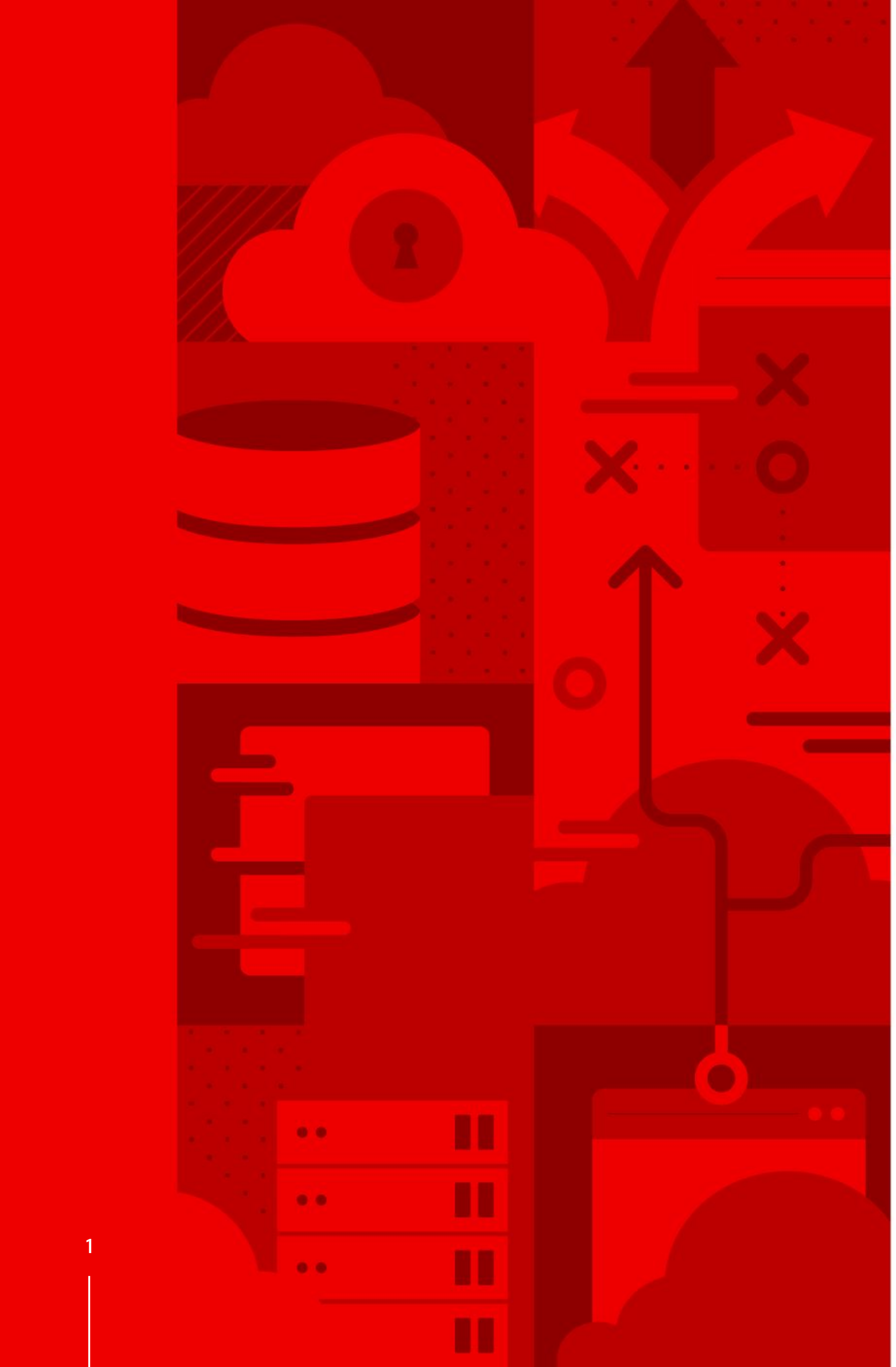


Machine Learning on OpenShift

From Data Scientist to Application Developer

Michael McCune
Principal Software Engineer

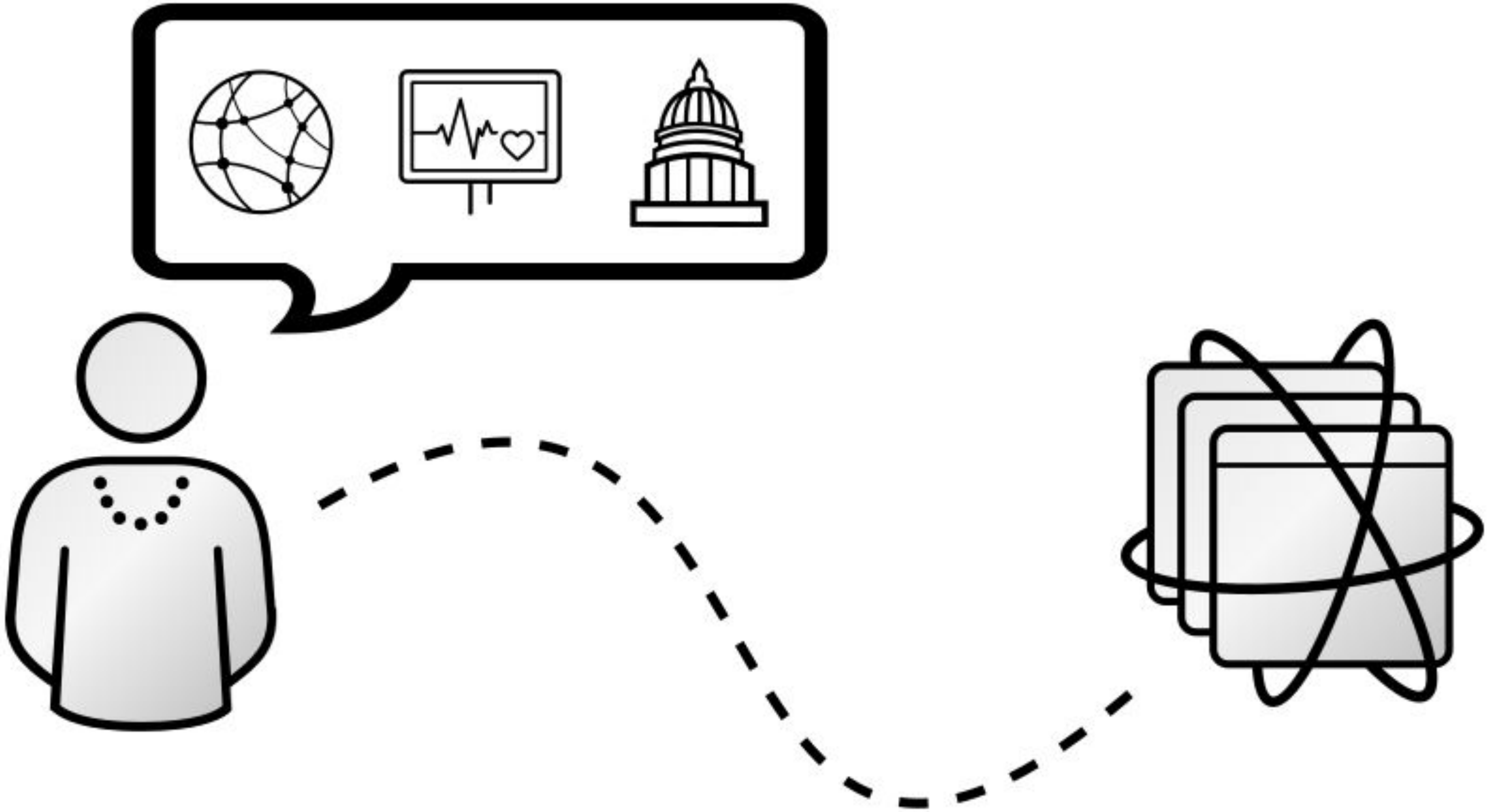
Forecast

- Introduction
- Technology Review
- Building Intelligent Applications
- Lessons Learned
- Next Steps

Who is this guy?

- Joined Red Hat 6 years ago
- Full stack, from embedded to orchestration
- Emerging technology at Red Hat
- Big Data on OpenStack and OpenShift

We are talking about a journey

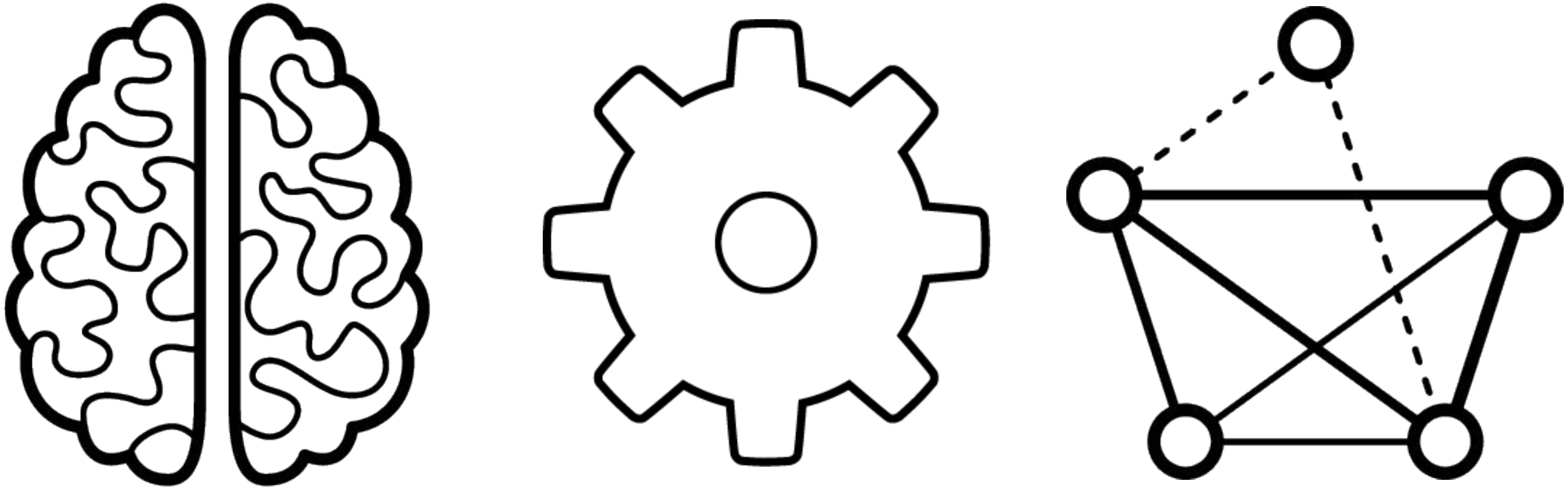


photography by Sam Hawley

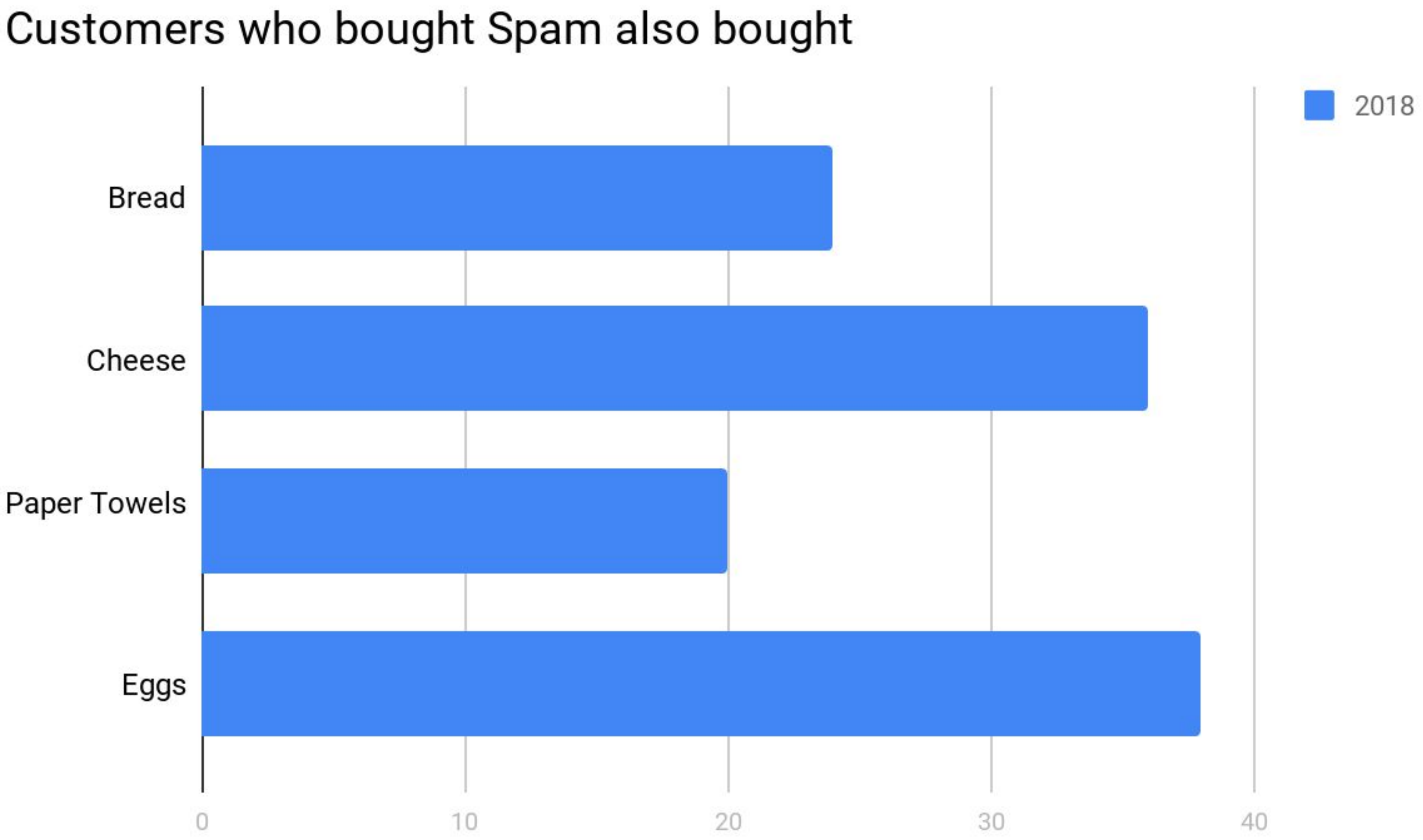
photography by Diliff and Janke

Level Set

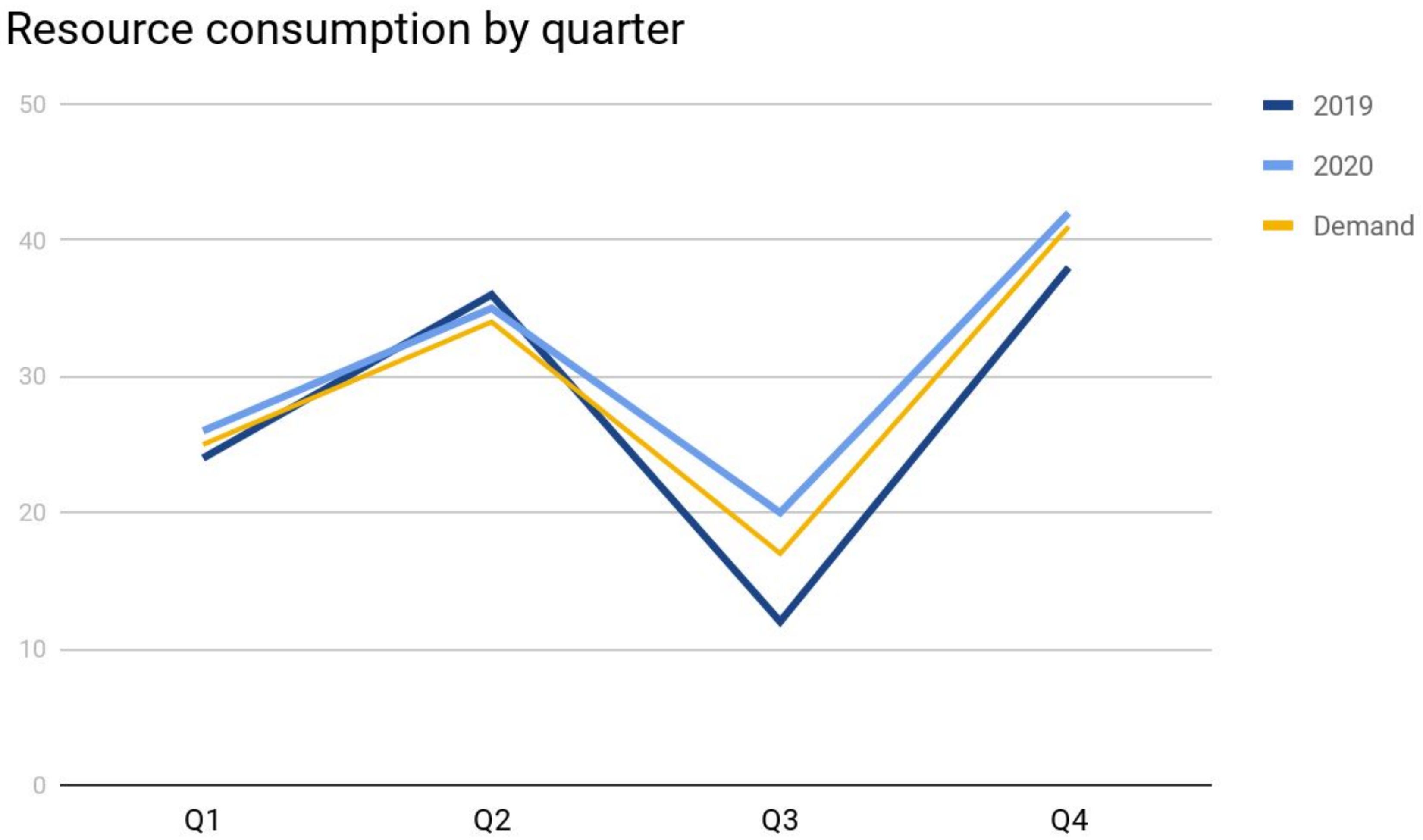
Machine Learning



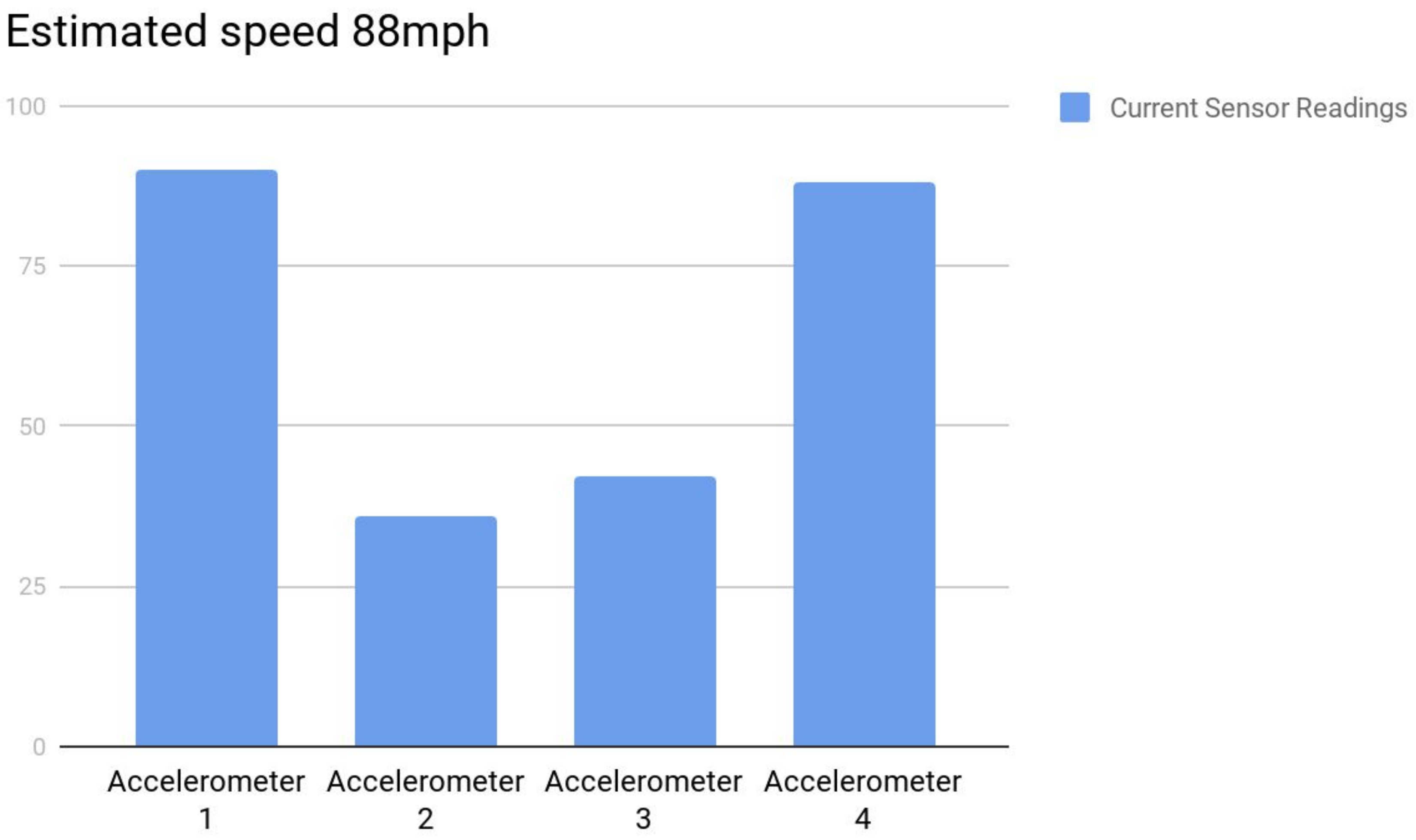
Identifying Trends



Predicting Results

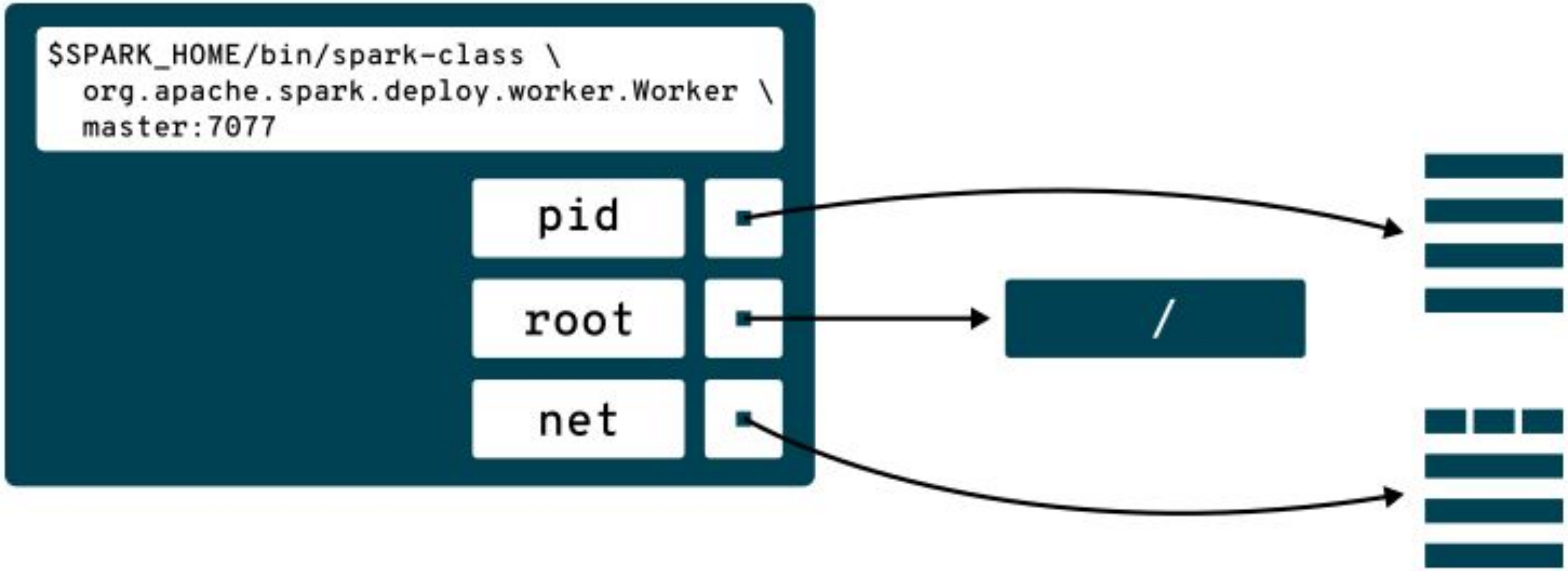


Estimating Values



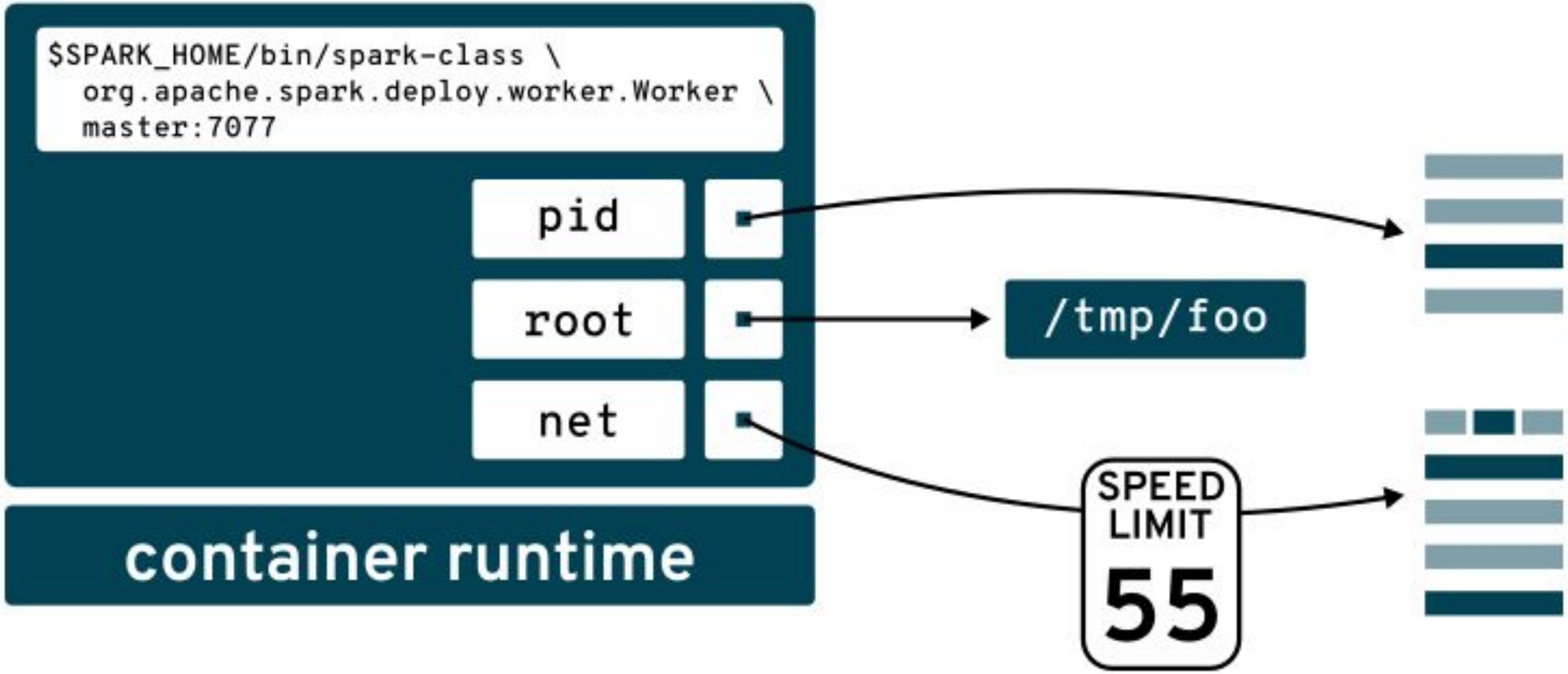
Kubernetes

Let’s talk containers

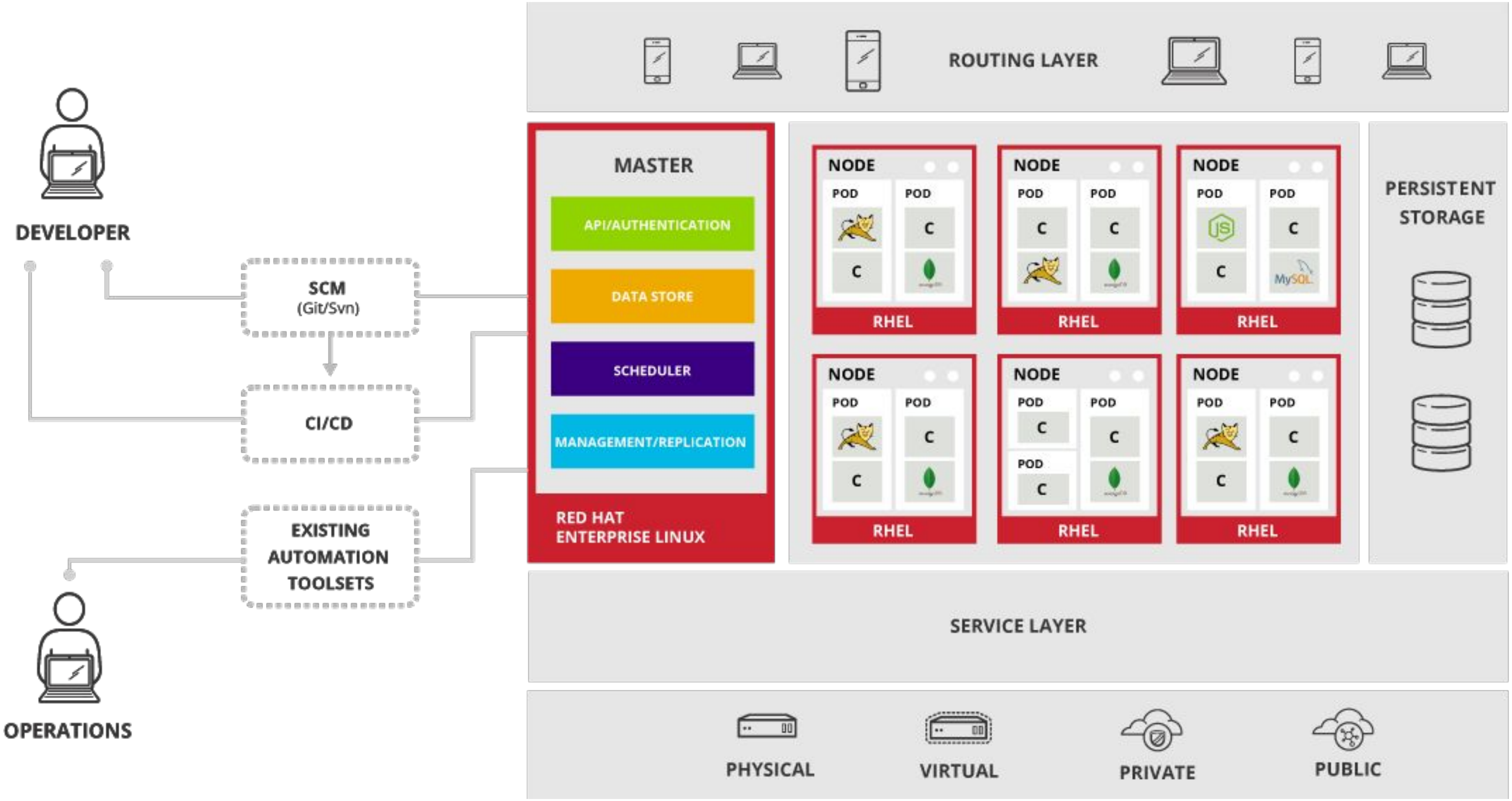


```
$SPARK_HOME/bin/spark-class \
  org.apache.spark.deploy.worker.Worker \
  master:7077
```

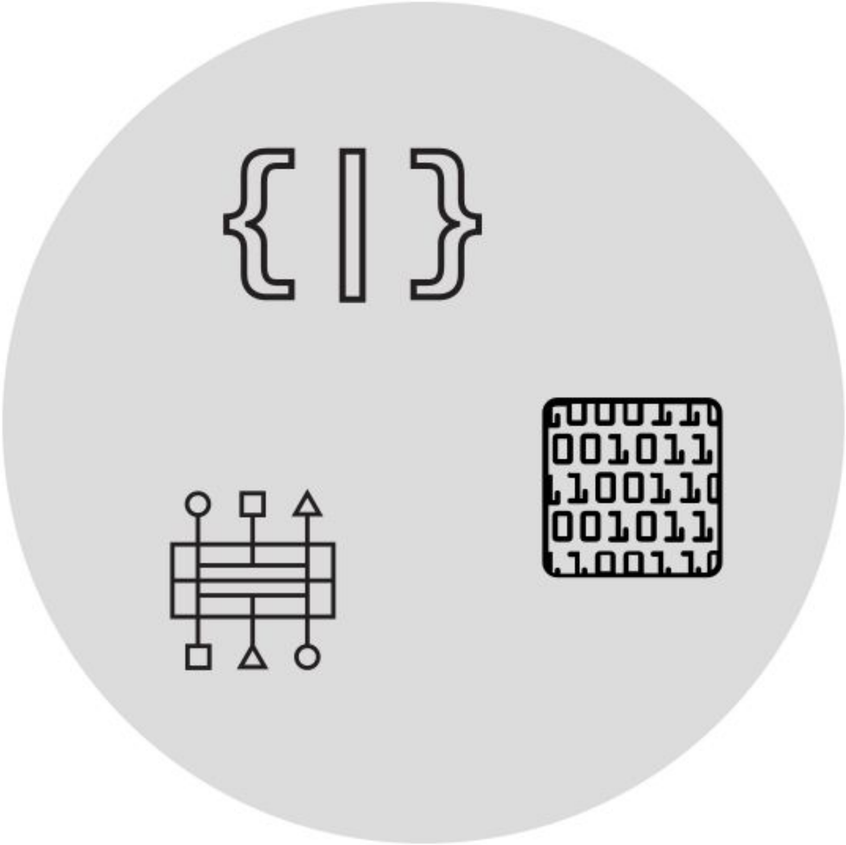
Let's talk containers



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

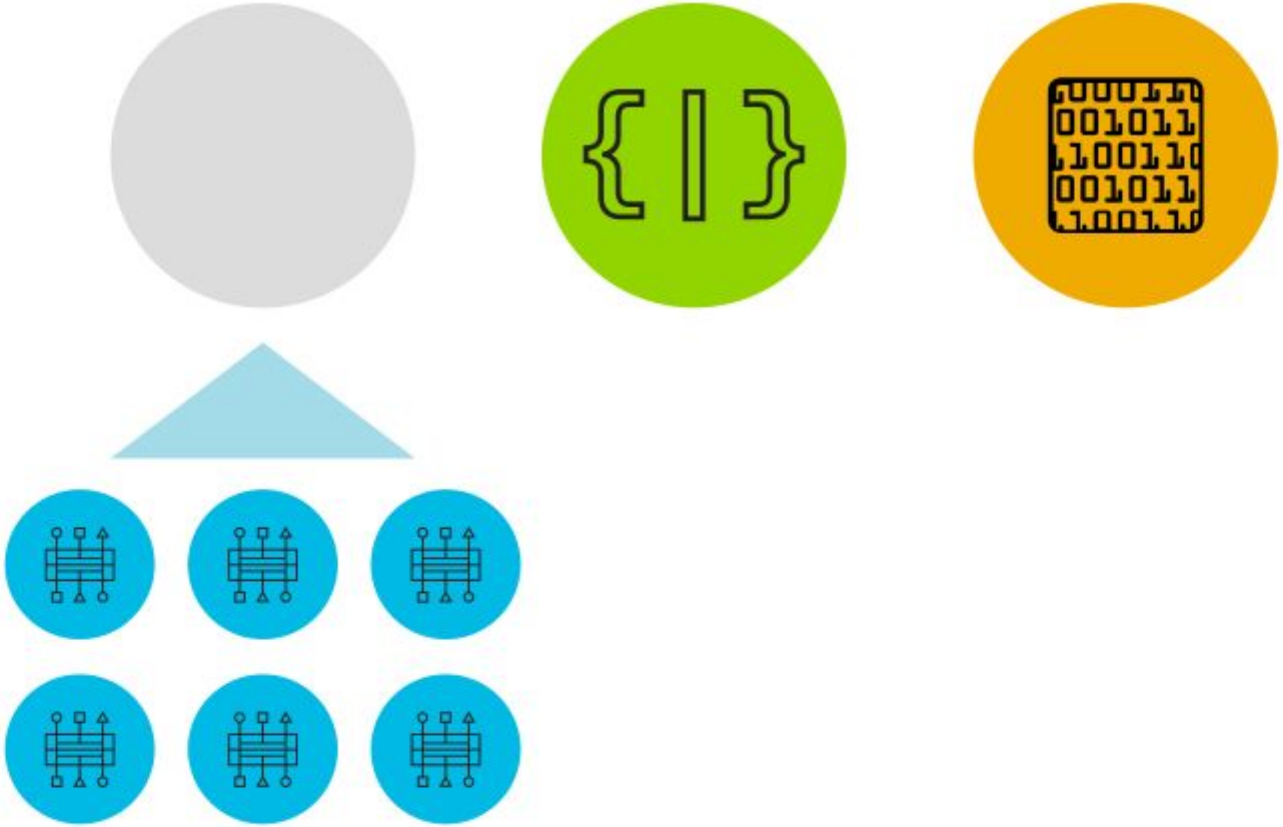

DEVELOPER
SCM
(Git/Svn)
CI/CD
EXISTING AUTOMATION TOOLSETS
OPERATIONS
ROUTING LAYER
MASTER
API/AUTHENTICATION
DATA STORE
SCHEDULER
MANAGEMENT/REPLICATION
RED HAT ENTERPRISE LINUX
NODE
POD
C
RHEL
MySQL
PERSISTENT STORAGE
SERVICE LAYER
PHYSICAL
VIRTUAL
PRIVATE
PUBLIC

Microservice Architectures

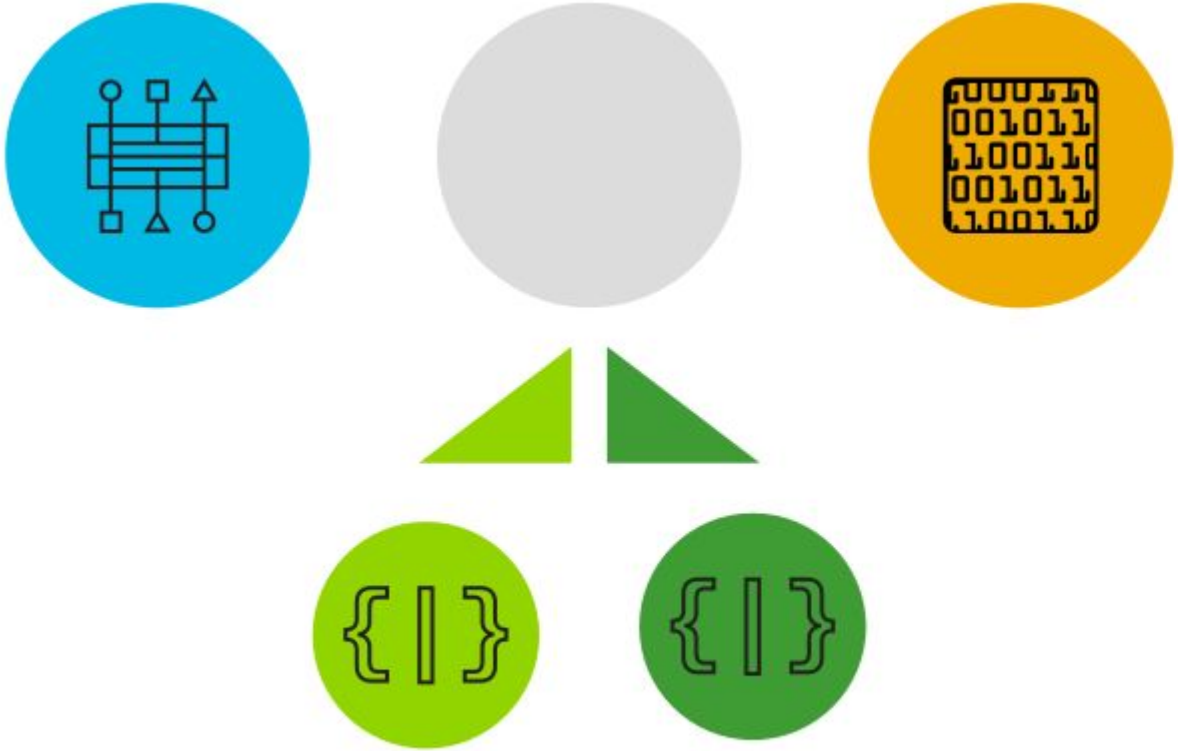


Composable

Scalable

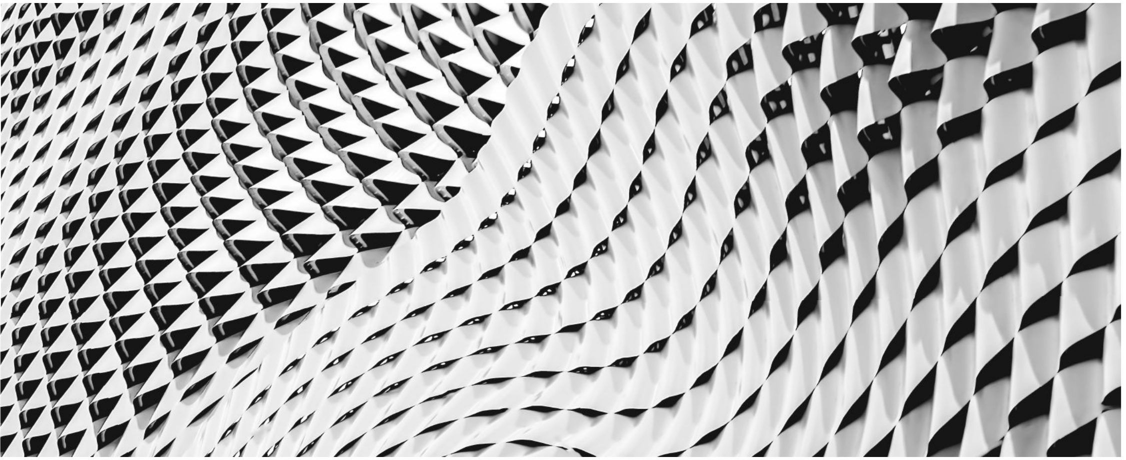


Flexible



Resilient

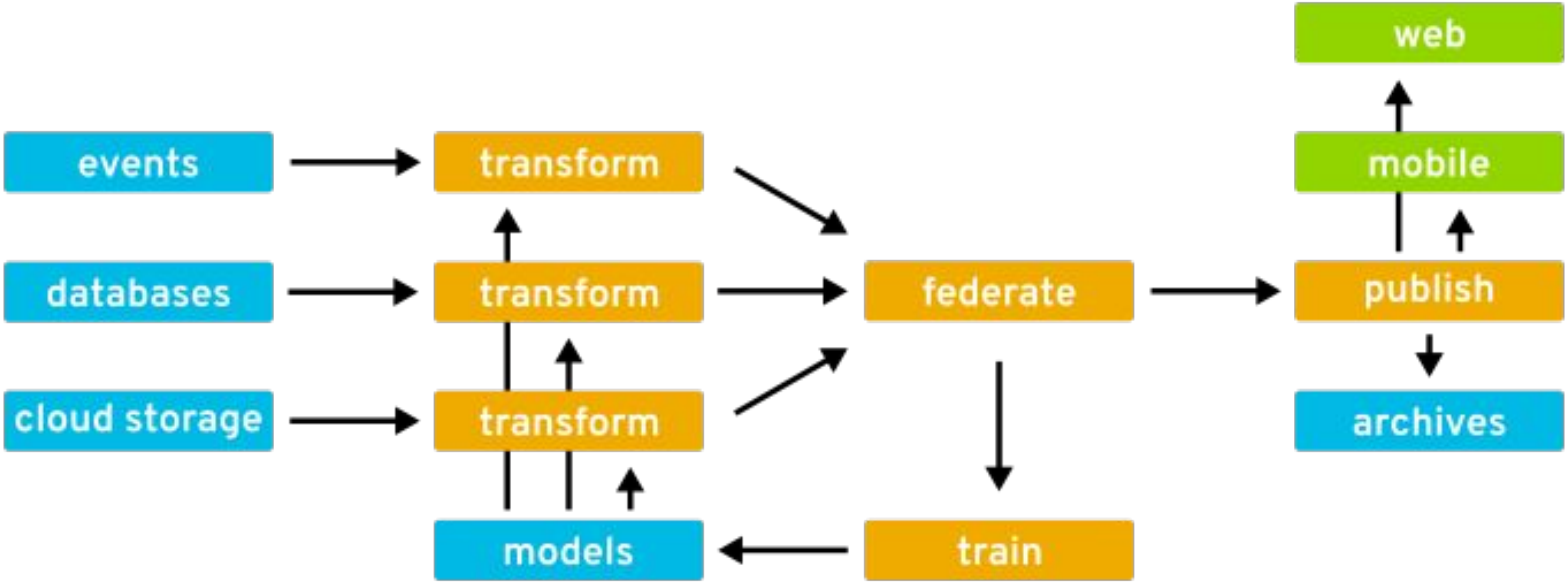
Intelligent Applications



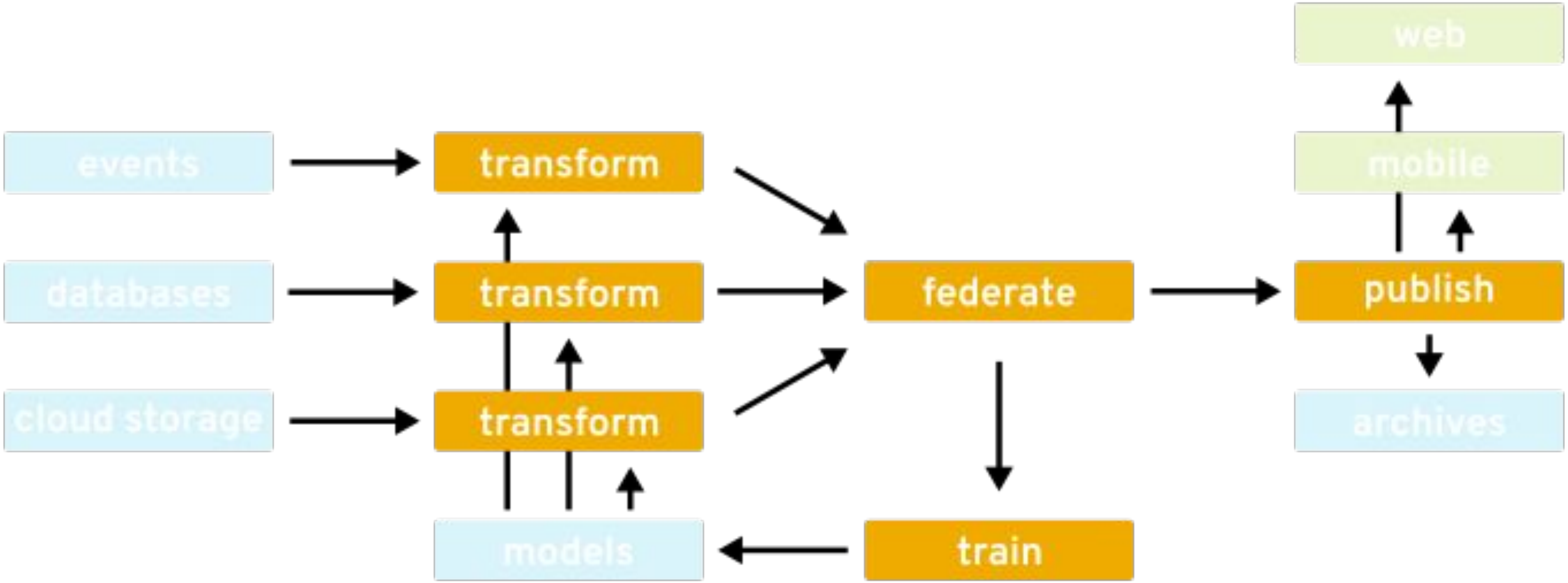
Intelligent applications **collect and learn from data** to provide **improved functionality** with **longevity and popularity**.

Red Hat

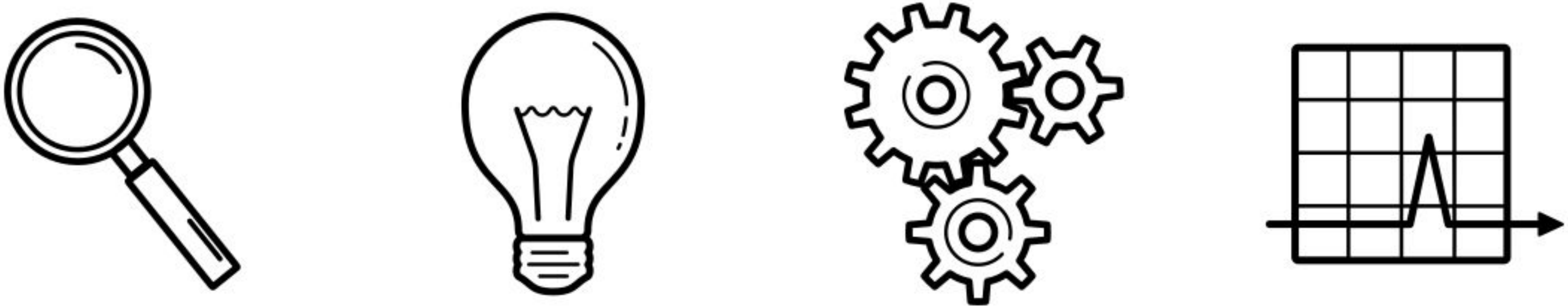
Intelligent Application Pipeline



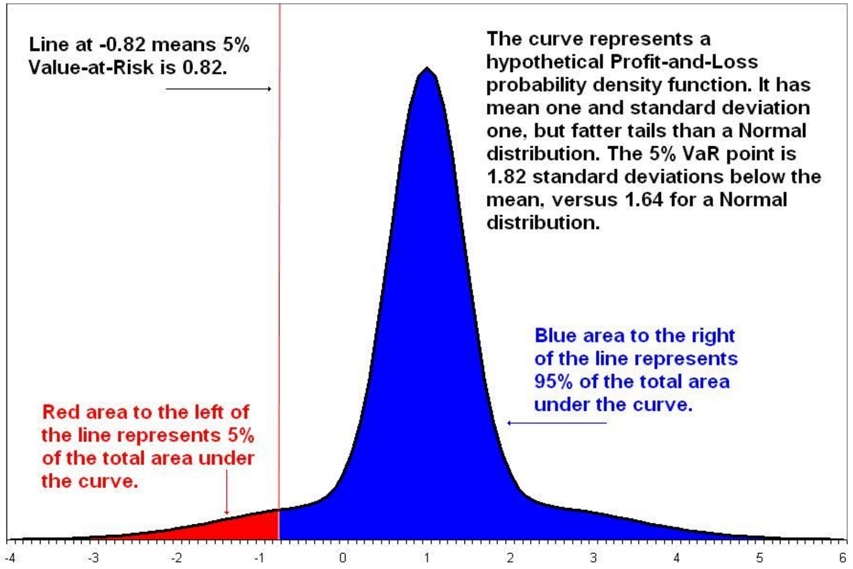
Service Based Components



Intelligent Application Lifecycle



Case Study
Value at Risk



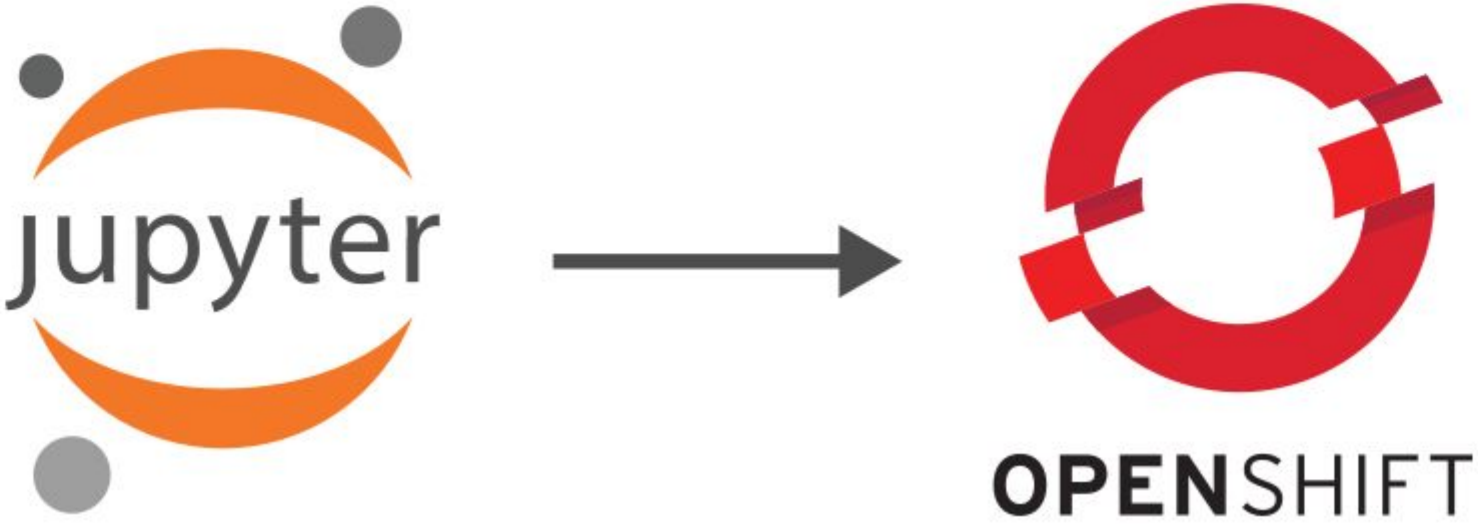
Line at -0.82 means 5% Value-at-Risk is 0.82.

The curve represents a hypothetical Profit-and-Loss probability density function. It has mean one and standard deviation one, but fatter tails than a Normal distribution. The 5% VaR point is 1.82 standard deviations below the mean, versus 1.64 for a Normal distribution.

Blue area to the right of the line represents 95% of the total area under the curve.

Red area to the left of the line represents 5% of the total area under the curve.

Going Cloud Native

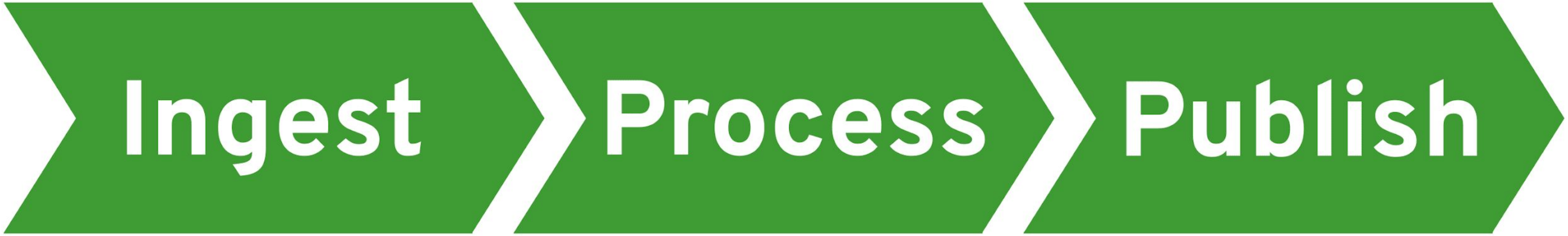


What is Cloud Native?

CLOUD NATIVE
COMPUTING FOUNDATION

- Containerized
- Dynamically orchestrated
- Microservice architectures
- learn more at **cncf.io/about/faq**

What will your application do?

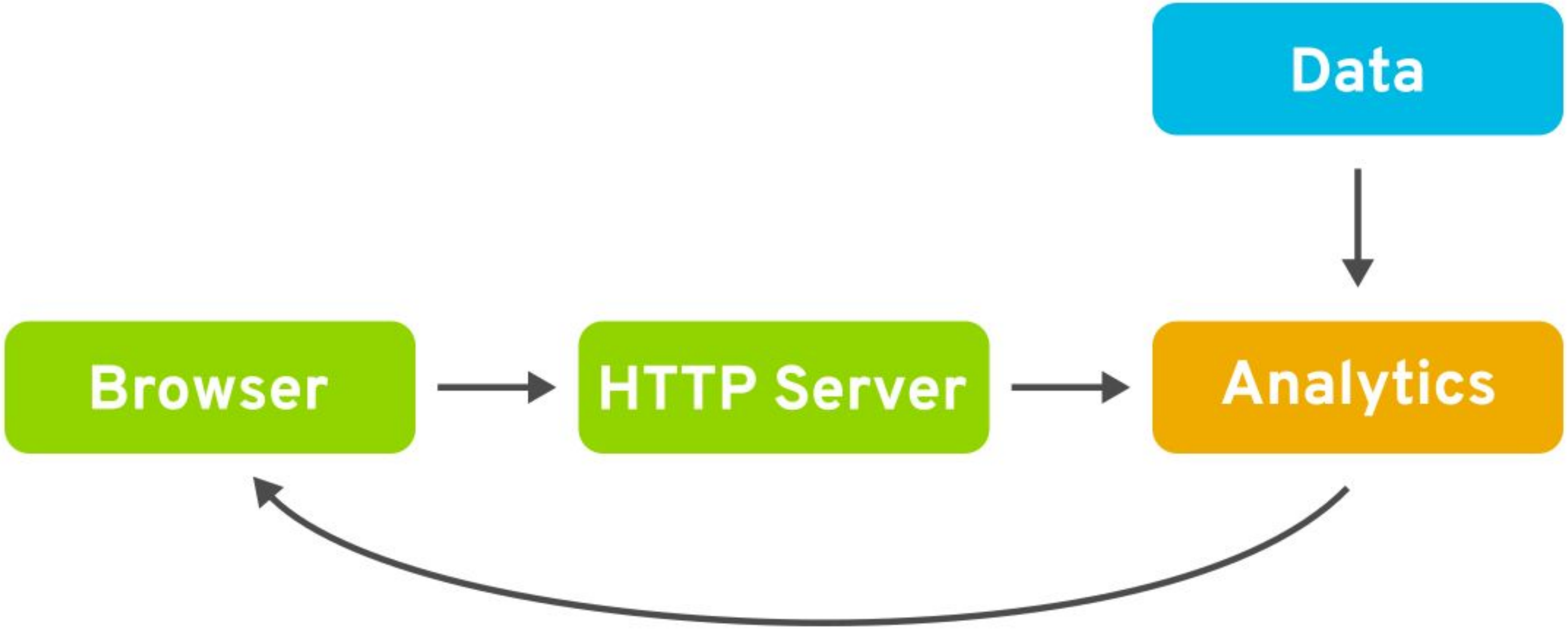


Ingest

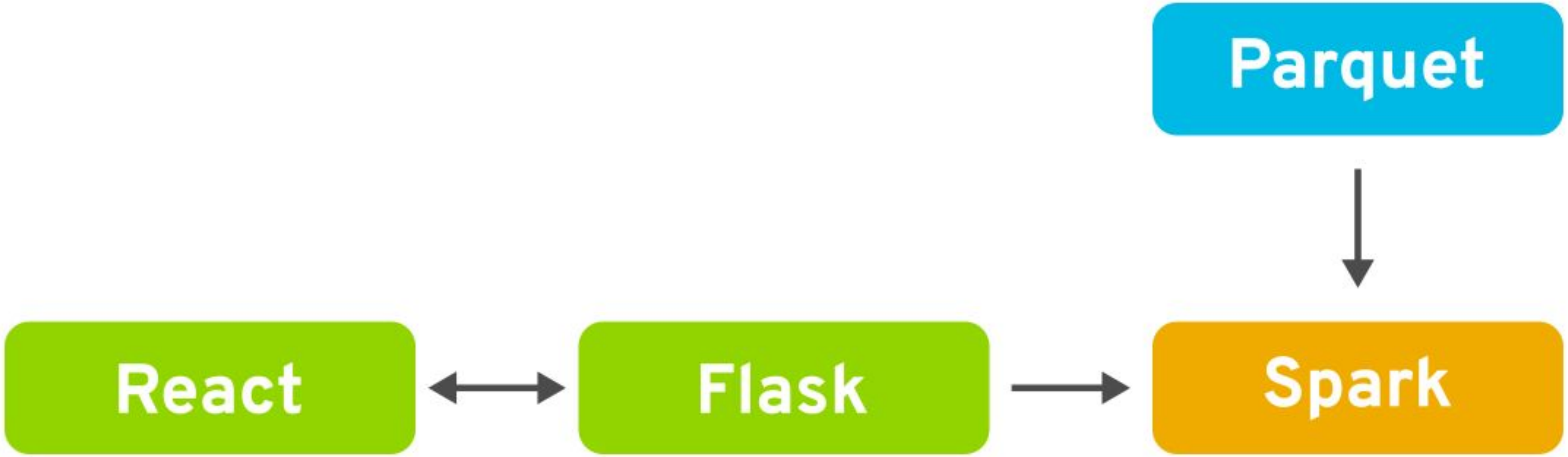
Process

Publish

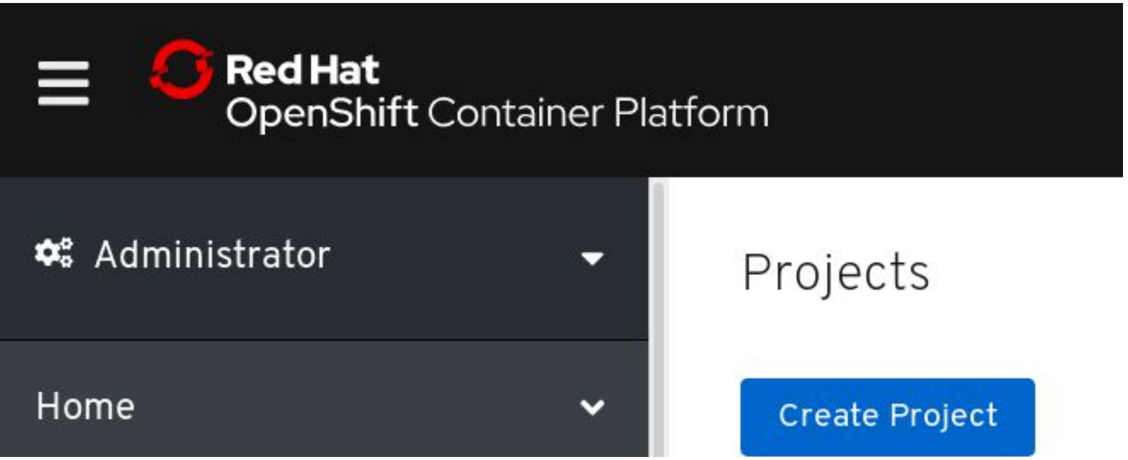
Storyboard your architecture



Choose wisely

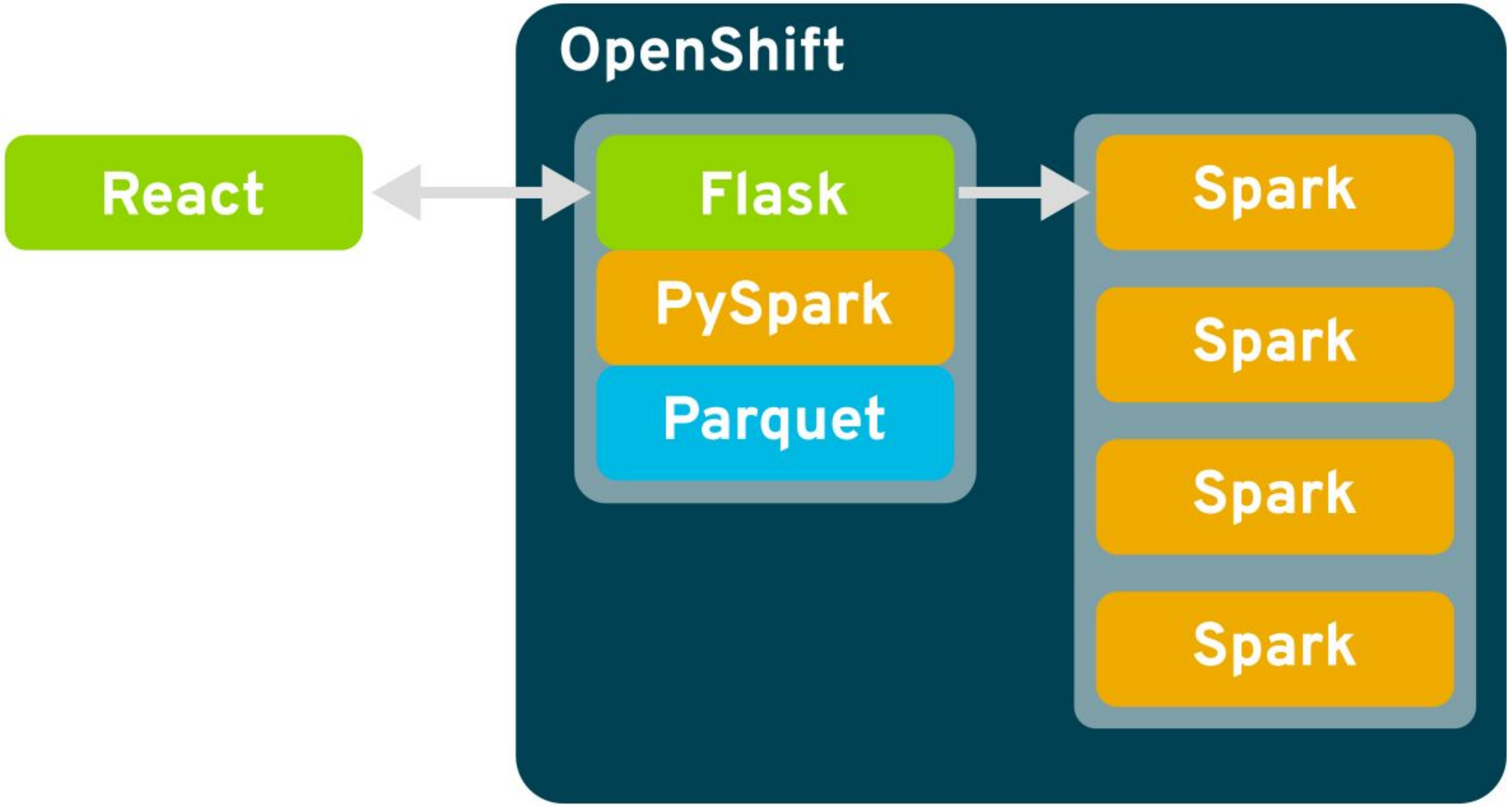


VAR Demo

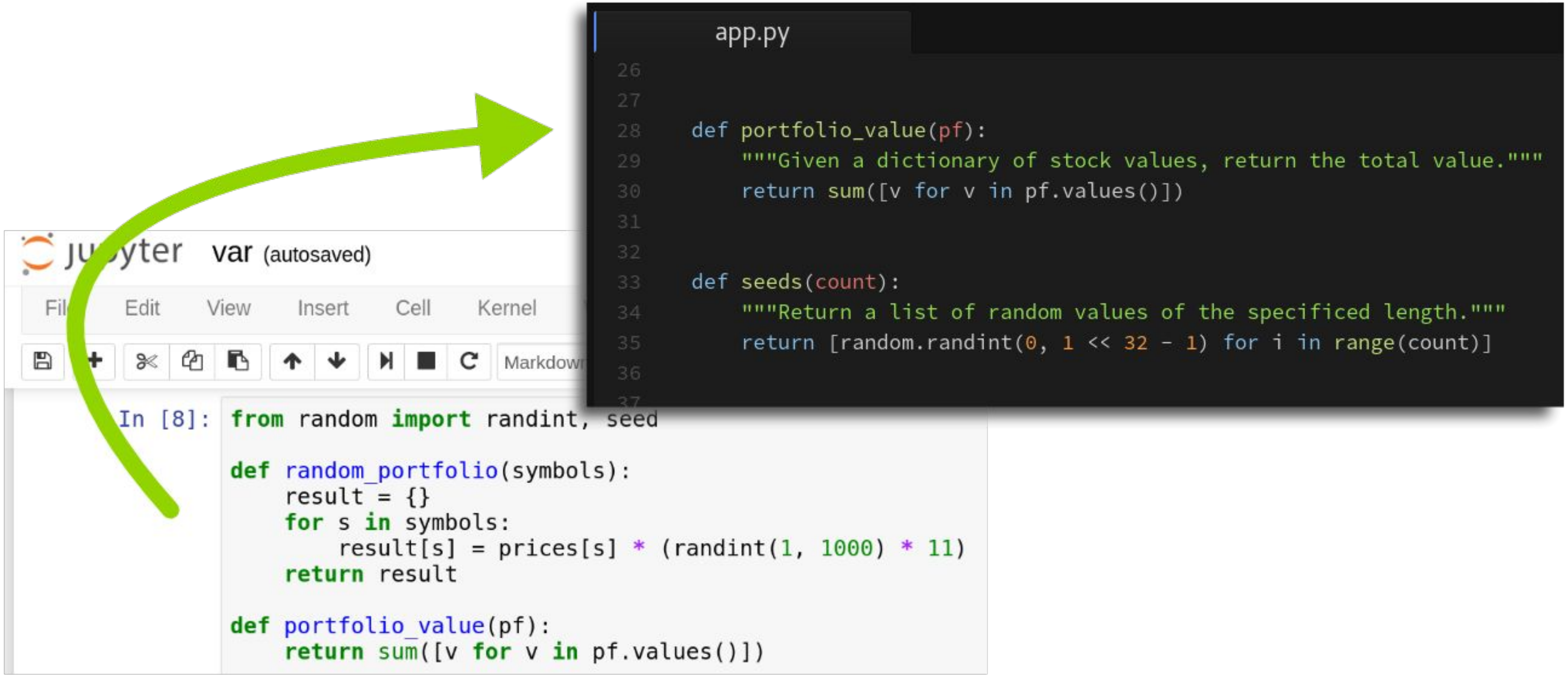


Red Hat OpenShift Container Platform

General architecture



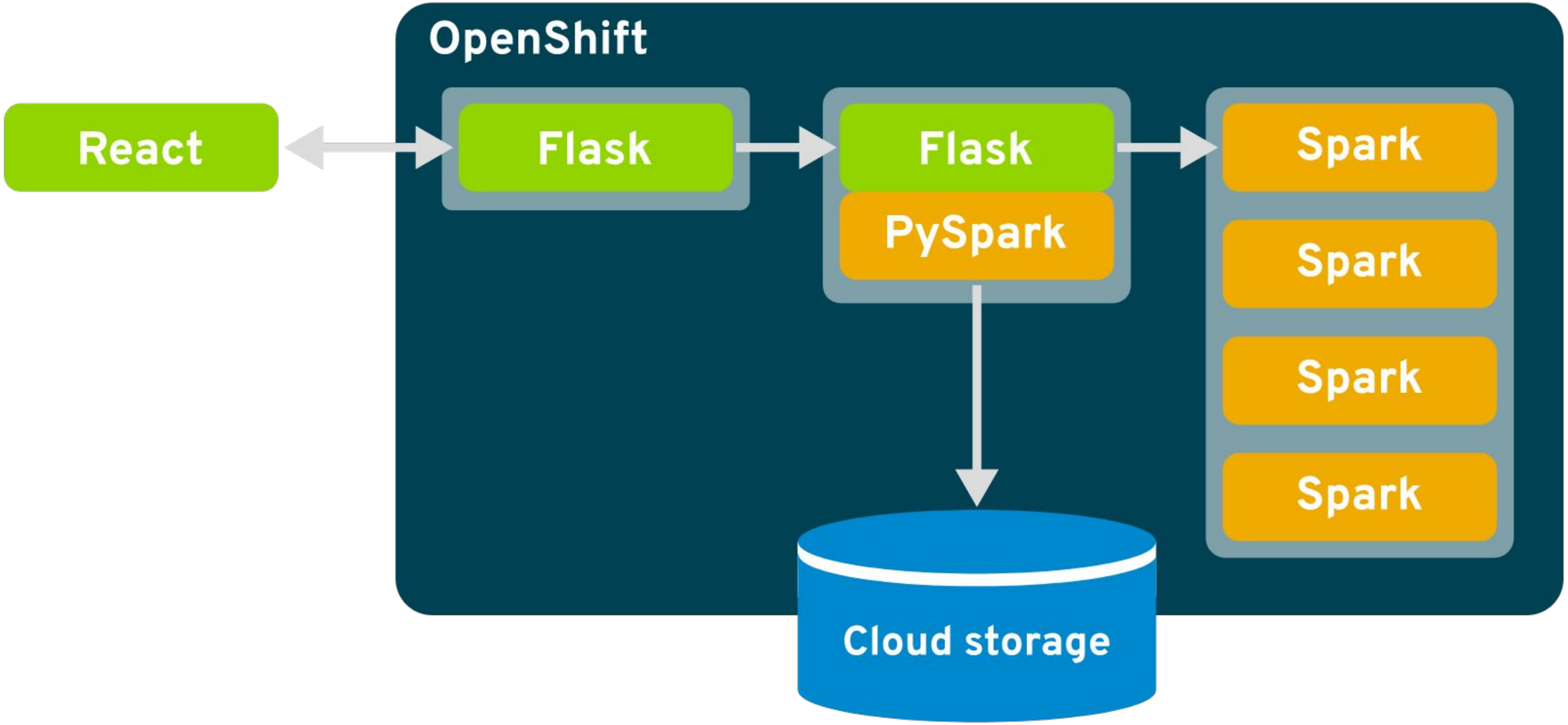
How was it built?



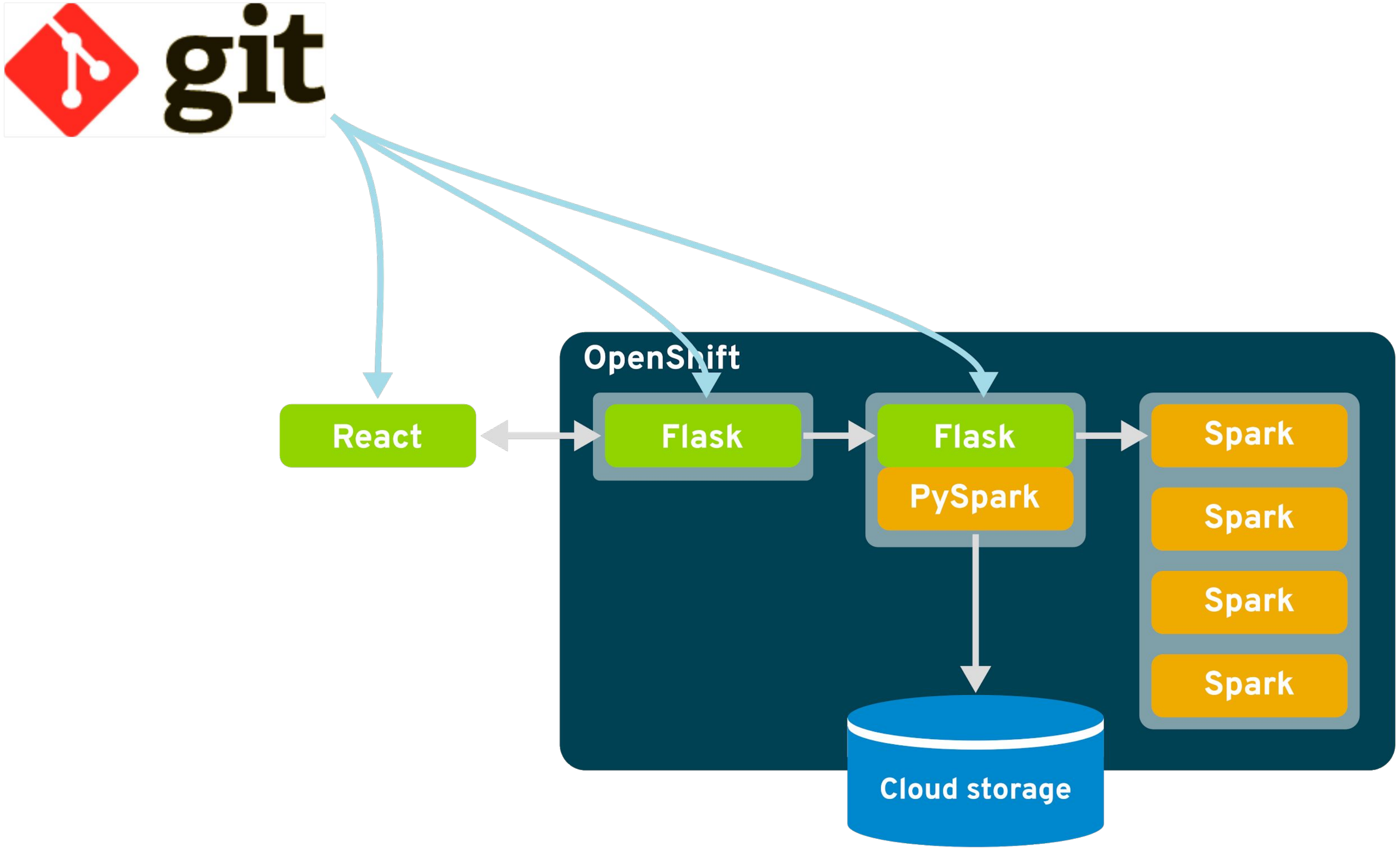
```python
def portfolio_value(pf):
    """Given a dictionary of stock values, return the total value."""
    return sum([v for v in pf.values()])


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

```python
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()])
```

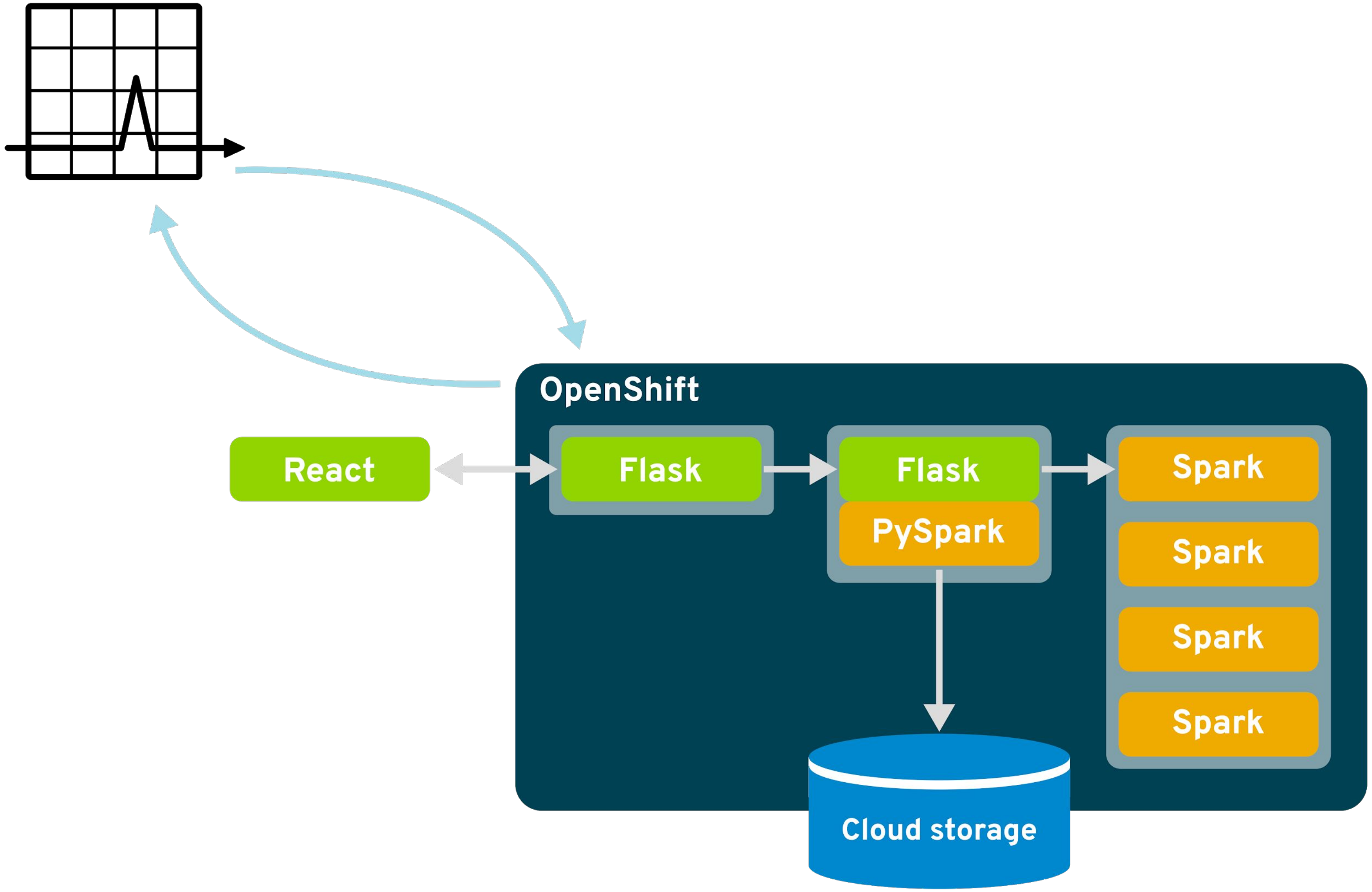
Moving out of alpha



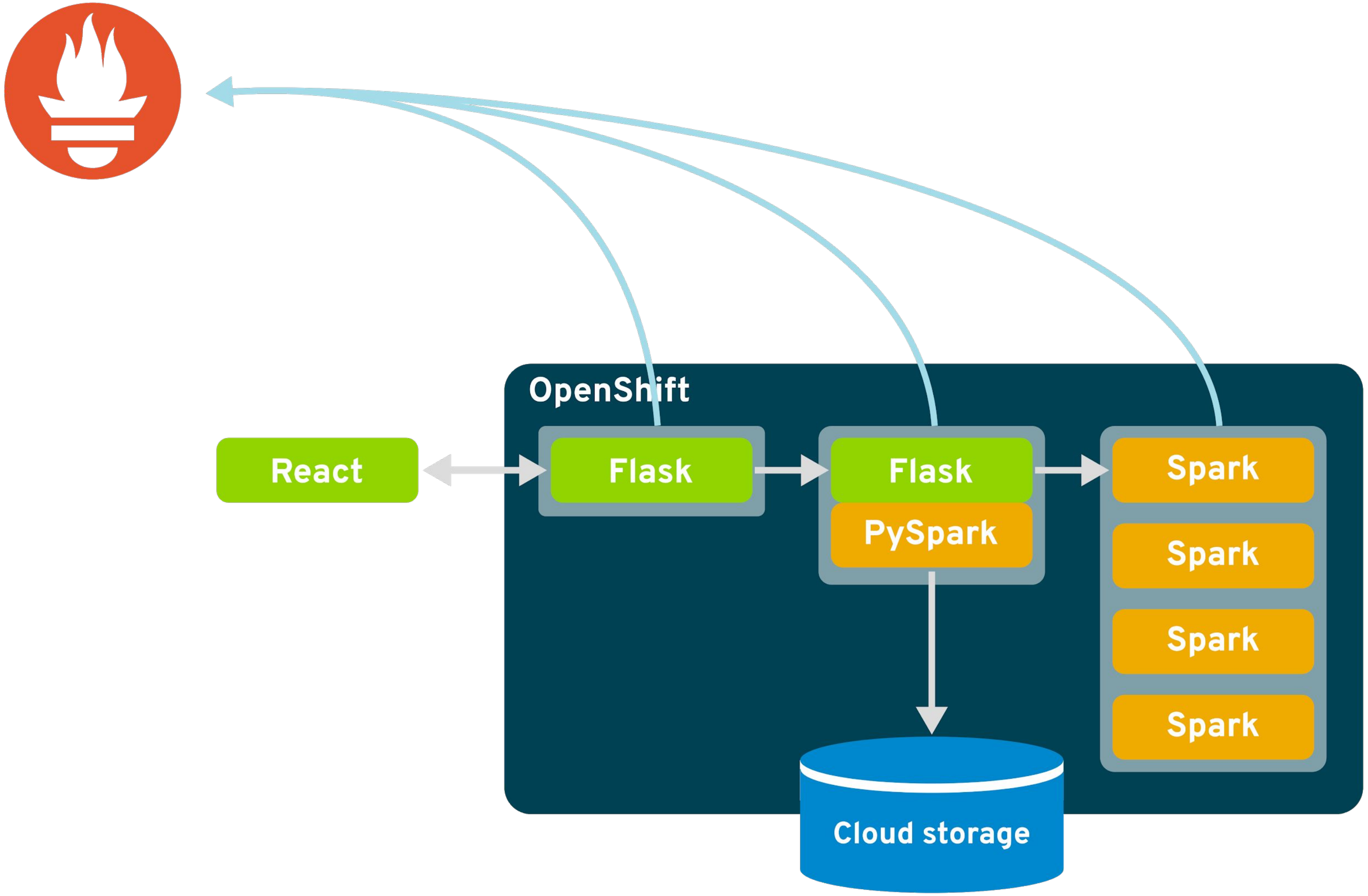
Layer in DevOps



Augment with lifecycle monitoring

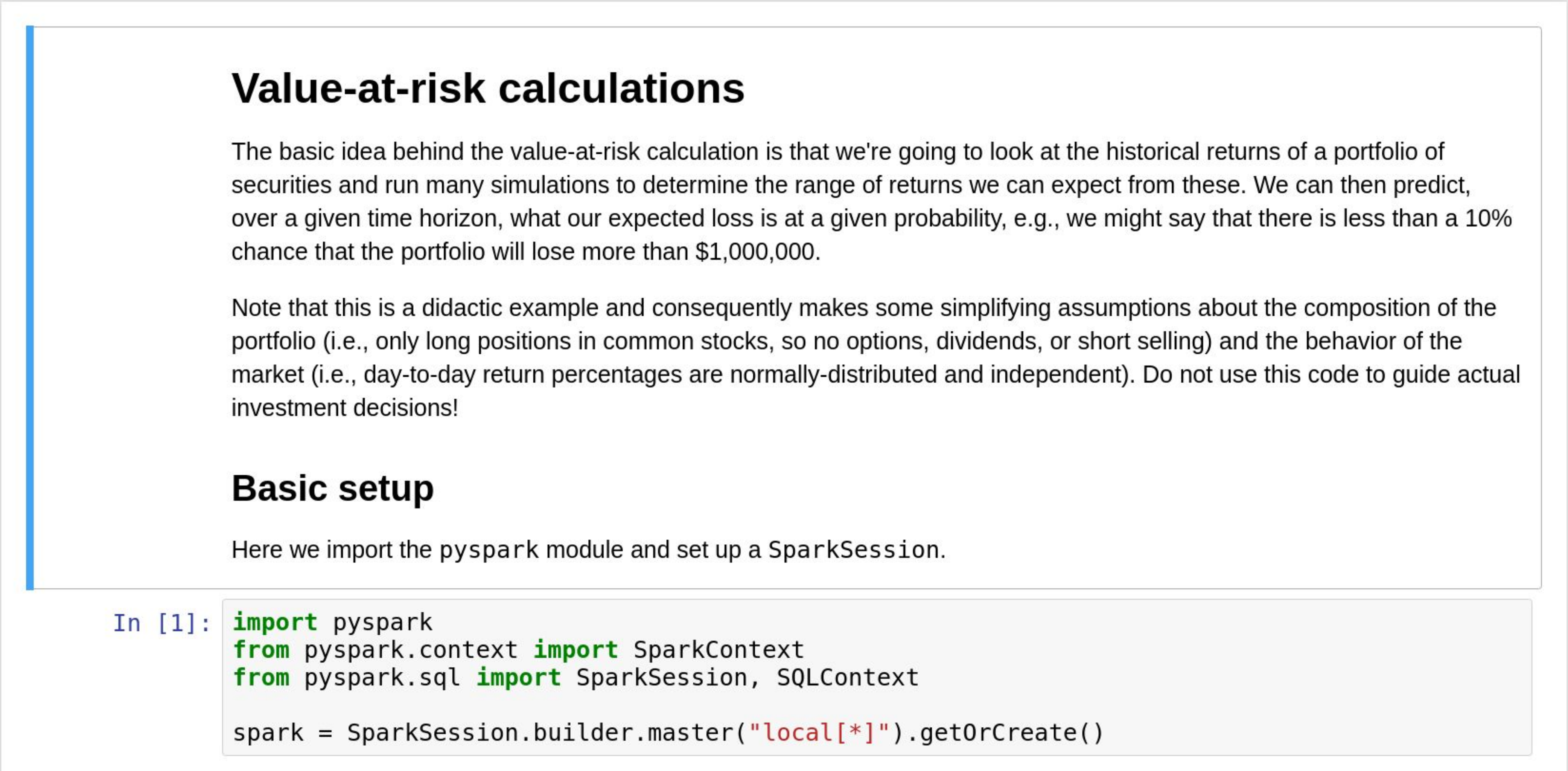


Harvest customized metrics



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.

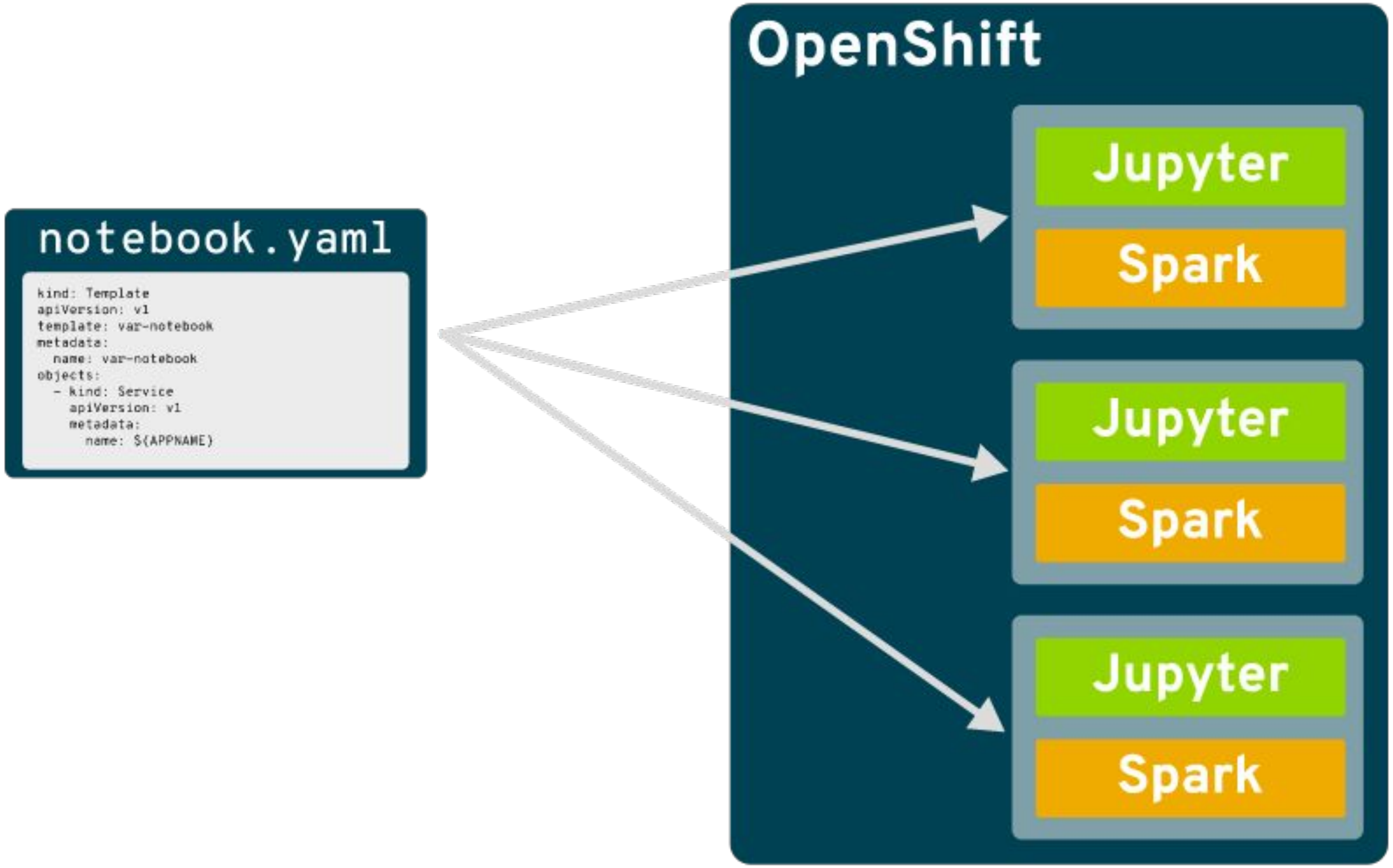
Note that this is a didactic example and consequently makes some simplifying assumptions about the composition of the portfolio (i.e., only long positions in common stocks, so no options, dividends, or short selling) and the behavior of the market (i.e., day-to-day return percentages are normally-distributed 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`.

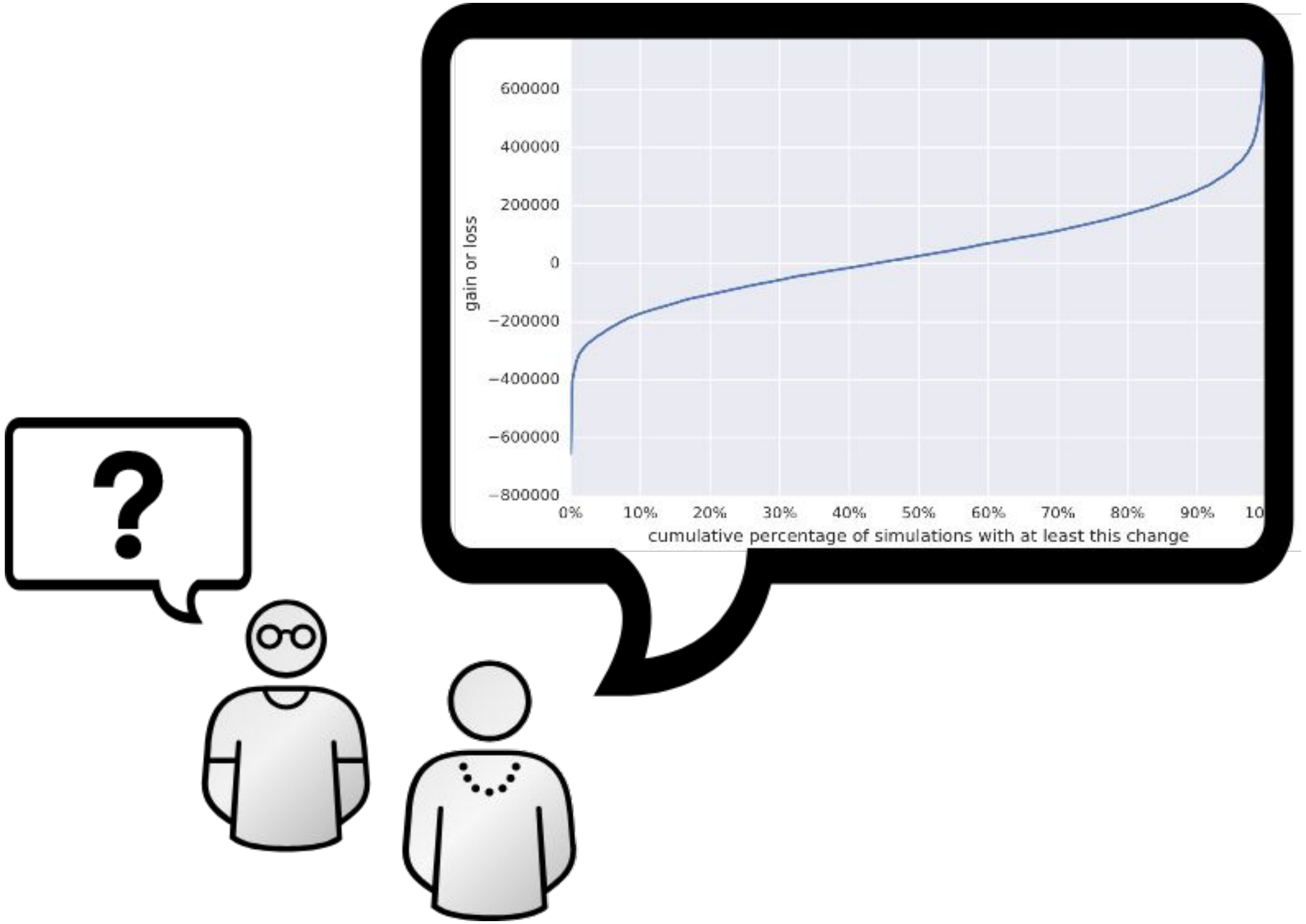
```
In [1]: import pyspark
        from pyspark.context import SparkContext
        from pyspark.sql import SparkSession, SQLContext

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

Templated Repeatability

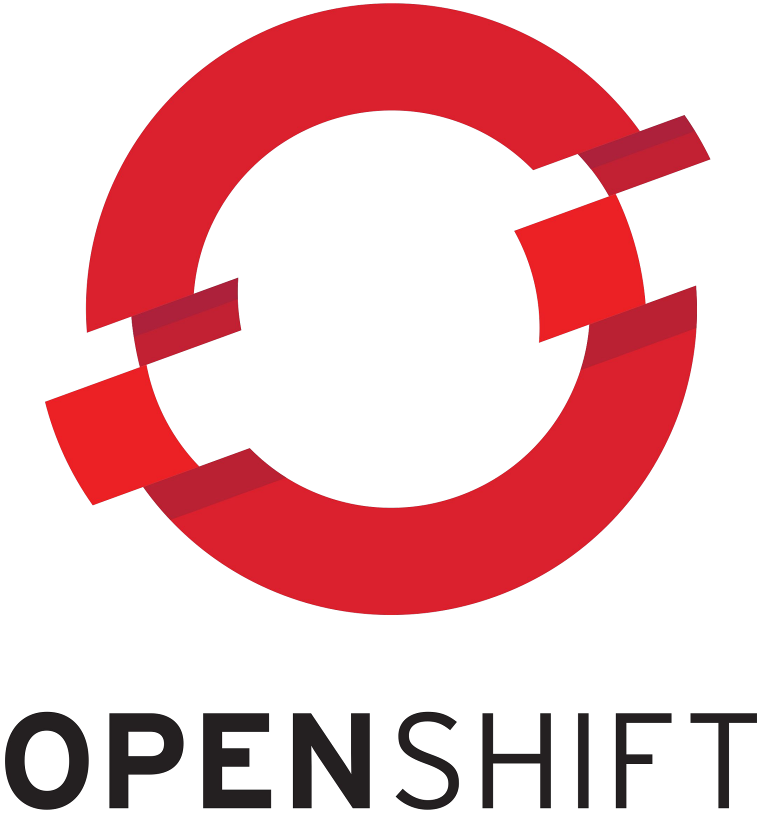


```
kind: Template
apiVersion: v1
template: var-notebook
metadata:
  name: var-notebook
objects:
  - kind: Service
    apiVersion: v1
    metadata:
      name: ${APPNAME}
```

Your greatest tool on the journey



Second greatest tool?



What Next?

AI/ML on OpenShift

www.openshift.com/learn/topics/ai-ml

Open Data Hub

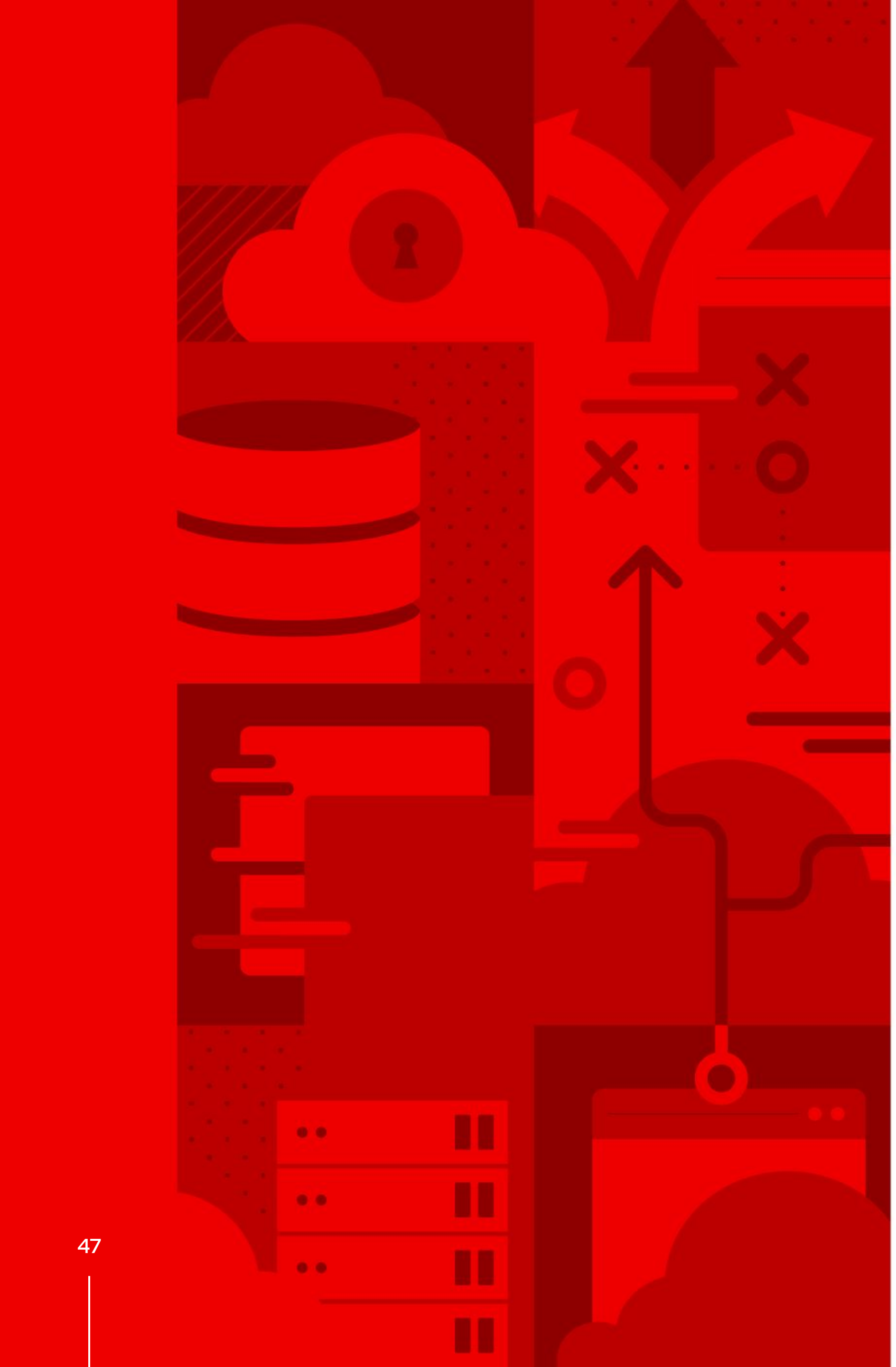
OPEN DATA HUB

AI Platform powered by Open Source

opendatahub.io

Radanalytics

radanalytics.io



Thank you

Keep in touch

elmiko@redhat.com

@elmiko@mastodon.technology

github.com/elmiko/rhug-artifacts