THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

FROM NOTEBOOK TO CLOUD NATIVE

MICHAEL MCCUNE

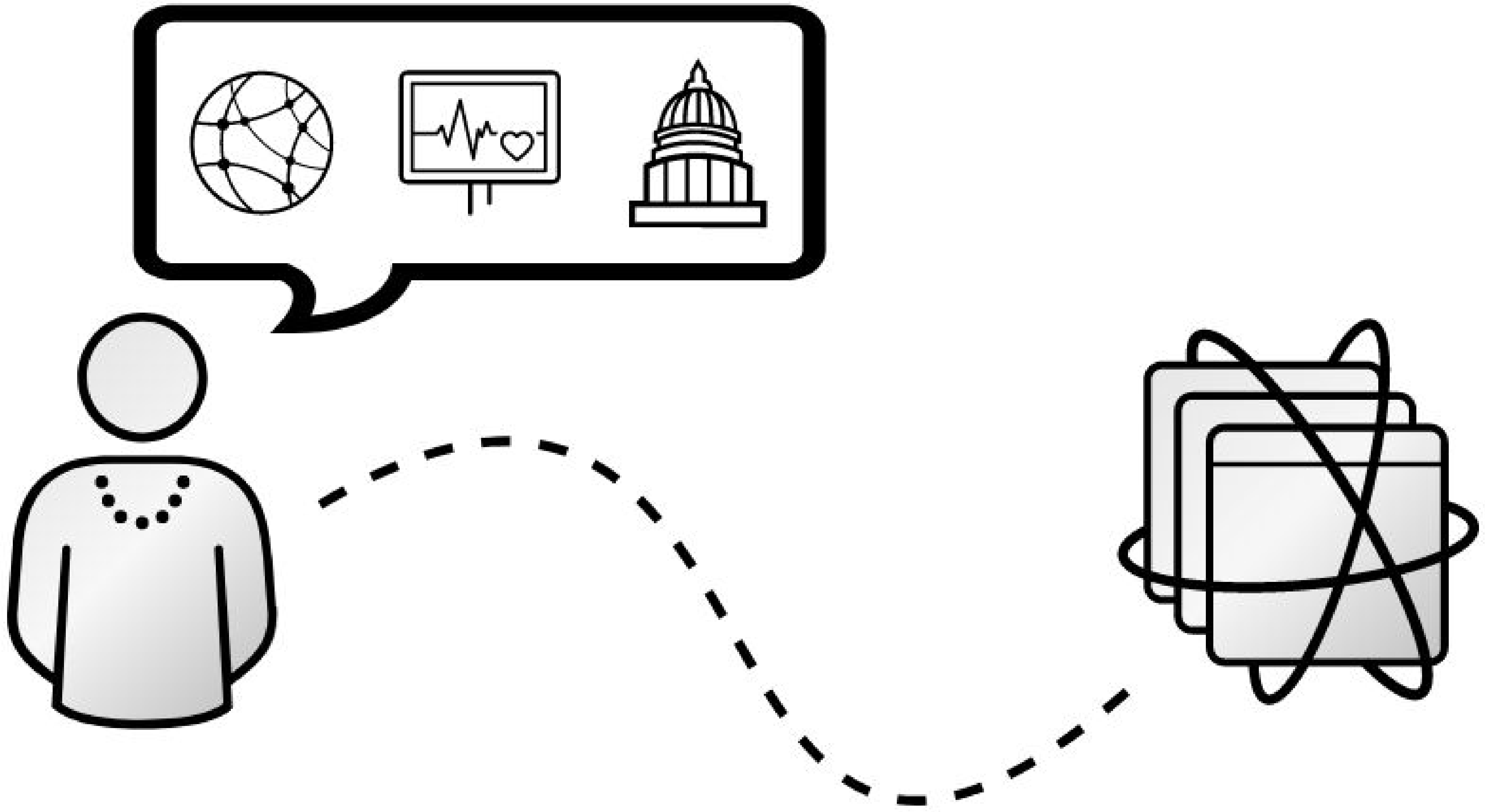
@elmiko@mastodon.technology

THE LINUX FOUNDATION

WHO AM I

- embedded to orchestration
- emerging technology @ Red Hat
- big data on OpenStack and OpenShift

WE'RE TALKING ABOUT A JOURNEY



#OSSummit

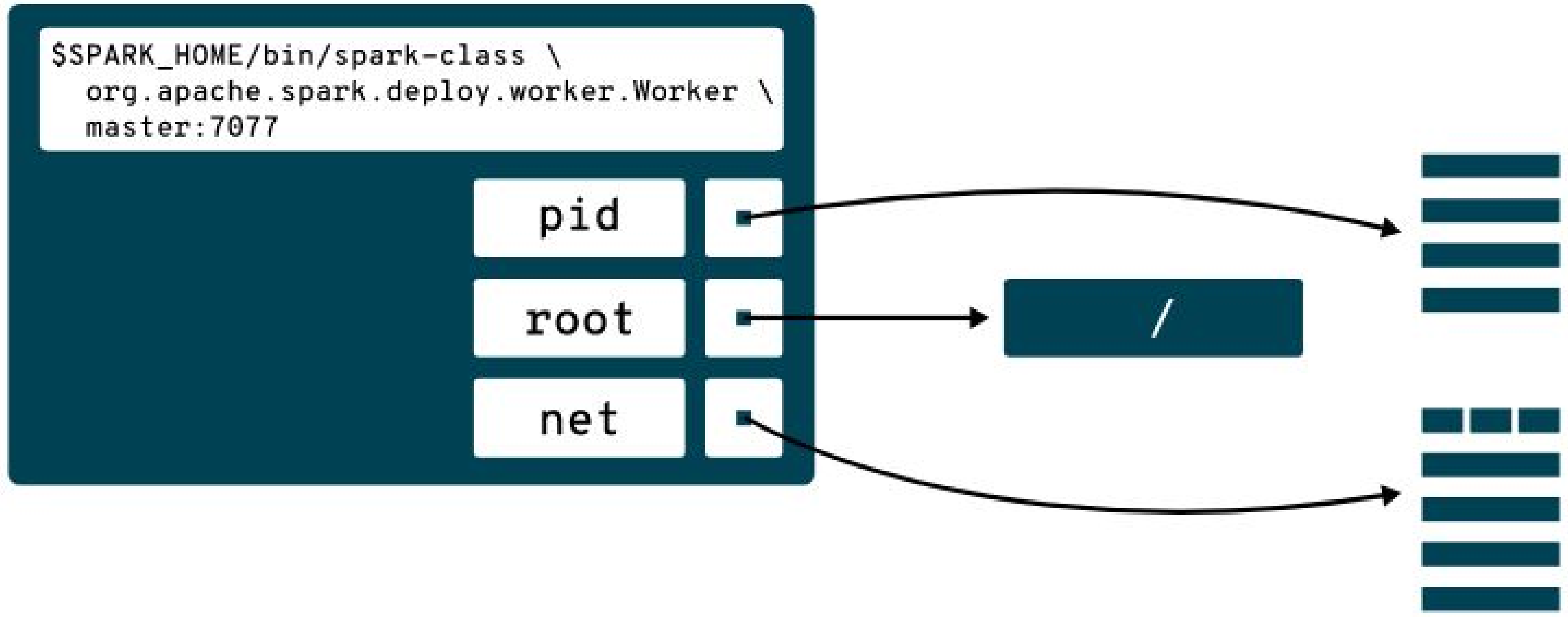
photography by Sam Hawley

#OSSummit

#OSSummit

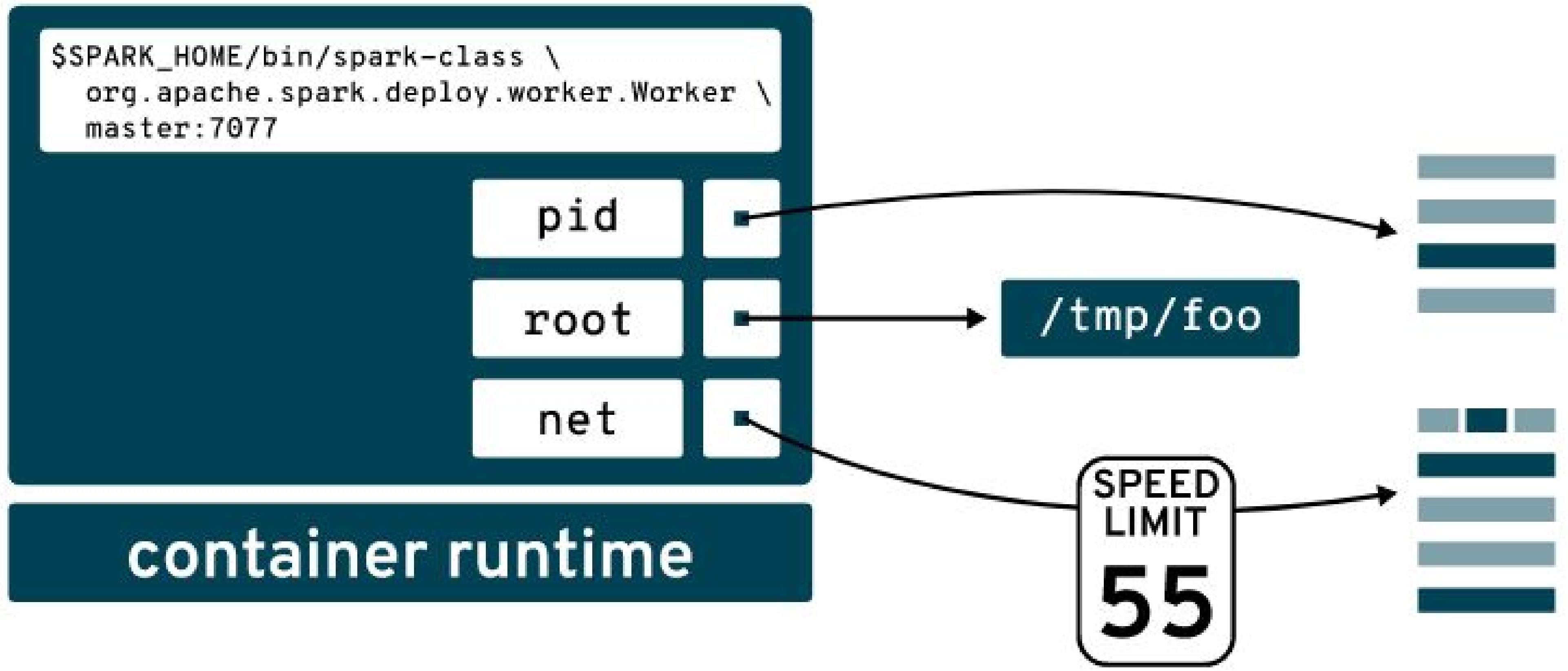
#OSSummit

LET'S TALK CONTAINERS



```
$SPARK_HOME/bin/spark-class \
  org.apache.spark.deploy.worker.Worker \
  master:7077
```

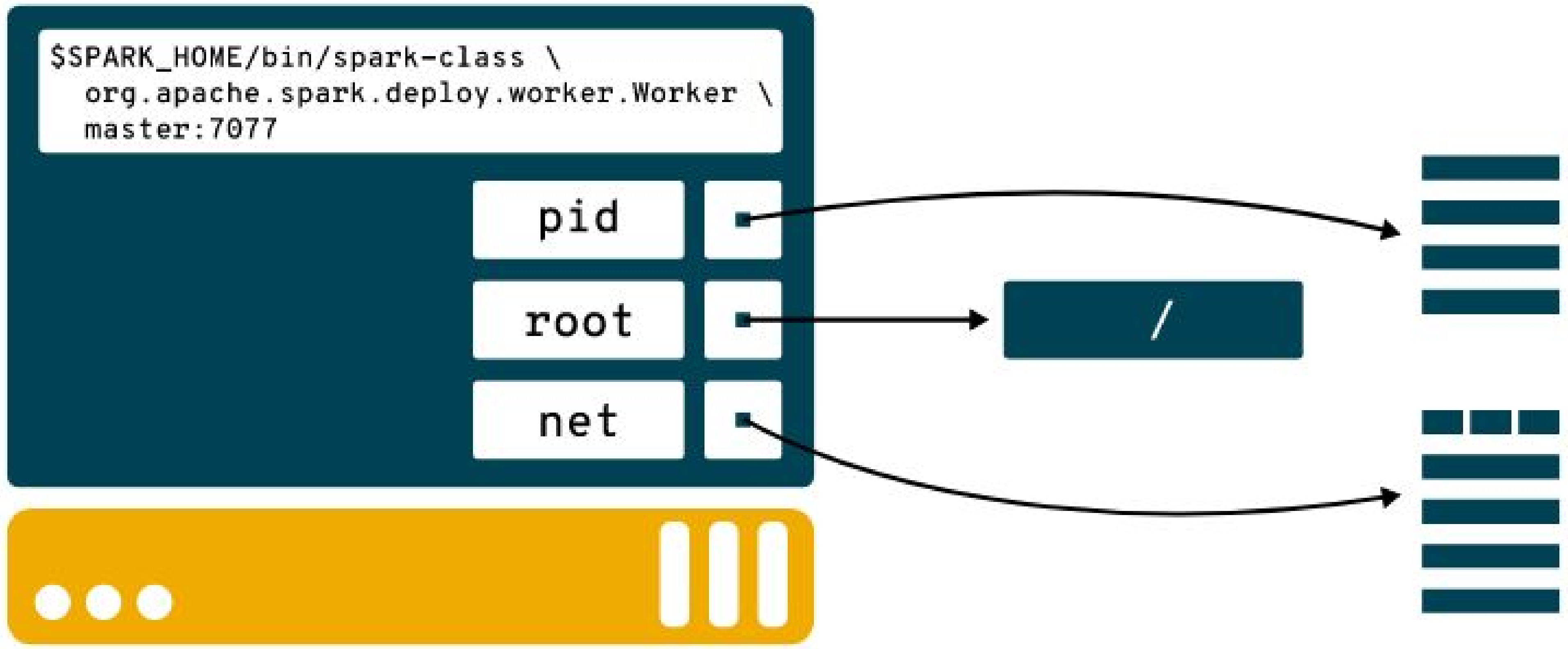
LET'S TALK CONTAINERS



```
$SPARK_HOME/bin/spark-class \
  org.apache.spark.deploy.worker.Worker \
  master:7077
```

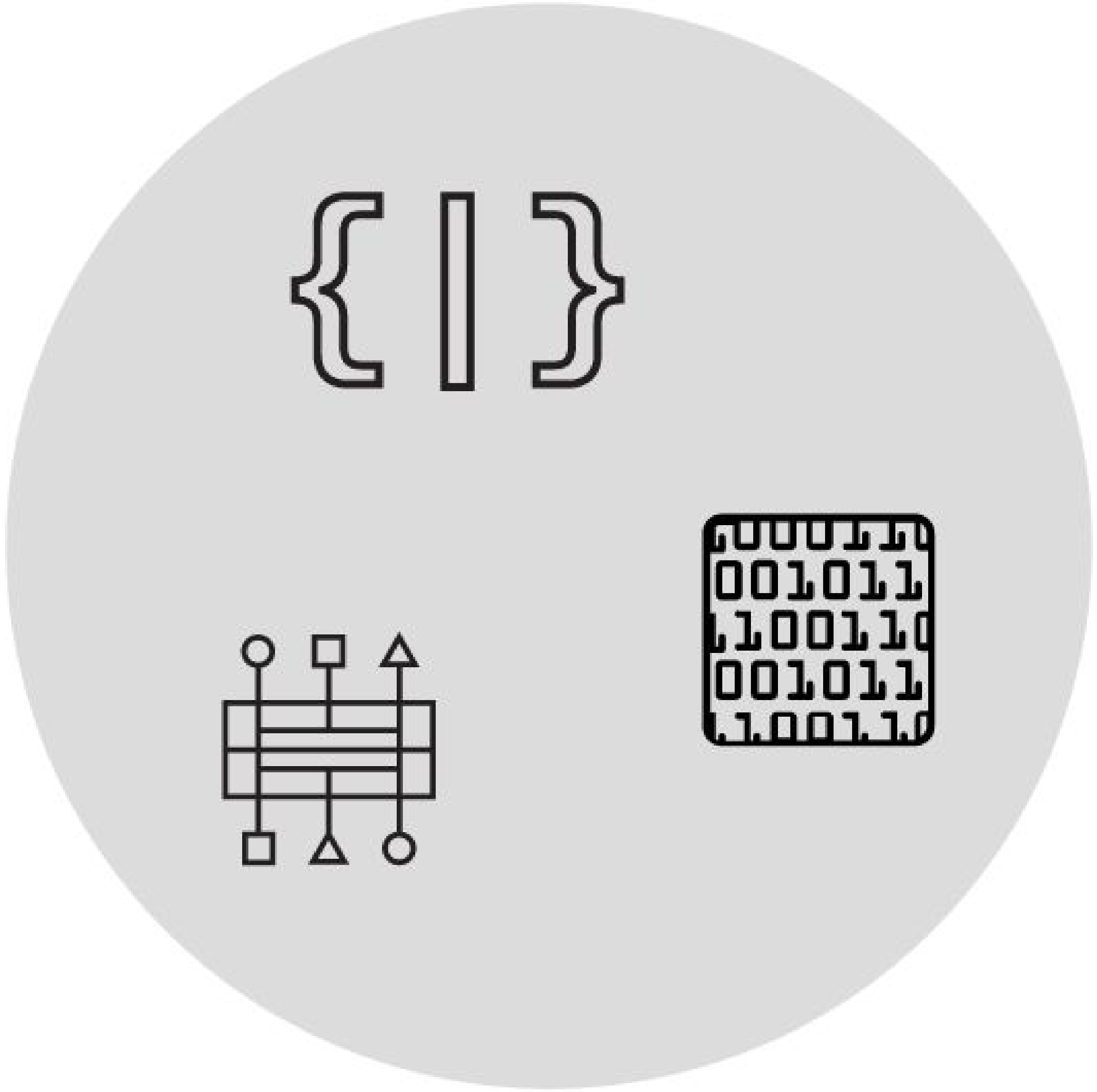
#OSSummit

LET'S TALK CONTAINERS



```
$SPARK_HOME/bin/spark-class \
  org.apache.spark.deploy.worker.Worker \
  master:7077
```

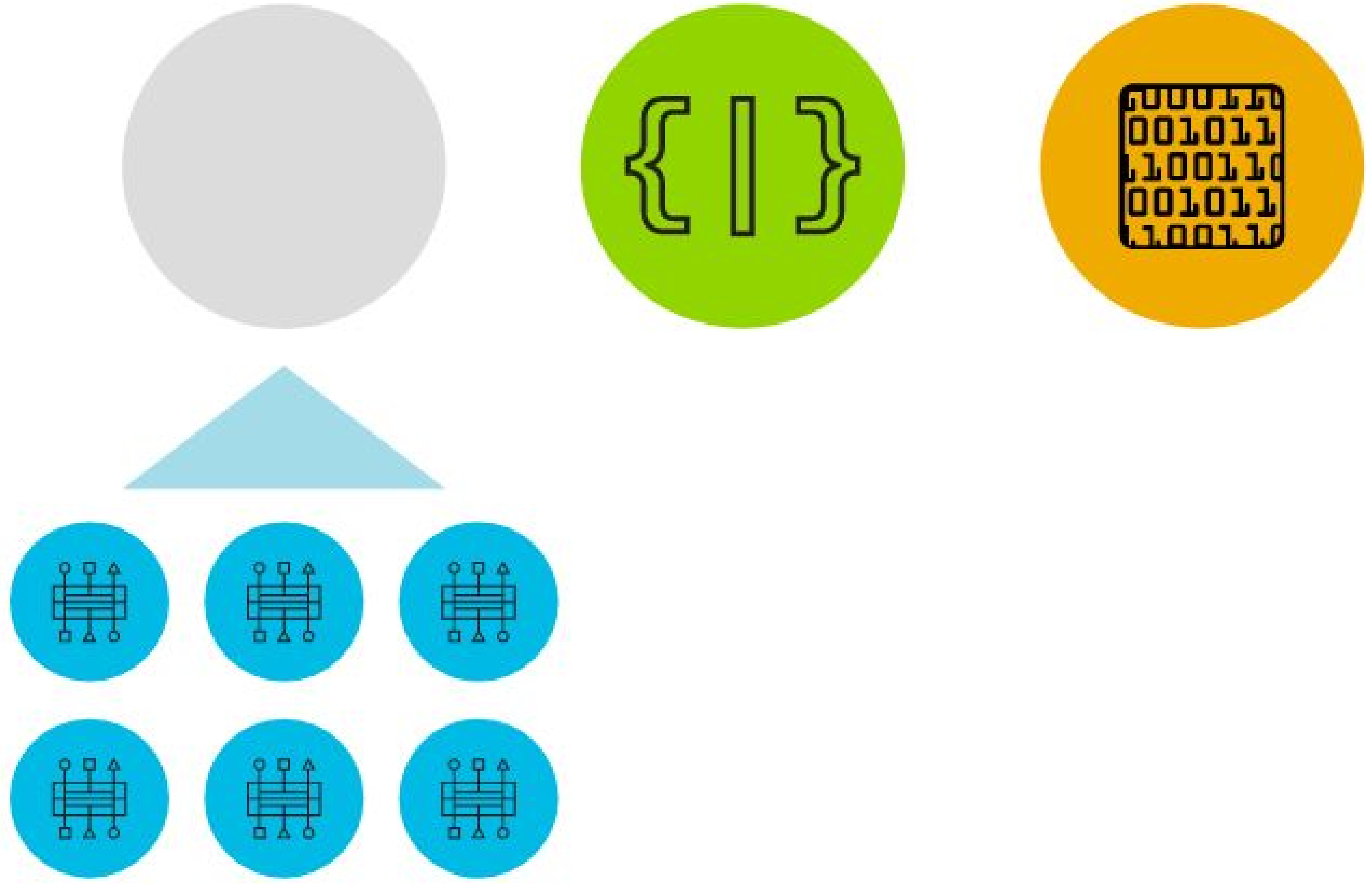
WHAT ABOUT MICROSERVICES?



WHAT ABOUT MICROSERVICES?

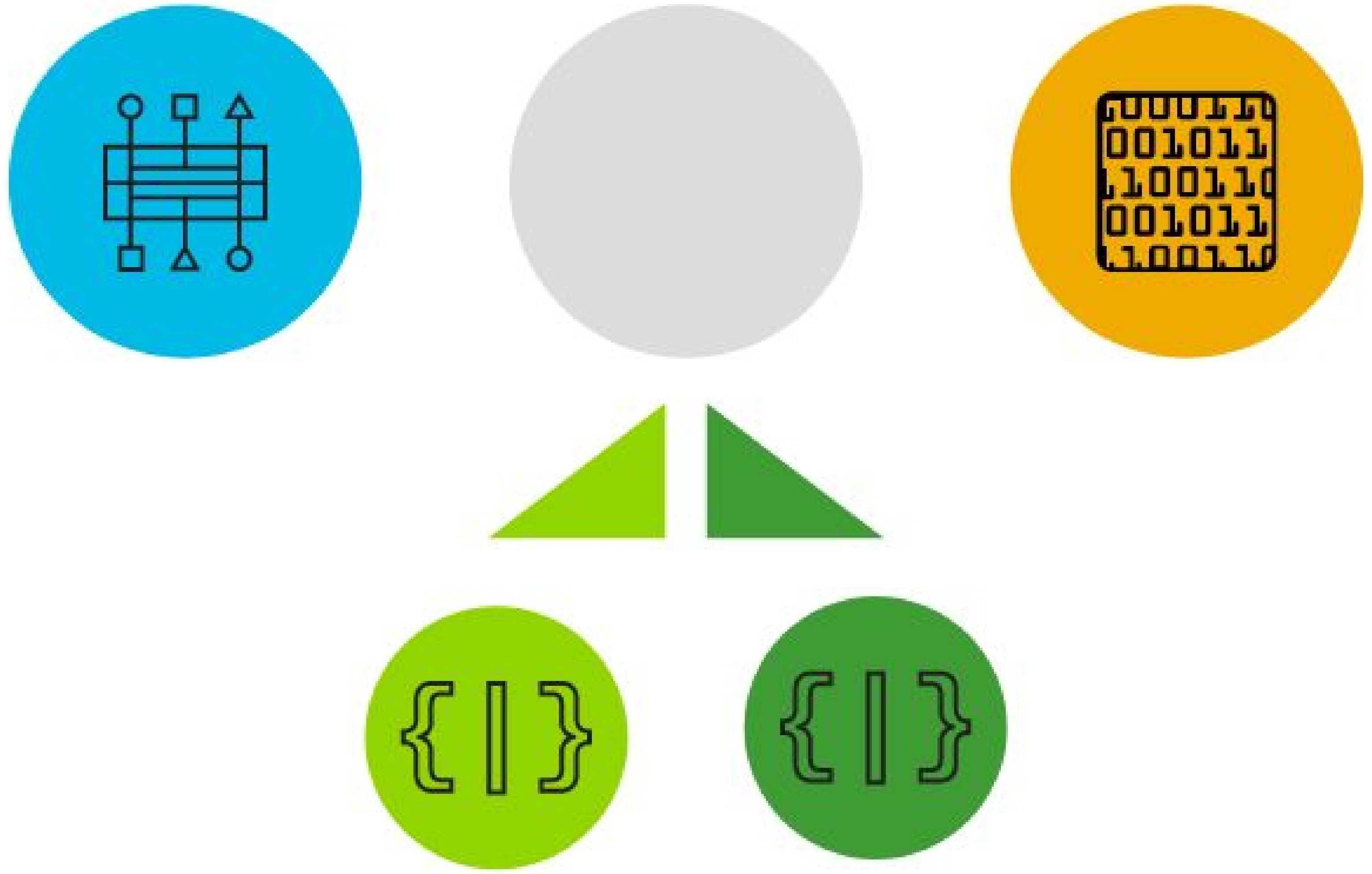
- modular and flexible
- stateful vs stateless
- network resilient

WHAT ABOUT MICROSERVICES?



#OSSummit

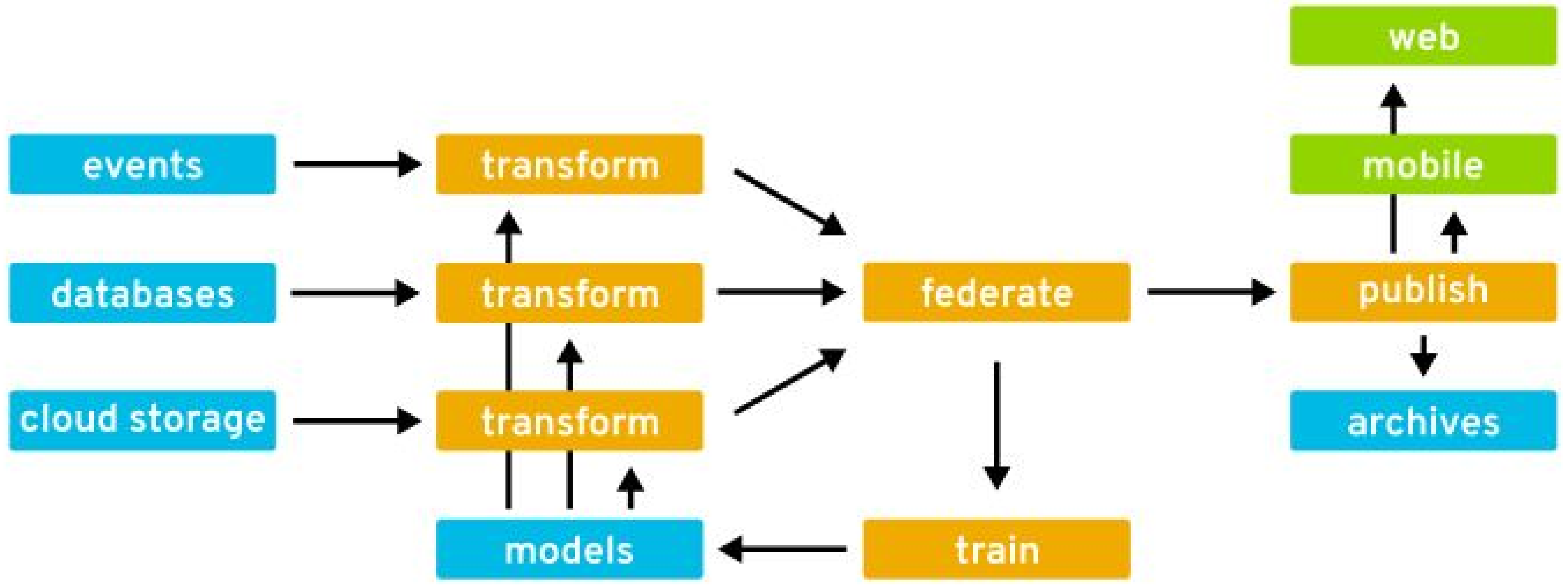
WHAT ABOUT MICROSERVICES?



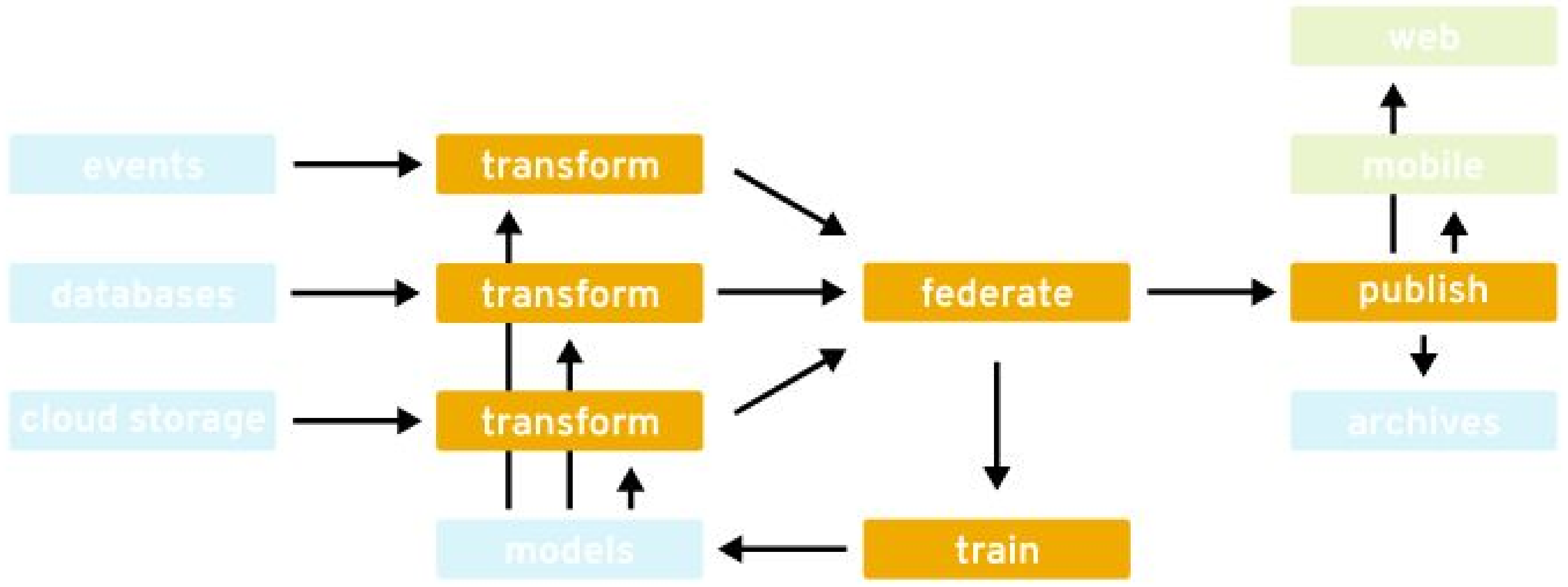
WHAT ABOUT MICROSERVICES?

001011
100110
001011

INTELLIGENT APPLICATIONS

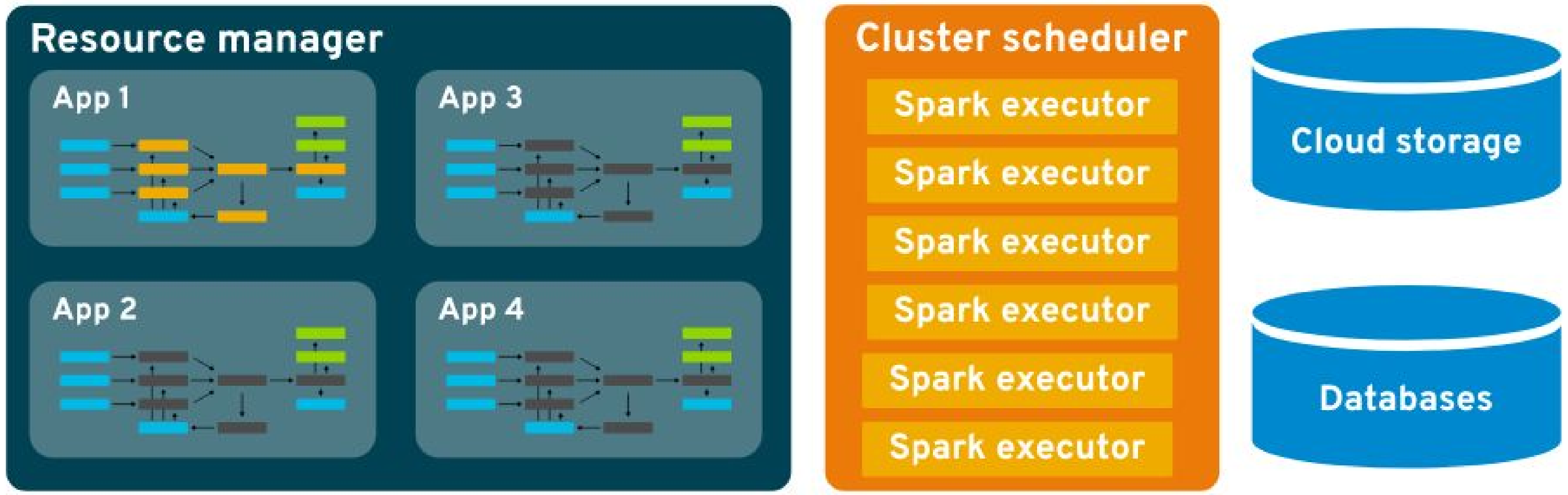


INTELLIGENT APPLICATIONS

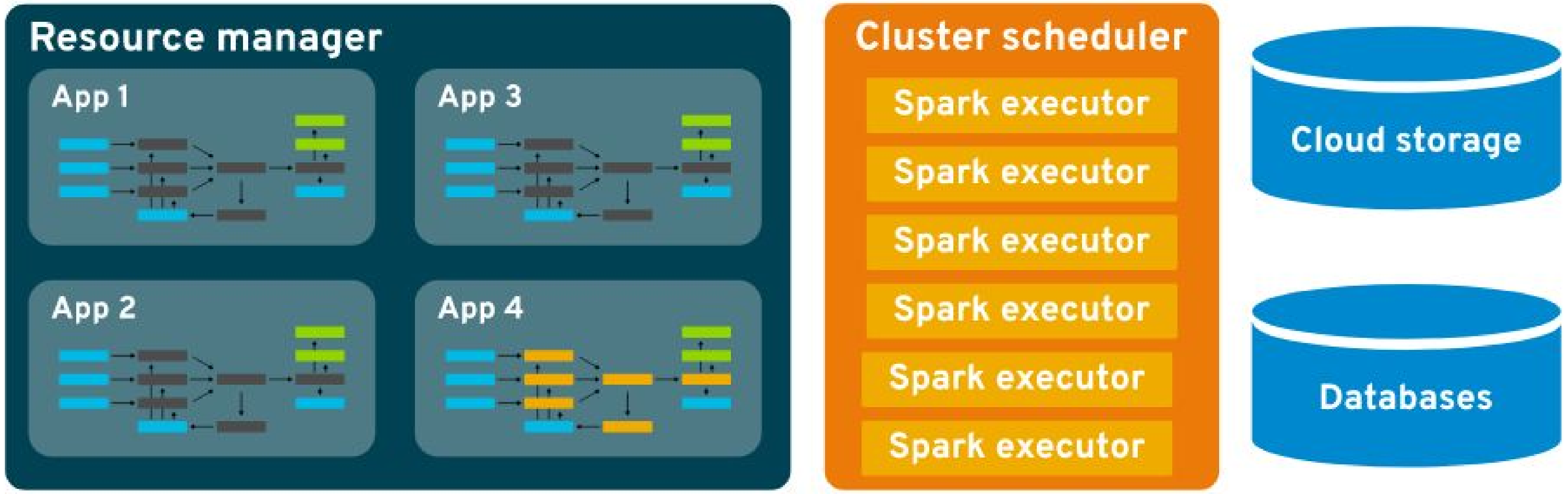


#OSSummit

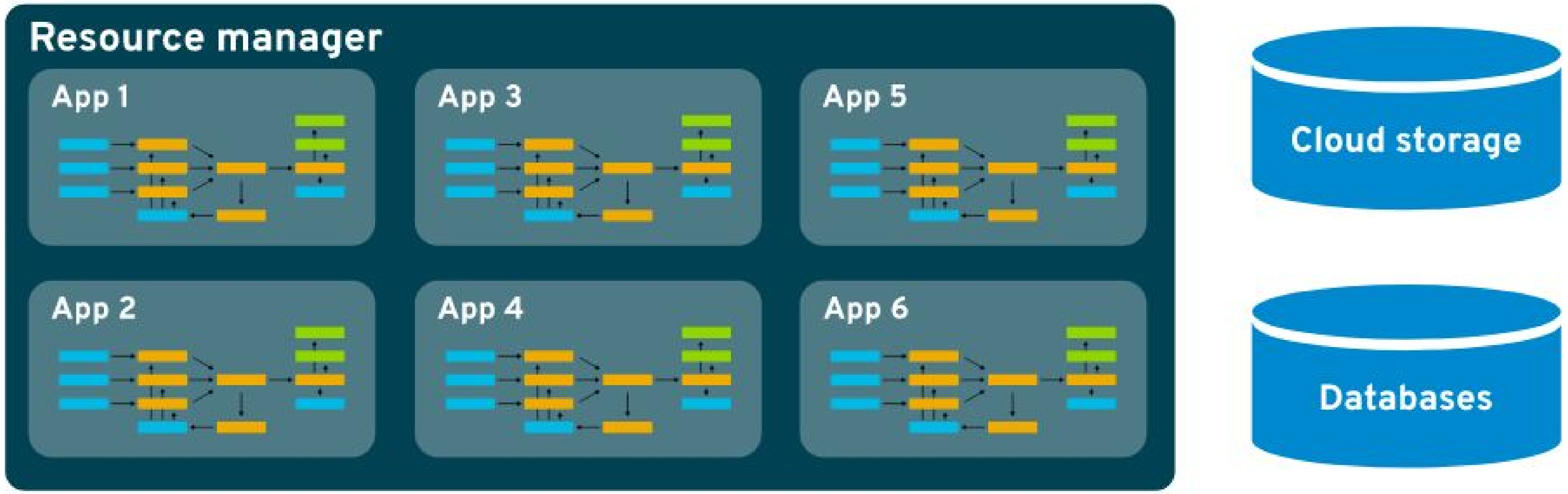
MONOLITHIC CLUSTERS



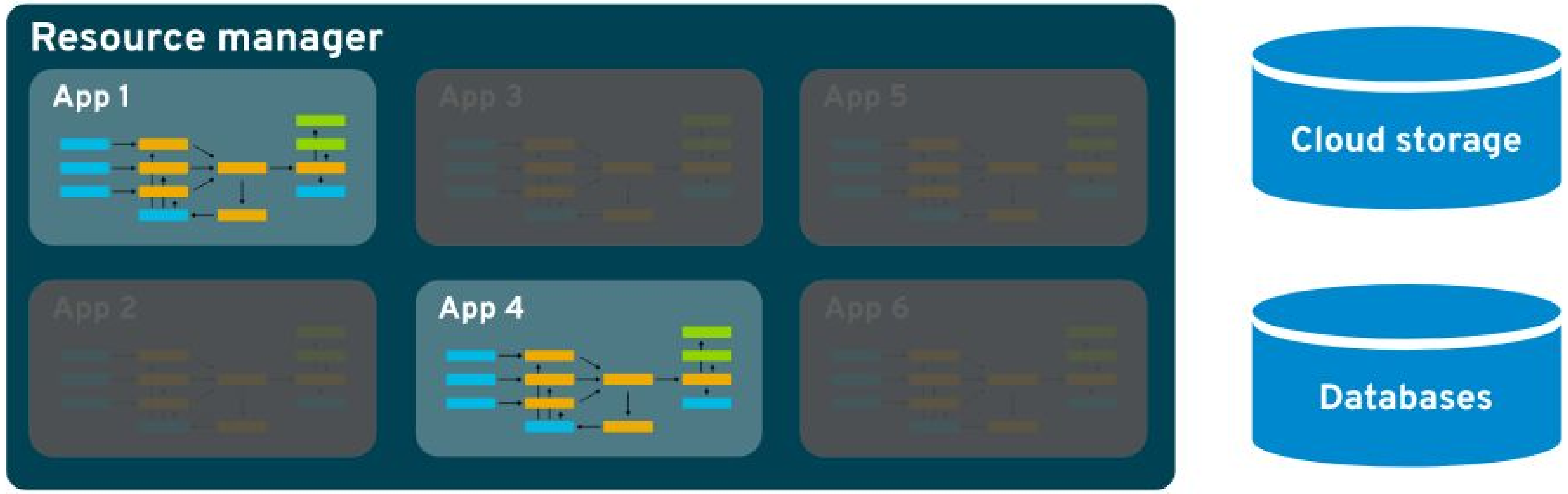
MONOLITHIC CLUSTERS



CLOUD NATIVE CLUSTERS



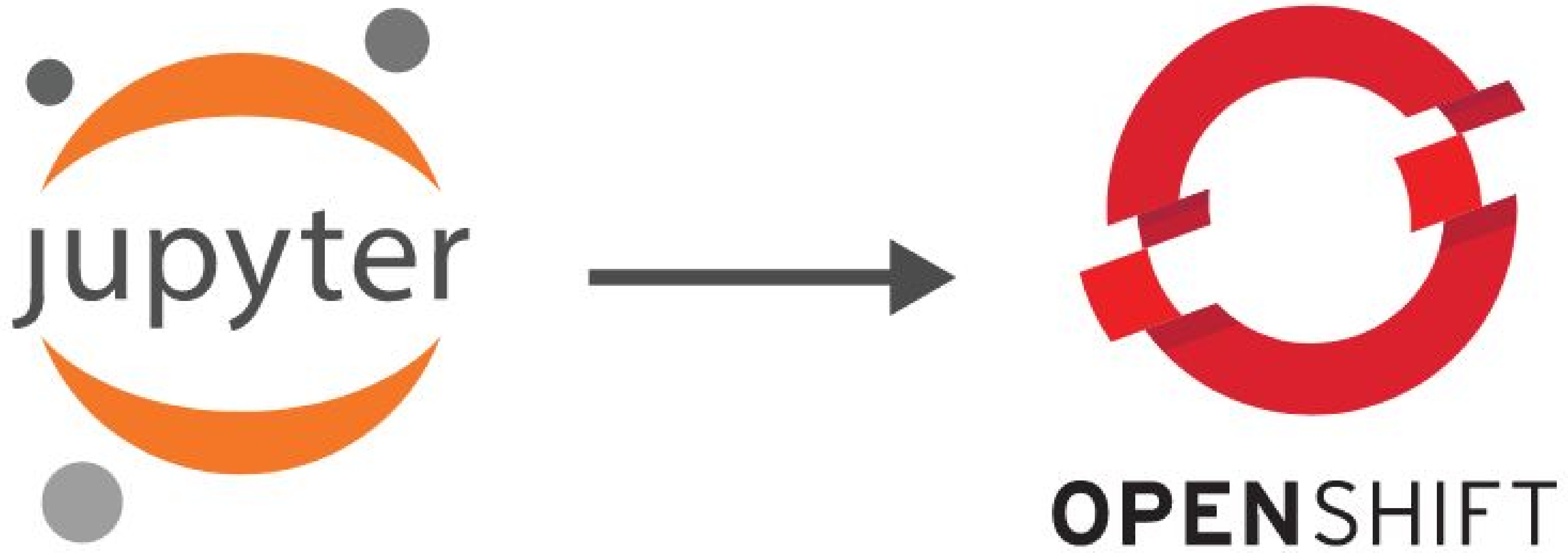
CLOUD NATIVE CLUSTERS



INTELLIGENT APPLICATION LIFECYCLES

NOTEBOOKS IN ACTION

GOING CLOUD NATIVE

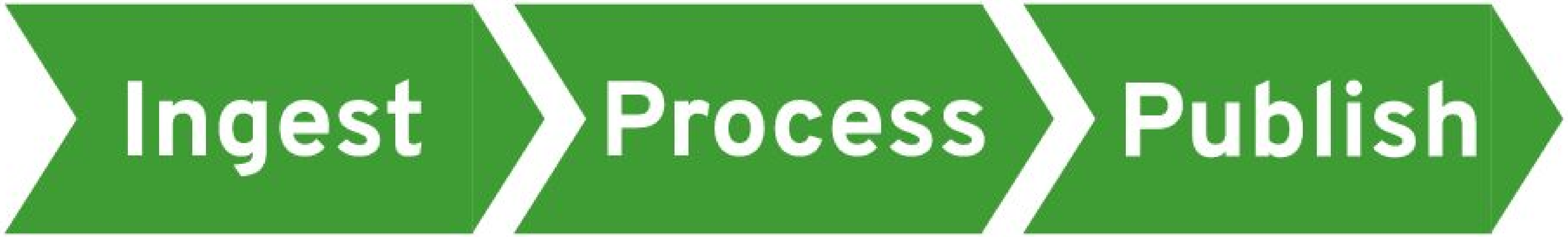


WHAT IS CLOUD NATIVE?

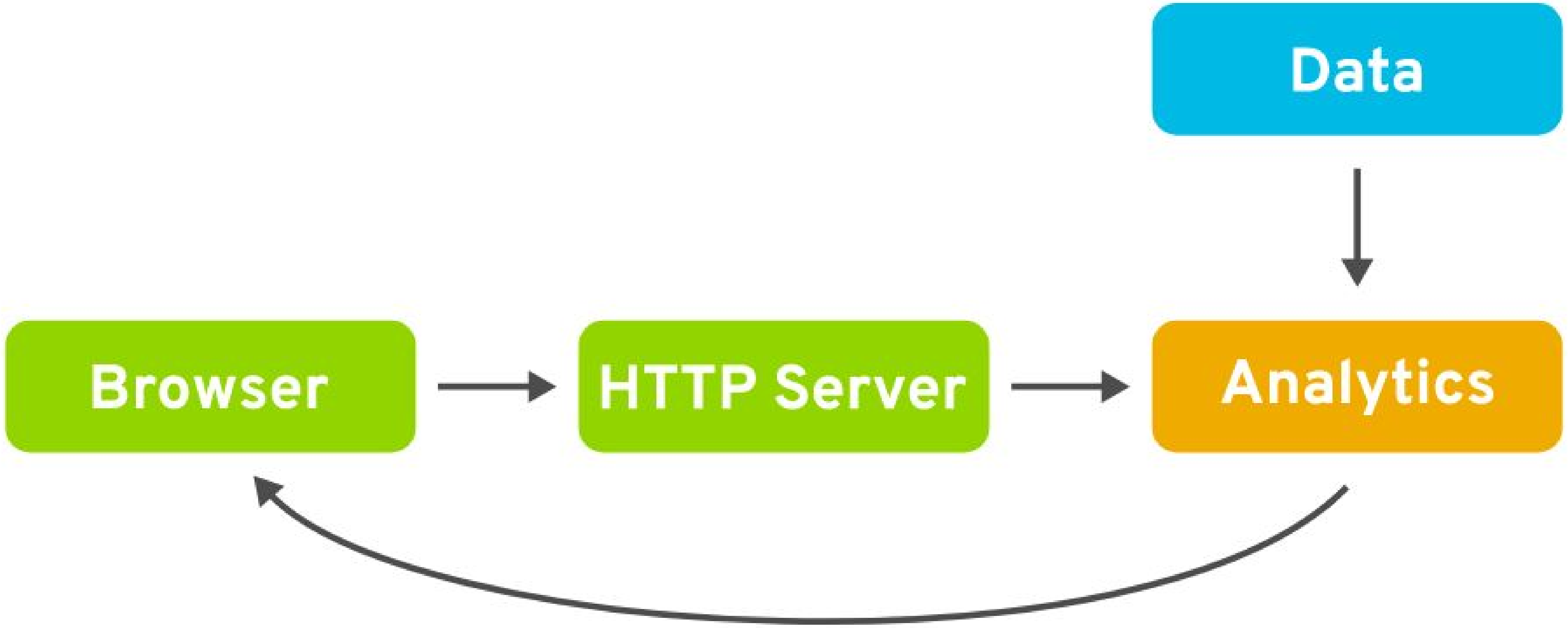
CLOUD NATIVE
COMPUTING FOUNDATION

- Containerized
- Dynamically orchestrated
- Microservice oriented
- **cncf.io/about/faq**

WHAT WILL YOUR APPLICATION DO?

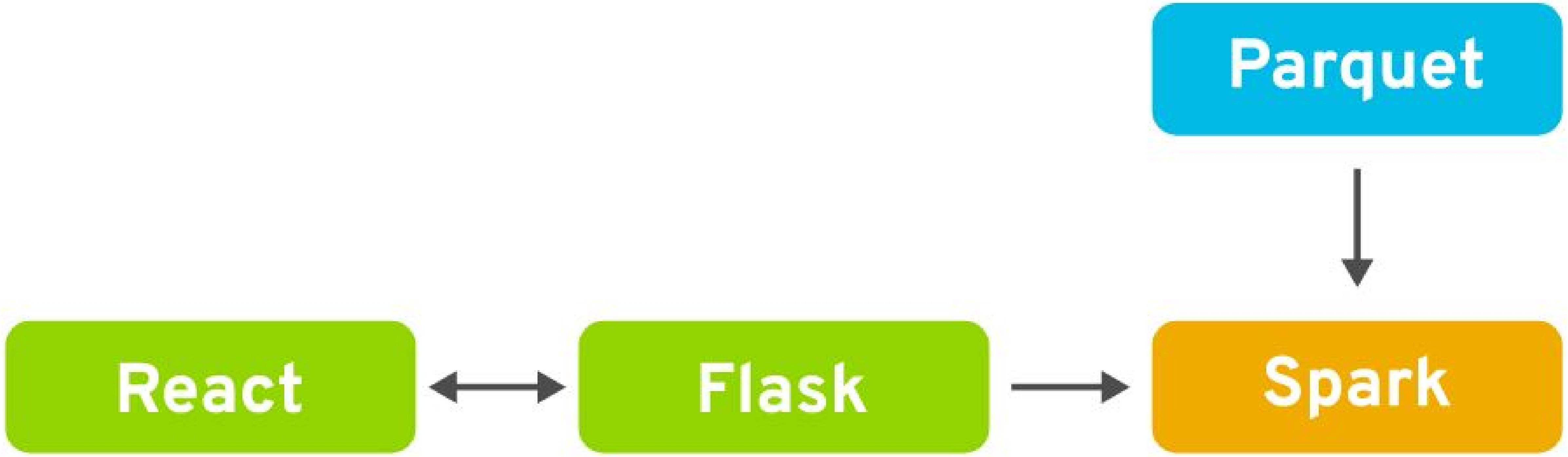


STORYBOARD YOUR ARCHITECTURE



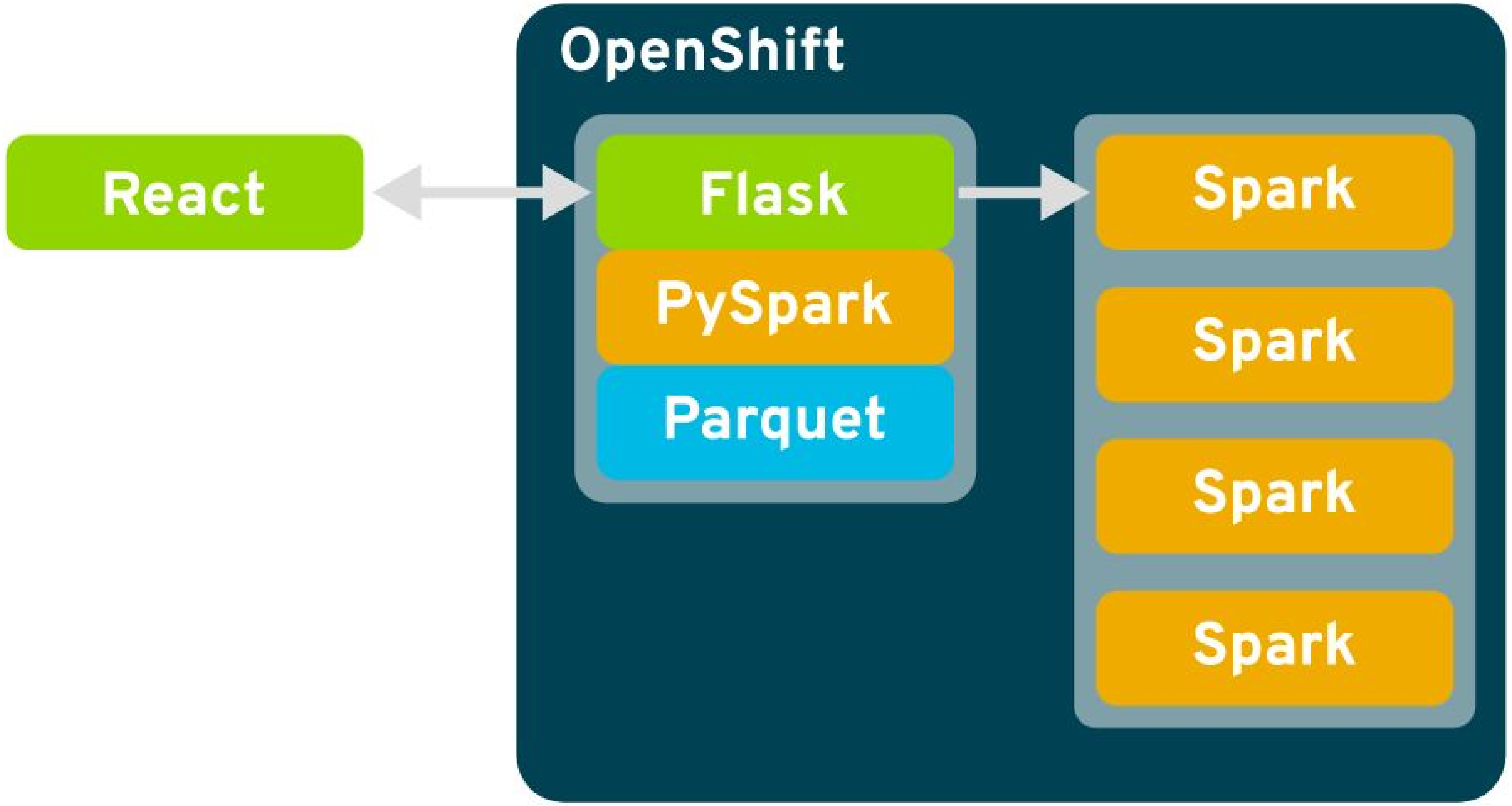
#OSSummit

CONSTRUCT YOUR APPLICATION

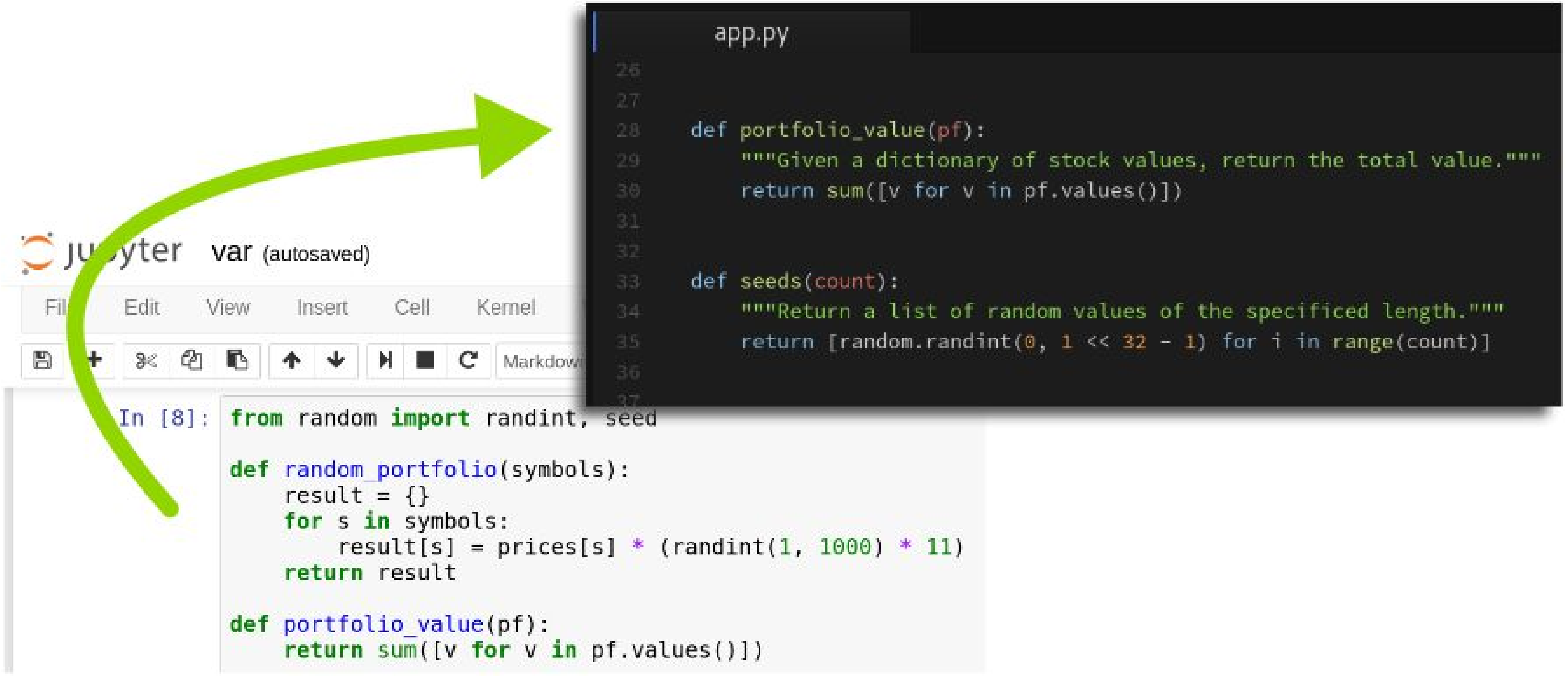


CASE STUDY: VAR-SANDBOX

GENERAL ARCHITECTURE



HOW WAS IT BUILT?

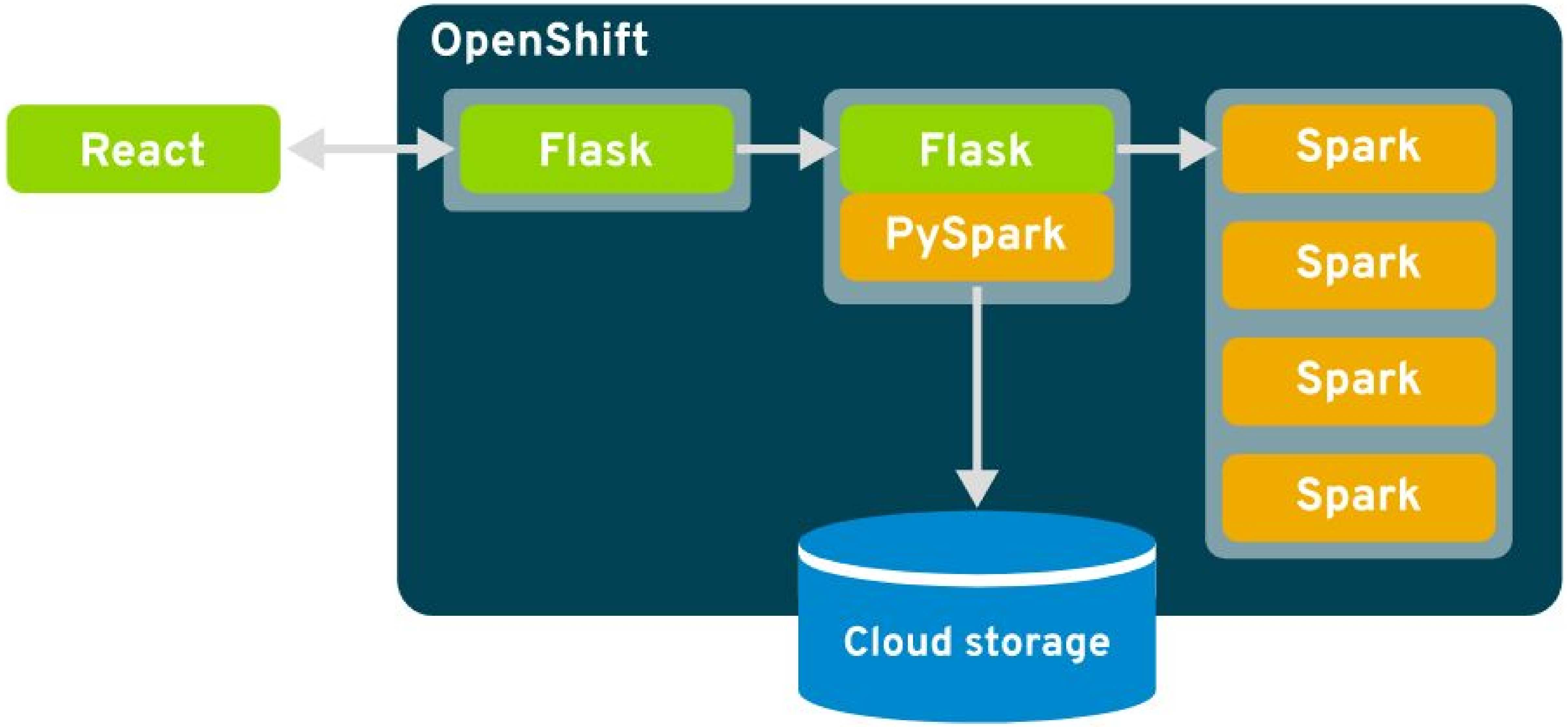


```
In [8]: from random import randint, seed

        def random_portfolio(symbols):
            result = {}
            for s in symbols:
                result[s] = prices[s] * (randint(1, 1000) * 11)
            return result

        def portfolio_value(pf):
            return sum([v for v in pf.values()])
```

```python
def portfolio_value(pf):
    """Given a dictionary of stock values, return the total value."""
    return sum([v for v in pf.values()])


def seeds(count):
    """Return a list of random values of the specified length."""
    return [random.randint(0, 1 << 32 - 1) for i in range(count)]
```

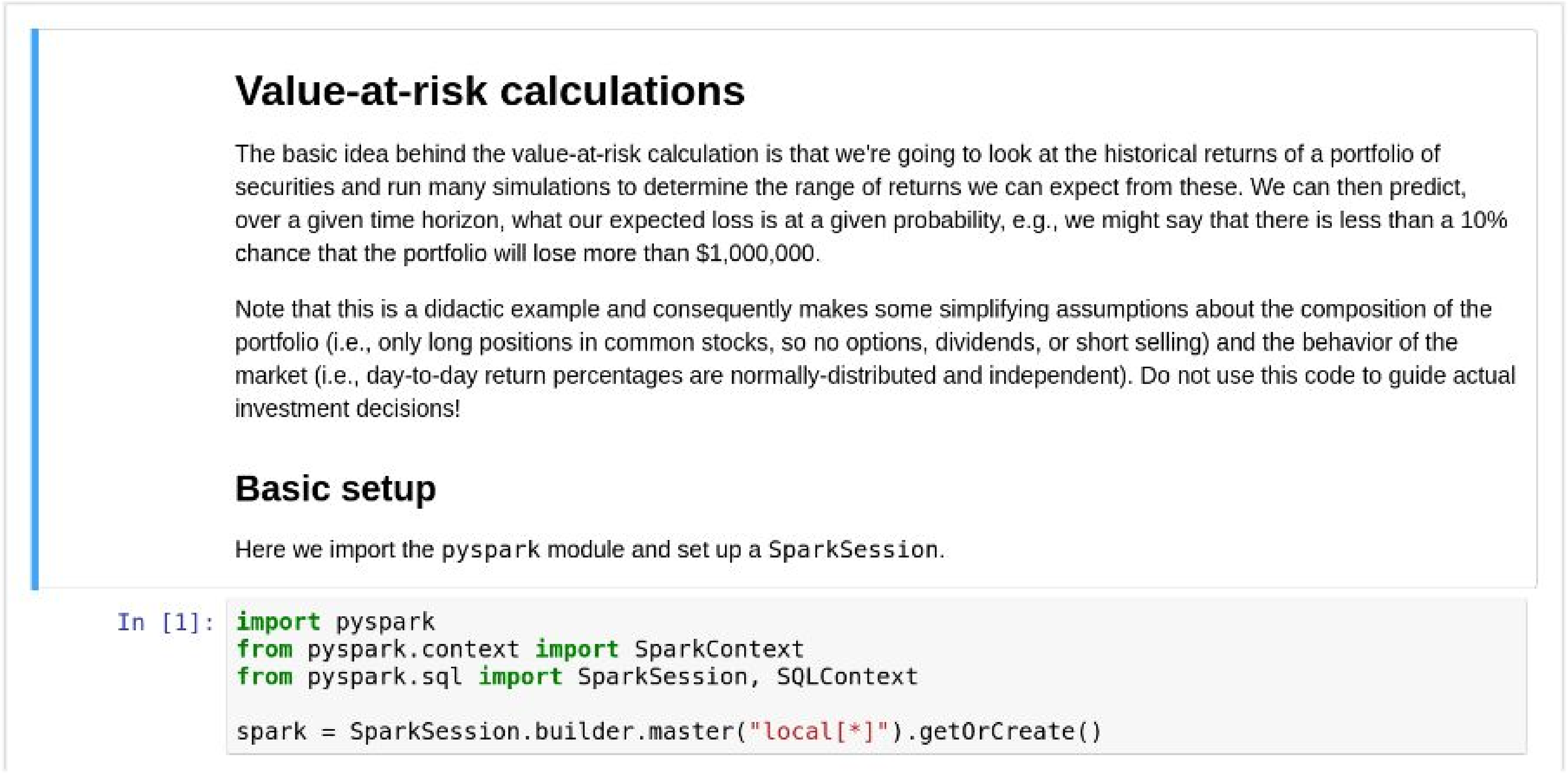
INTO THE SANDBOX

MOVING OUT OF ALPHA



LESSONS LEARNED

CODE COMMENTS 3.0



Value-at-risk calculations

The basic idea behind the value-at-risk calculation is that we're going to look at the historical returns of a portfolio of securities and run many simulations to determine the range of returns we can expect from these. We can then predict, over a given time horizon, what our expected loss is at a given probability, e.g., we might say that there is less than a 10% chance that the portfolio will lose more than $1,000,000.

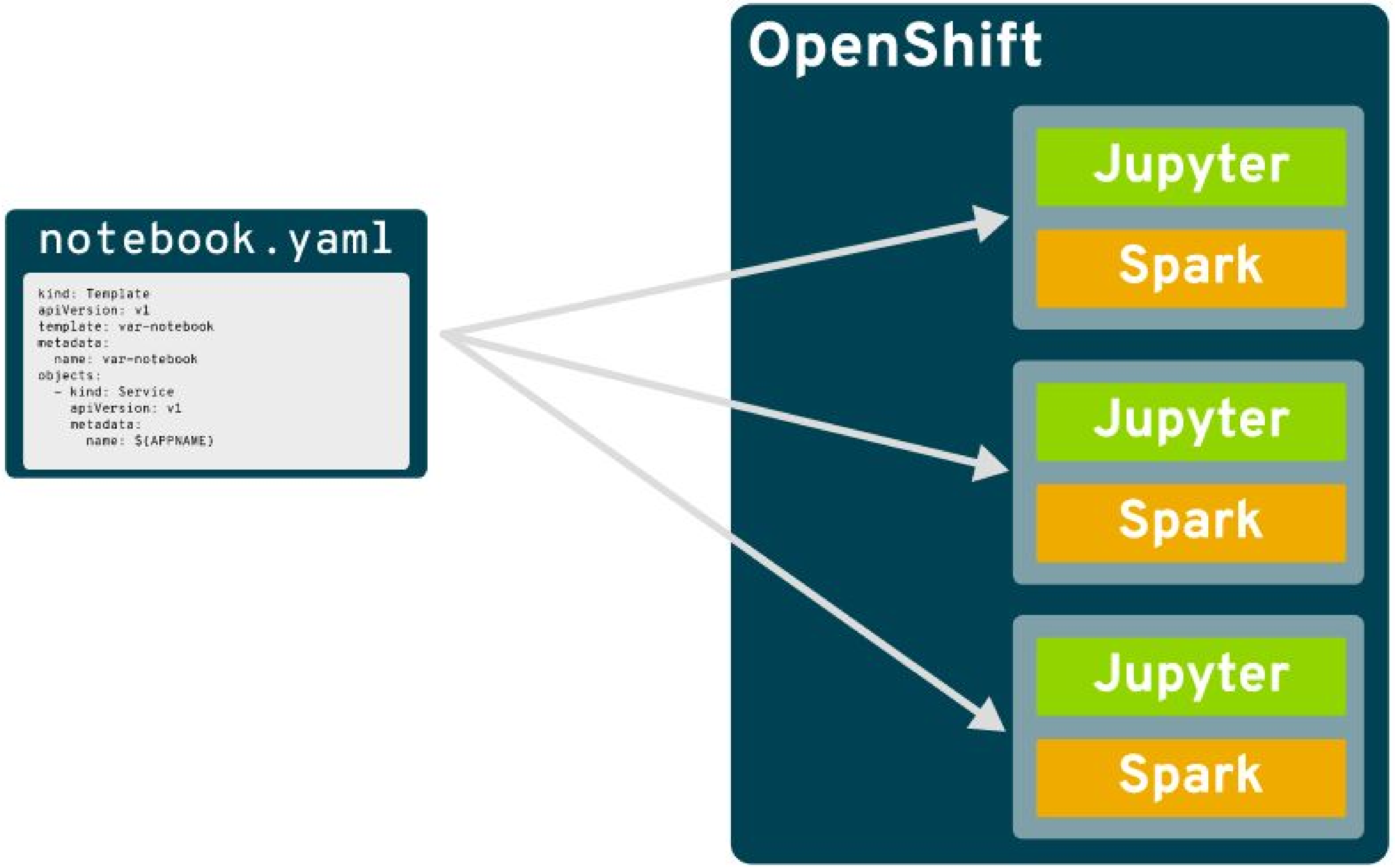
Note that this is a didactic example and consequently makes some simplifying assumptions about the composition of the portfolio (i.e., only long positions in common stocks, so no options, dividends, or short selling) and the behavior of the market (i.e., day-to-day return percentages are normally-distributed and independent). Do not use this code to guide actual investment decisions!

Basic setup

Here we import the `pyspark` module and set up a `SparkSession`.

In [1]:
```
import pyspark
from pyspark.context import SparkContext
from pyspark.sql import SparkSession, SQLContext

spark = SparkSession.builder.master("local[*]").getOrCreate()
```

TEMPLATED REPEATABILITY

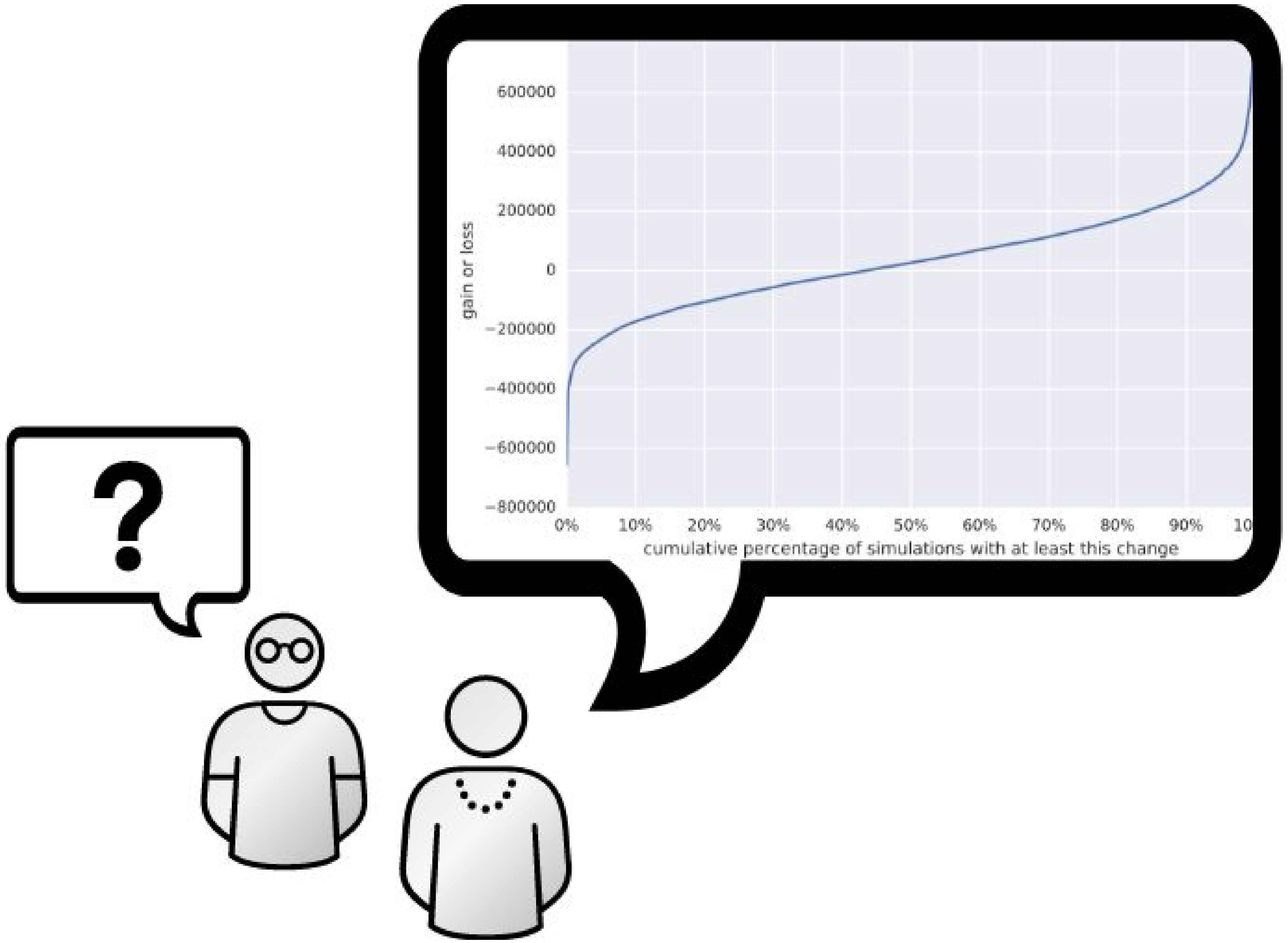


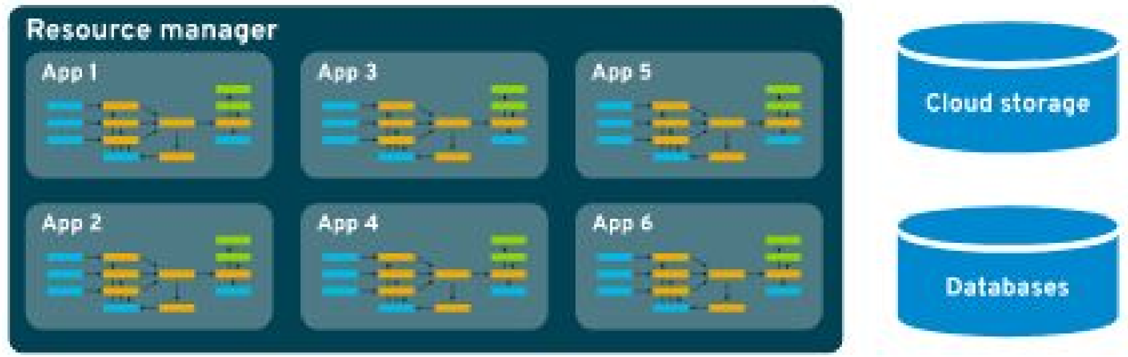
```
kind: Template
apiVersion: v1
template: var-notebook
metadata:
  name: var-notebook
objects:
  - kind: Service
    apiVersion: v1
    metadata:
      name: ${APPNAME}
```

IT'S NOT ALL ROSES

- tricky Spark configurations
- notebooks add infrastructure
- code can be difficult

YOUR GREATEST TOOL ON THE JOURNEY





Ingest
Process
Publish

msm@redhat.com
elmiko.github.com

radanalytics.io