#RedHatOSD

RED HAT FORUMS

Challenges and benefits of a Cloud Native content delivery platform

Niko Usai
System Architect & Innovation Evangelist - Sourcesense
Raffaele Camanzo
Business Operations Manager - Sourcesense
03/12/2019

Red Hat | SOURCESENSE

SOURCESENSE

CORPORATE ADVISORY

Assessment & Migration
Methodologies & Processes

DIGITAL EVOLUTION

Cloud Native
Microservices
DevOps
Machine Learning

Agenda

#RedHatOSD

Business case

Solution

Cloud Native

Leading Italian financial newspaper

With several online products

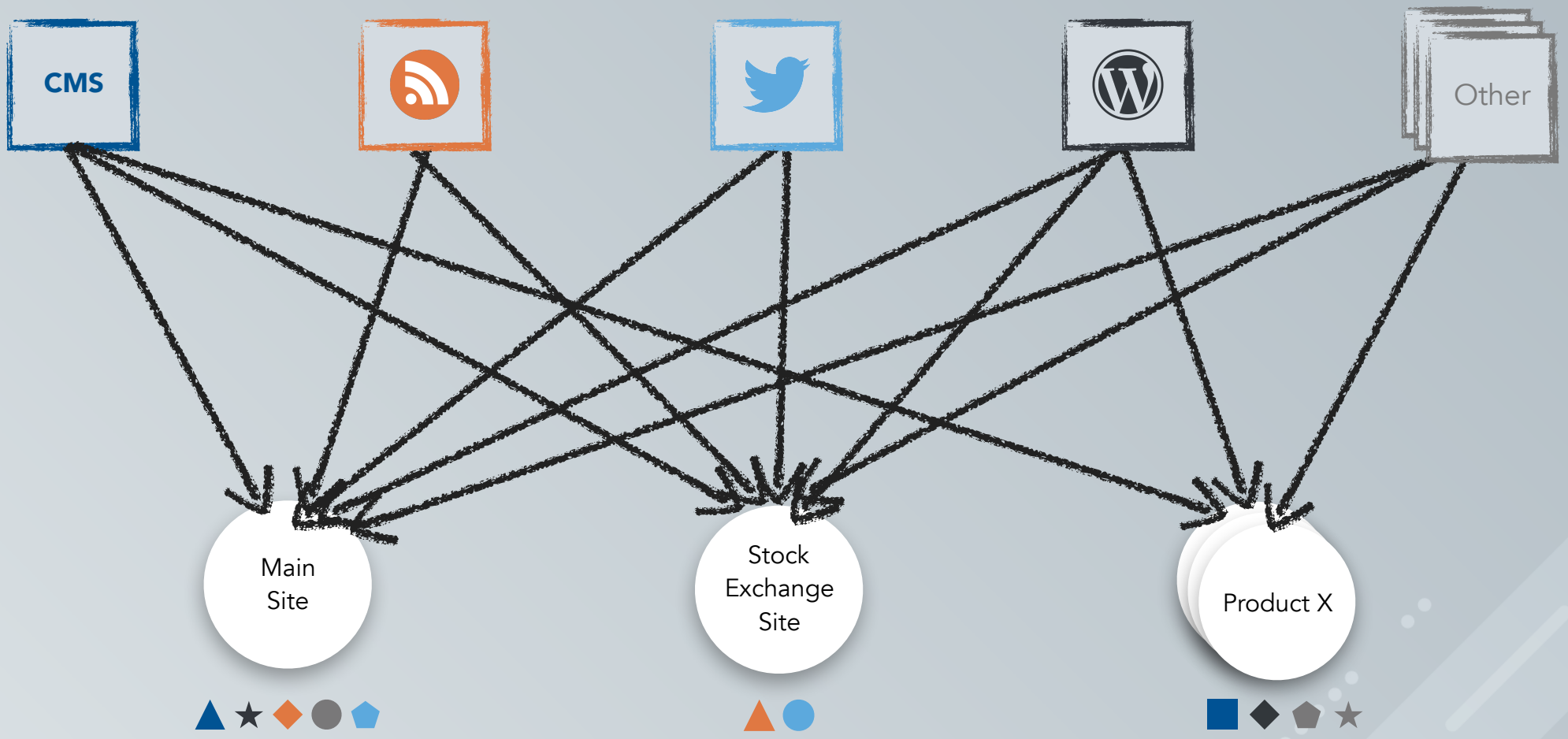
- Review the publishing pipeline
- Improve ease of release of new products
- Enhance scalability

Content from several sources

Many different products

composed by a mix of content sources

Complex publishing process



What doesn't works?

- Changes are expensive
  Tight coupling between the sources and the product
- Slow development cycle
  Every new product has to reinvent the wheel to consume from a source
- No unified view of the whole content
  No unique source of truth
- Hard to scale
  Due to interconnected legacy systems

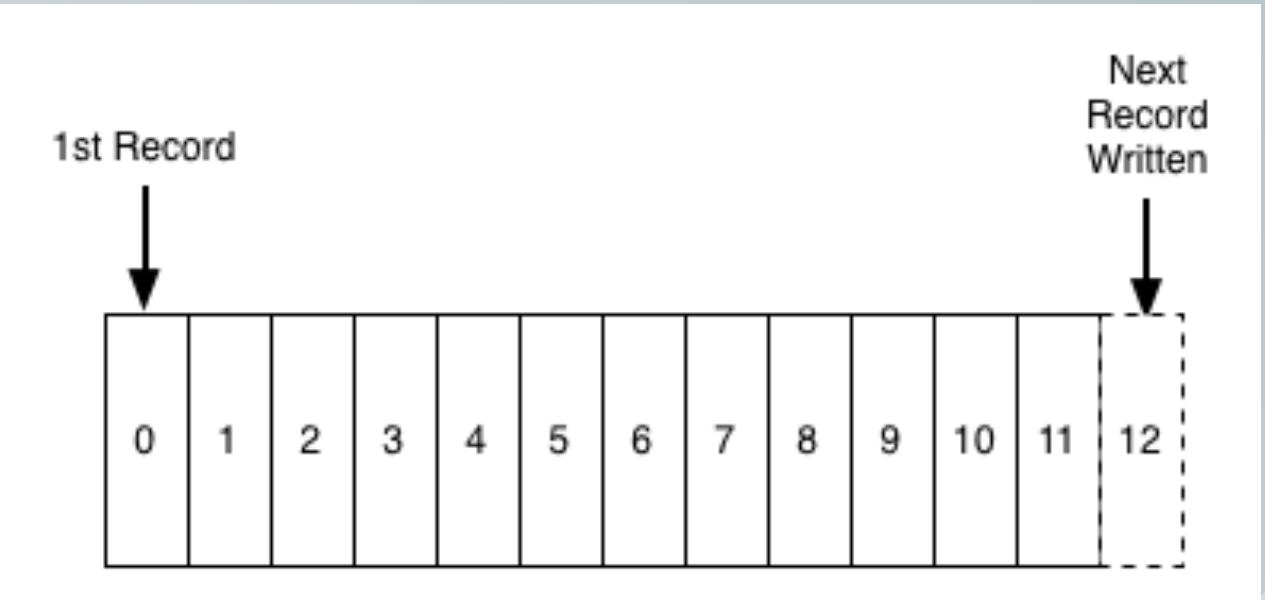
SOLUTION

Composition of 2 patterns

- Log-based Architecture
- Backend For Frontend

What is the log?

It is an append-only, totally-ordered sequence of records ordered by time



What is it good for?

- **Database replica**
  Internal structure to propagate change to distributed database
- **Data integration**
  Evolution of ETL data flow
- **Realtime Processing**
  Infrastructure for continuous data processing -> Stream Processing
- **Distributed system design**
  Event driven systems and CQRS

What is it good for?

- Database replica
  Internal structure to propagate change to distributed database
- Data integration
  Evolution of ETL data flow
- Realtime Processing
  Infrastructure for continuous data processing -> Stream Processing
- Distributed system design
  Event driven systems and CQRS

Using the log as "source of truth"

We can produce a **persistent**, **re-playable** record of content **history**.

Consumer of the log can create **different** materialised **views** of the source content

No distinction between **past** data and **live** data

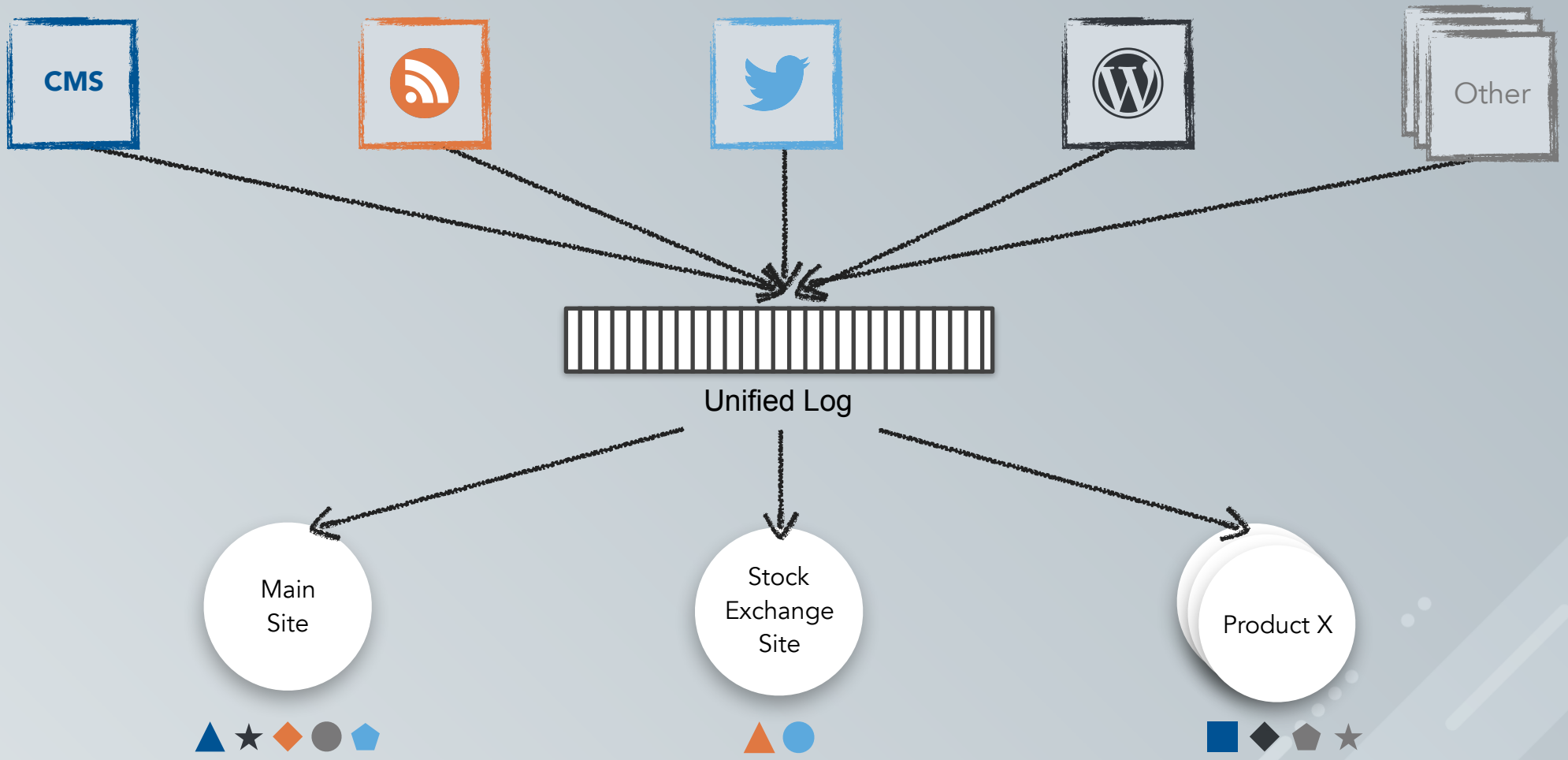
Log-based Architecture

It is a **distributed, horizontally-scalable** and **fault-tolerant** implementation of a write-ahead **log**

Developed at **LinkedIn** for data integration /processing

"It's Okay To Store Data In Apache Kafka" cit. **Jay Kreps**

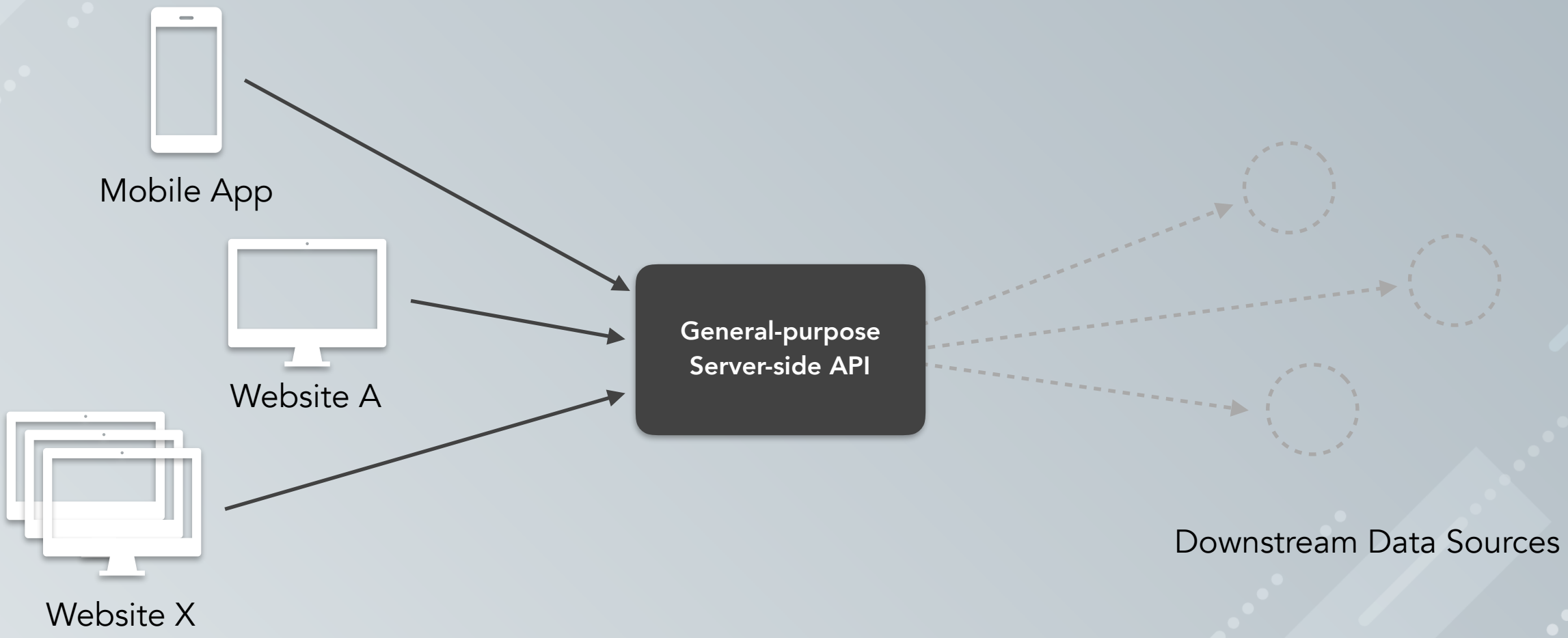
The whole content as a source of truth



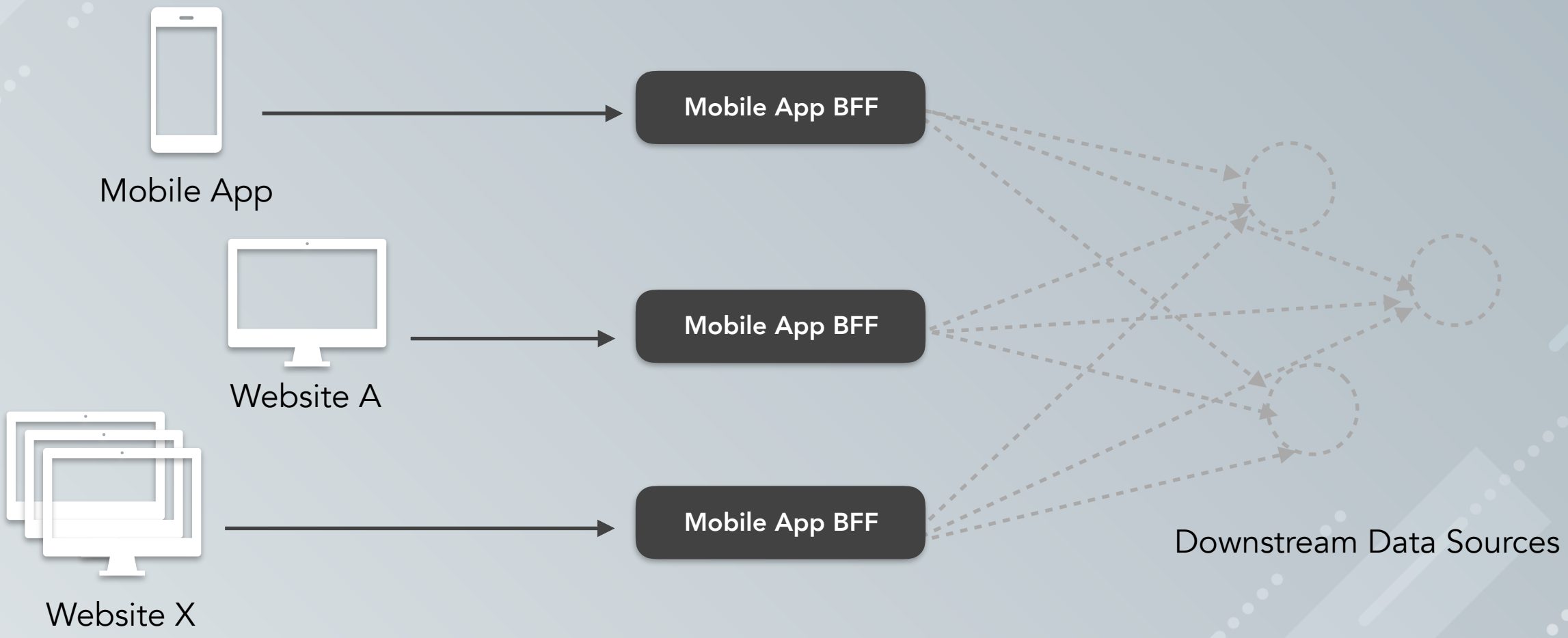
Composition of 2 patterns

- Log-based Architecture
- Backend For Frontend

Traditional General-Purpose Backend



BBF Approach



Advantages

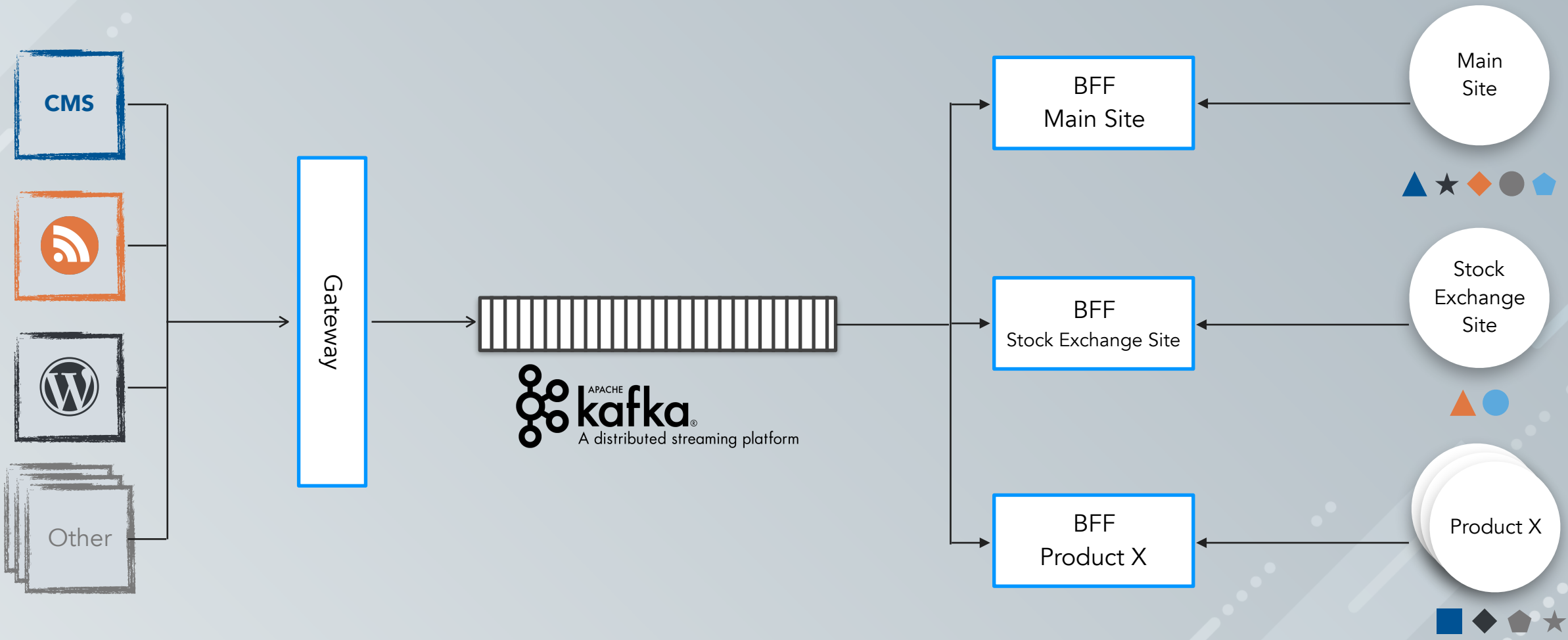
- **Performance**
  The client of the API gets **exactly** what it needs
- **Scalability optimisation**
  Different client scale independently so their BFF
- **Team scaling**
  Team can work vertically, no side effect impacts for new features

Wrapping it up

With a *log-based* publishing pipeline we have **materialised views** of aggregated content for the **specific product** from a single **source of truth**

We then create a BFF on top of that derived data, so we can **scale it horizontally** easily

Wrapping it up



What doesn't works?

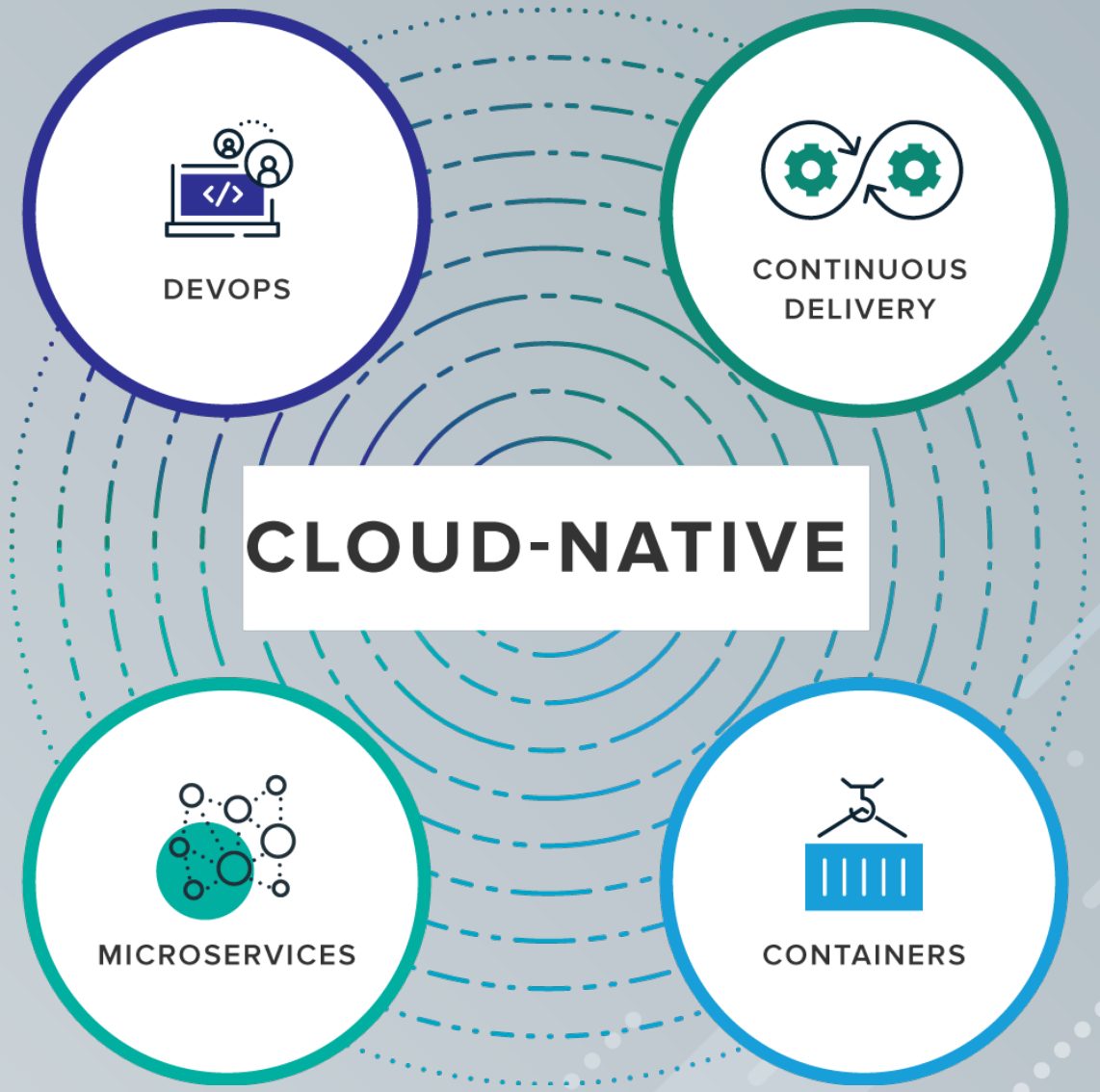
- Changes are expensive
  Tight coupling between the sources and the product
- Slow development cycle
  Every new product has to reinvent the wheel to consume from a source
- No unified view of the whole content
  No unique source of truth
- Hard to scale
  Due to interconnected legacy systems

CLOUD NATIVE

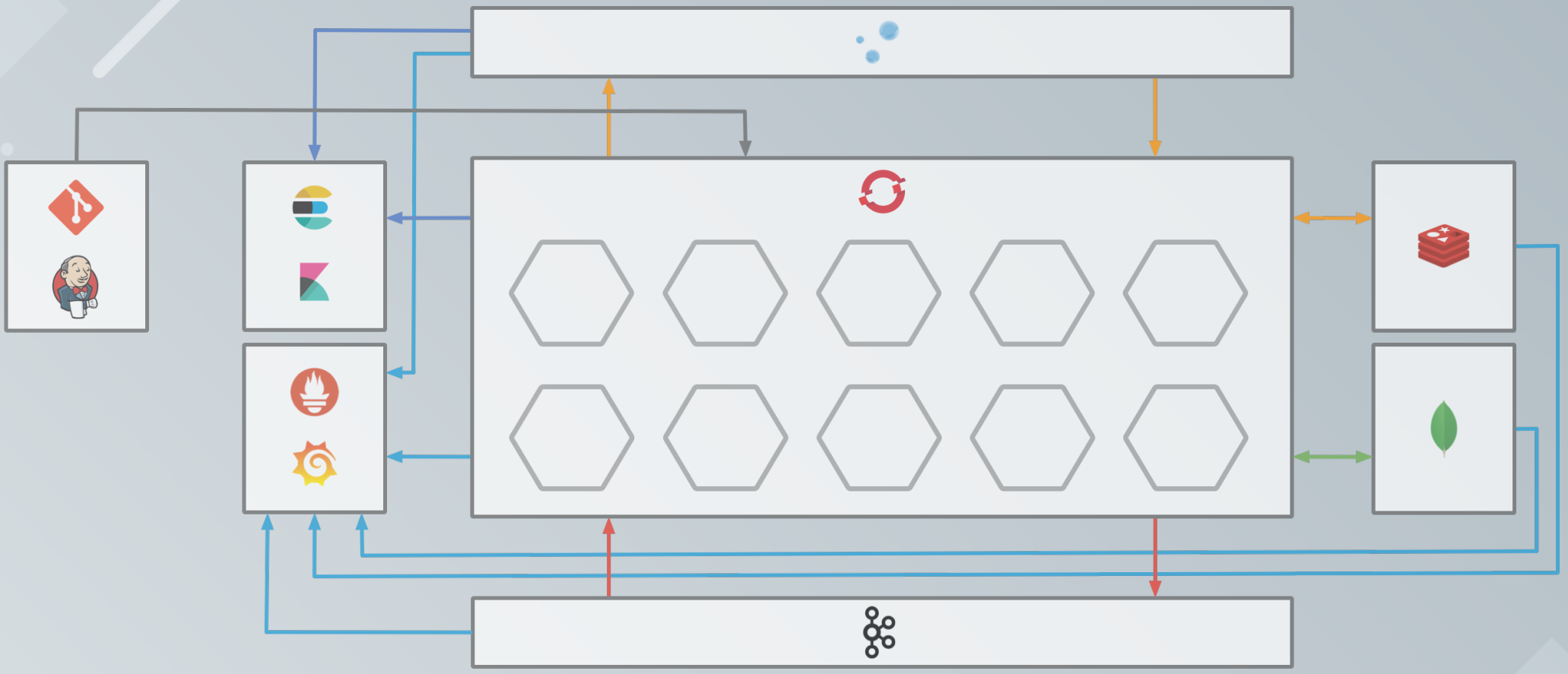
Cloud Native

https://github.com/cncf/toc/blob/master/DEFINITION.md

- Scalable applications
- Resilient, manageable and observable loosely coupled systems
- Infrastructure automation



Cloud Native



Cache | Data projection persistence (R) | System state changelog (W) | Monitoring and alerting (system) | Logging and monitoring (application) | CI/CD

Openshift installation

- 15 workload node
- 4 Infrastructure node (router & registry)
- 2 master node

Openshift benefits

- **Efficient use of hardware**
  Workload management and autoscaling
- **Observability**
  We collect logs and metrics from both infrastructure and applications, aggregate them to monitor and perform analysis
- **CI/CD**
  Fully automated software development pipelines

#RedHatOSD

RED HAT FORUMS

THANK YOU

linkedin.com/company/Red-Hat

facebook.com/RedHatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat | SOURCESENSE