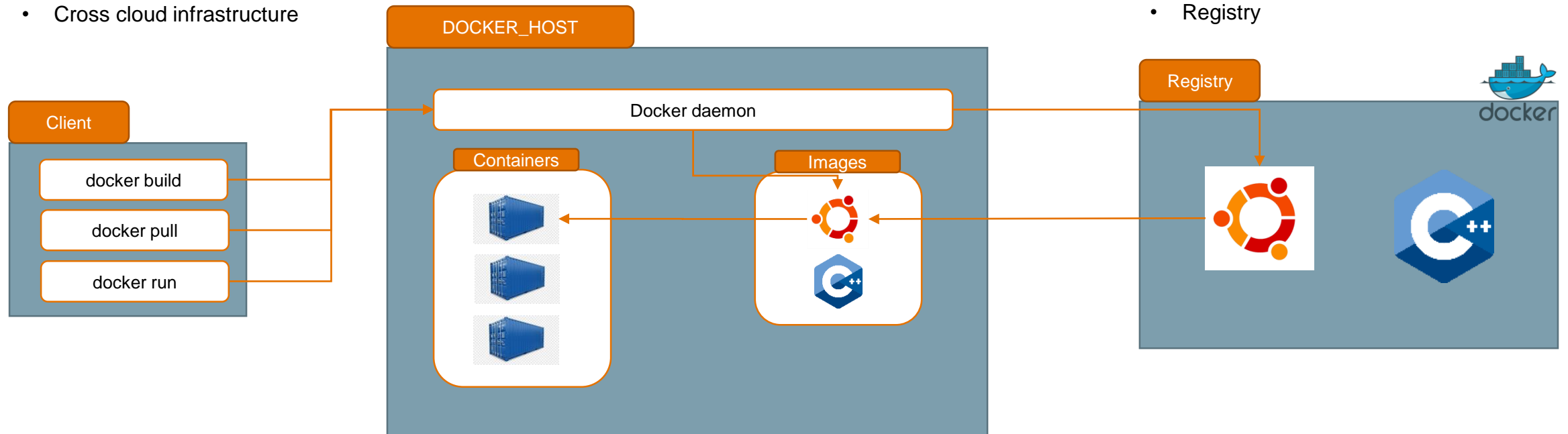
Docker Features

- Extremely fast and elegant isolation framework
- Inexpensive
- Fast boot/shutdown
- Low CPU/memory overhead
- Cross cloud infrastructure

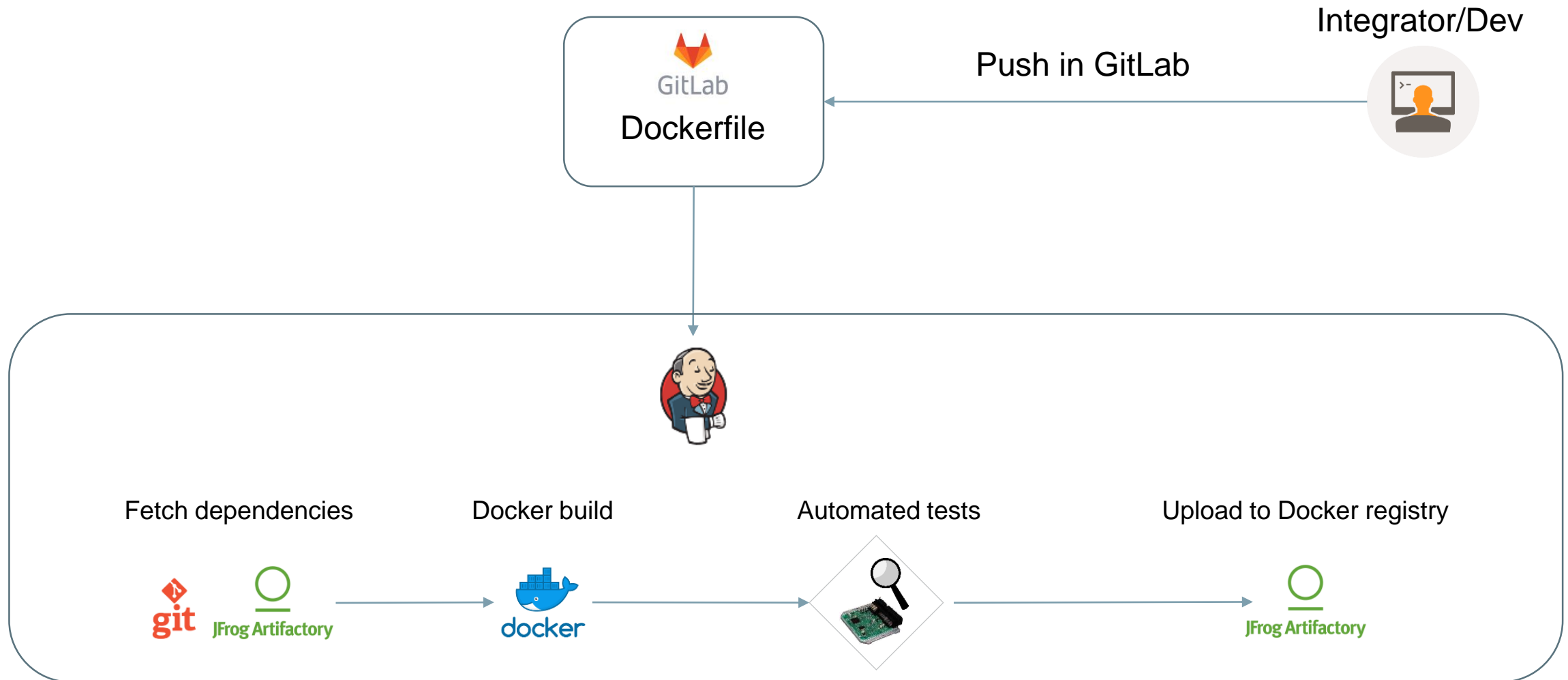


Docker Components

- Docker Client
- Docker Daemon
- Docker Containers
- Docker Images
- Registry



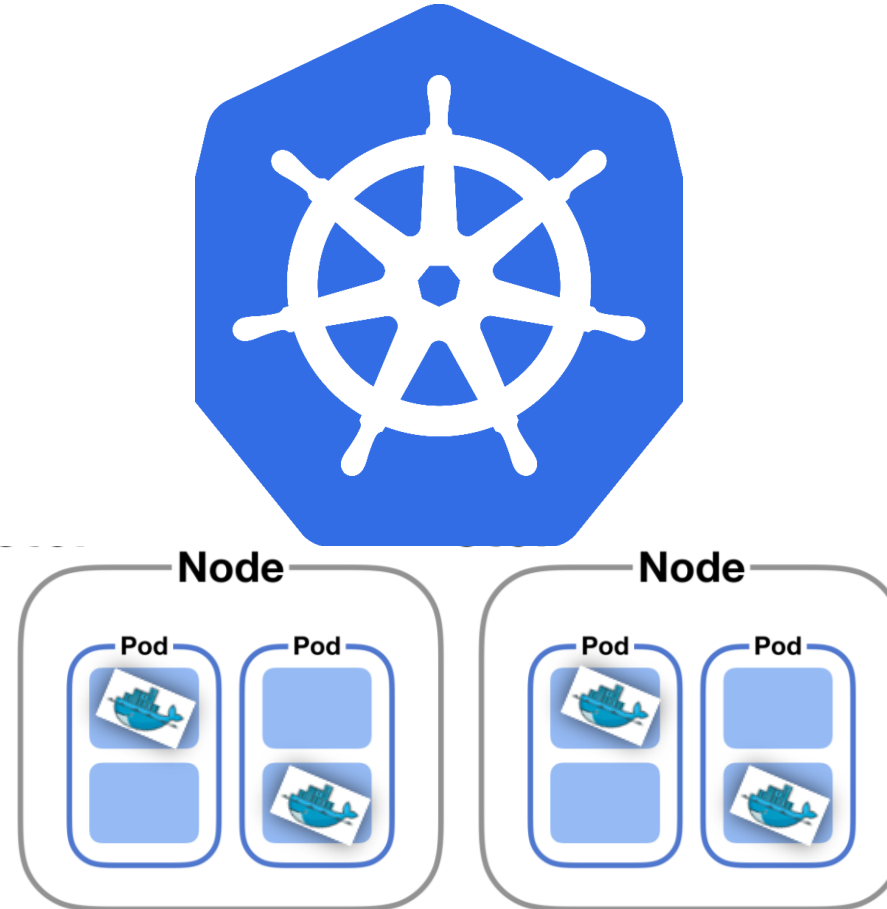
Docker image creation



Introduction to Kubernetes

Kubernetes Features

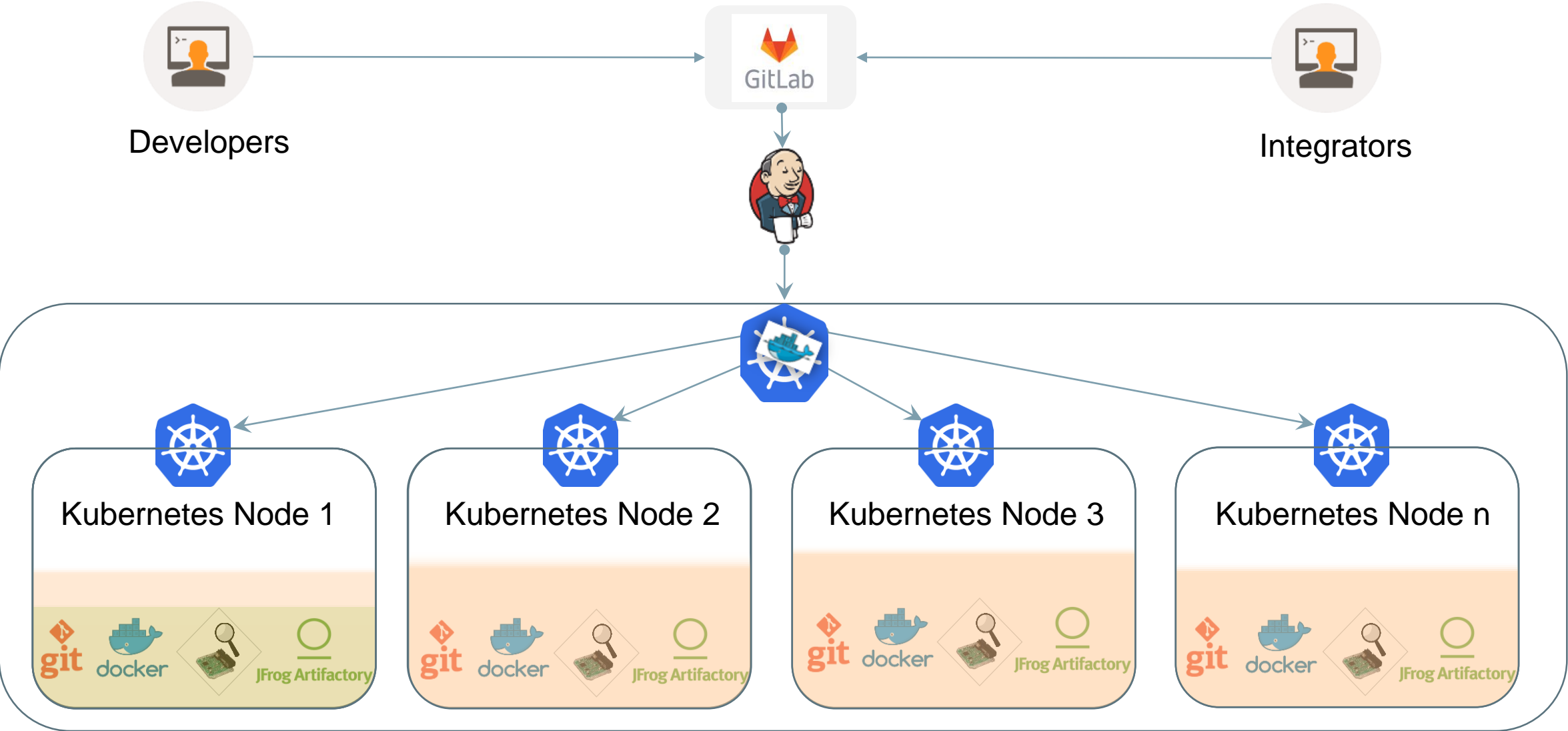
- Horizontal Scaling
- Self-healing
- Ingress Controller
- High Availability
- Services
- Load Balancing



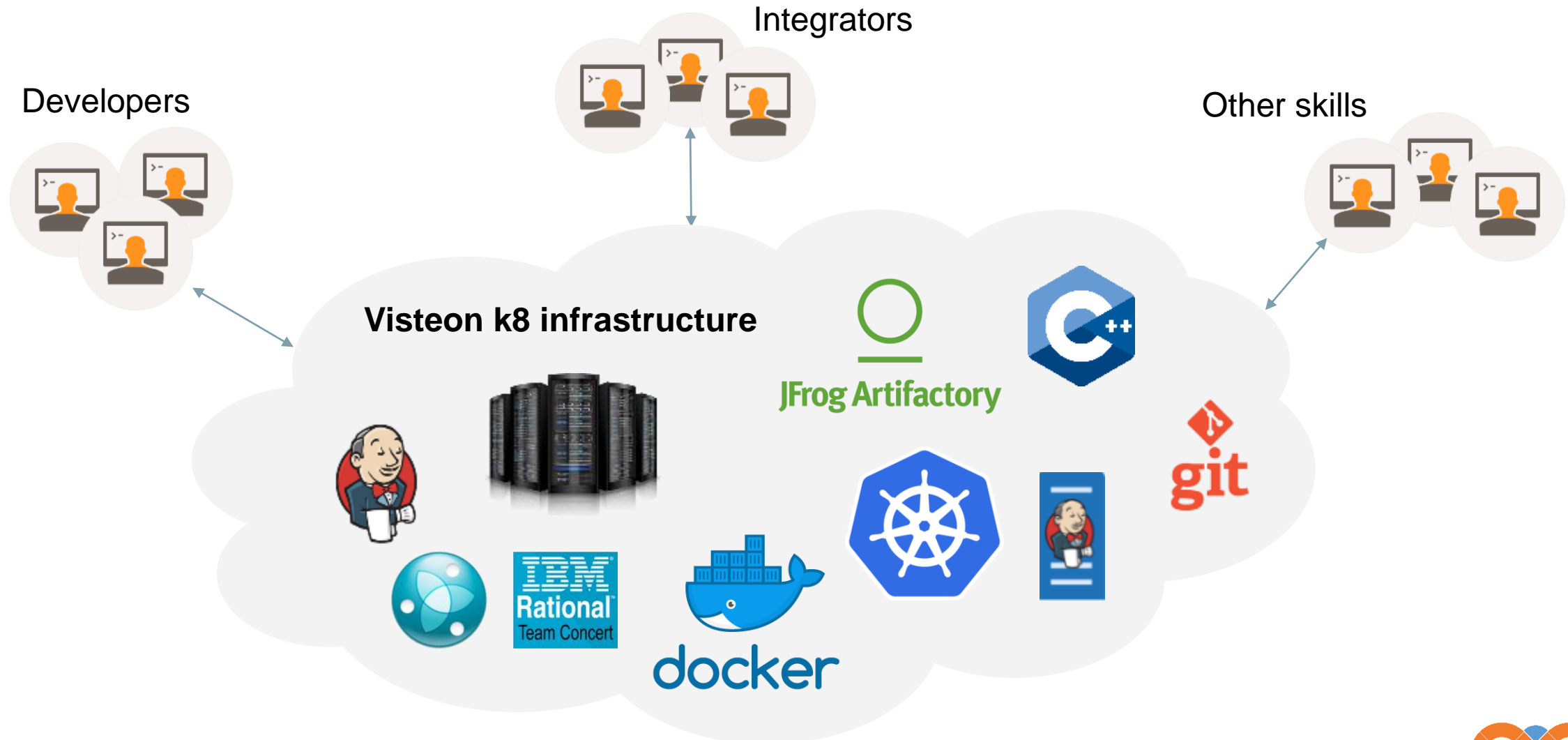
Kubernetes Components

- API Server
- Controller
- Scheduler
- Etcd
- Pods
- Kubelet

Kubernetes cluster Build Management



Kubernetes cluster cross projects and skills usage



DEVOPS

What is DevOps?

Specifics in the Automotive industry

DevOps in Visteon

Infrastructure efficiency

➤ **CI/CD Deployment**

Monitoring

Q&A

21



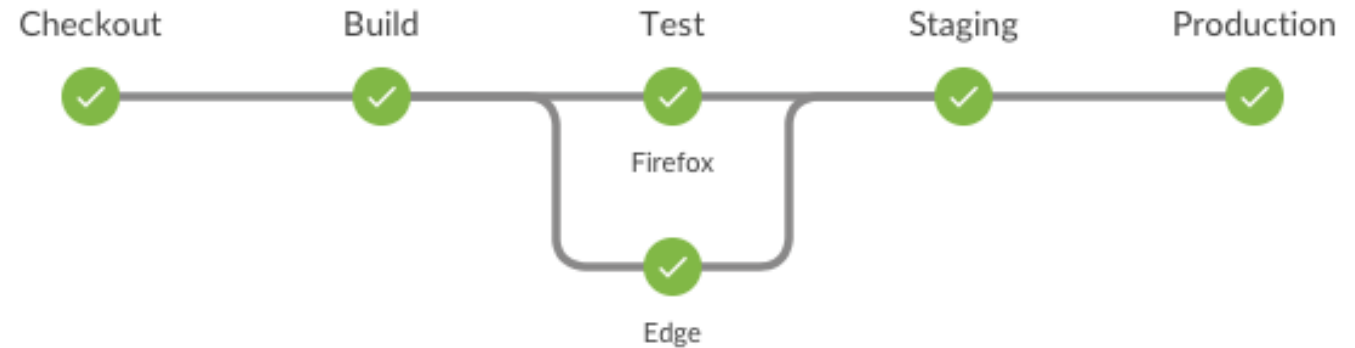
What is Jenkins?

- Jenkins is open source automation server
- Written in Java
- Server based application
- Orchestrate almost entire Software Development Cycle
- Continuously build and test your software projects
- Thousands open source plugins that integrate Jenkins with other DevOps Tools like Git, Docker, Kubernetes and much more!



What is Jenkins Pipelines?

- Jenkins Plugins features
- Chains different stages (pipes) together
- Configured as a code
- Flexible and fully customizable flow
- Provides ability to parallel building
- Ability to pause and wait for human input
- Allows to restart from specific stage (pipe)
- Can survive Jenkins restart / crash



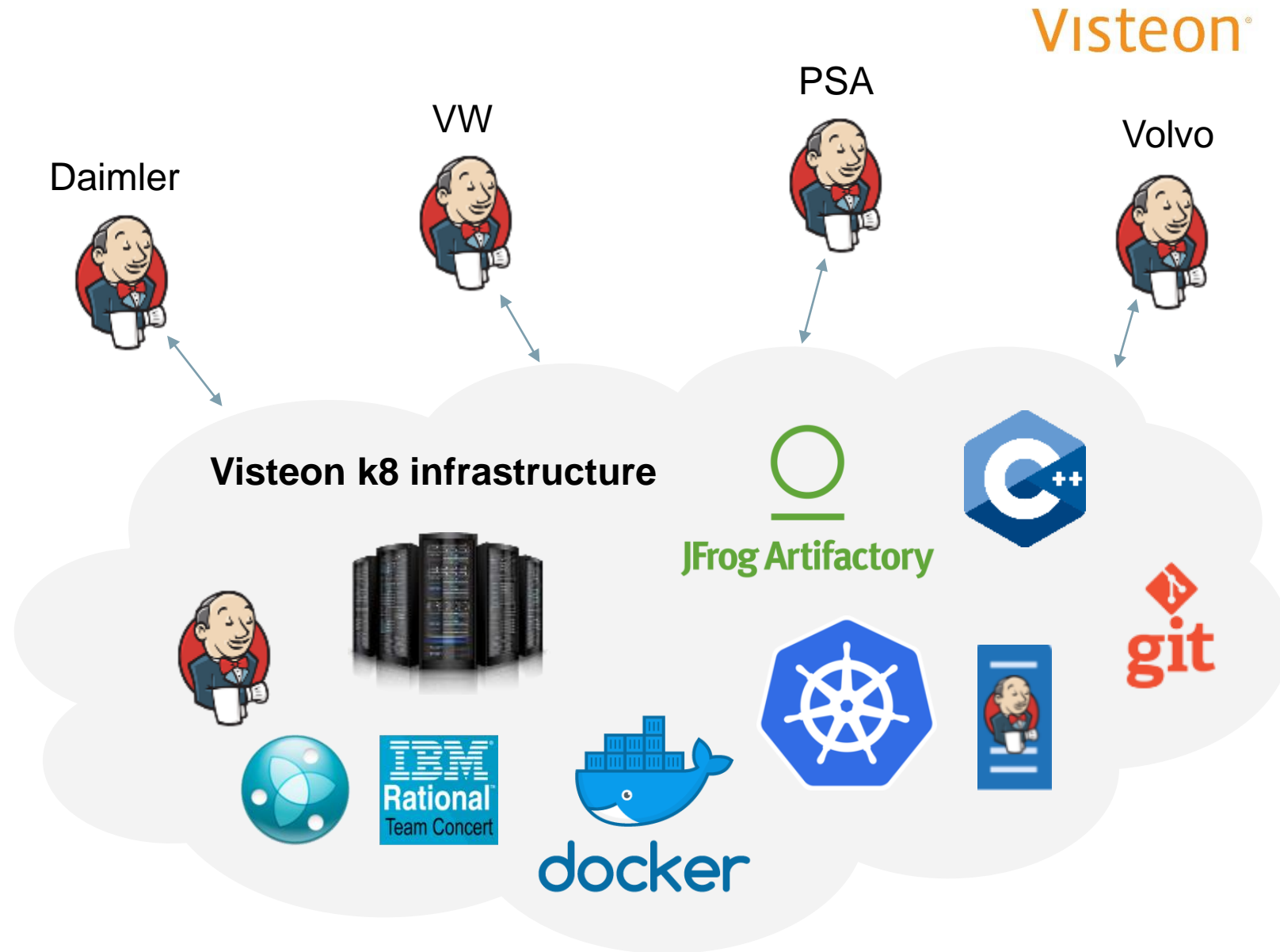
What is Jenkins Shared Library?

- A separate SCM repo that contains reusable custom steps that can be called from Pipelines
- Loaded and used as code libraries for Jenkins Pipelines
- Modifications made to a shared library custom step are applied to all Pipelines that call that custom step
- Configured once per Jenkins instance
- Cloned at build time



Challenges

- How to maintain and manage existing and newly upcoming Jenkins masters
- How to maintain the growing number of projects onboarded on the k8s cluster
- How to reduce overhead of the DevOps team
- How to increase the reusability across different projects



Inspired by Lord of the Rings...

Visteon®

Jenkins Master

You will serve
me..!

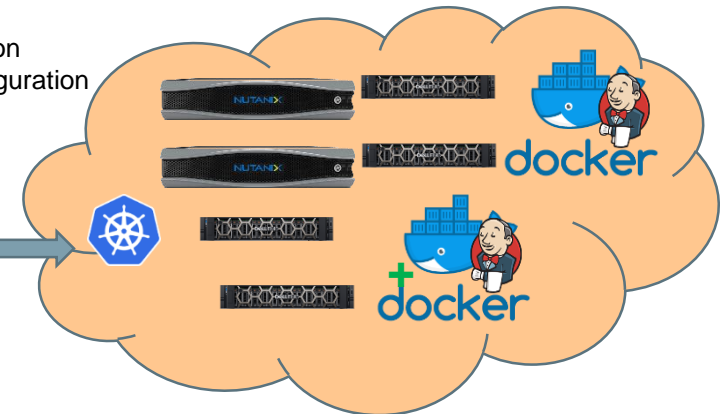


Jenkins Master as a code

```
1 apiVersion: v1
2 kind: PersistentVolumeClaim
3 metadata:
4   name: jenkins-master-{{jenkinsMasterName}}-pvc
5   namespace: jenkins-master
6 spec:
7   accessModes:
8     - ReadWriteMany
9   resources:
10     requests:
11       storage: 50Gi
12 ---
13 kind: PersistentVolume
14 apiVersion: v1
15 metadata:
16   name: jenkins-master-{{jenkinsMasterName}}-pv
17 spec:
18   capacity:
19     storage: 50Gi
20   nfs:
21     server: {{nfsServer}}
22     path: {{nfsServerPath}}
23   accessModes:
24     - ReadWriteMany
25   persistentVolumeReclaimPolicy: Recycle
26   volumeMode: Filesystem
27   status:
28     phase: Available
29 ---
30 apiVersion: apps/v1
31 kind: Deployment
32 metadata:
33   name: jenkins-master-{{jenkinsMasterName}}
34   namespace: jenkins-master
35 spec:
36   replicas: 1
37   selector:
38     matchLabels:
39       app: jenkins-master-{{jenkinsMasterName}}
40   template:
41     metadata:
42       labels:
43         app: jenkins-master-{{jenkinsMasterName}}
44     spec:
```

Predefined configurations as template:

- Jenkins Master with Docker
- Preinstalled plugins
- LDAP configured
- Admin user permissions setup
- Initial Jenkins Job for structure creation
- Optimal Jenkins Master service configuration
- Easy Jenkins Master version update



Jenkins takes 3 parameters:

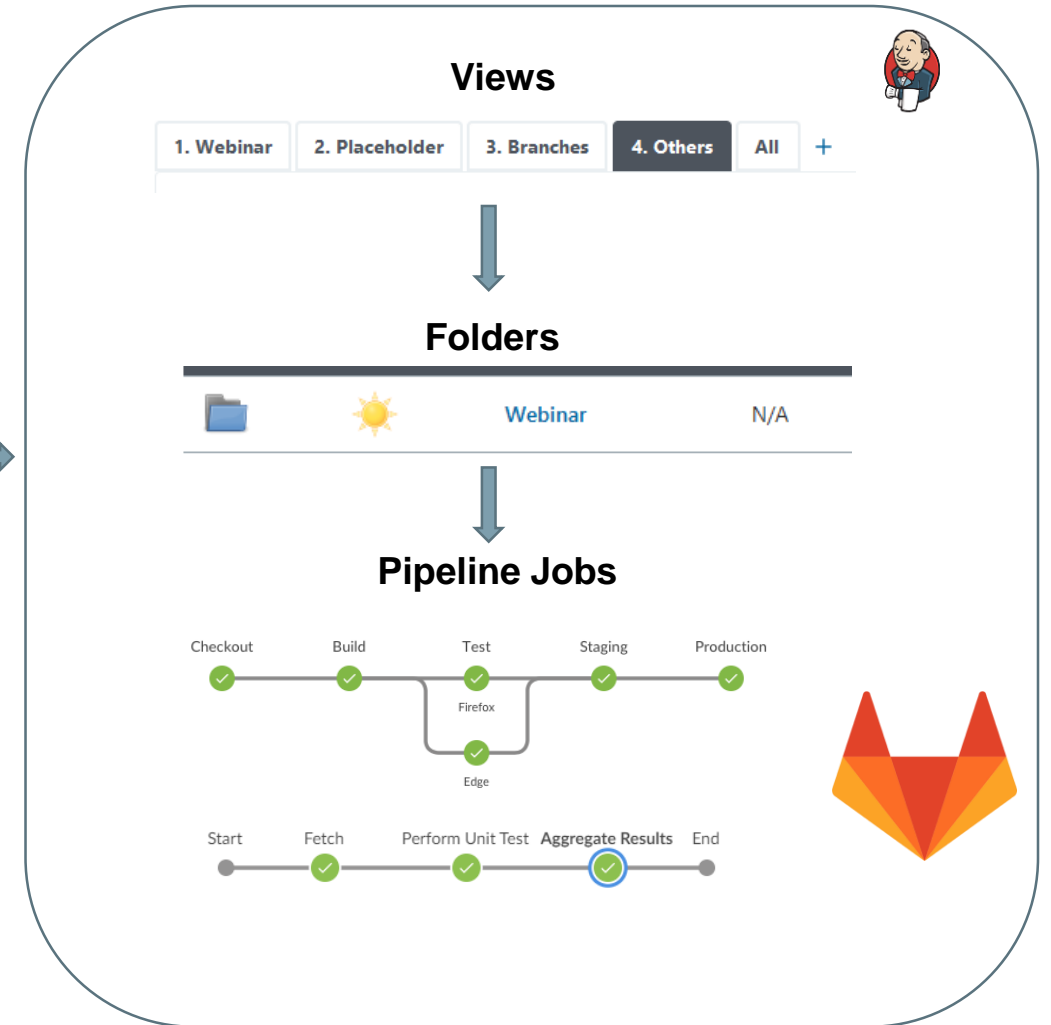
- **jenkinsMasterName** - the name of the new Jenkins Master instance
- **jenkinsPort** - http port of the new Jenkins Master instance (range 30010-30100)
- **agentPort** - tcp port of the new Jenkins Master agents (range 31010-31100)

Jenkins Structure as a code

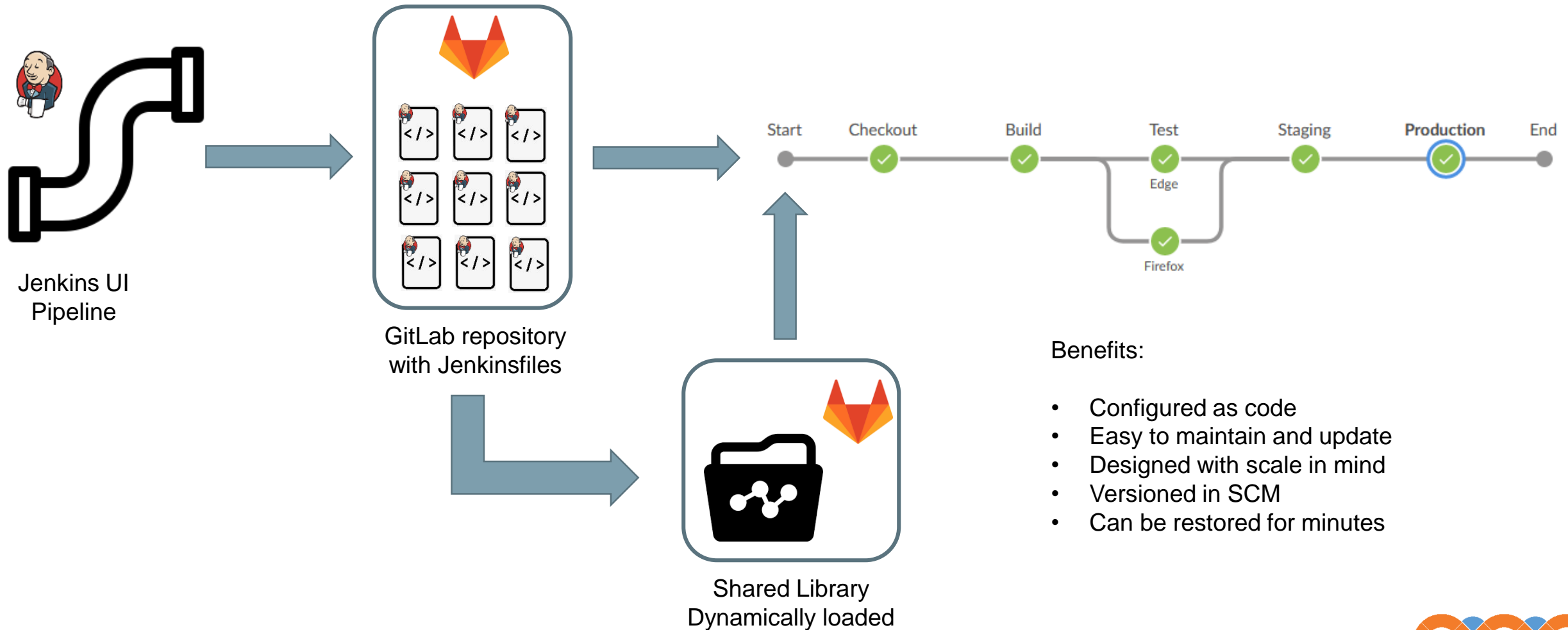


Jenkins DSL
template job

```
4 // 1. Integration
5 folder("integration") { displayName("Integration") }
6   folder("integration/${project}") {}
7   pipelineJob("integration/${project}/build") {
8     displayName("Build")
9     definition {
10       cpsScm {
11         scm {
12           git {
13             remote {
14               url("${repo}")
15               credentials("${credential}")
16             }
17             branches("${branch}")
18             scriptPath("${jenkinsfiles}/${project}/integration/build.jenkinsfile")
19             extensions {} // required as otherwise it may try to tag the repo, which you may not want
20           }
21         }
22       }
23     }
24   }
25   pipelineJob("integration/${project}/test") {
26     displayName("Test")
27     definition {
28       cpsScm {
29         scm {
30           git {
31             remote {
32               url("${repo}")
33               credentials("${credential}")
34             }
35             branches("${branch}")
36             scriptPath("${jenkinsfiles}/${project}/integration/test.jenkinsfile")
37             extensions {} // required as otherwise it may try to tag the repo, which you may not want
38           }
39         }
40       }
41     }
42   }
43   pipelineJob("integration/${project}/promotion") {
44     displayName("Promotion")
45   }
46 }
```



Pipelines as a Code



DEVOPS

What is DevOps?

Specifics in the Automotive industry

DevOps in Visteon

Infrastructure efficiency

CI/CD Deployment

➤ **Monitoring**

Q&A



What is monitoring?

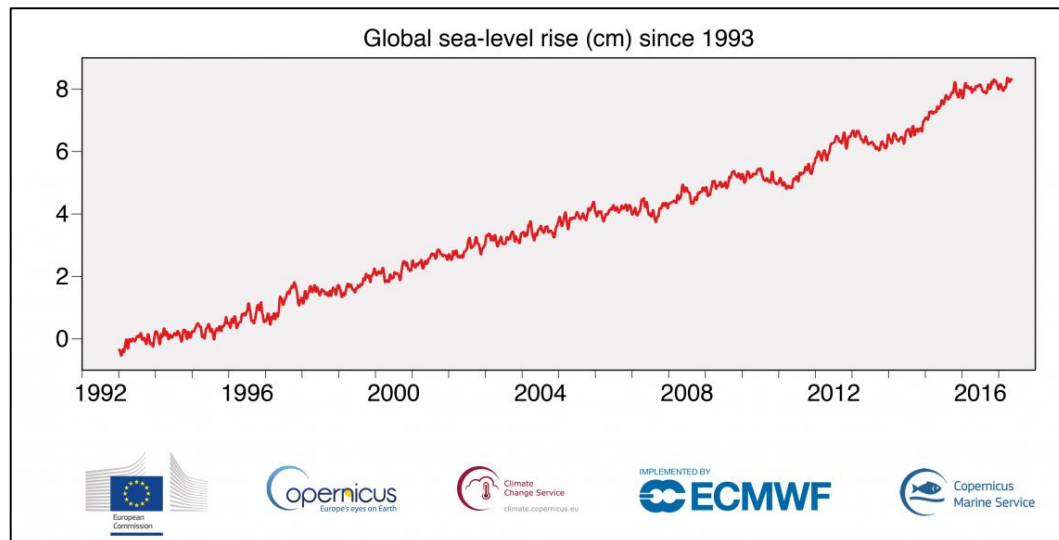
monitor *verb*

Oxford **Learner's Dictionaries**

🔊 /'mɒnɪtə(r)/

🔊 /'mɑːnɪtə/

- 1 to watch and check something over a period of time in order to see how it develops, so that you can make any necessary changes
 - **monitor something** *Each student's progress is closely monitored.*
 - *The authorities will continue to monitor the situation.*
 - **monitor somebody** *The patient is carefully monitored.*
 - **monitor what, how, etc...** *We need to monitor how the situation develops.*

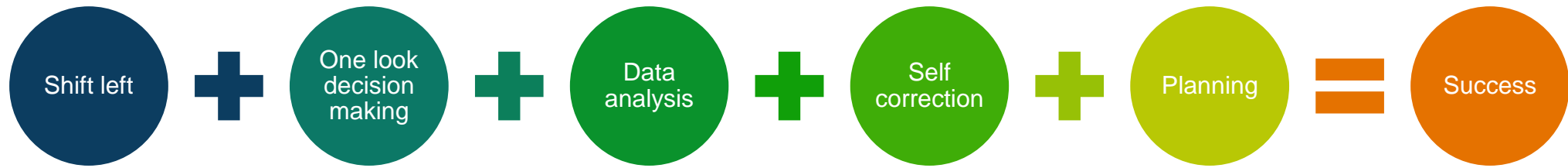
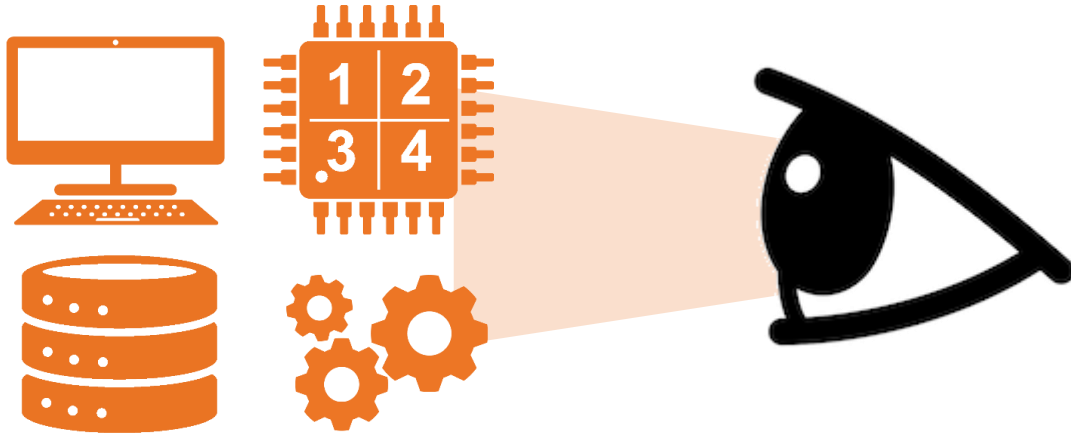


© Mazda Motor Corp.



Why monitoring?

Visteon®



What to monitor?

Visteon®



Infrastructure



Processes



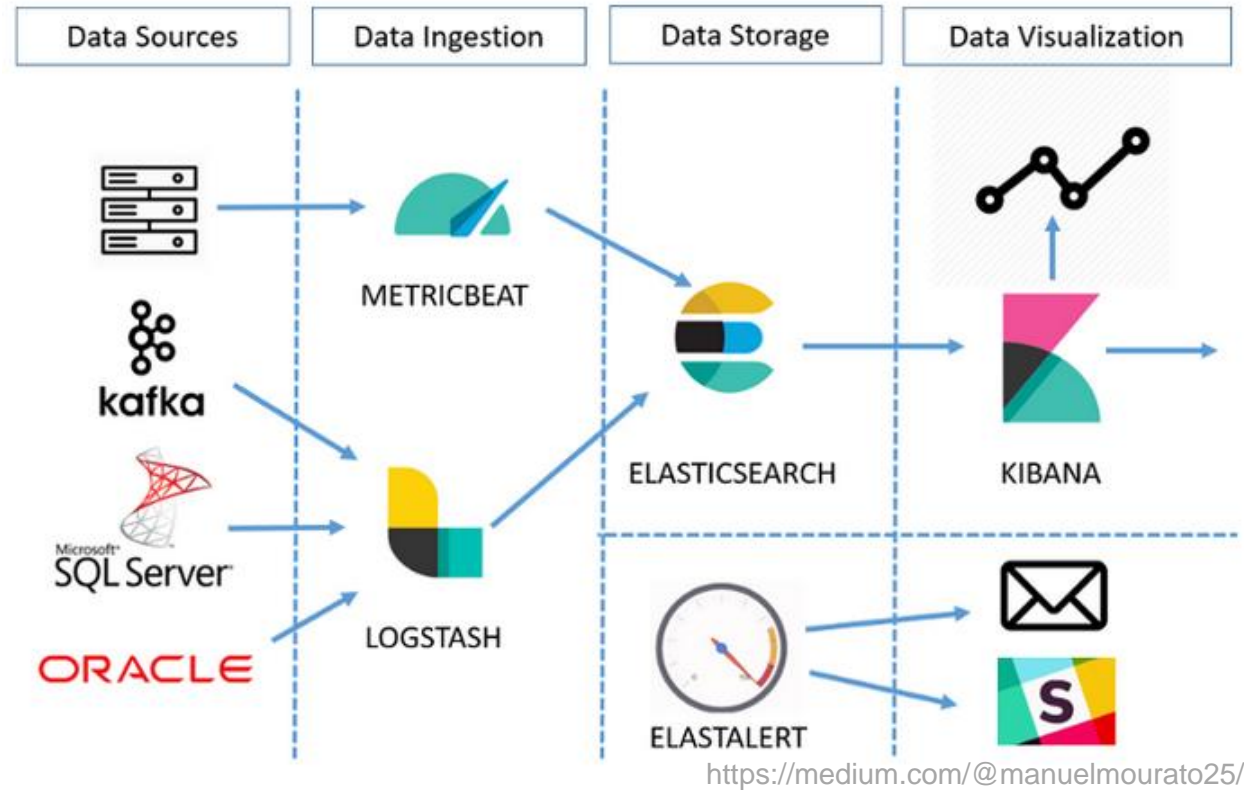
Software quality and test results



Monitoring system overview

Typical 4-layer architecture:

- Data sources
 - Reusing existing data
- Data collection services
 - Event driven
 - Stateless
- Data storage
 - Distributed and persistent
 - Timeseries (indexed by time)
- Data visualization
 - Dashboards/Metrics
 - Alerts
 - Custom processing (e.g. analytics)



Our approach

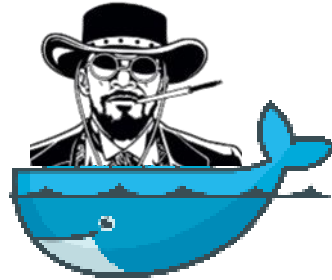
Data Sources



JFrog Artifactory



Data Collection



Data Storage



Data Visualization



A blend of adapted open source and home grown solutions



Dashboards

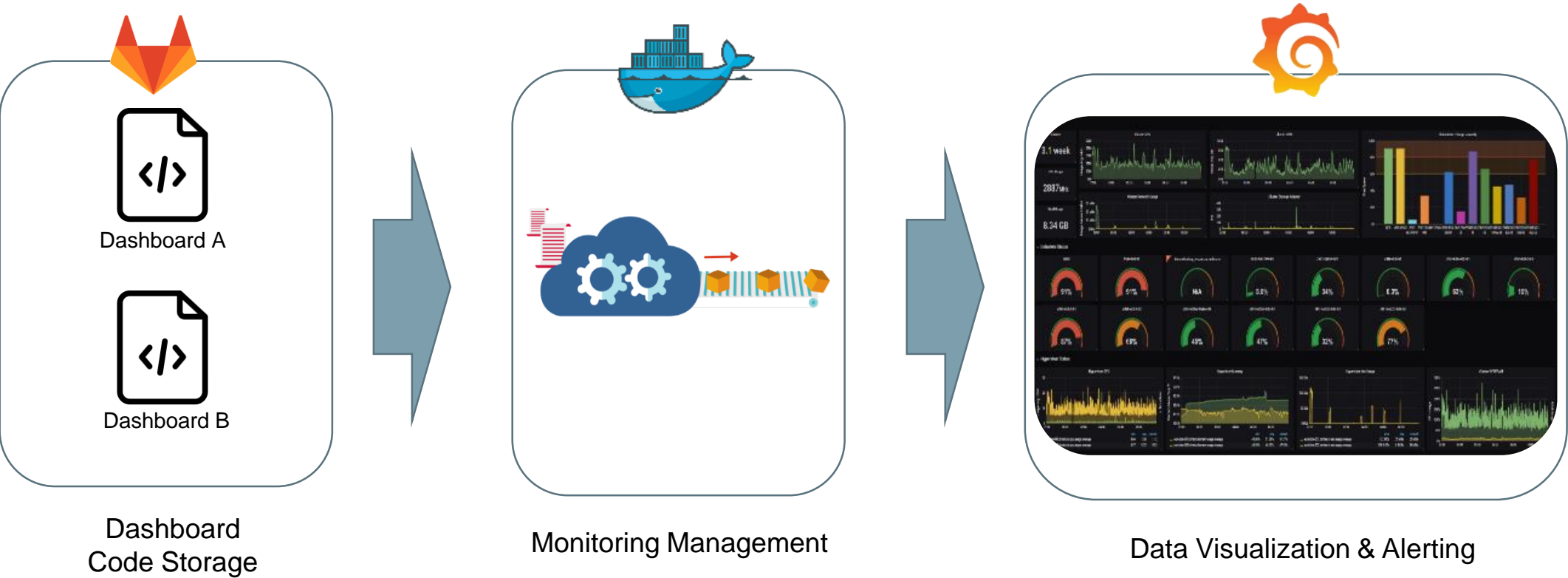
- Built with Grafana
- Data visualization from different data sources
- Primary data analysis interface
- Rich APIs
- Alerting via email or custom push event
- Infinite flexibility by highly customizable panels



<https://www.reddit.com/r/grafana/>



Dashboard as a code

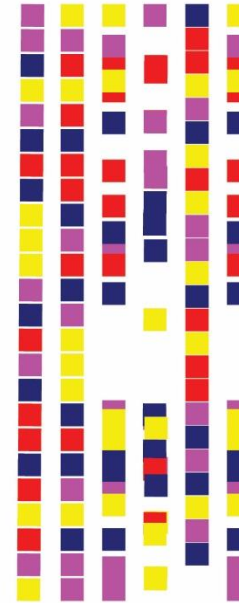


- Big data analytics
- Trend detection
- Predictive maintenance
- Machine learning
- Heterogeneous

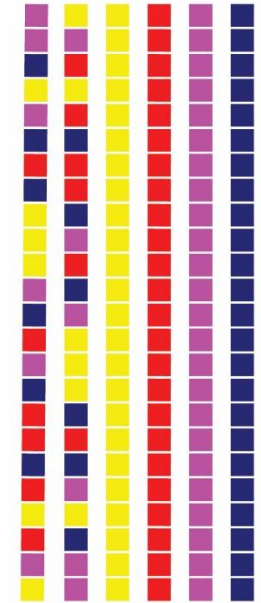
BIG DATA



ANALYTICS



DECISIONS

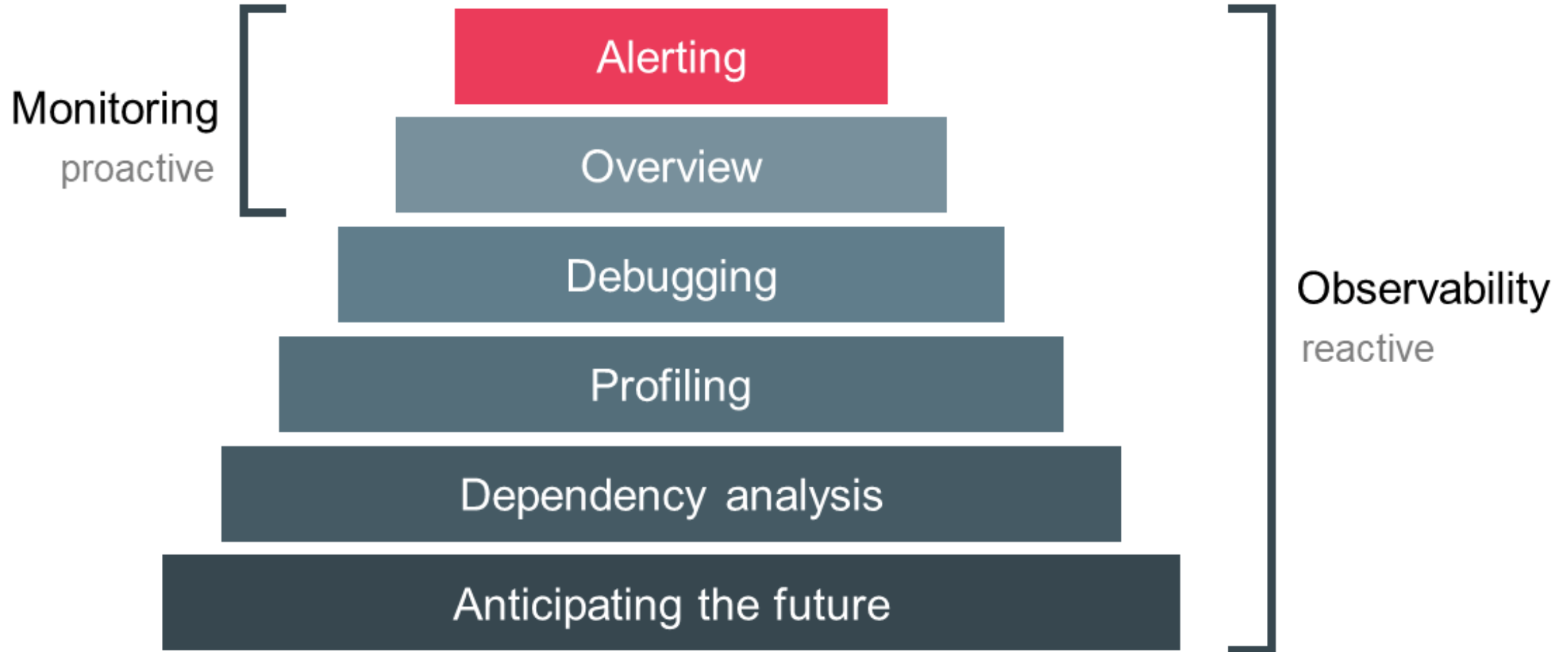


Utilize the available data to create more value



Monitoring and beyond...

Visteon®



DEVOPS

What is DevOps?

Specifics in the Automotive industry

DevOps in Visteon

Infrastructure efficiency

CI/CD Deployment

Monitoring

➤ **Q&A**



Small part of our products on the roads

Visteon®





Visteon®

