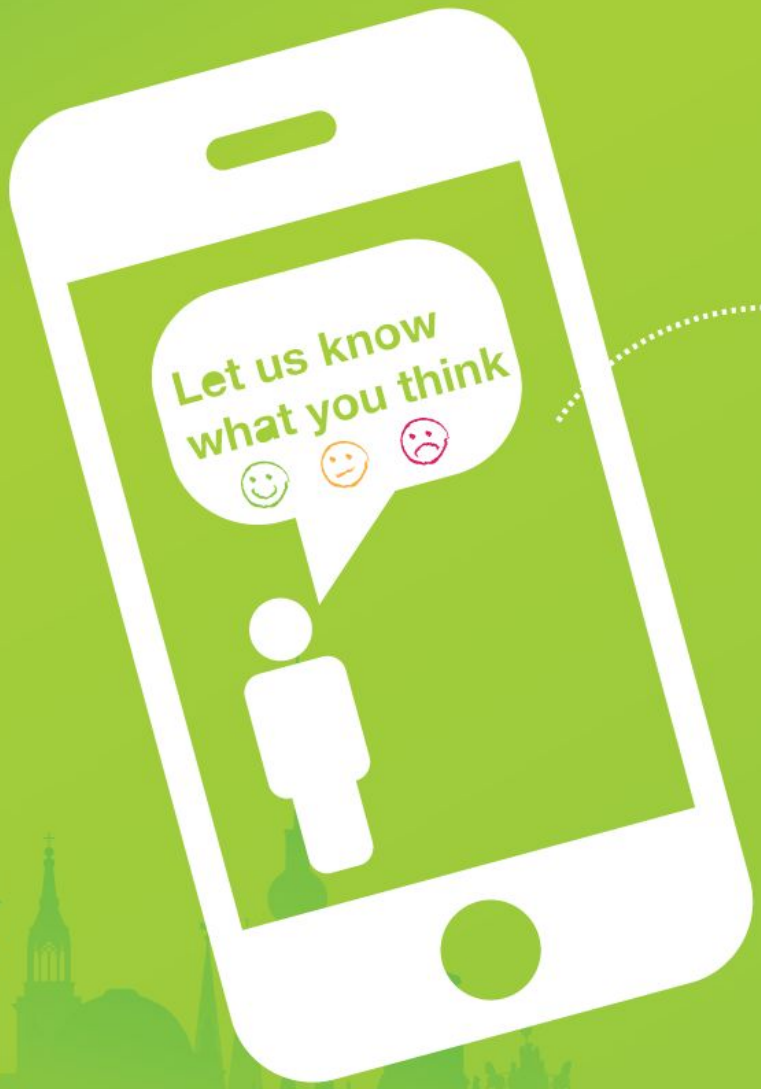
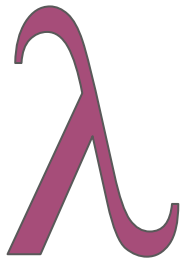
goto; conference



Click ‘engage’
to rate sessions
and ask questions

christian.stamm@otto.de
@christian_stamm

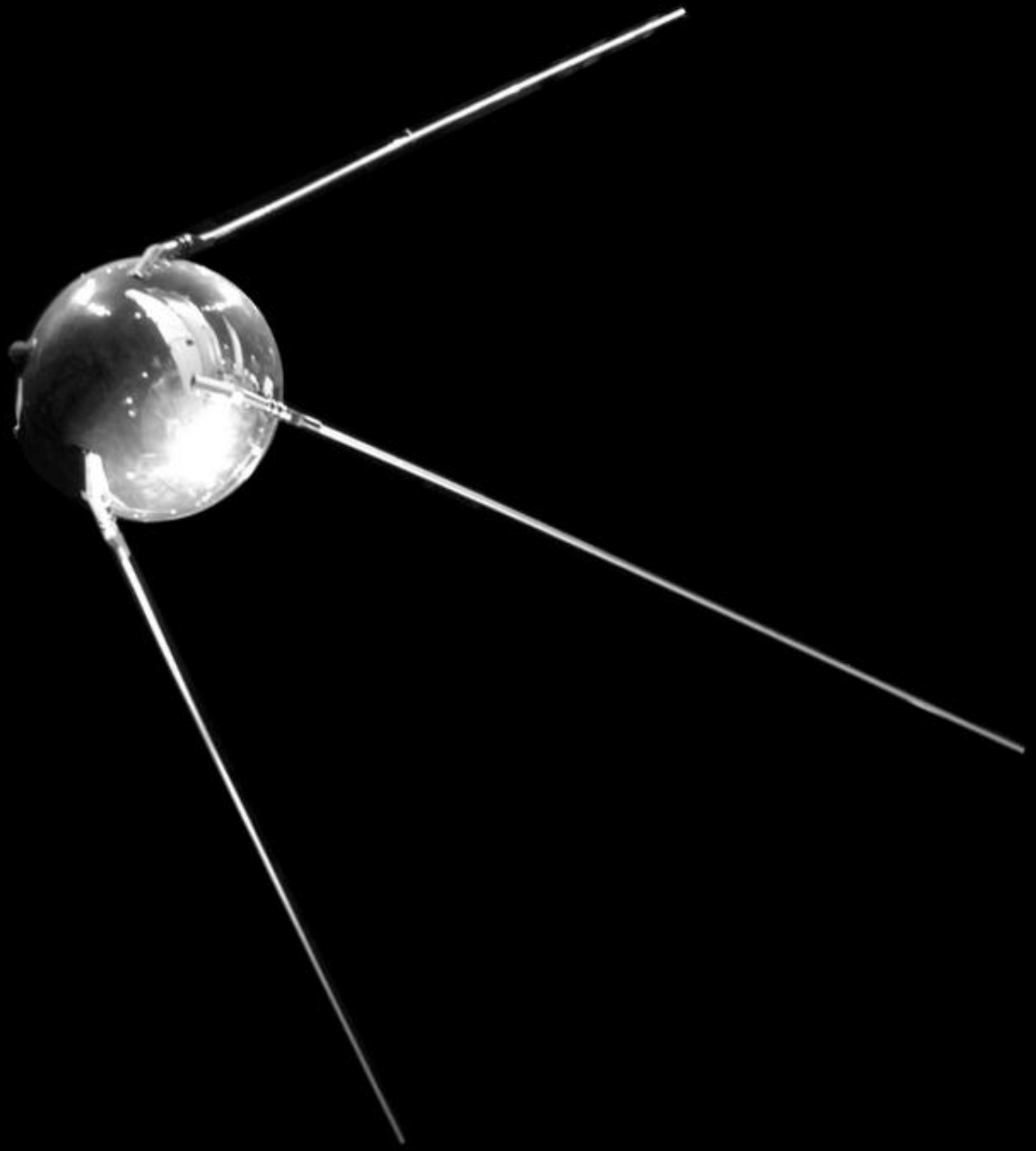


Lambdas all the way down.

How functional programming made our work more efficient and more fun.

christian.stamm@otto.de
@christian_stamm
#GOTOber
4.12.2015

OTTO



1950: first OTTO catalogue

- 28 pairs of shoes
- 14 pages
- pictures glued in

W. Otto

Art.568 orthopädischer Frauenschuh schwarz oder braun Boxcalf mit Wildledereinsatz, verlängerte Hinterkappen, Größe 3-8.

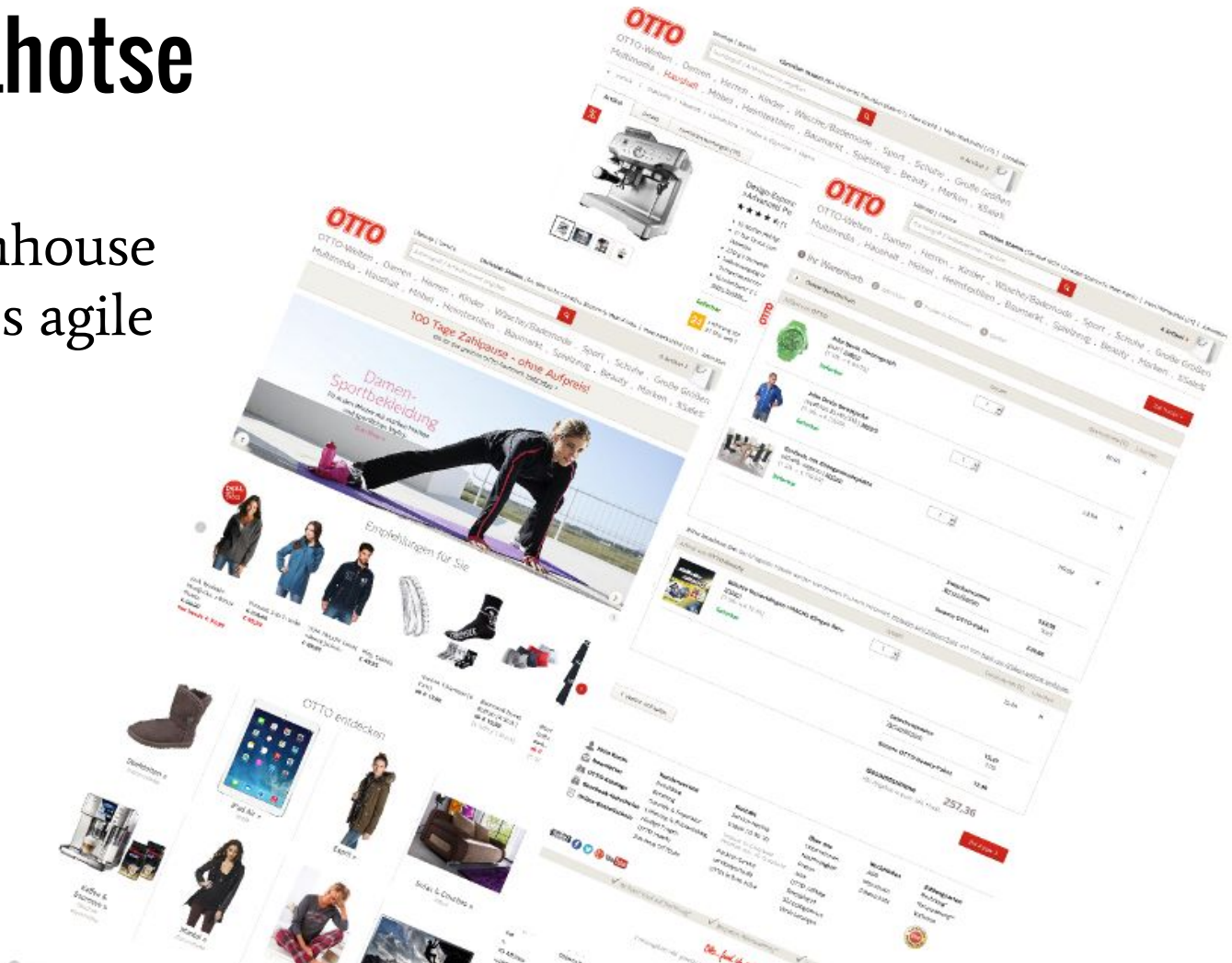
Art.565 Pumps schwarz Samtcalf mit Lackbesatz Absatzhöhe 5 cm, Größe 3-7.

1995: otto.de is online

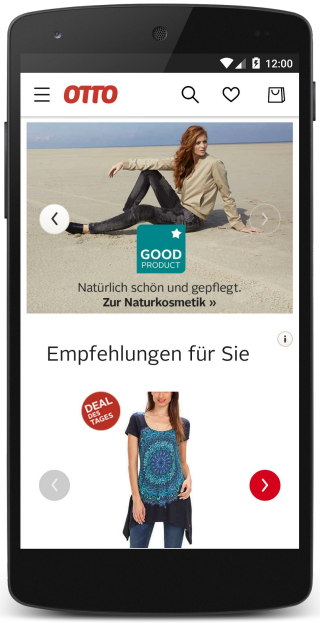
2001-2013: Intershop

since 2011: Lhotse

- developed inhouse
- otto becomes agile



The times they are a changing



~500.000 cm²

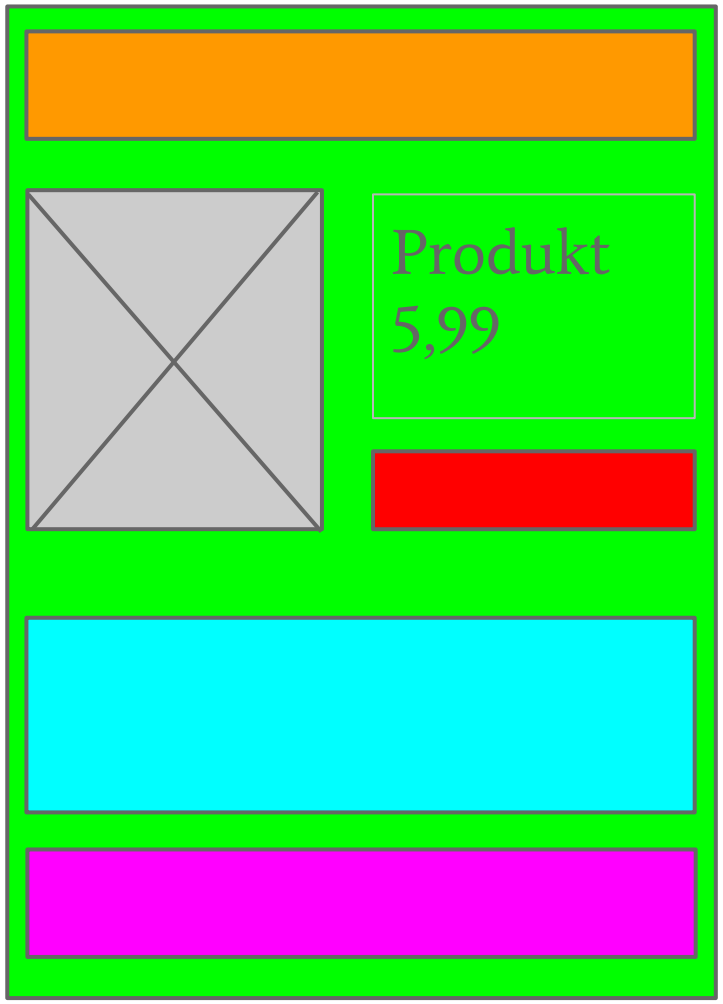
~70 cm²

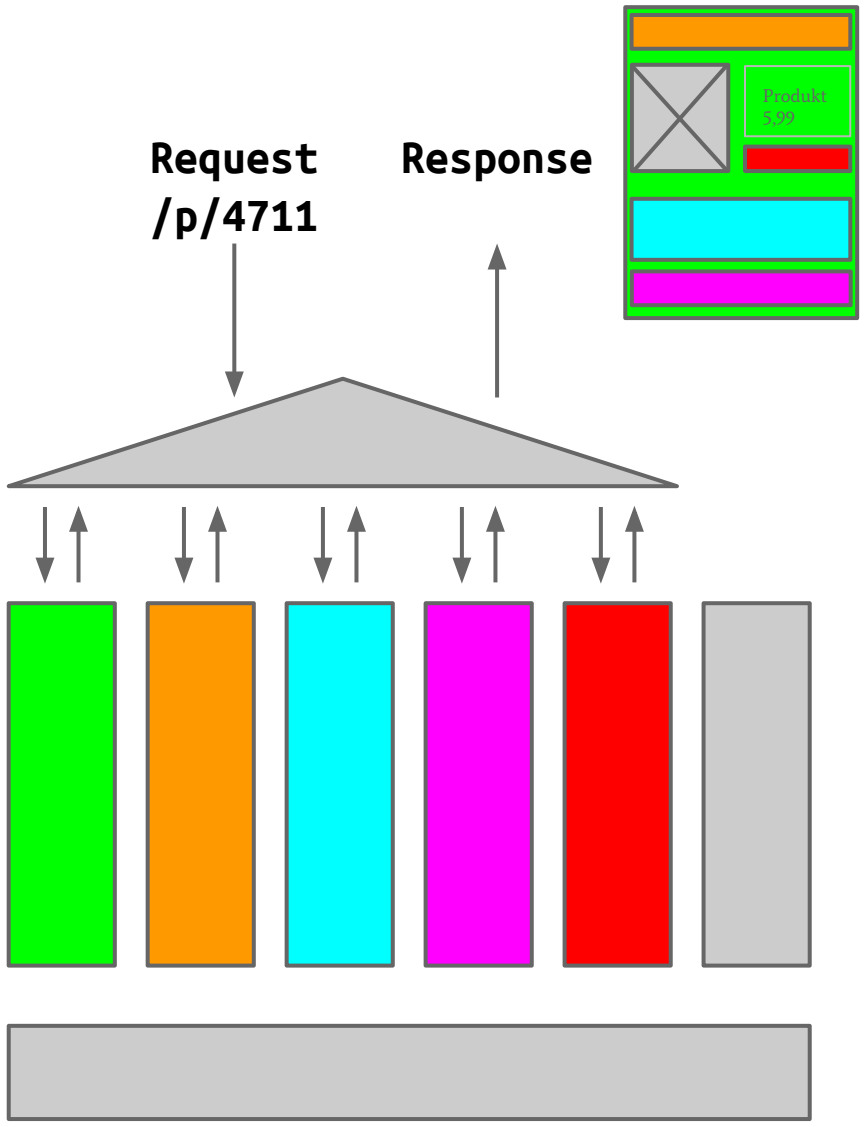
- ➔ The architecture of otto.de
- ➔ The Lambda Architecture: Facade/Batch/Streaming
- ➔ Static vs. dynamic typing: the epic struggle
- ➔ Functional programming in Infrastructure & CD
- ➔ How functional programming spreads at otto.de

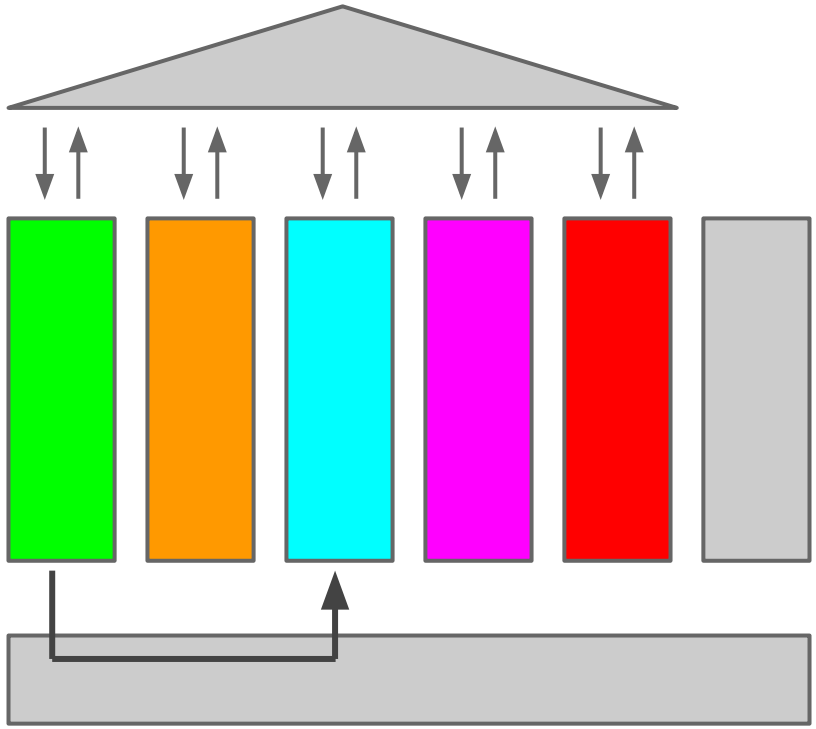
AZƏRBAYCAN
50 qəp
Vostok 1
2011

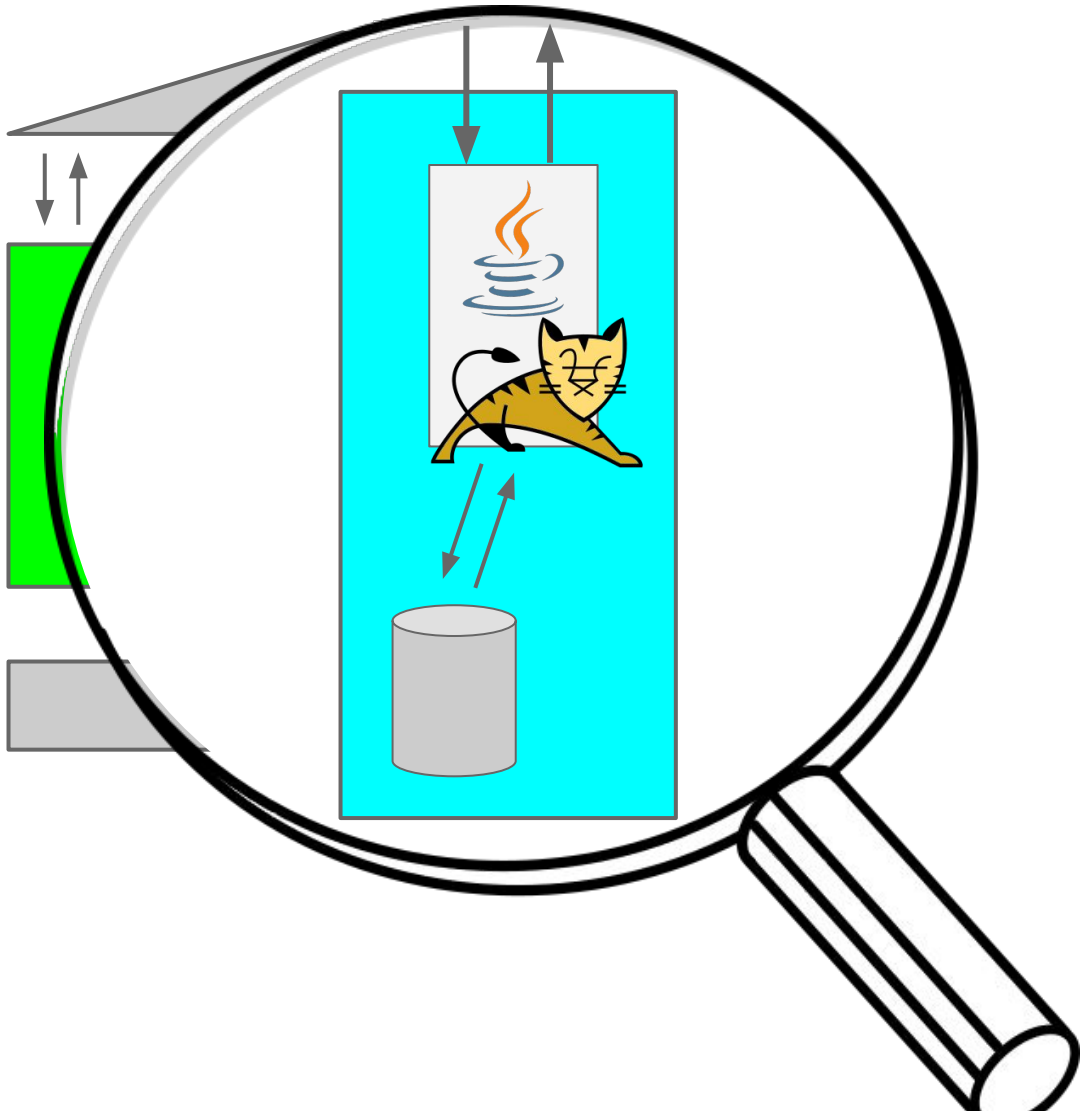
DER ERSTE WELTRAUMFLIEGER JURI ALEXEJEWITSCH GAGARIN / 12. 4. 1961
CCCP
10
DEUTSCHE DEMOKRATISCHE REPUBLIK
URBSCHAT

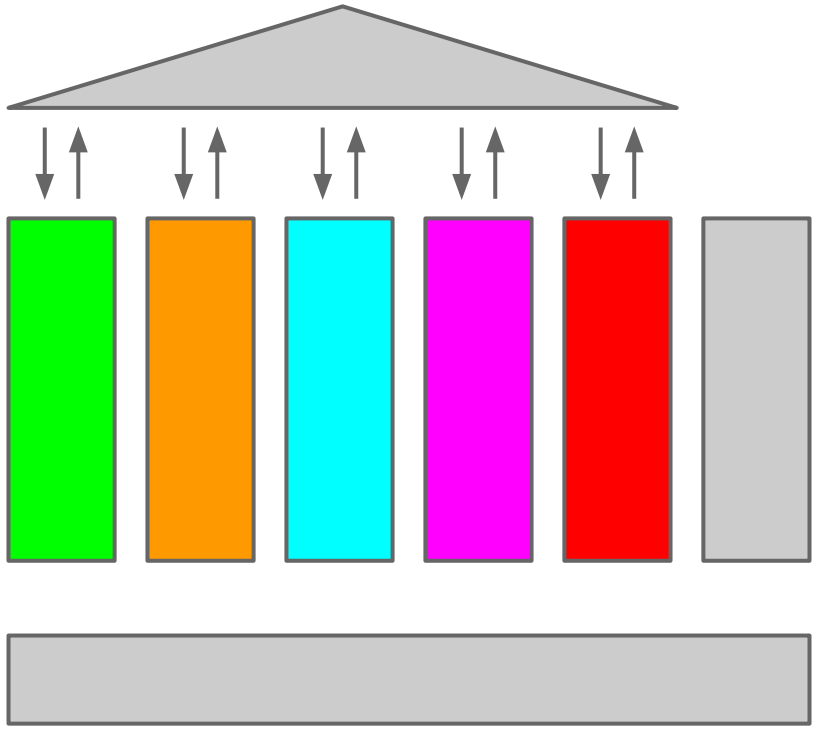
The architecture of otto.de


Produkt
5,99


Request
/p/4711
Response
Produkt
5,99

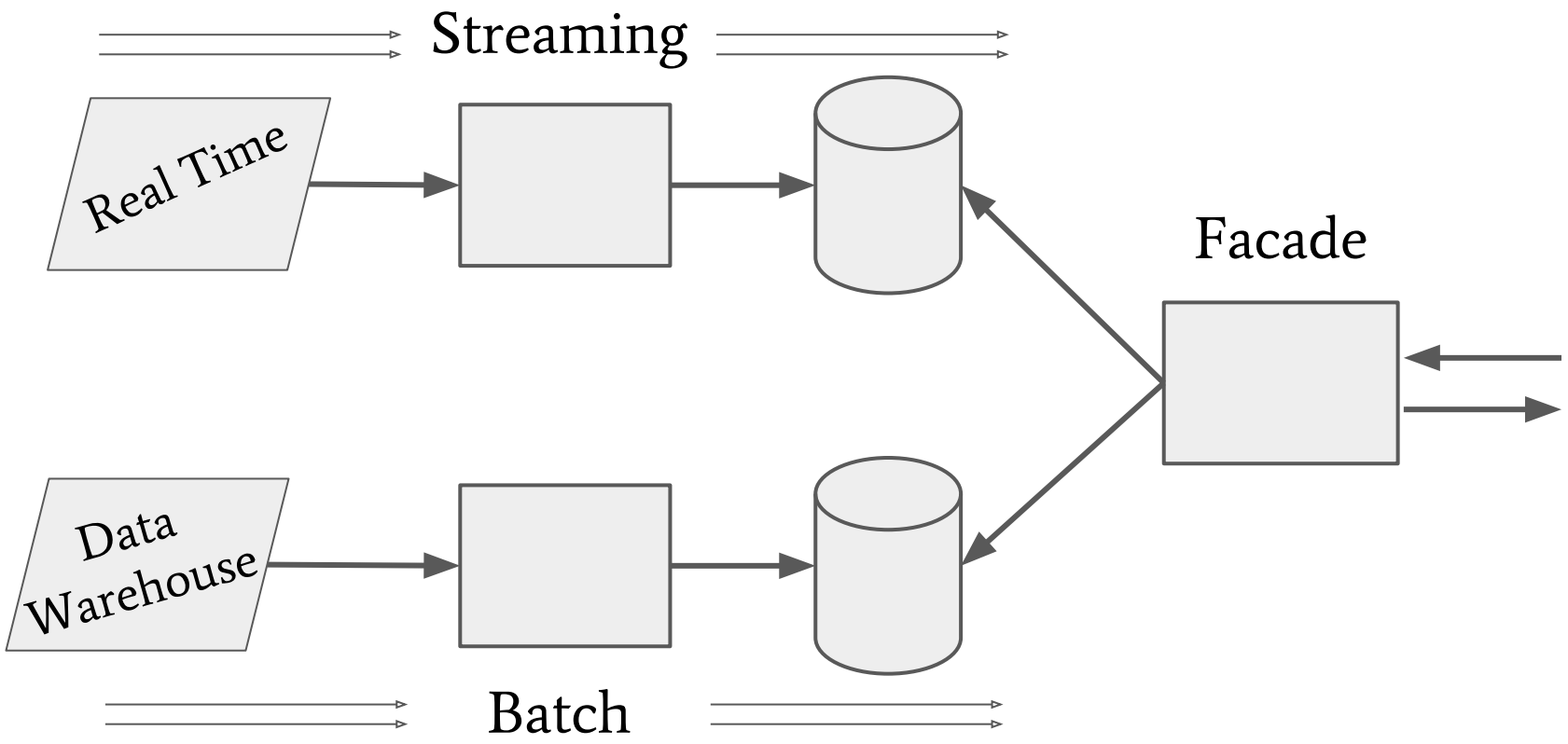


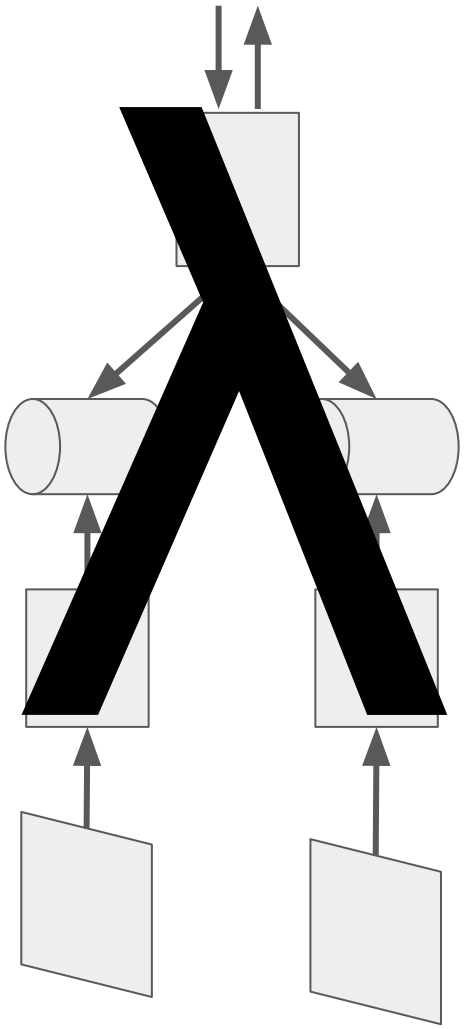


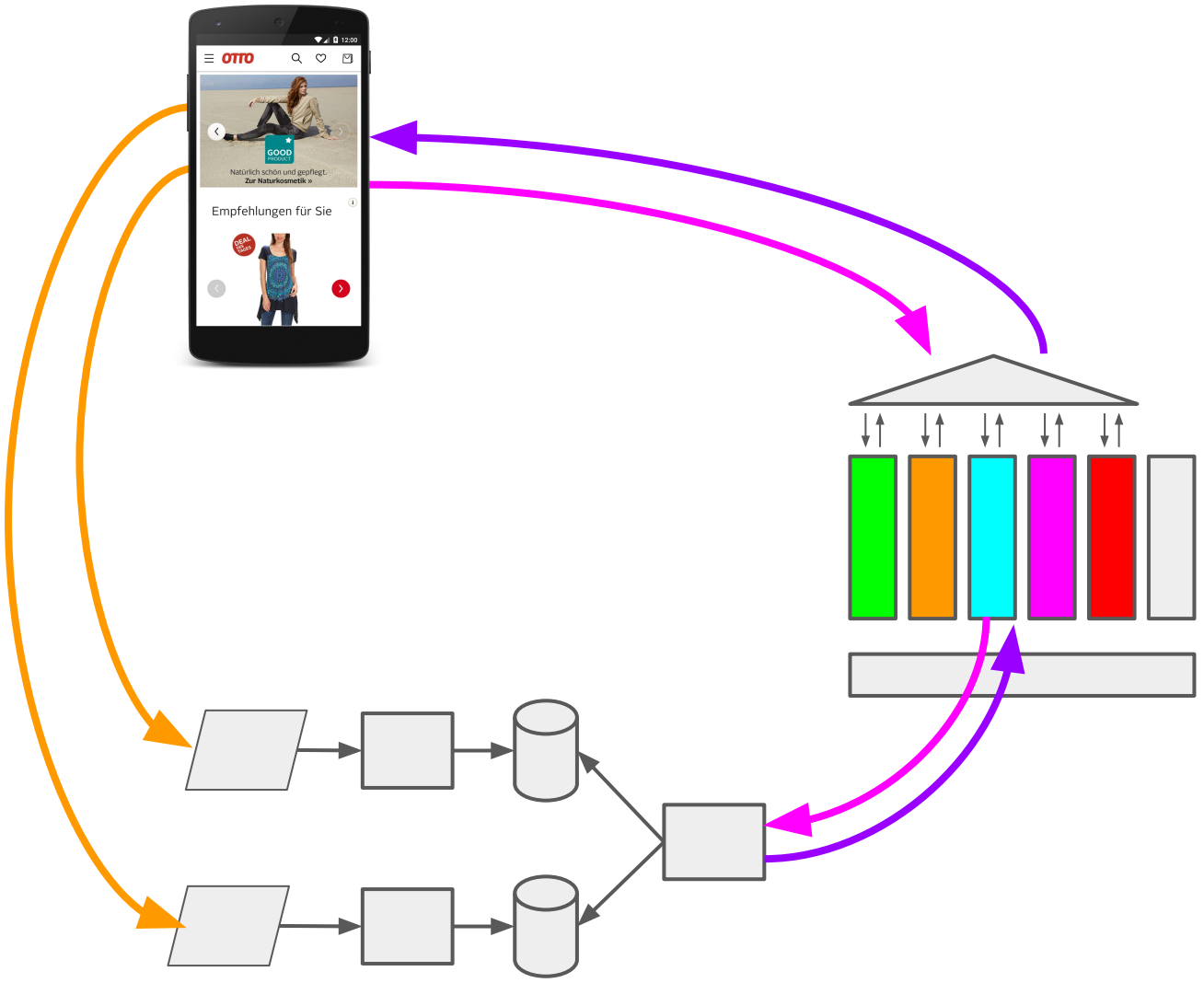


СОЮЗ

The Lambda Architecture

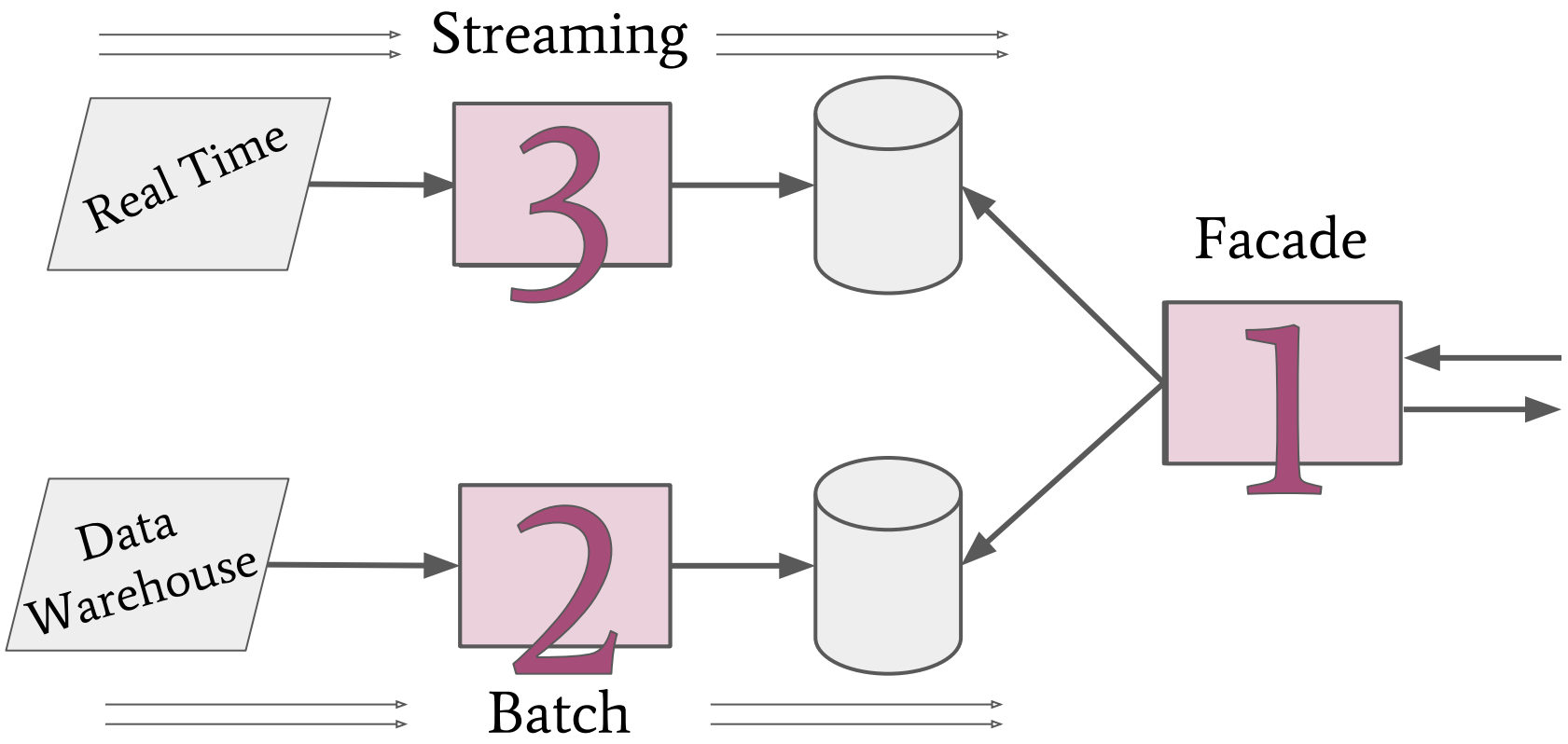

Streaming
Real Time
Facade
Data Warehouse
Batch

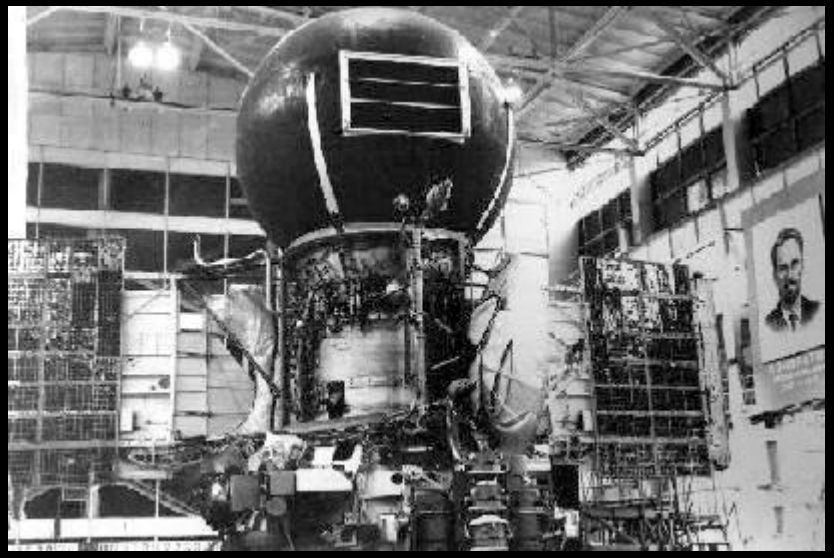



OTTO
GOOD
Natürlich schön und gepflegt.
Zur Naturkosmetik »
Empfehlungen für Sie

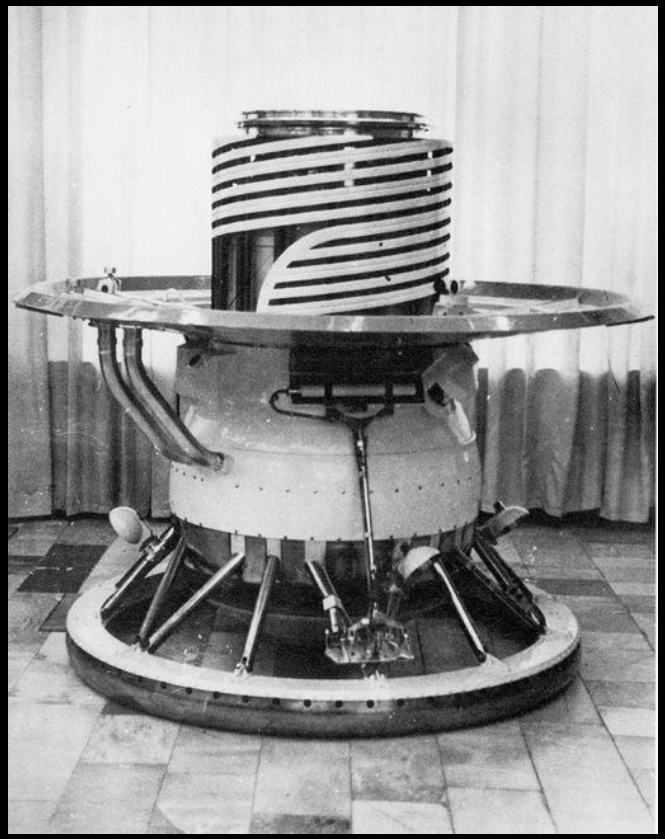
8 weeks

for a prototype

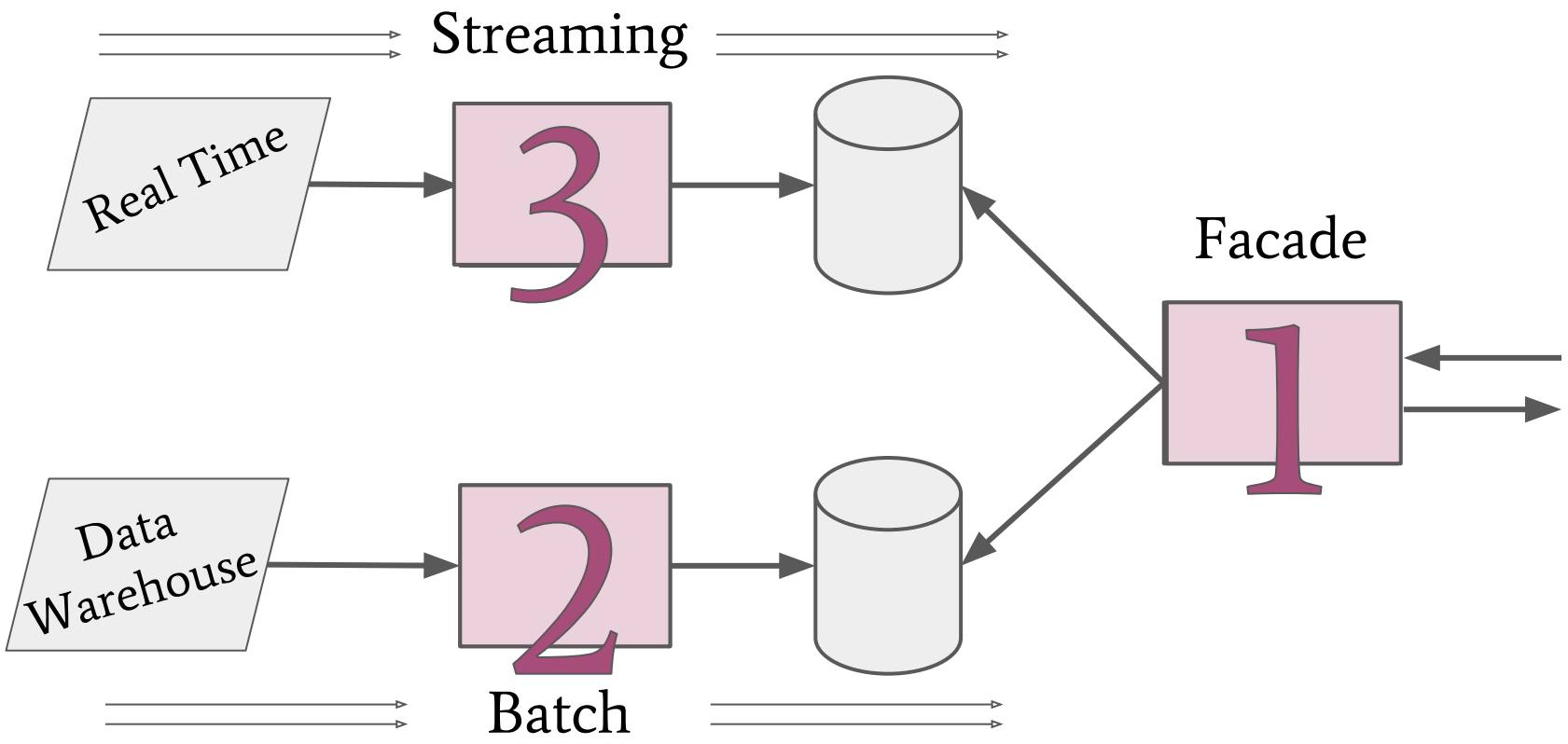

Streaming
Real Time
3
Facade
1
Data
Warehouse
2
Batch



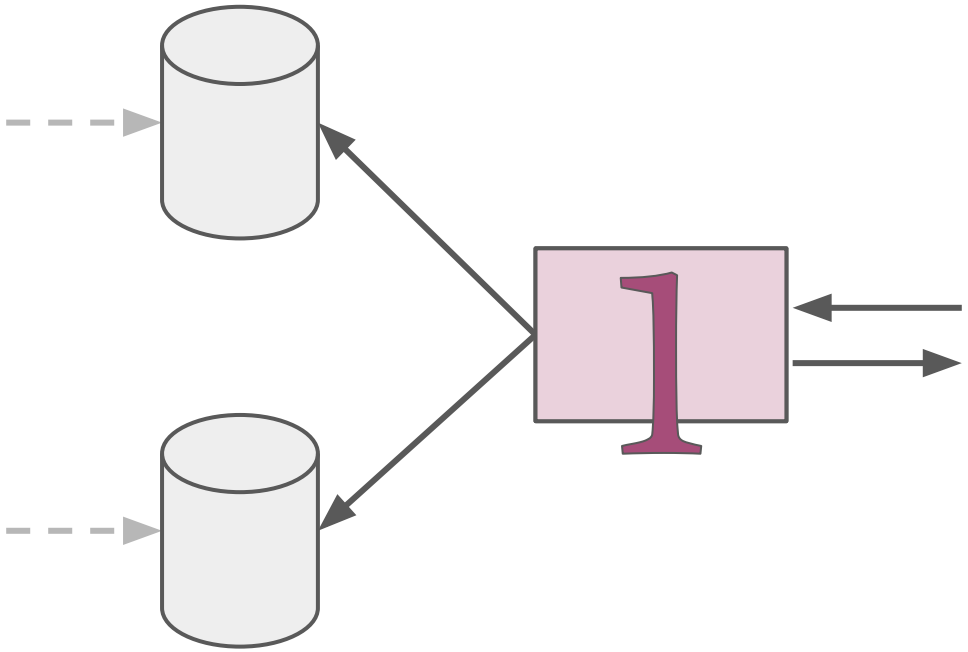
ВЕНЕРА
ЗЕМЛЯ
СССР
12-II-1961
СССР
ПОЧТА



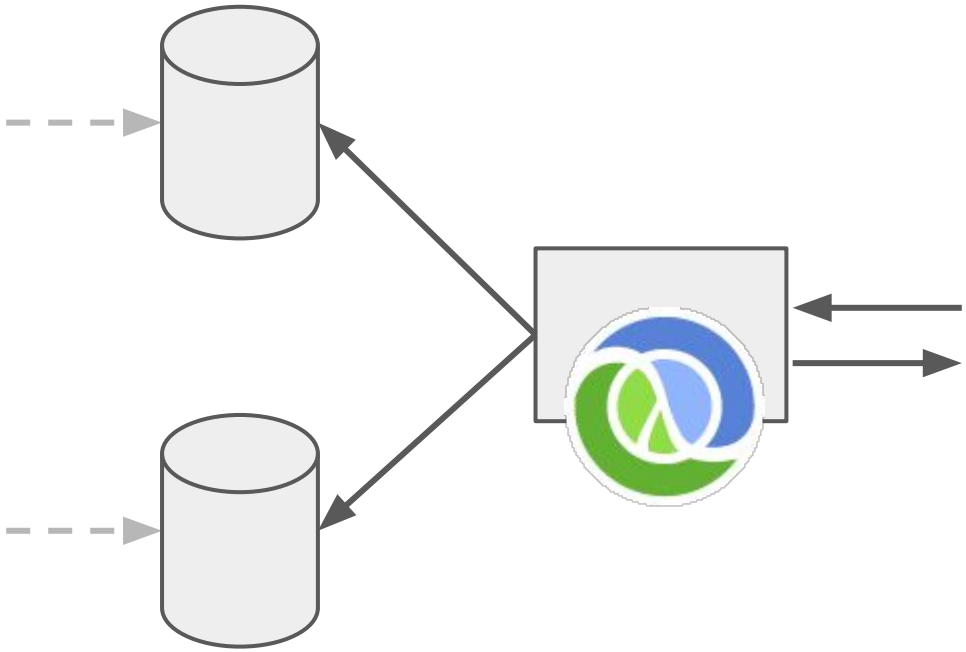
ВЕНЕРА-9 22.10.1975 ОБРАБОТКА ИППИ АН СССР 28.2.1976
ВЕНЕРА-10 25.10.1975 ОБРАБОТКА ИППИ АН СССР 28.2.1976


Streaming
Real Time
3
Facade
1
Data
Warehouse
2
Batch

The Facade


1

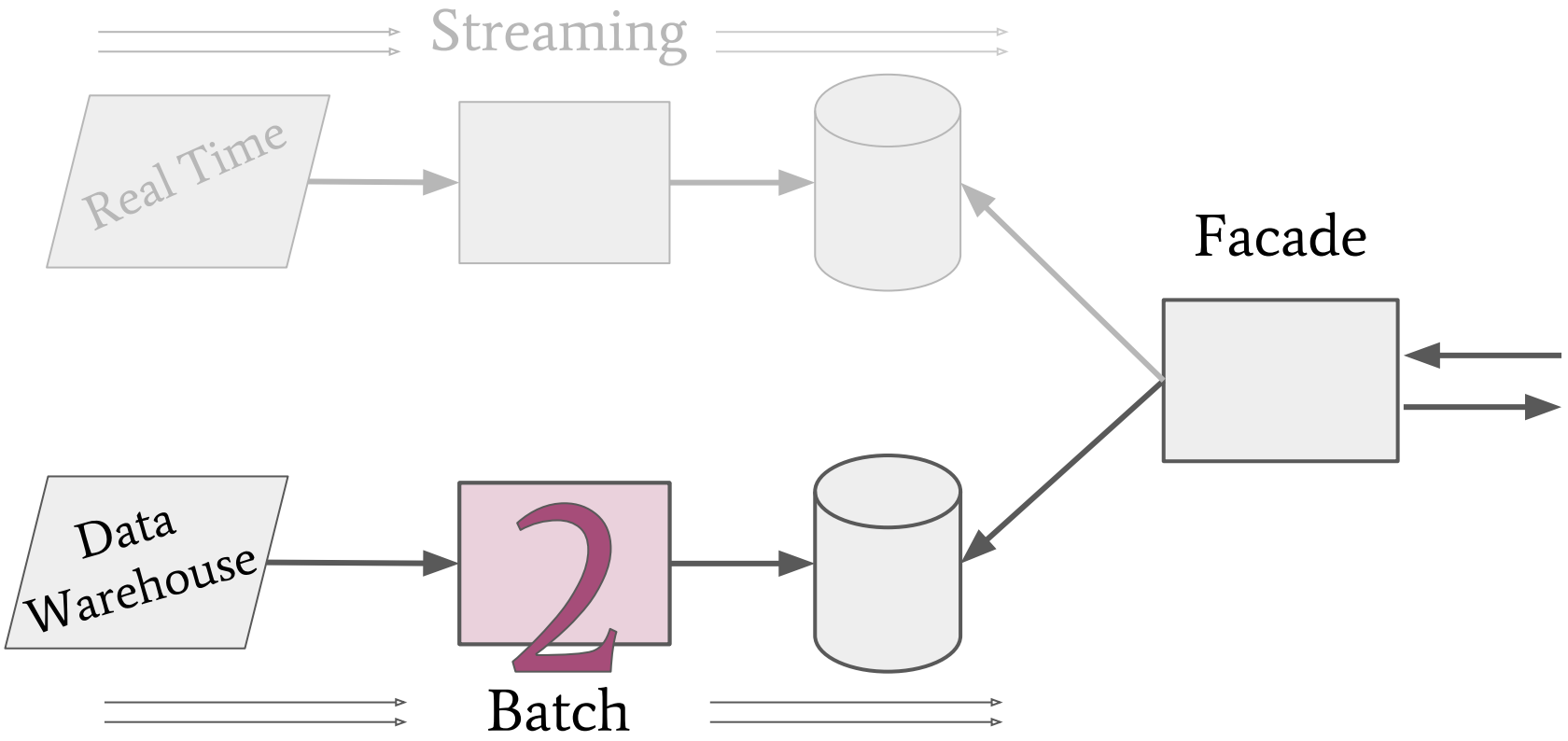
Clojure!



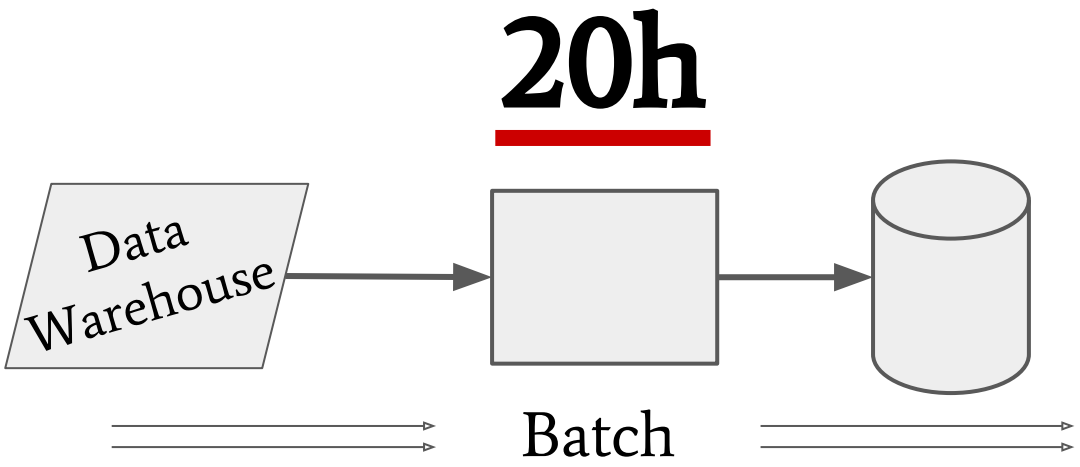
СССР
ПОЧТА 1971
„ЛУНОХОД-1"
К 16

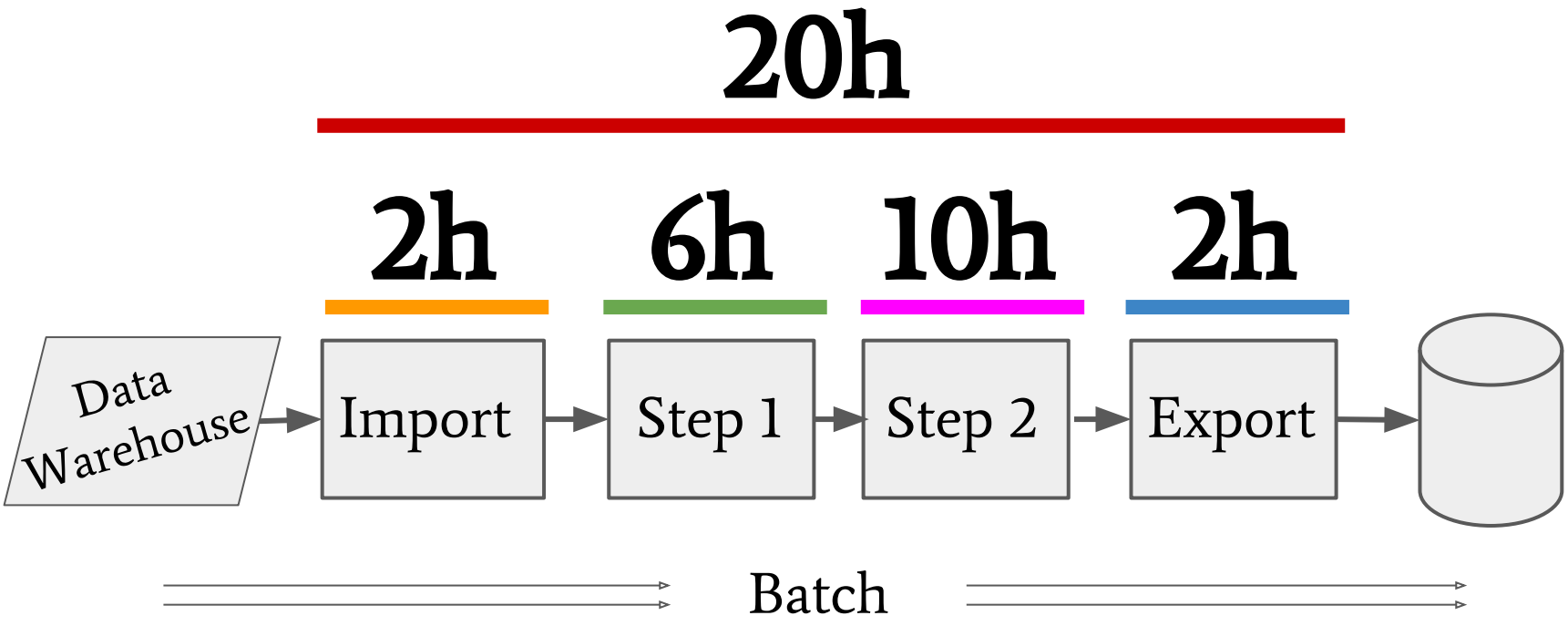
DEUTSCHE DEMOKRATISCHE REPUBLIK
20
1. MOBILES MONDLABOR · LUNOCHOD 1 ·

batch processing

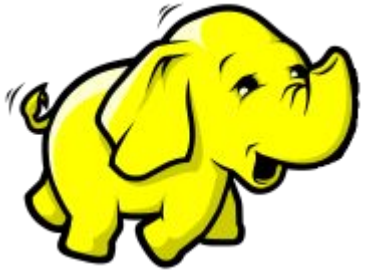

Streaming
Real Time
Facade
Data
Warehouse
2
Batch

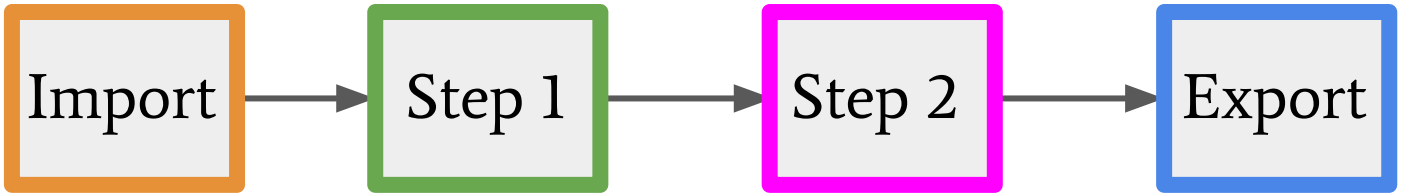
bash & python scripts


20h
Data Warehouse
Batch


20h
2h
6h
10h
2h
Data Warehouse
Import
Step 1
Step 2
Export
Batch

Immutable Filesystem

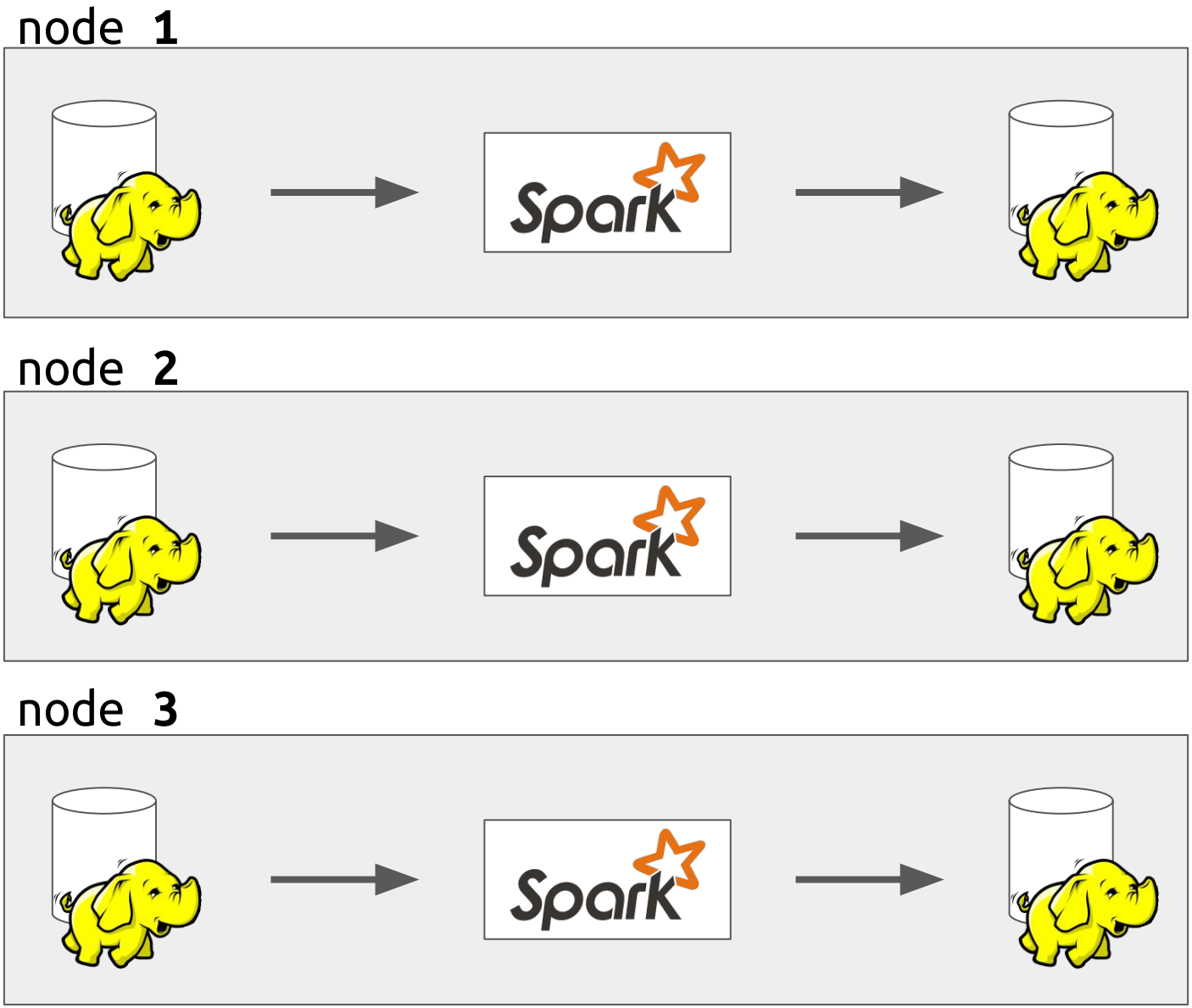




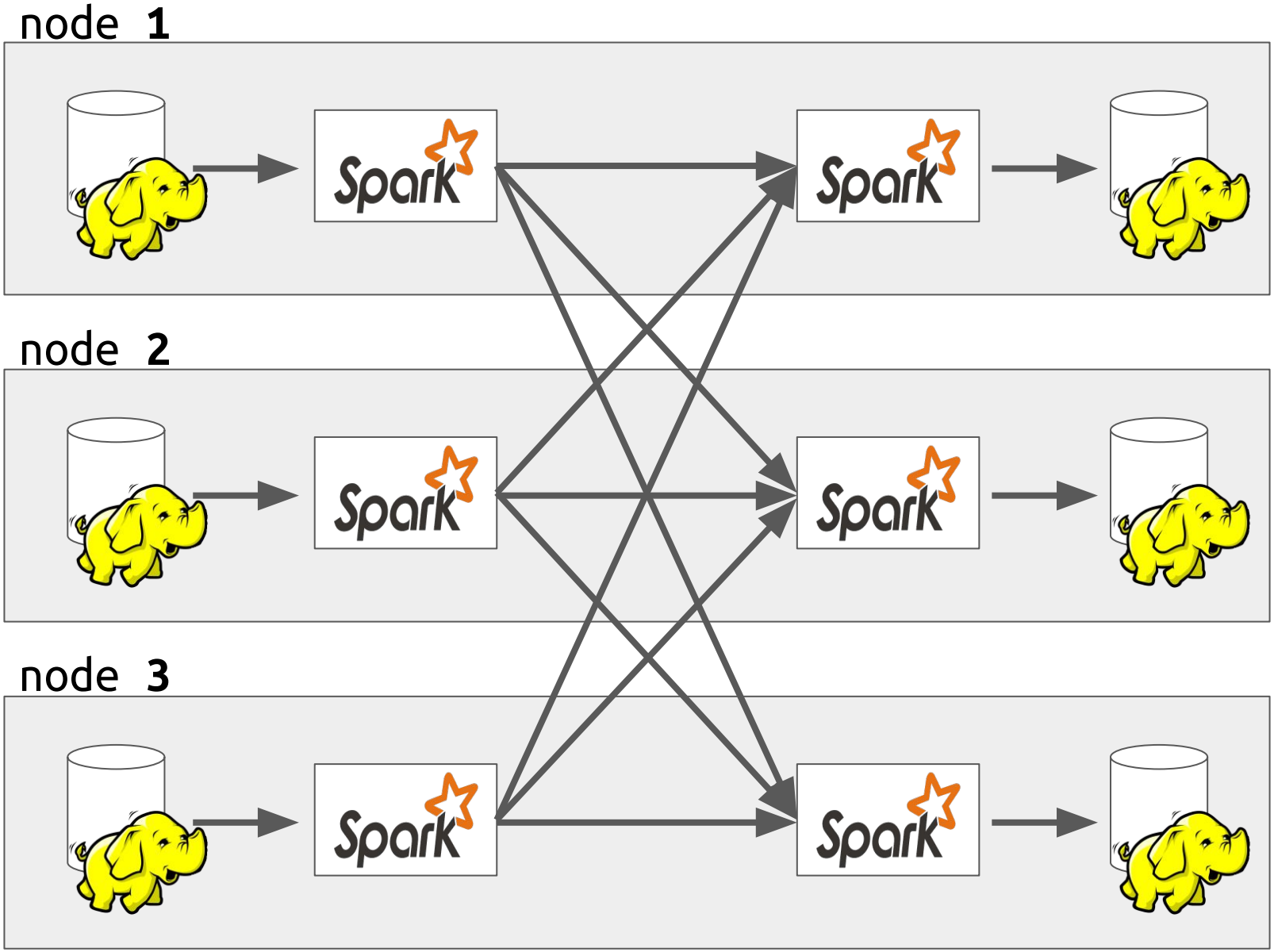

```
.
└── 000
    ├── input
    │   ├── data.csv
    │   └── _SUCCESS
    └── output
        ├── step_1
        │   ├── data.csv
        │   └── _SUCCESS
        ├── step_2
        │   ├── data.csv
        │   └── _SUCCESS
        └── export
            └── _SUCCESS
```

Apache Spark

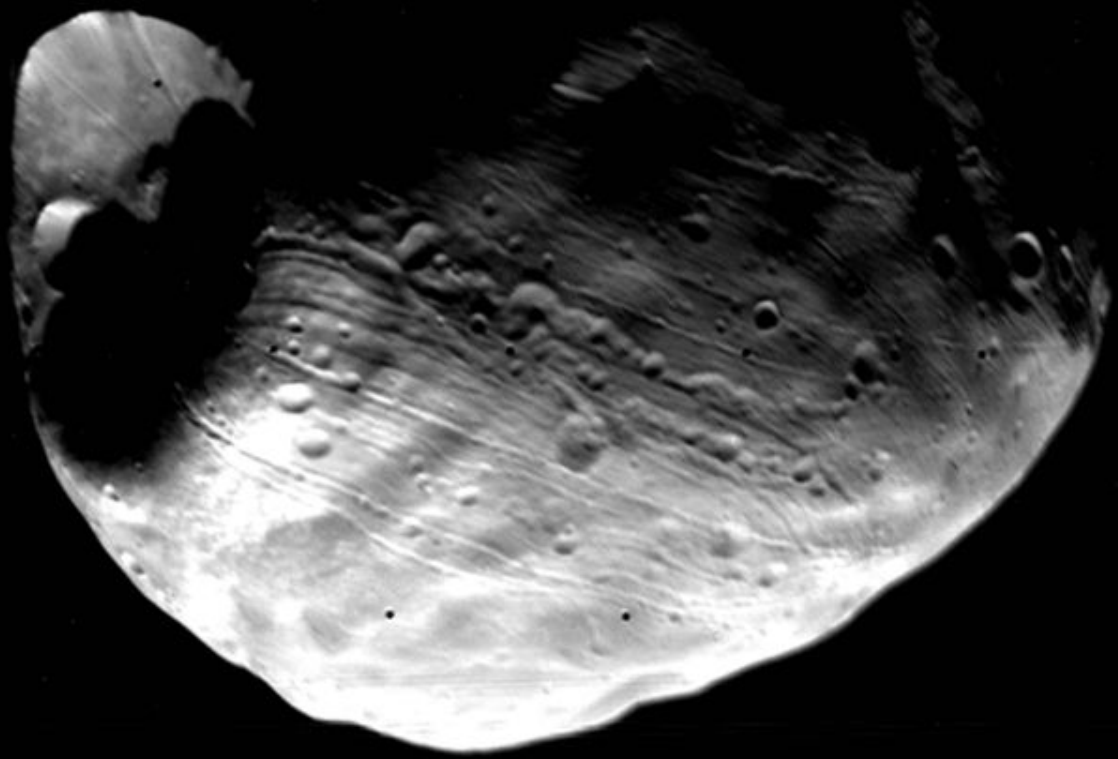
```
sc.textFile(inputPath)
   .map(doParse)
   .filter(onlyValid)
   .map(addExtraData)
   .map(calculate)
   .saveAsTextFile(outputPath)
```


node 1
Spark
node 2
Spark
node 3
Spark

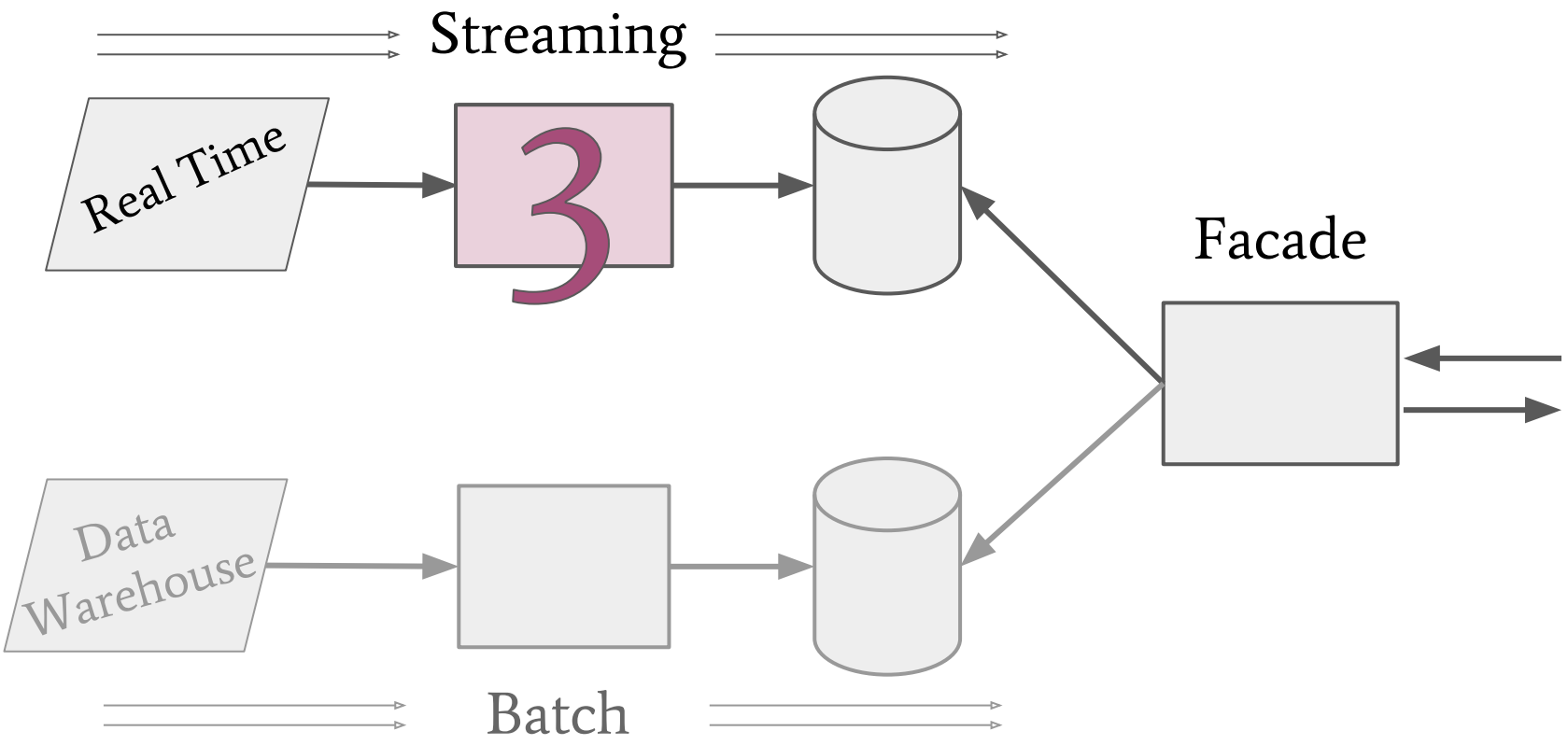
```
sc.textFile(inputPath)
   .map(doParse)
   .filter(onlyValid)
   .map(addExtraData)
   .groupBy(grouping)
   .map(calculate)
   .saveAsTextFile(outputPath)
```


node 1
Spark
Spark
node 2
Spark
Spark
node 3
Spark
Spark

ПОЧТА СССР 1988
10к
Международный
космический
проект
«ФОБОС»

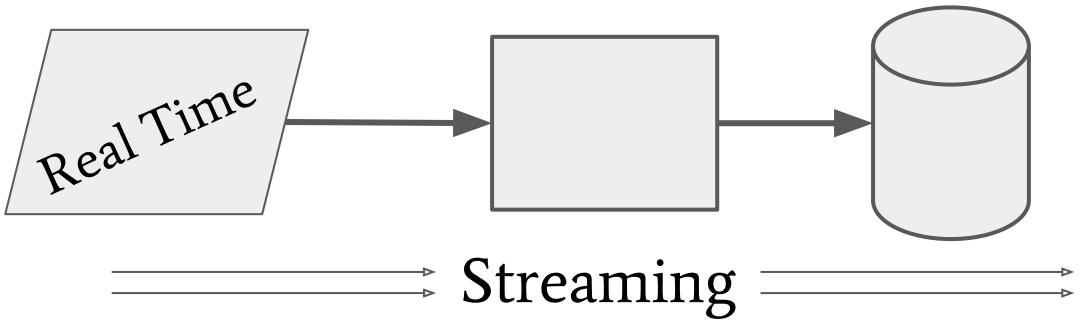


stream processing


Streaming
Real Time
3
Facade
Data
Warehouse
Batch

spark streaming

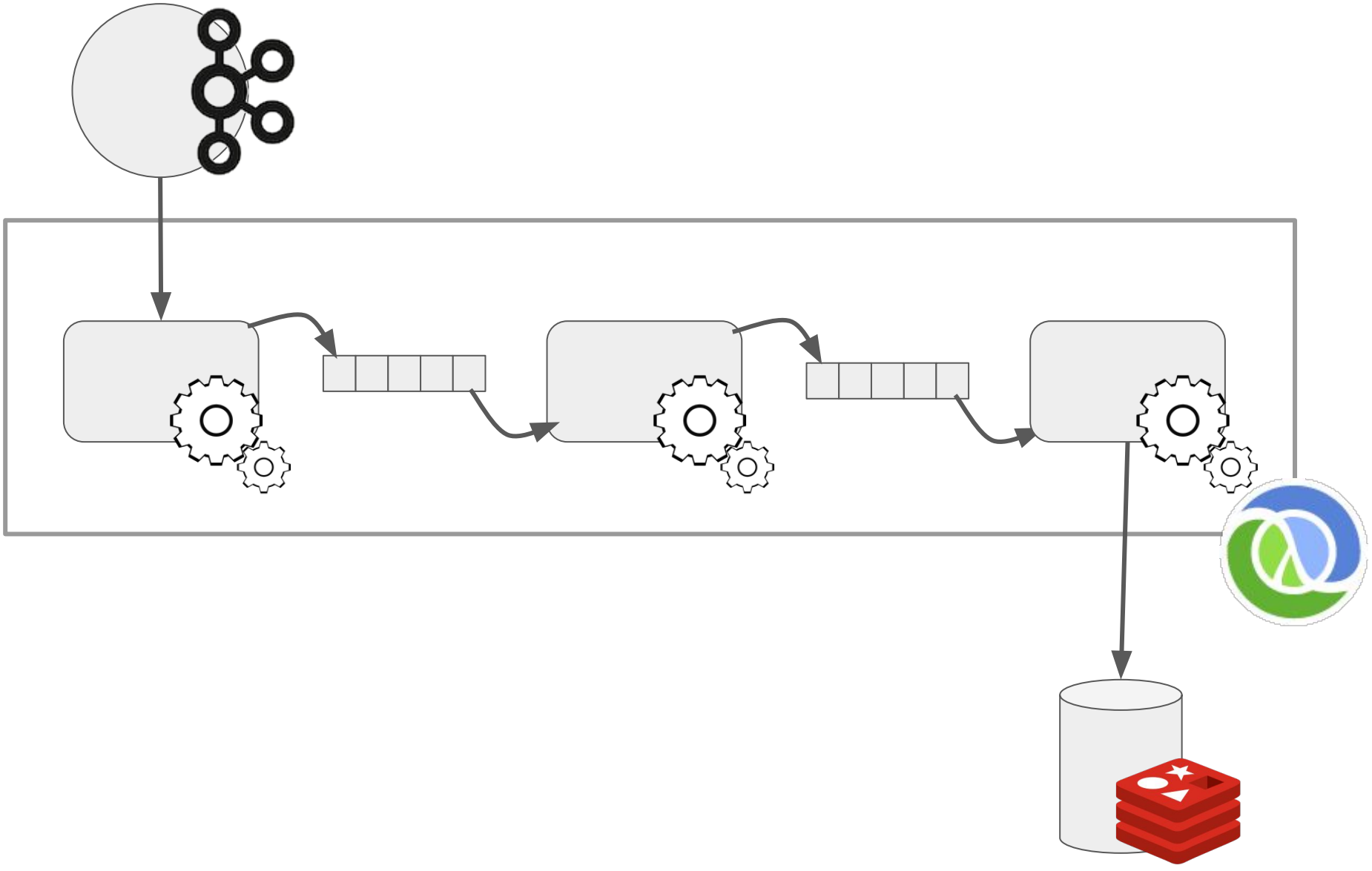
ls

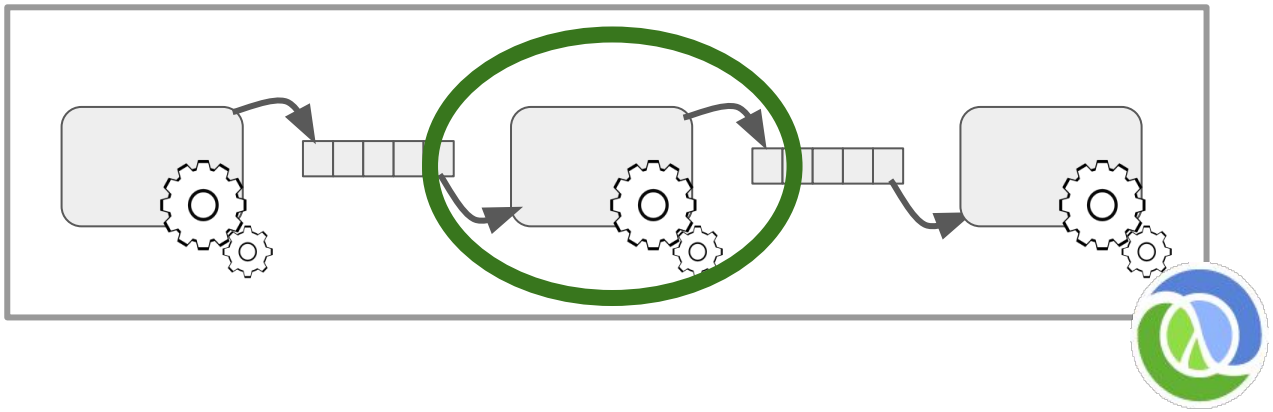


core.async

```
data.map(doParse)
    .filter(onlyValid)
    .map(addExtraData)
    .map(calculate)
```

```
(map calculate
  (map extra-data
    (filter only-valid
      (map parse data))))
```

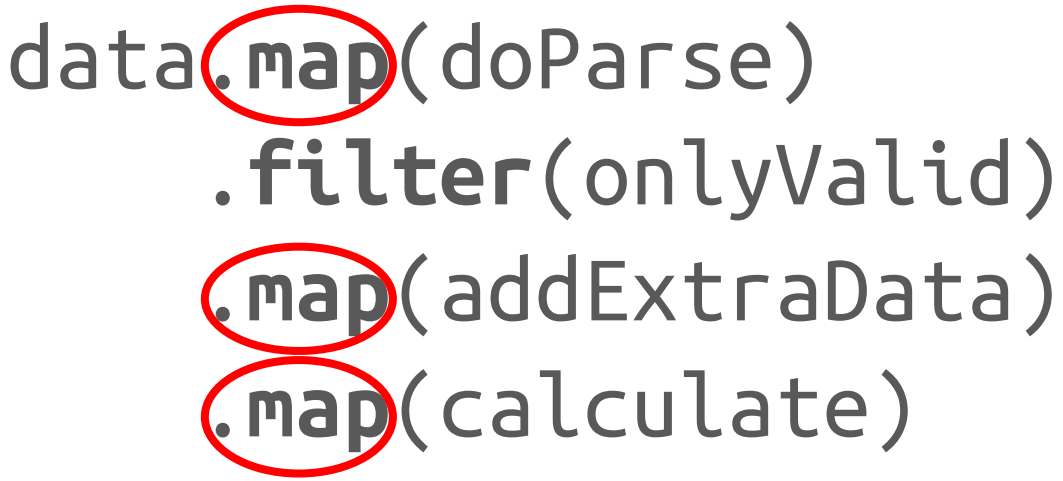




```
(async/pipeline
  parallelism
  input-chan
  (filter not-nil?
    (map process))
  output-chan)
```

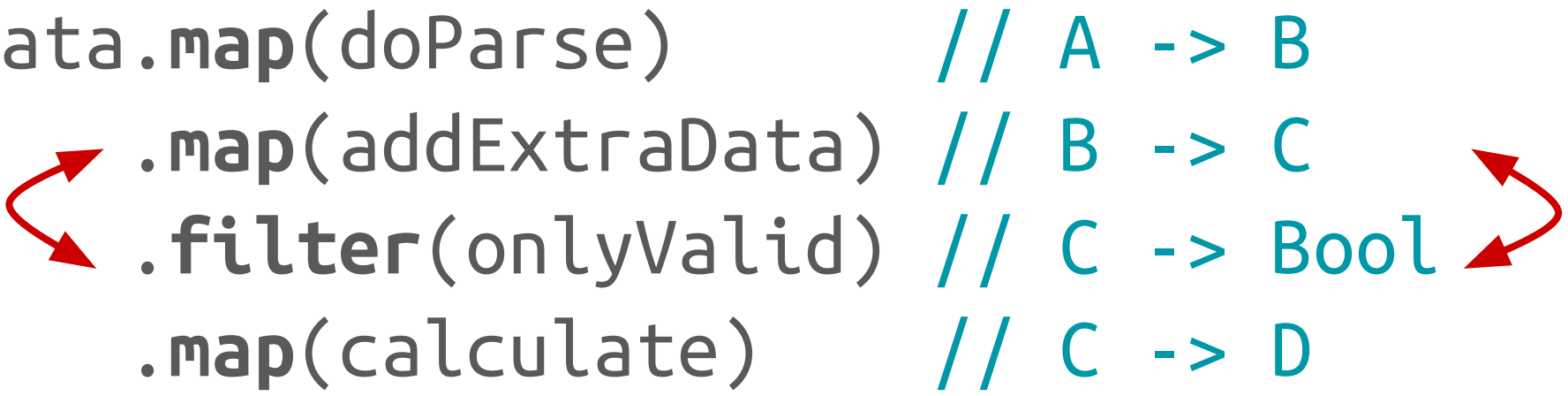
```clojure
(defn process [event]
  (some-> event
    (parse)
    (filter)
    (add-extra-data)
    (calculate)))
```

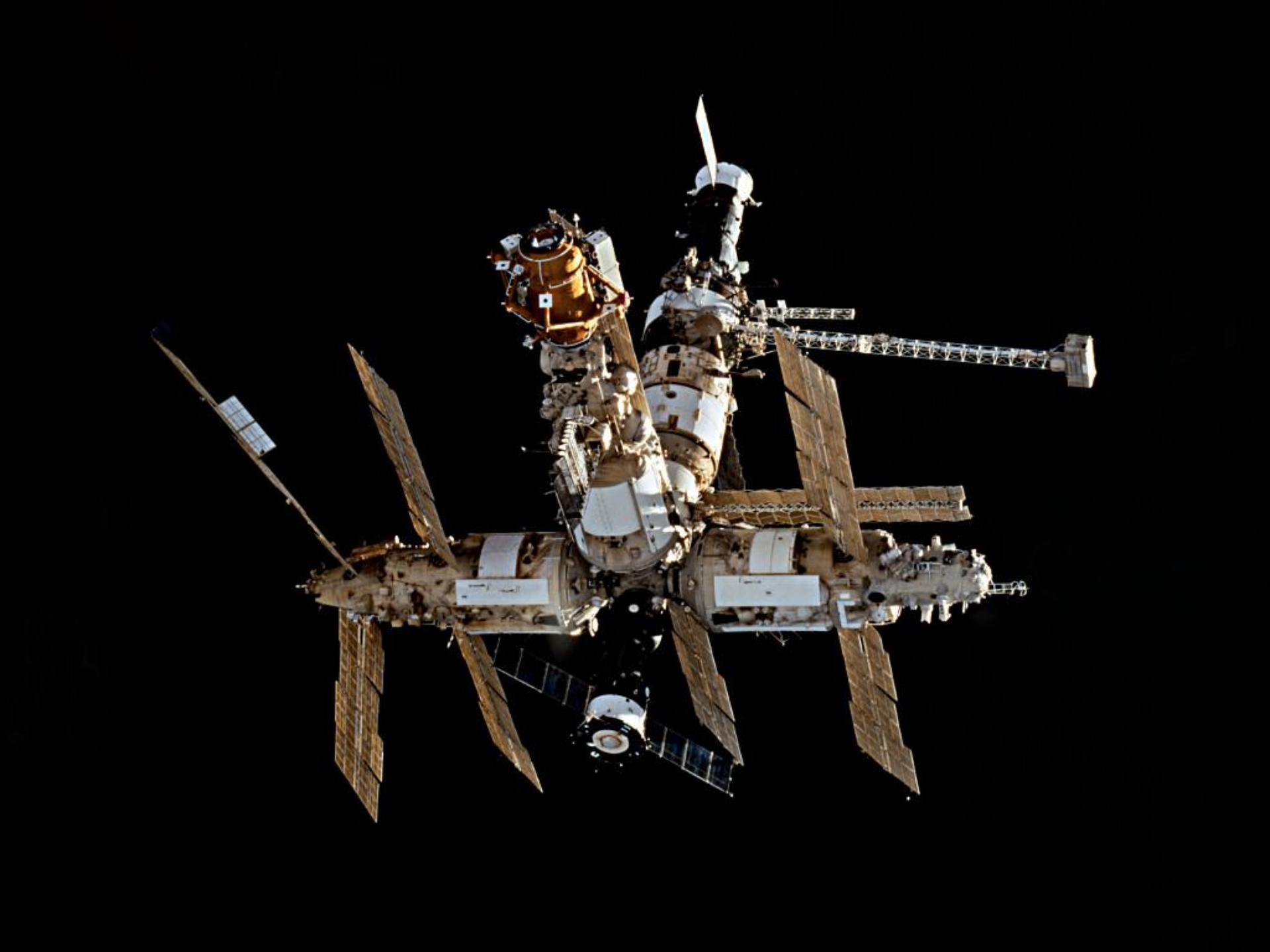
static vs. dynamic typing: the epic struggle

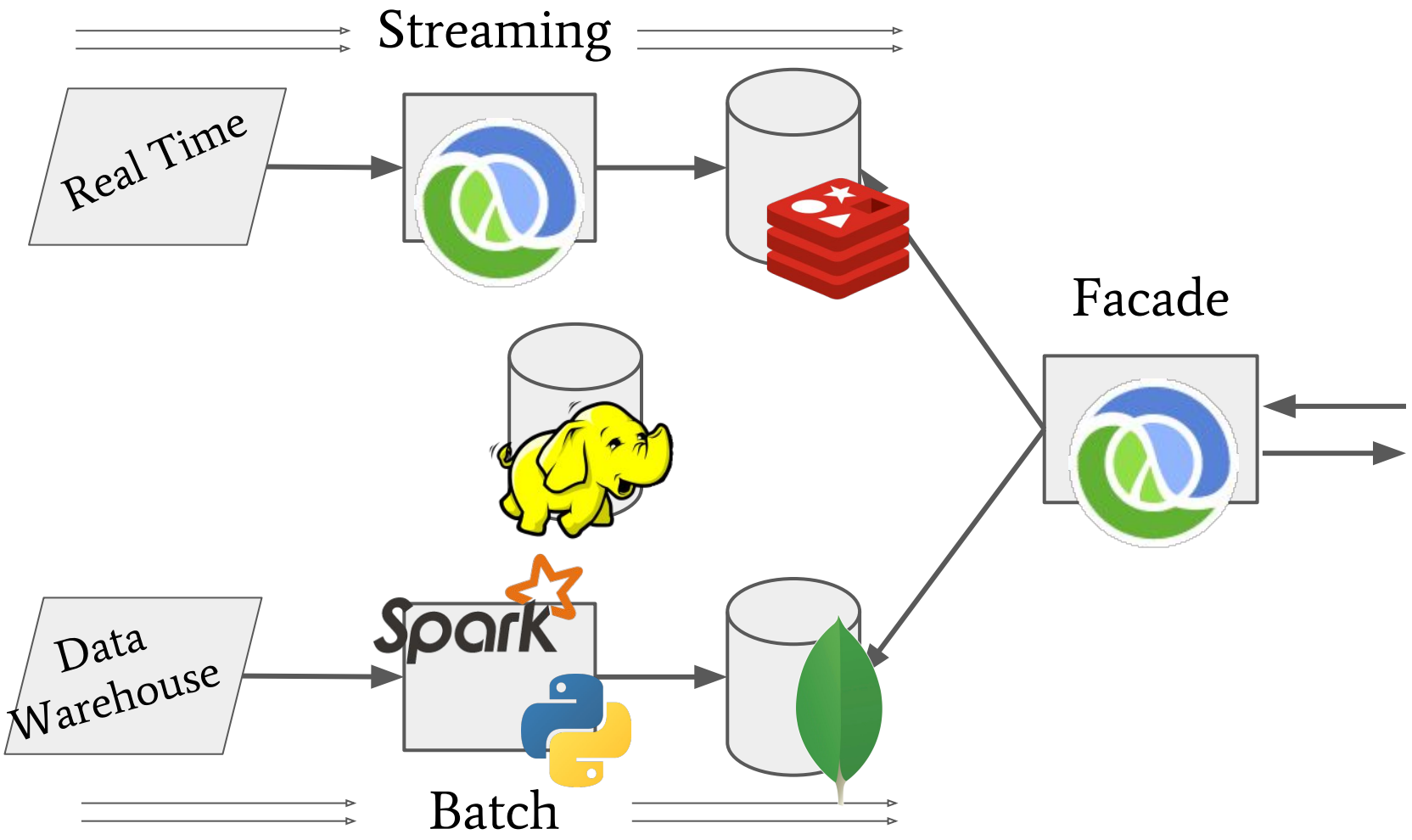

data.map(doParse)
.filter(onlyValid)
.map(addExtraData)
.map(calculate)

```
def addExtraData(x : ParsedType):EnrichedType = {
    val extra: ExtraType = fetchForId(x.id)
    EnrichedType(x, extra)
}
```

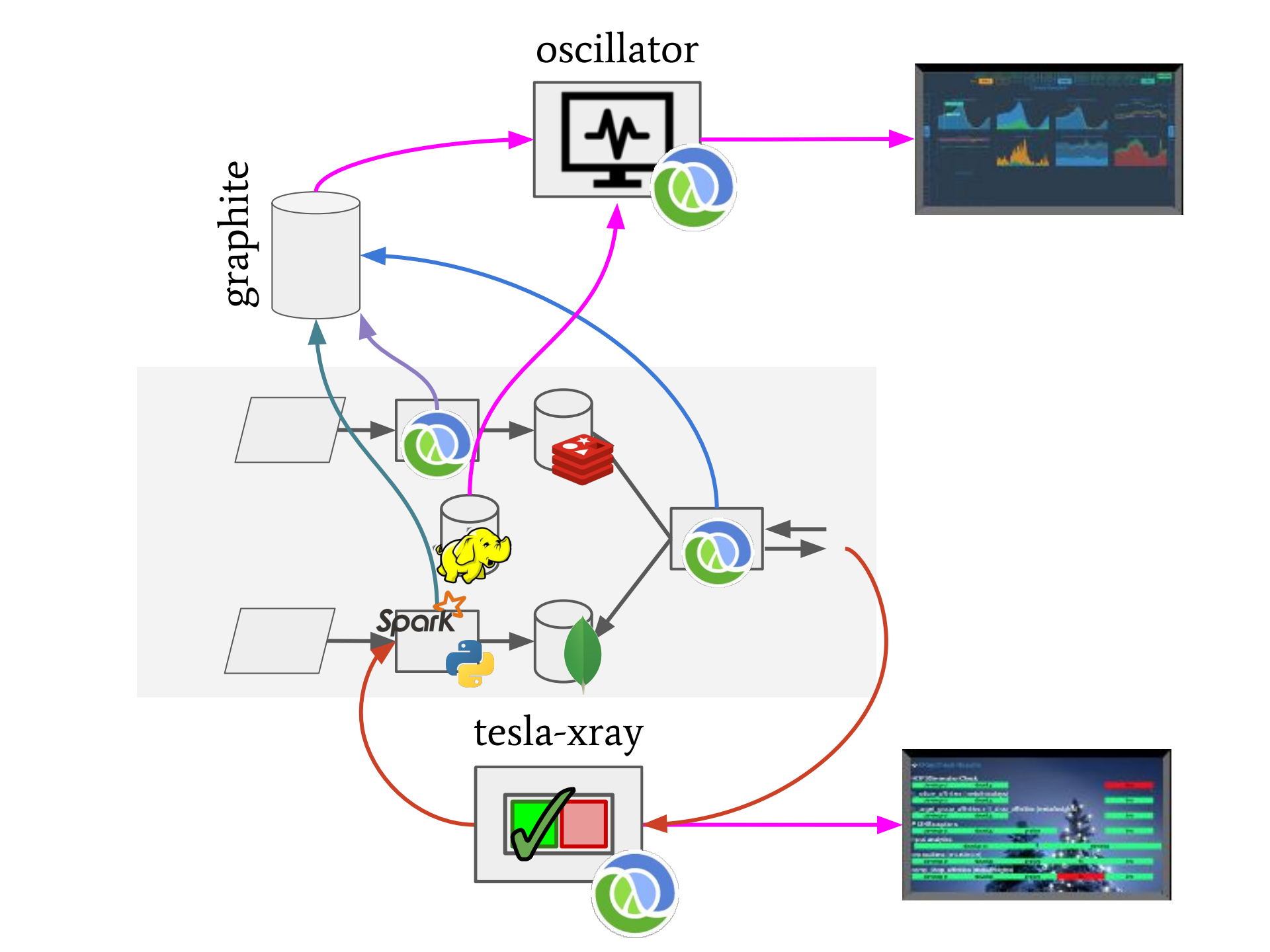
```clojure
(defn add-extra-data [event]
  (let [id (:id event)
        extra (fetch-for-id id)]
    (assoc event :extra extra)))
```


data.map(doParse) // A -> B
.map(addExtraData) // B -> C
.filter(onlyValid) // C -> Bool
.map(calculate) // C -> D




Streaming
Real Time
Facade
Data
Warehouse
Spark
Batch

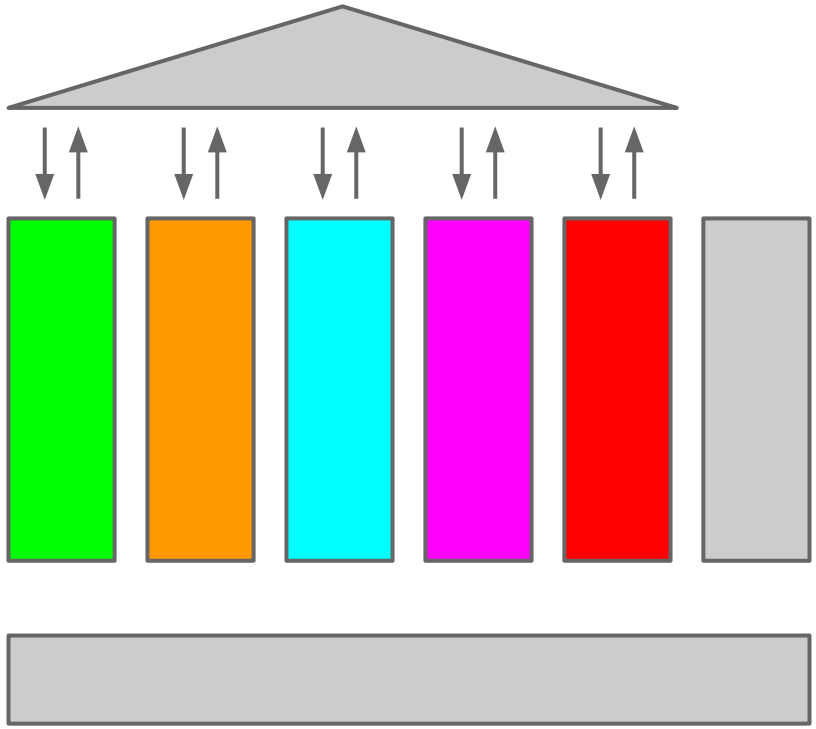
Functional programming in infrastructure & CD

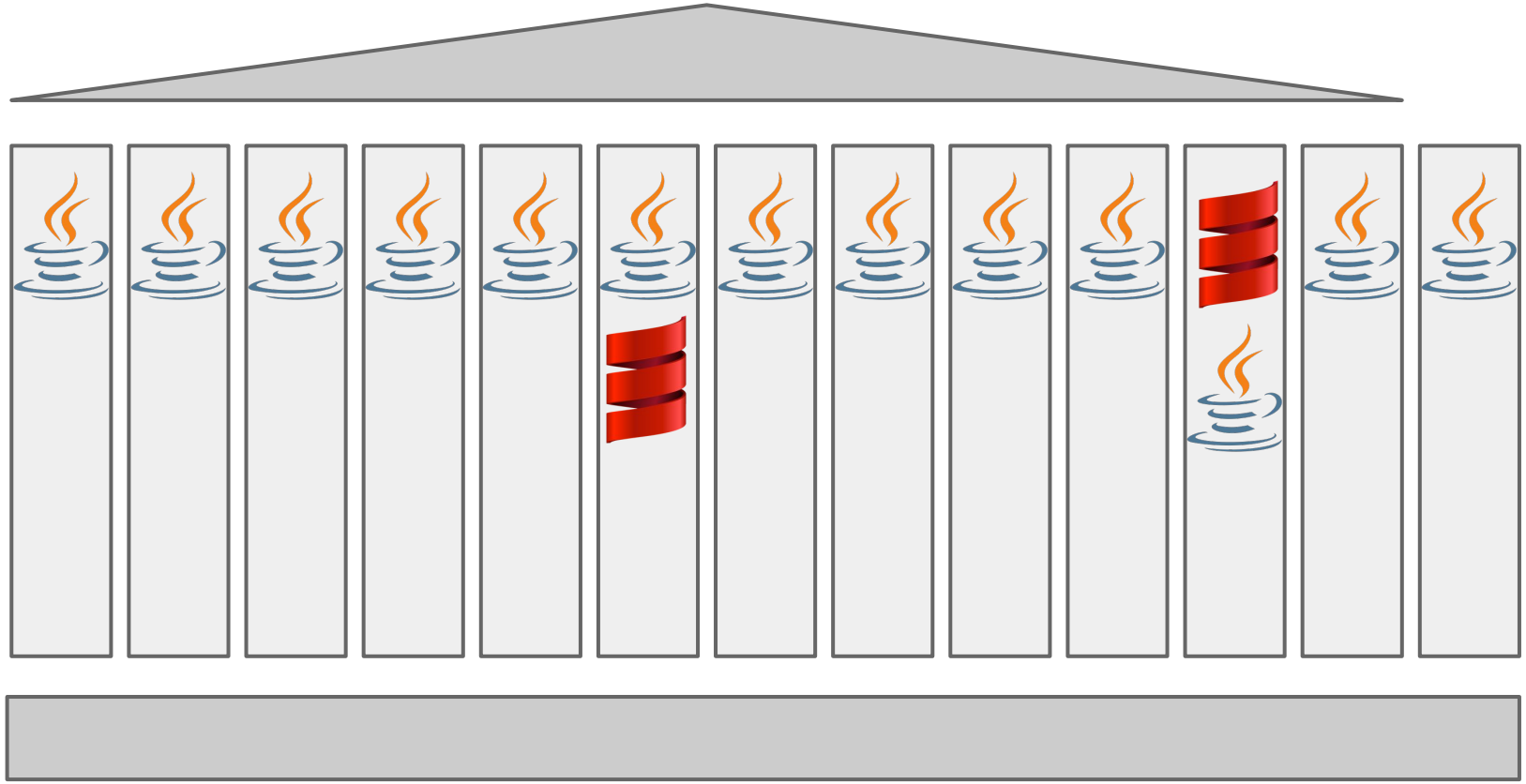

oscillator
graphite
Spark
tesla-xray

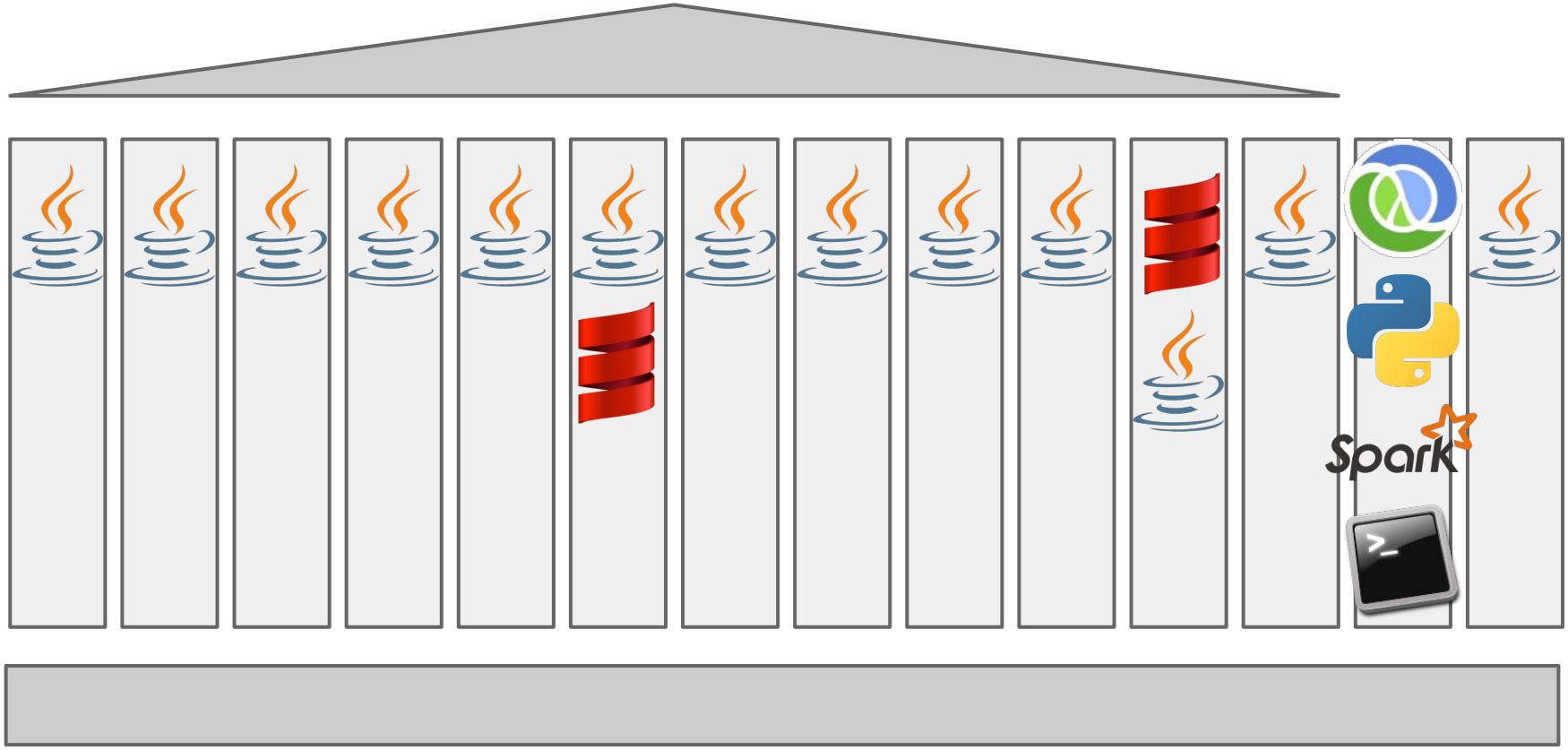
```
(def pipeline
  `((either
      wait-for-manual-trigger
      wait-for-repo)
    (with-repo
      build-artifact
      deploy-ci-and-test
      deploy-qa-and-test
      deploy-production)))
```

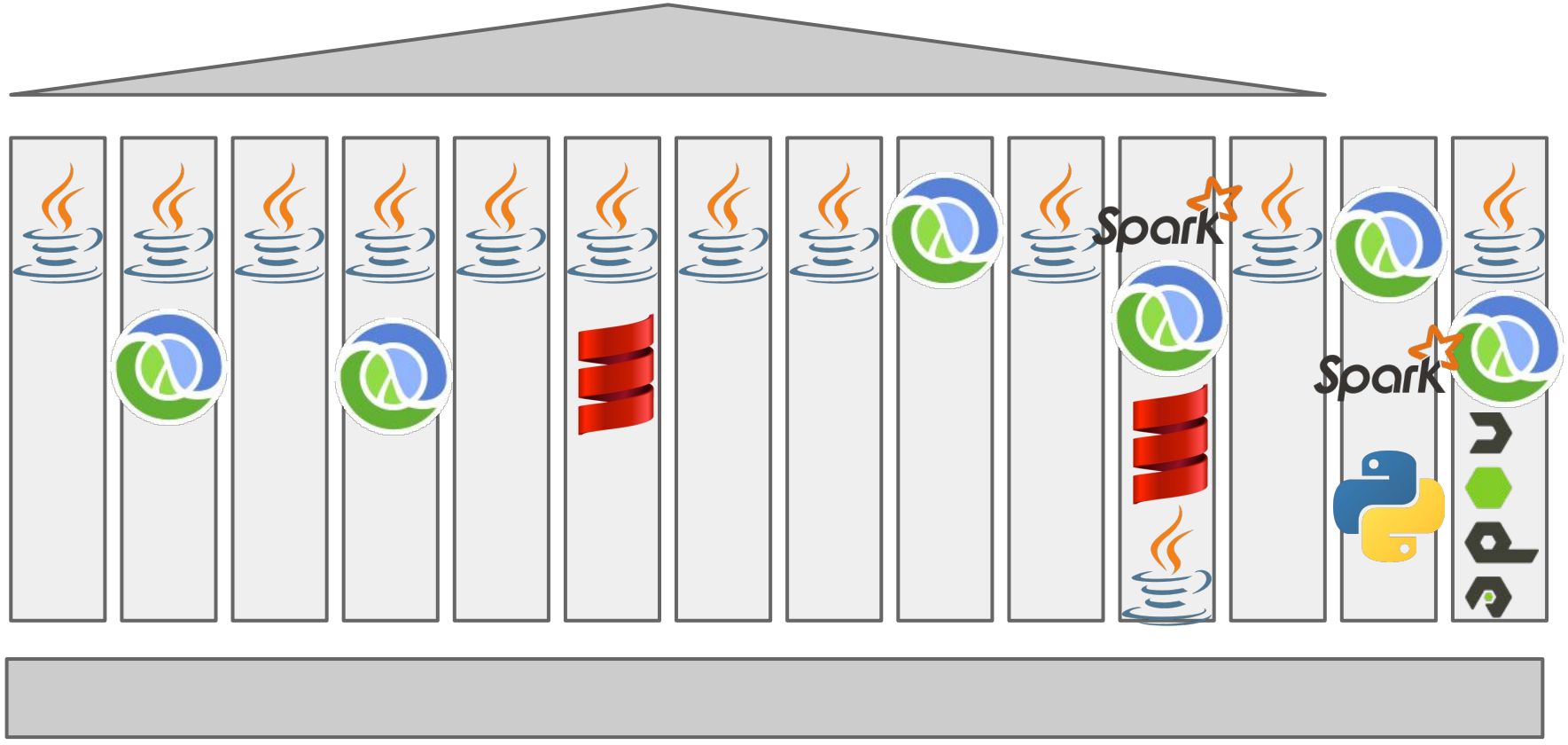
github.com/flosell/lambdacd

How functional programming spreads at otto.de






Spark


Spark
Spark

πάντα
ῥεῖ!

we code

github.com/otto-de

we blog

dev.otto.de

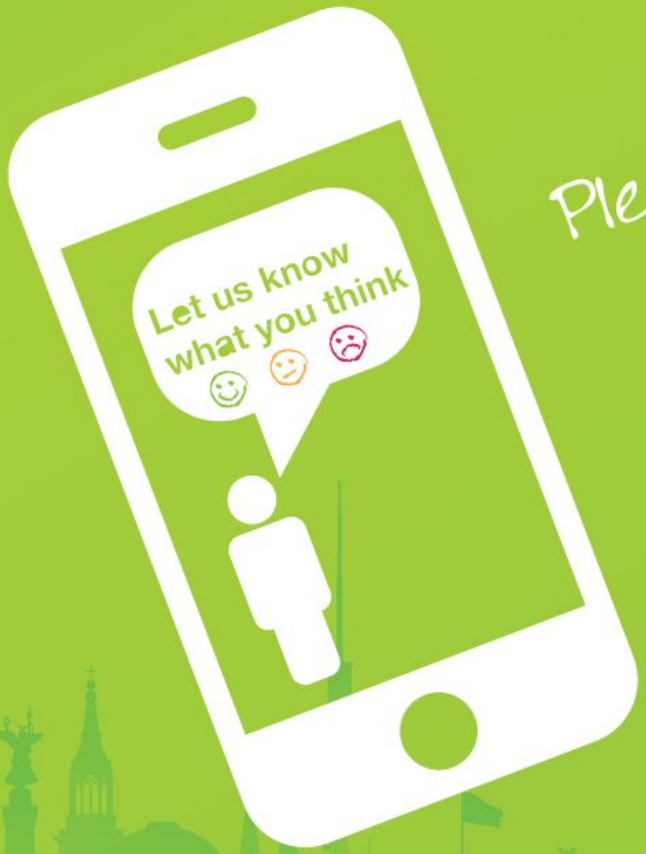
we hire

otto.jobs

christian.stamm@otto.de
@christian_stamm
#GOTOber
4.12.2015

OTTO

goto; conference



Please

Remember to rate session

Thank you!

christian.stamm@otto.de
@christian_stamm