Apache Spark in the Cloud

Zbyněk Roubalík
Senior Quality Engineer, Red Hat

February 15 2018

Technologies

- Apache Spark
- Docker
- Kubernetes
- OpenShift

Apache Spark in the Cloud

aka

How to create and deploy Apache Spark applications to cloud native environments like OpenShift

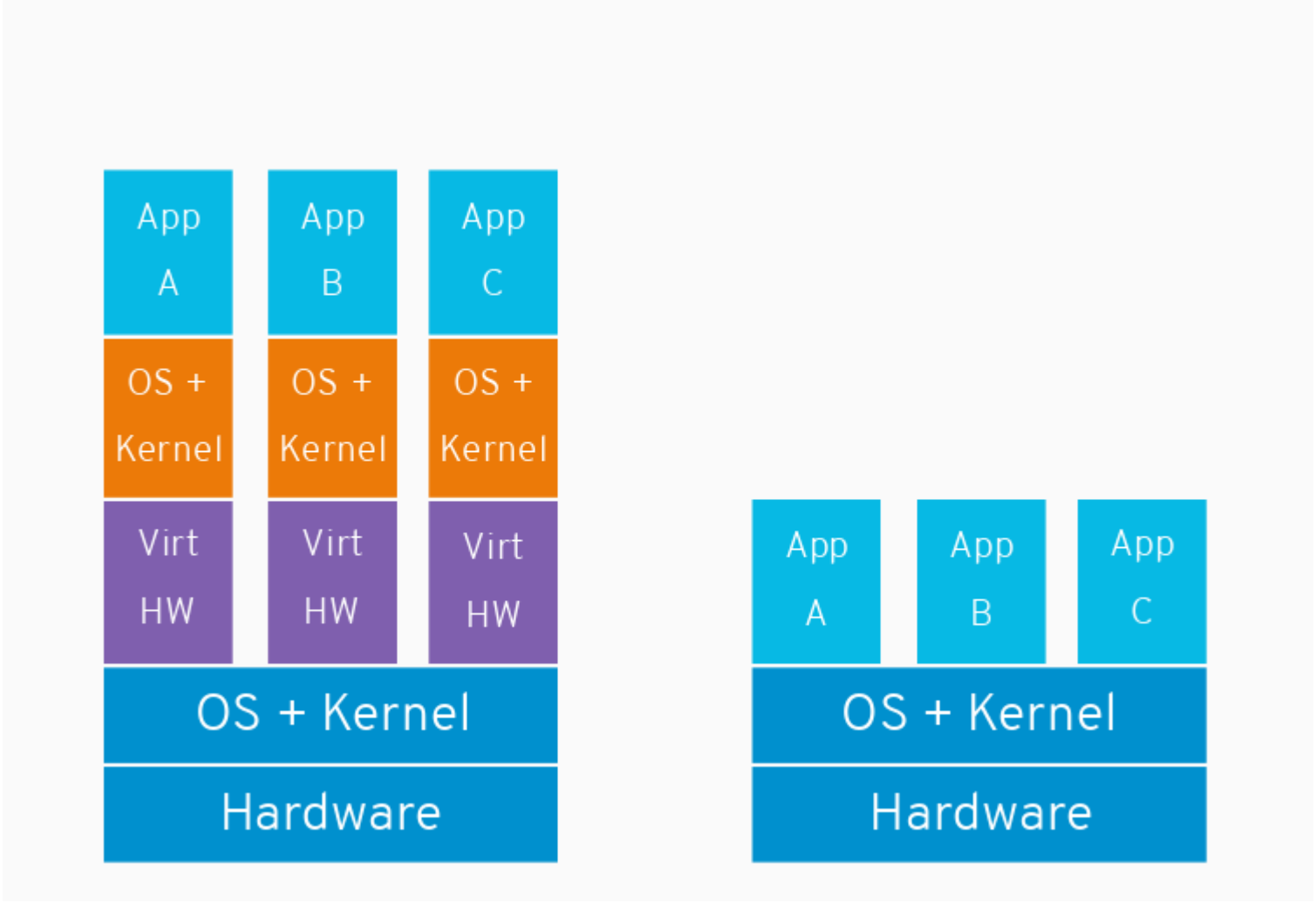
What is cloud native?

- Containerized
- Dynamically orchestrated
- Microservice oriented

- www.cncf.io/about/faq

Containers

- A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.
  - https://www.docker.com/what-container



VM vs **Containers**

Containers

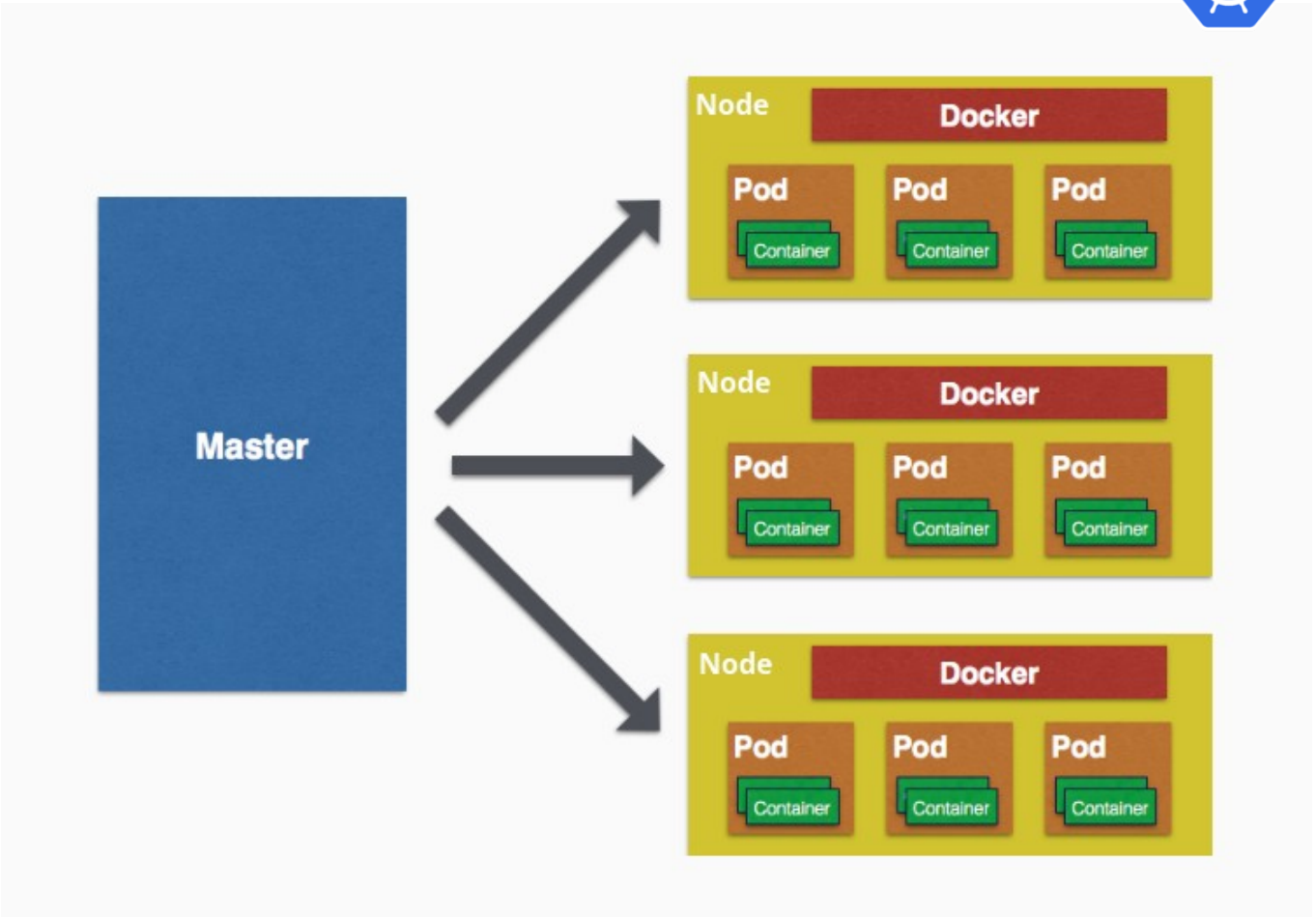
- Cloud vs standard deployment model
- Pets vs Cattle
- Developers + Operations (Admins) → DevOps
- Docker

Kubernetes

kubernetes

- Container cluster manager

Kubernetes

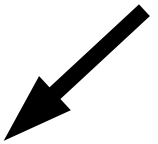


- Based on **etcd** – distributed clustered key value store
- Smallest deployable unit is Pod

OpenShift

RED HAT® OPENSHIFT

- Open Source Container Application Platform
- Focused on application (not just containers as a concept) and developer experience

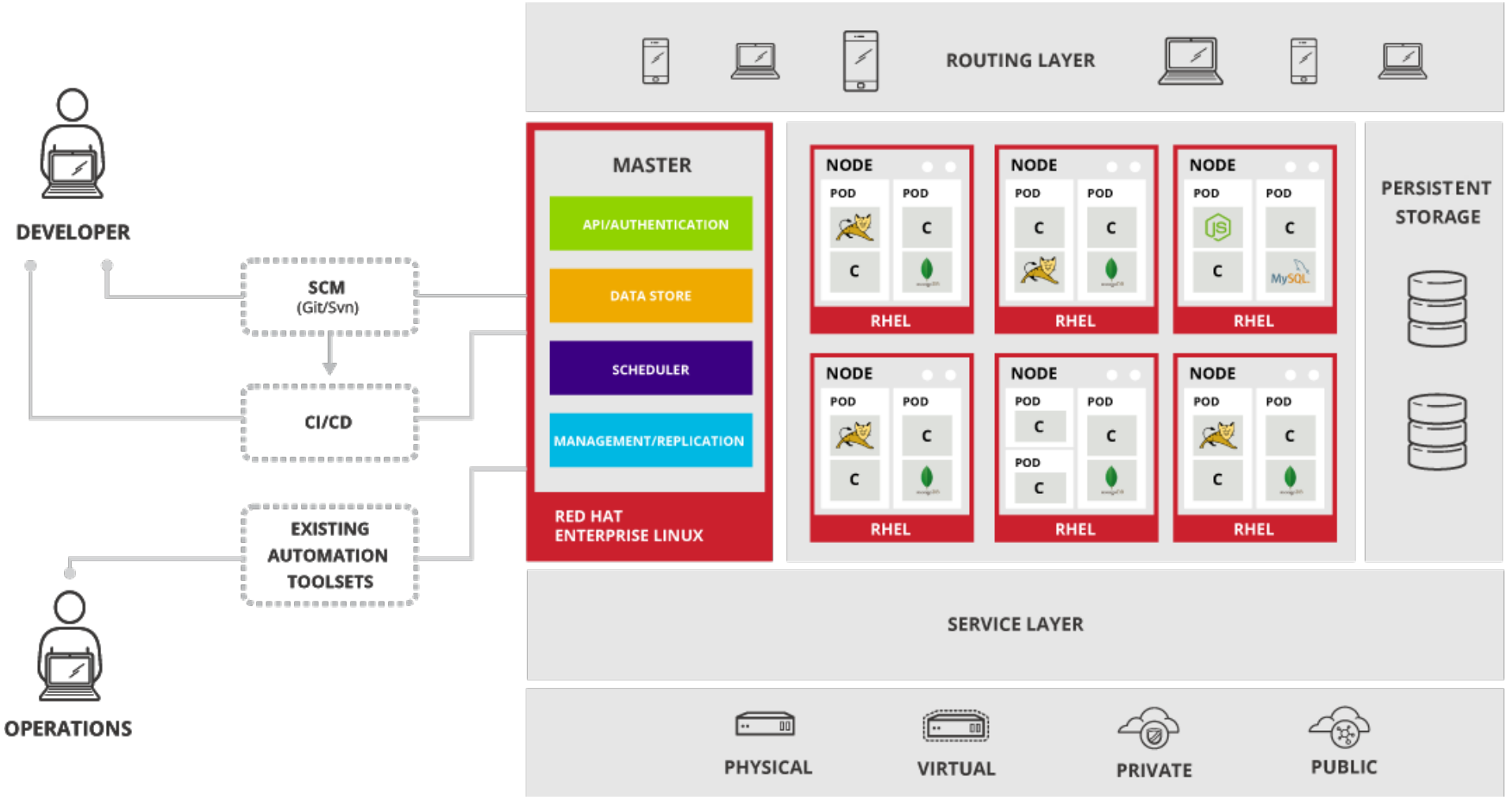


OpenShift

- Sits on the top of Kubernetes
- Source code, builds and deployments management
- **S2I - Source to Image**
- Application lifecycle management (CI/CD)
- Service catalog (Language runtimes, Middleware, Databases)
- Security

OpenShift architecture

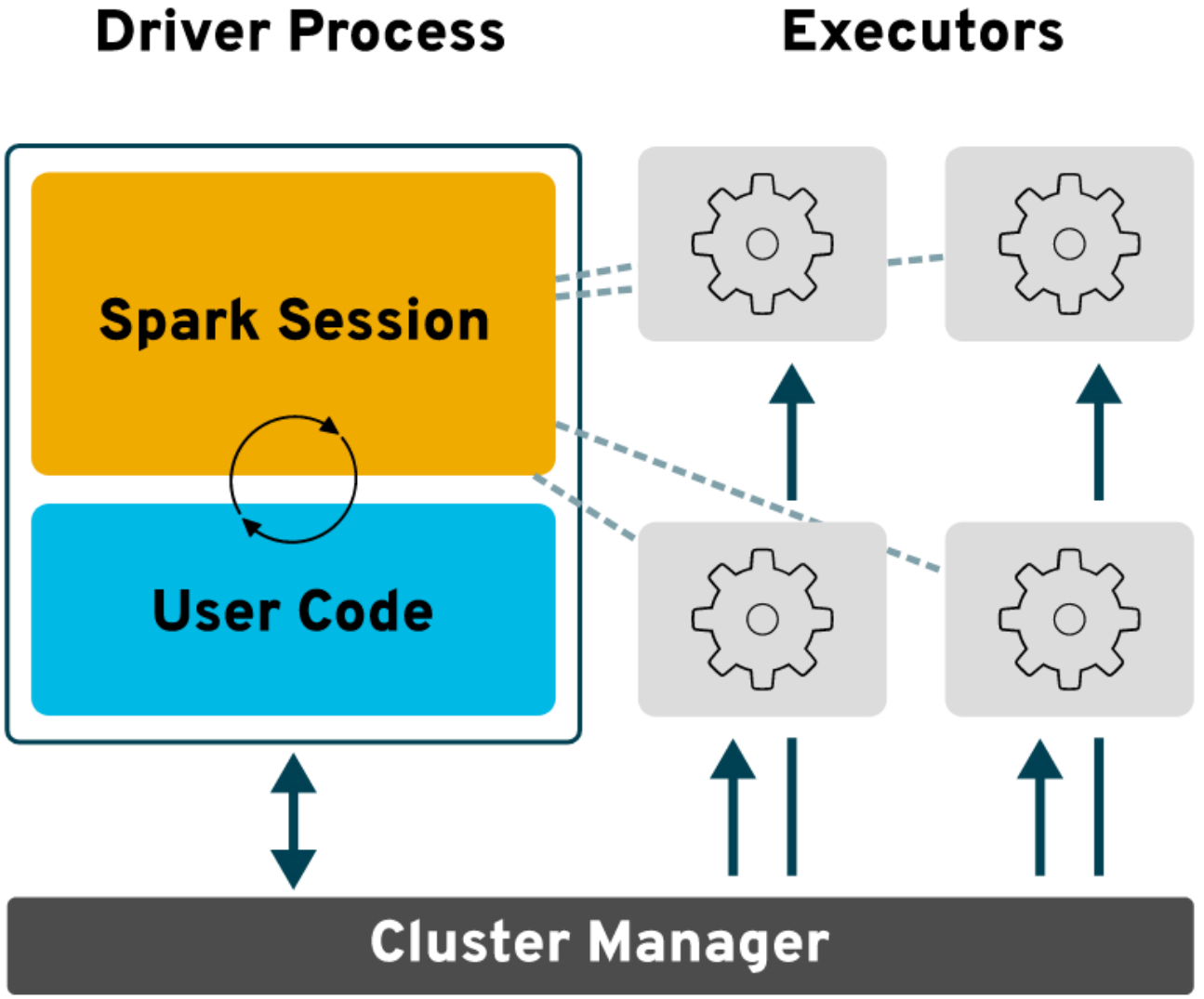
RED HAT® OPENSHIFT



Apache Spark

- Fast and general engine for large-scale data processing
- Distributed computation system
- Provides high-level APIs in Java, Scala, Python and R
- Supports a rich set of tools for Big Data, AI, ML
  - **Spark SQL** for SQL and structured data processing
  - **MLlib** for machine learning
  - **GraphX** for graph processing
  - **Spark Streaming**
  - **...**

General Spark architecture



How to interact with Spark

- Run an application

```
spark-submit --master=local[1] MyApp.py
```

- Start a REPL
  - Scala

```
spark-shell
```

  - Python

```
pyspark
```

  - R

```
sparkR
```


The fundamental Spark abstraction

Resilient distributed dataset (RDD)

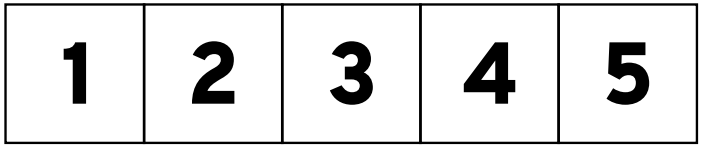
- are partitioned, lazy and immutable homogenous collections
  - partitioned
  - lazy
  - immutable

Resilient distributed dataset in action

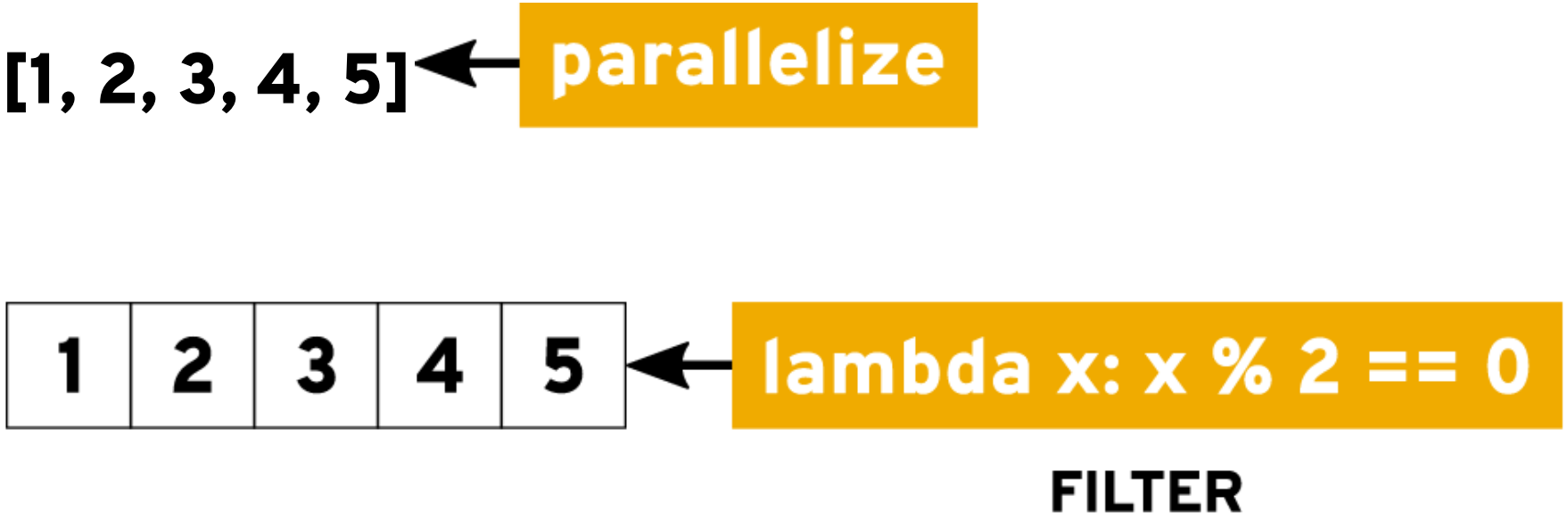
[1, 2, 3, 4, 5]

Resilient distributed dataset in action

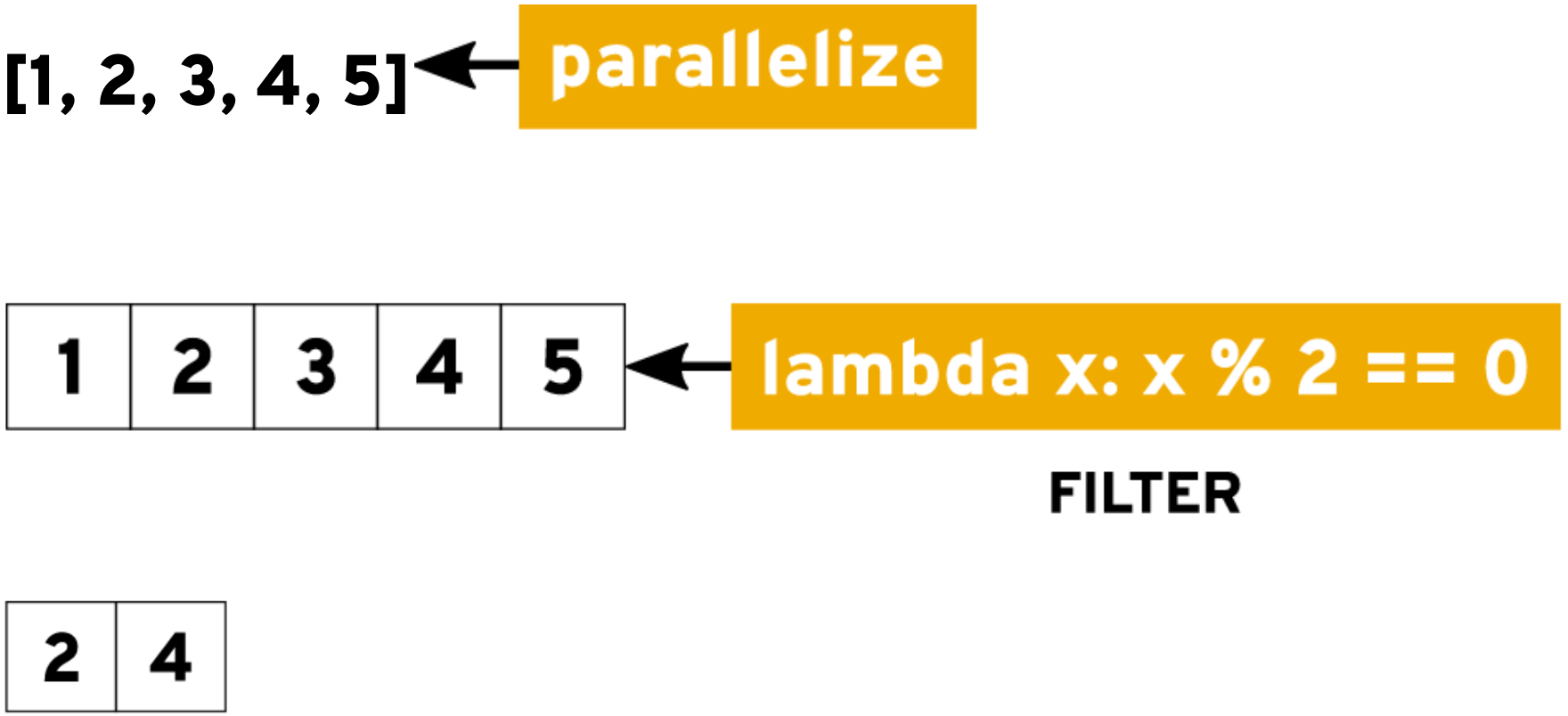
Resilient distributed dataset in action



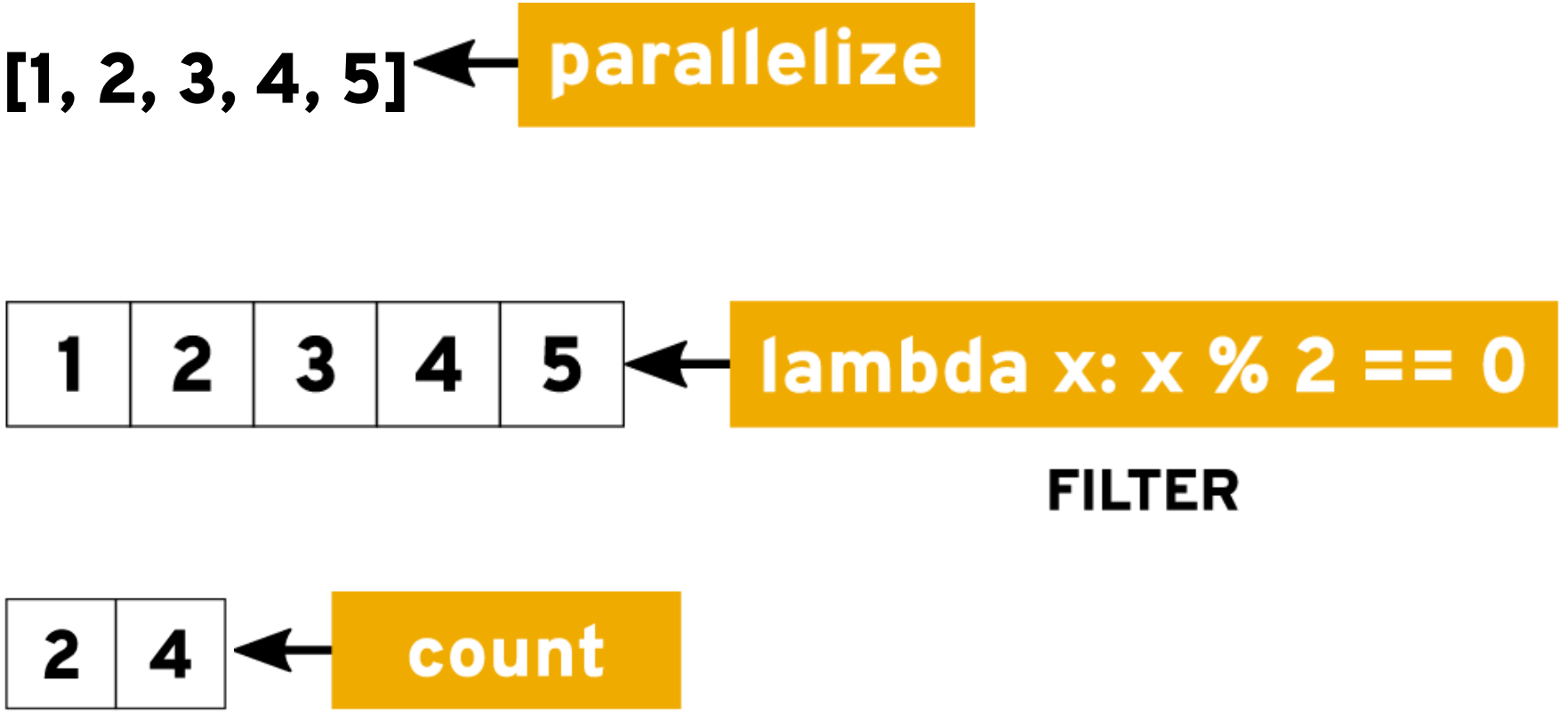
Resilient distributed dataset in action



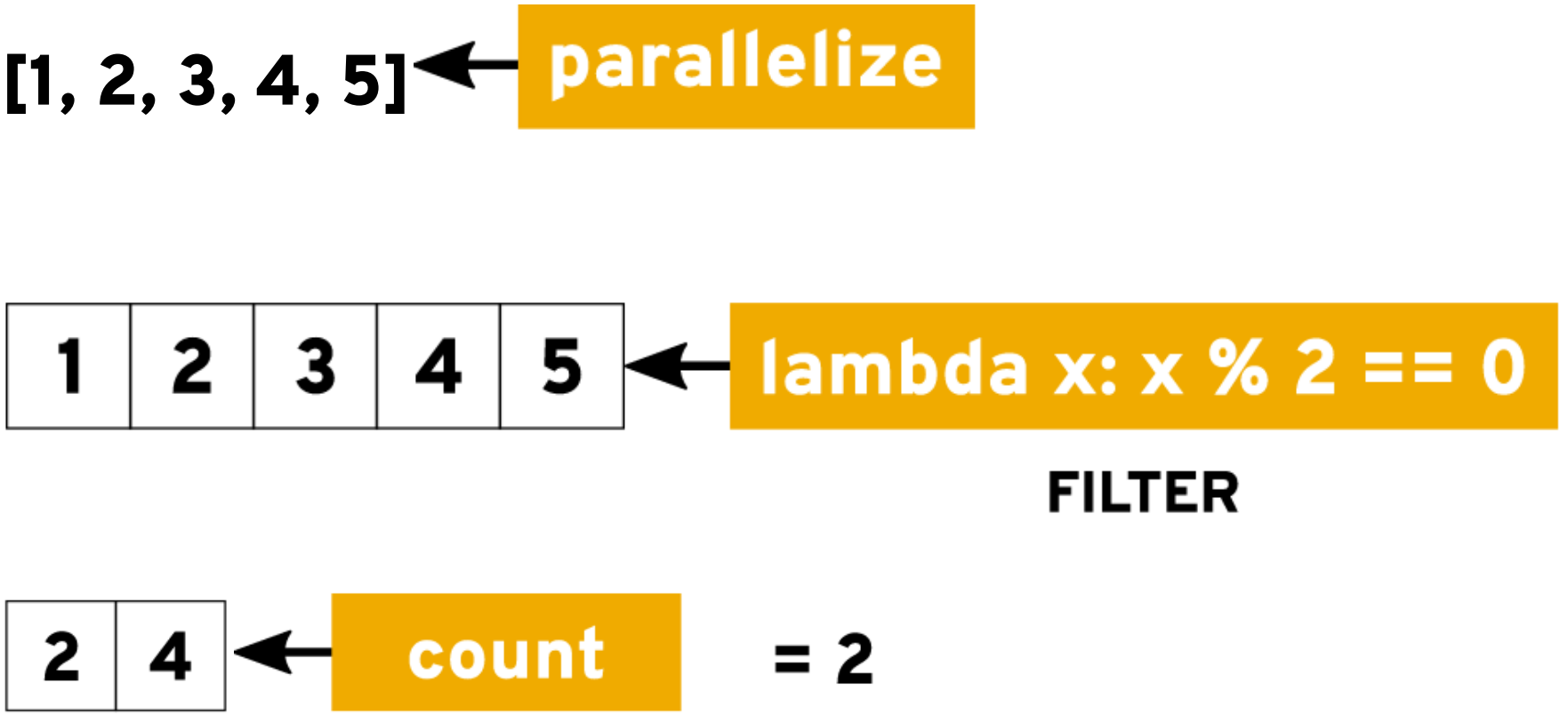
Resilient distributed dataset in action



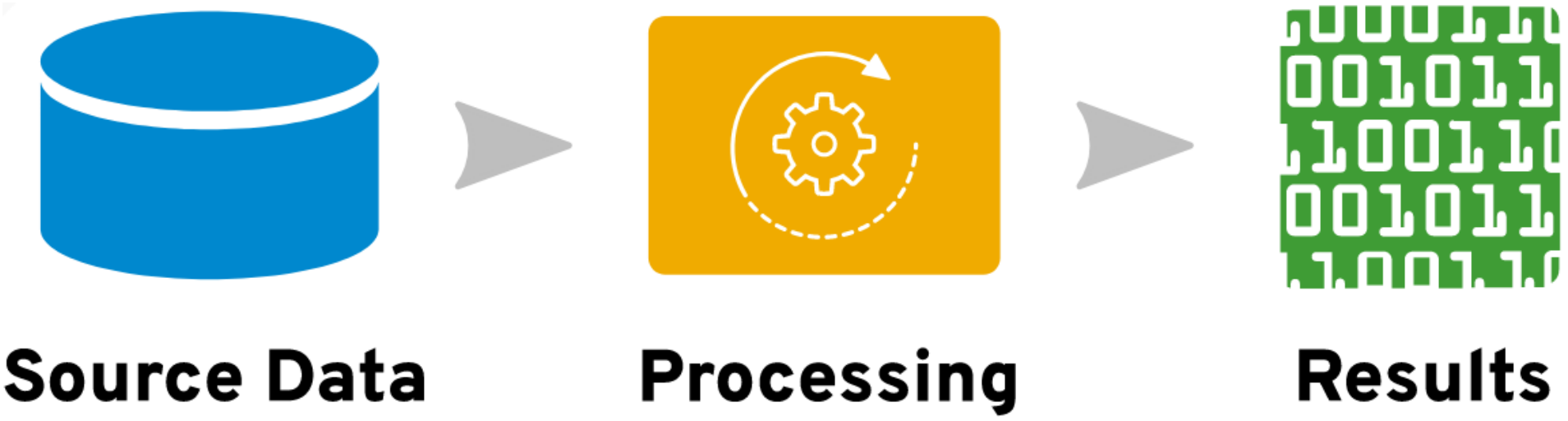
Resilient distributed dataset in action



Resilient distributed dataset in action



What is Spark application?

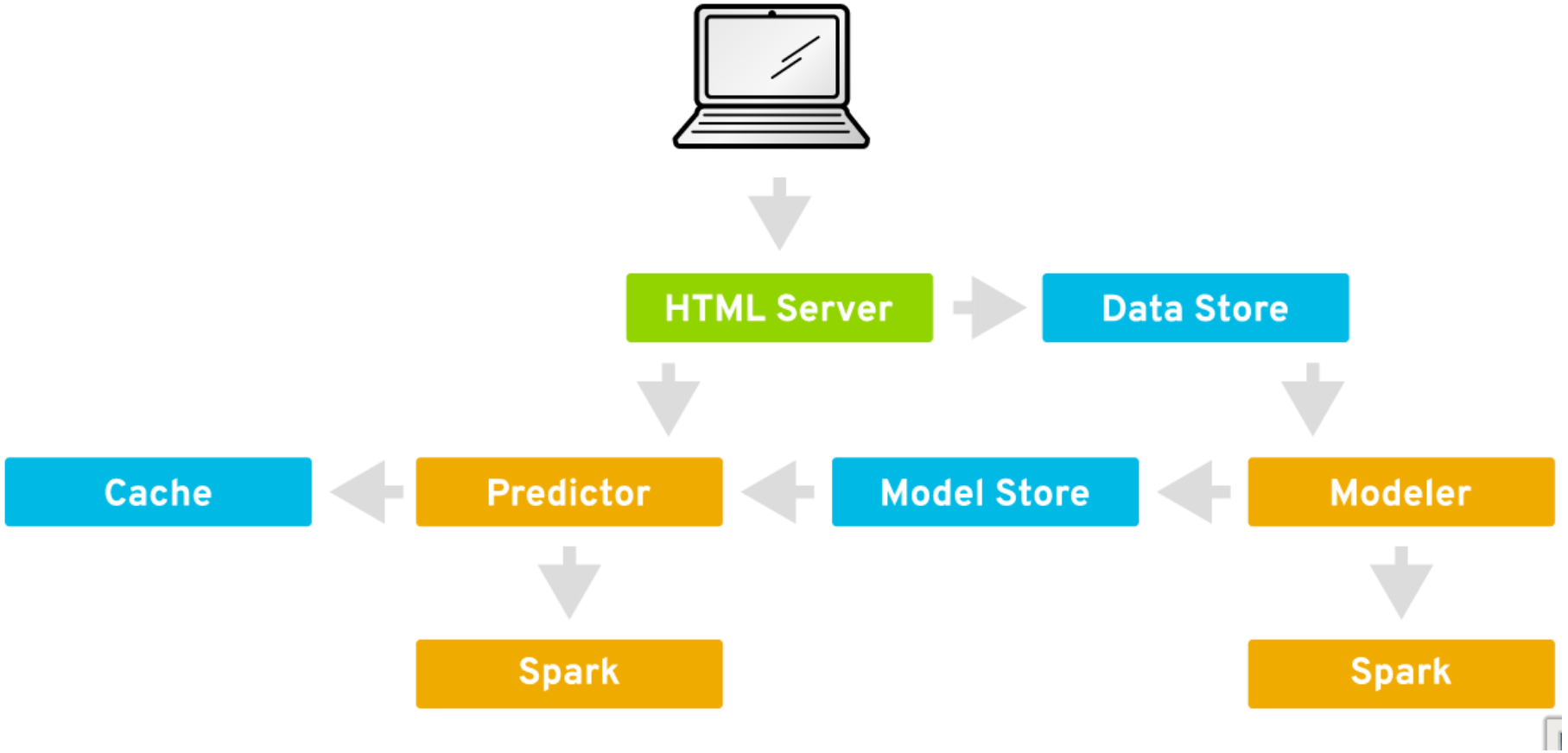


simple.py application

- Even numbers count

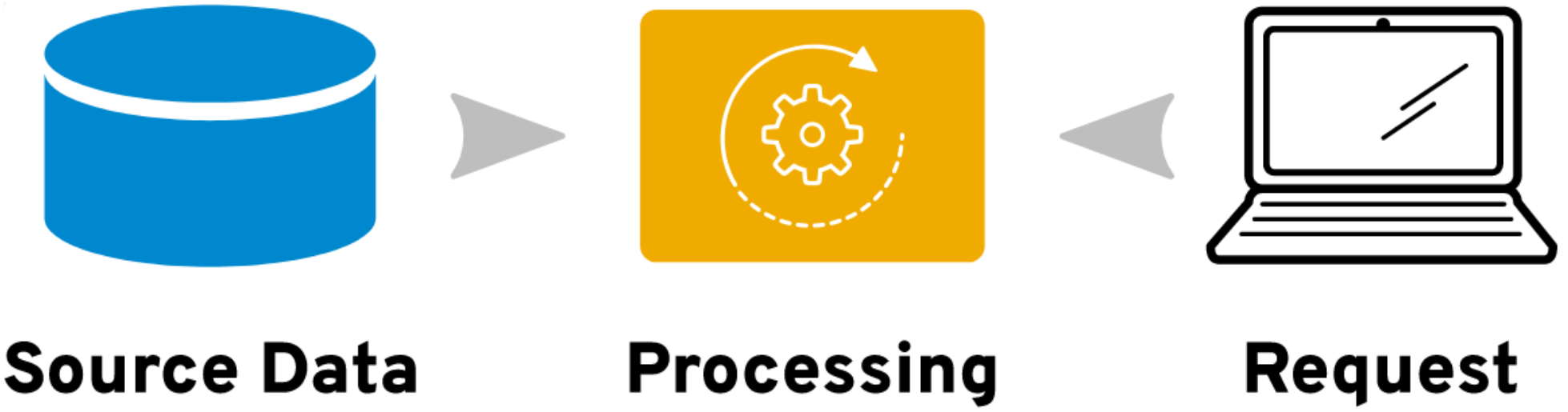
```
 1 import sys
 2 from pyspark.sql import SparkSession
 3
 4 spark = SparkSession.builder.appName("simple").getOrCreate()
 5
 6 data = range(int(sys.argv[1]))
 7
 8 evens = spark.sparkContext.parallelize(data)\
 9         .filter(lambda x: x%2 == 0)\
10         .count()
11
12 print("Out of 0-{} there are {} even numbers."
13       .format(sys.argv[1], evens))
```


A little more complex application

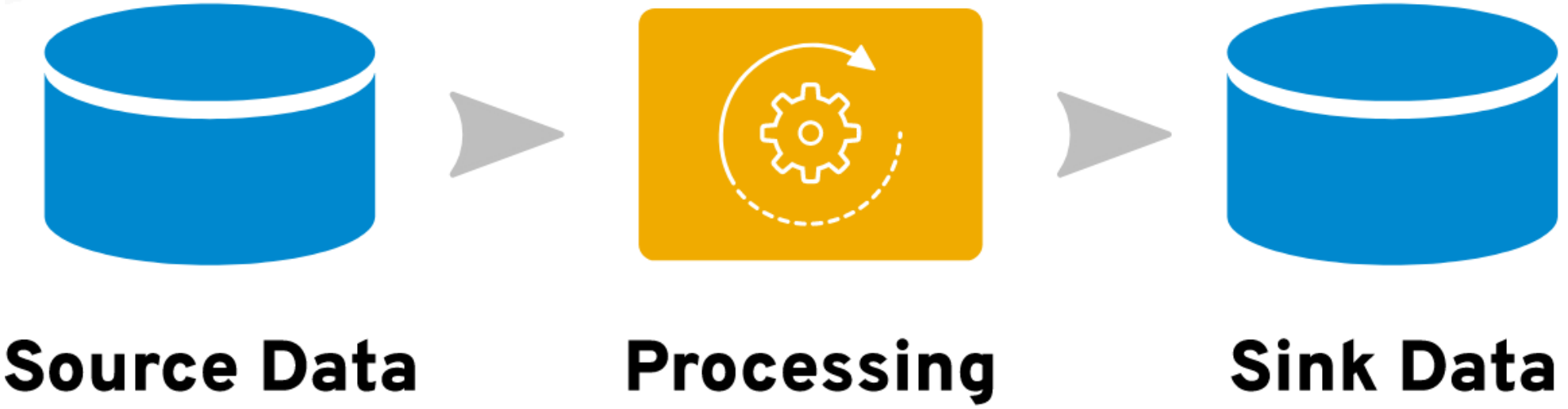


Designing a Spark microservice

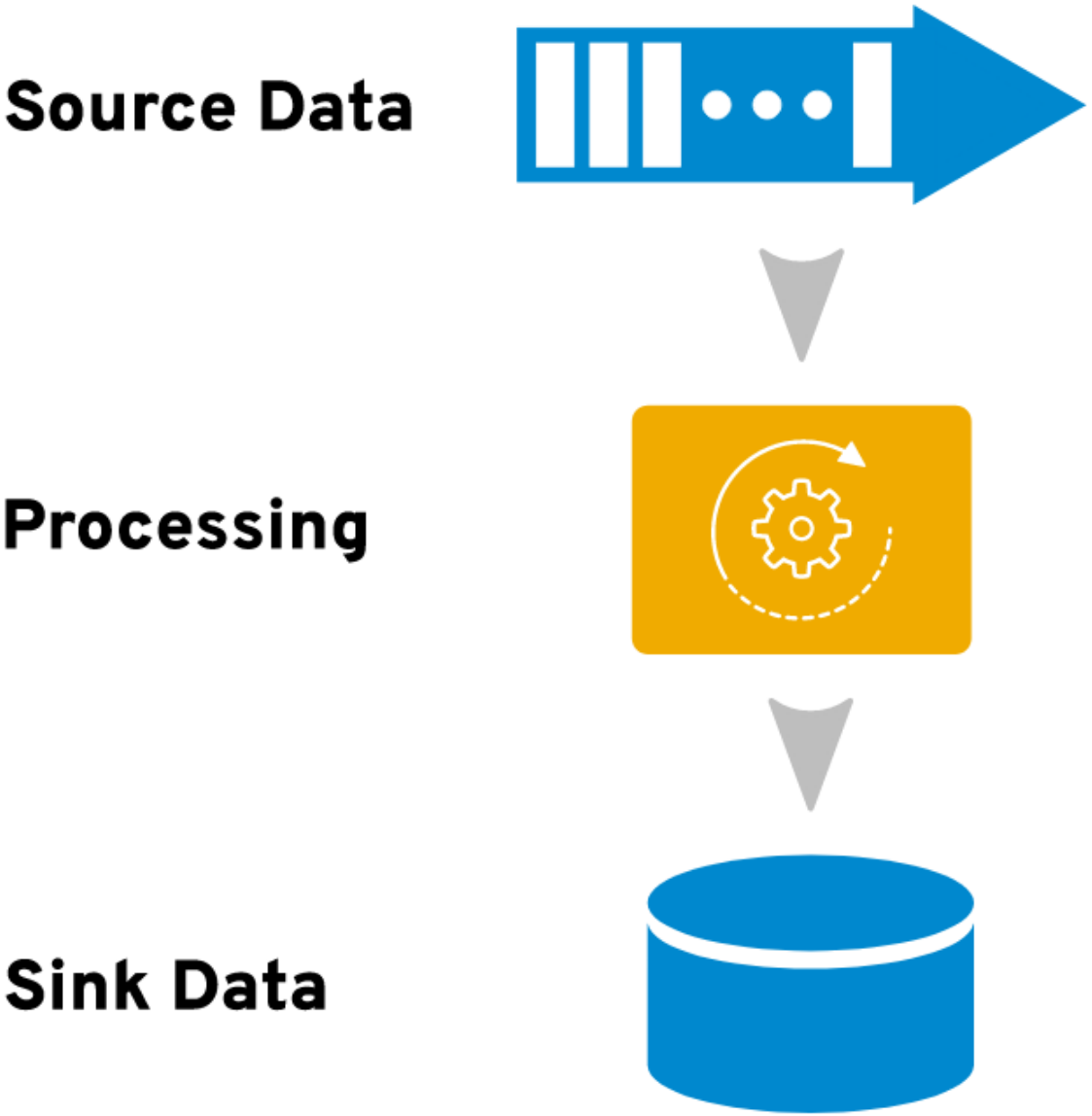
On demand batch processing



Continuous batch processing

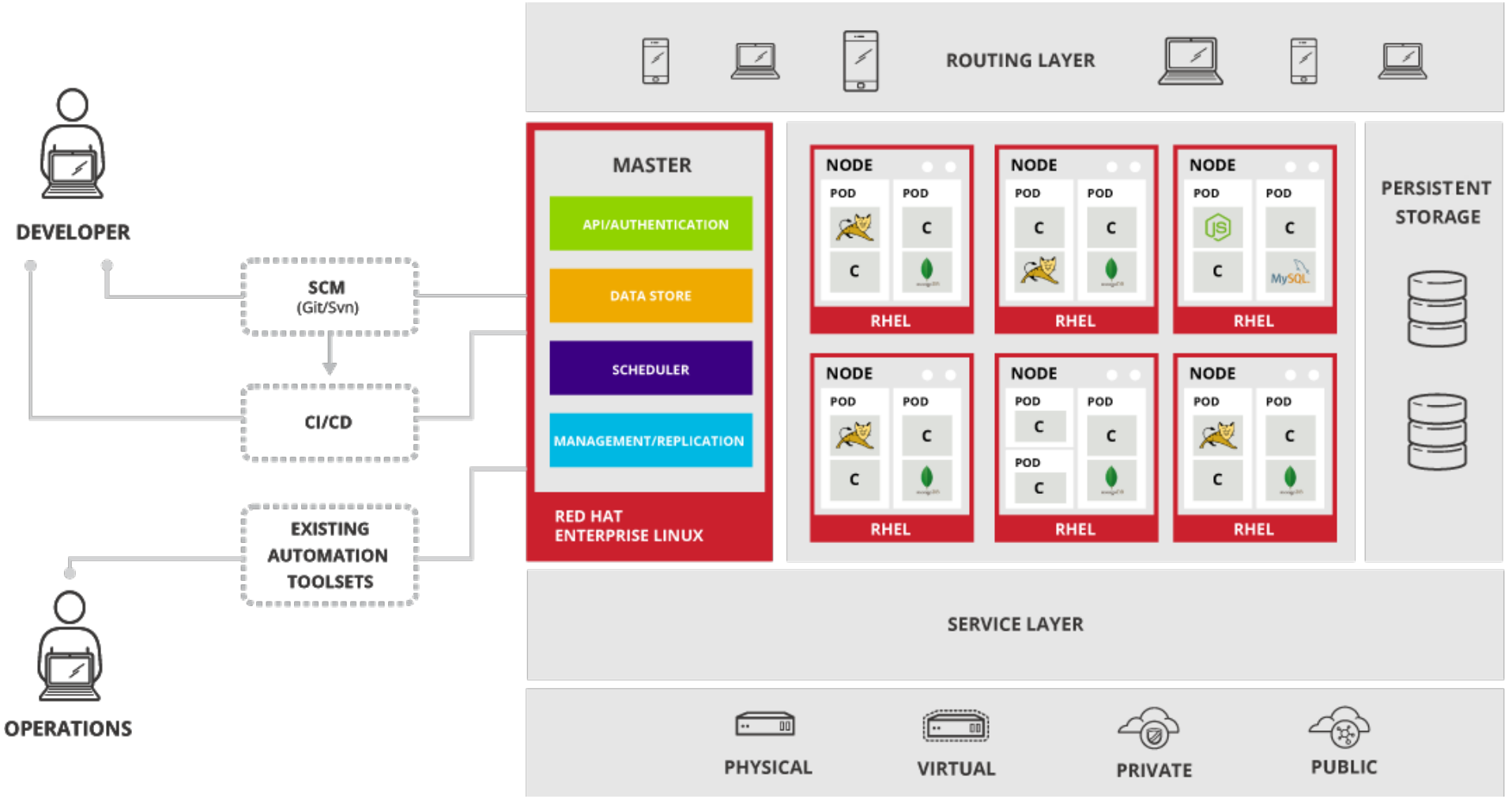


Stream processing

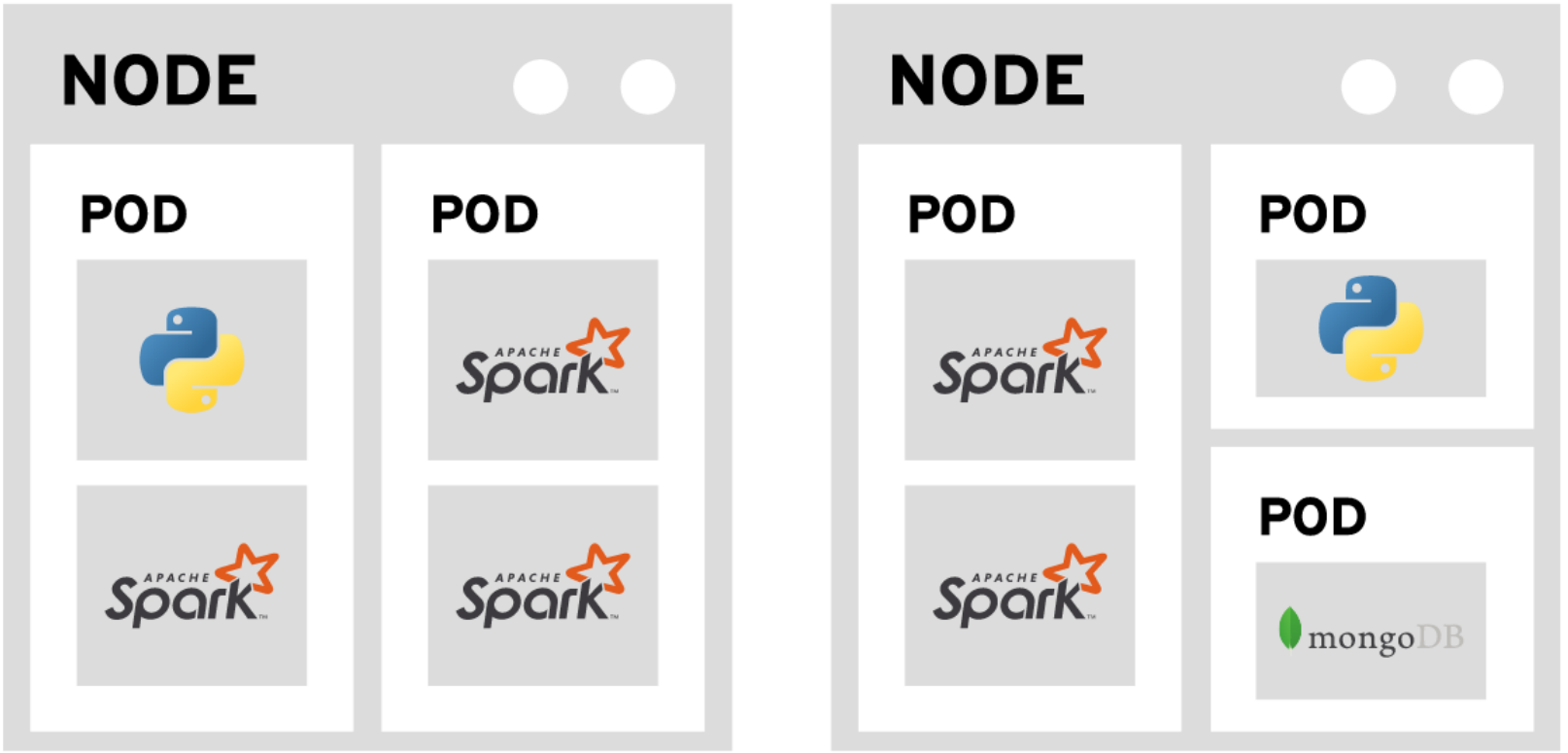


OpenShift architecture - recall

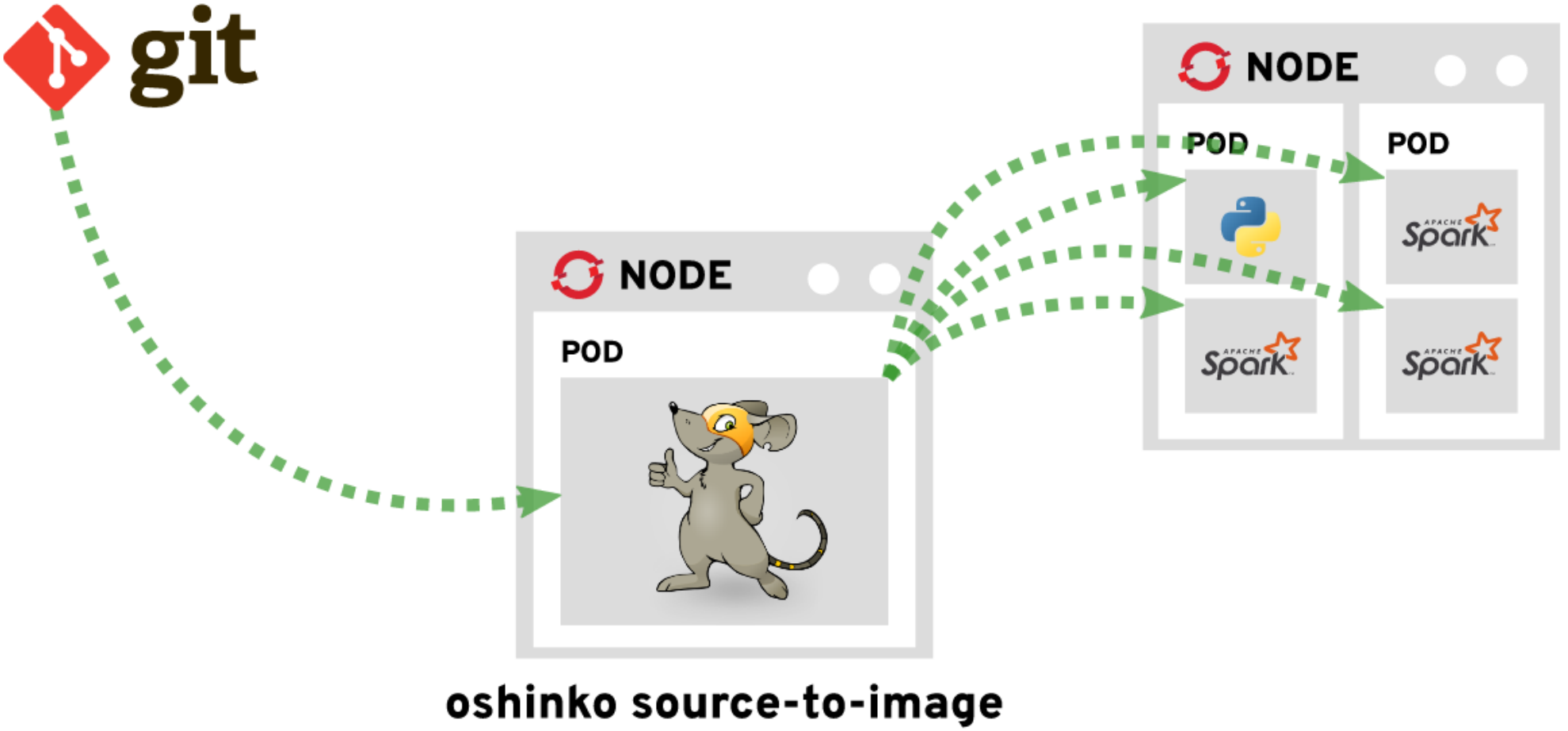
RED HAT® OPENSHIFT



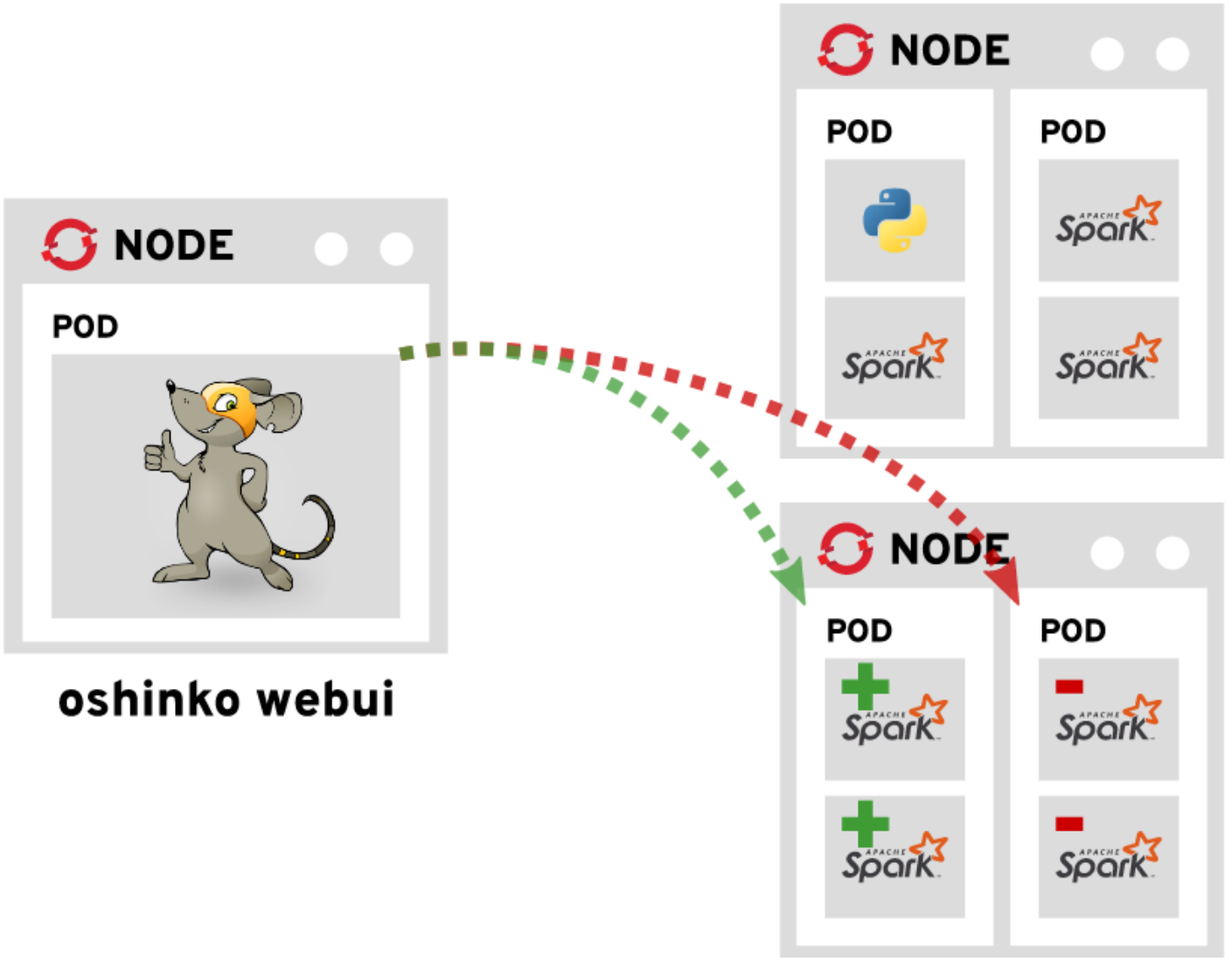
Spark on OpenShift



Oshinko - Integrating Spark and OpenShift



Oshinko - Integrating Spark and OpenShift



Demo time

Takeaways

- Containers
- Kubernetes
- OpenShift
- Apache Spark
- Oshinko tooling

Спасибі!

www.github.com/radanalyticsio

zroubali@redhat.com