

AN OVERVIEW OF

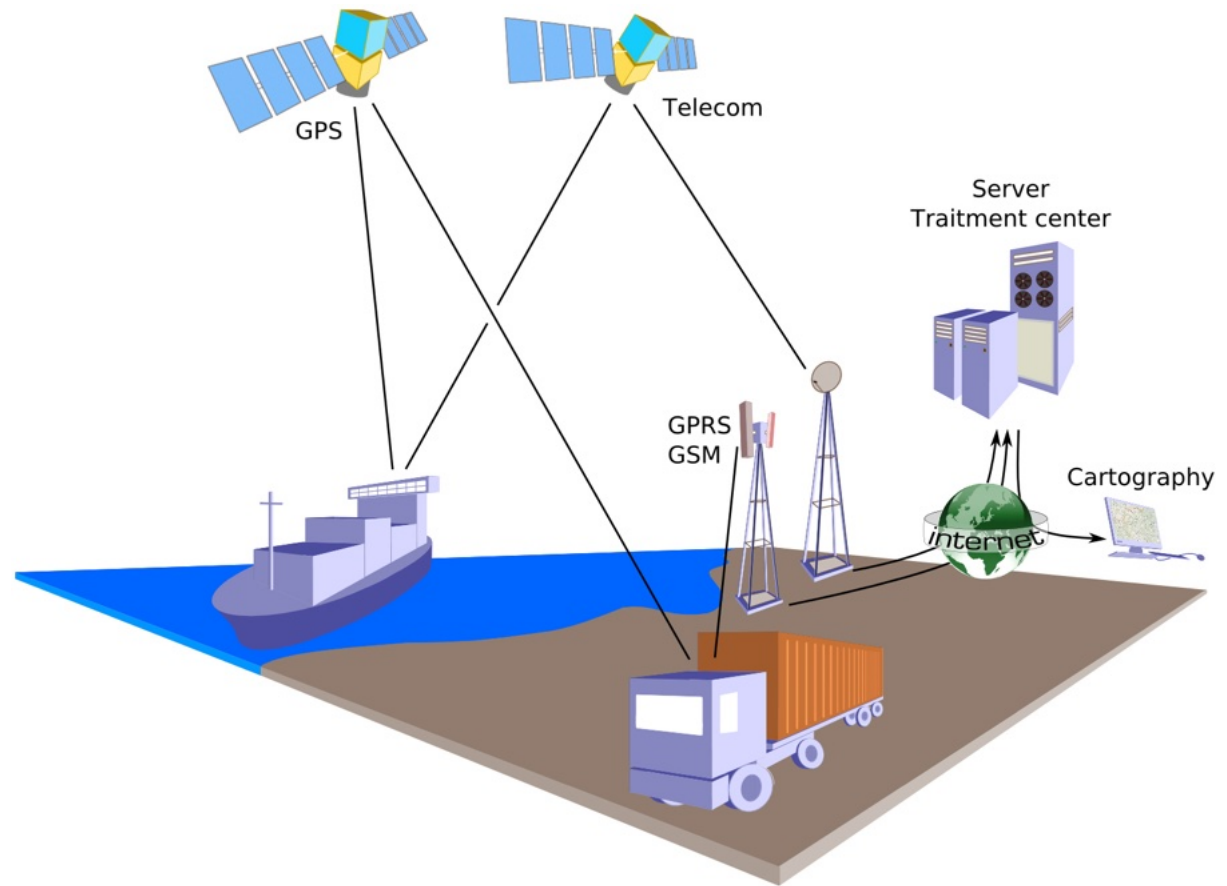
SPRING XD



Erik Lupander
Callista Enterprise AB

ON THE AGENDA...

- Big Data & Data Ingestion
- Spring XD
- Demo



According to IBM

$2,5 \times 10^{18}$

bytes of information is produced each day

That's roughly

~ 2.5 million 1-terabyte hard drives



~ 3,8 billion CD-ROM discs



~1,7 trillion 3.5" floppy discs



BIG DATA

- Big Data can be about ...
 - Collecting data
 - Analyzing data
 - Acting on data
 - And more depending in your context ...
- Machine Learning
 - Is often applied on huge datasets
 - Is often used for drawing conclusions from datasets
 - Fascinating subject
 - But not what this presentation is about
- Spring XD
 - Is about collecting data

For the sake of context, a few real-world Big Data examples.

BIG DATA — EXAMPLES

General Electric Jet Engine

- ~ one terabyte per flight



BIG DATA — EXAMPLES

2014 U.S GP Formula 1 race

- 18 cars
- 150 sensors per car
- 243 terabytes of data



BIG DATA — EXAMPLES

A modern connected car

- ~ 25 gb per hour



BIG DATA - APPLICATIONS

Oil rigs

- \$150 000 per incident
- 25 000 metrics per second
- Machine Learning
 - Predictive maintenance
 - Early warning
 - Reduce failures
 - Progress anticipation



BIG DATA - APPLICATIONS

Mining industry

Dundee Precious Metals

- Sensor data mining and machine learning
 - Reduce production outages
 - \$60 to \$40 per ton raw material extraction cost



Commodity futures

Example from Spring One 2GX:

Predict price of corn, soybean and wheat futures for an undisclosed agricultural company using Twitter posts.

- Ingests ~ 50 million tweets per day using Spring XD
- By using various data analysis technologies together with other data sets, crop futures could be forecasted.



DATA INGESTION

- Data Ingestion is about collecting "Big Data"
 - Distributed & Scalable
 - Pipelining
- Data is often heterogenous
 - Protocols (HTTP, JMS, JDBC, TCP/UDP, FTP, ...)
 - Format (JSON, XML, plain text, binary, ...)
- Data may need processing before being sunk into the Data Lake
 - Filtering
 - Transformation
 - Aggregation
 - Splitting
 - Routing
 - (E.g. traditional Integration Patterns)

DATA INGESTION VS INTEGRATION FRAMEWORKS

- Data Ingestion is used for different purposes than Integration Platforms
- Focus is on scalability over correctness:
 - Aggregated data over single data items
 - E.g. metrics
 - Used for enabling analysis of data rather than handling your core business process
 - No focus on guaranteed or eventual data consistency
 - E.g. no transaction, retry or rollback semantics
- Strictly one-way
 - Can't return data to the remote party

SPRING EXTREME DATA

- So, what is Spring XD (eXtreme Data)?

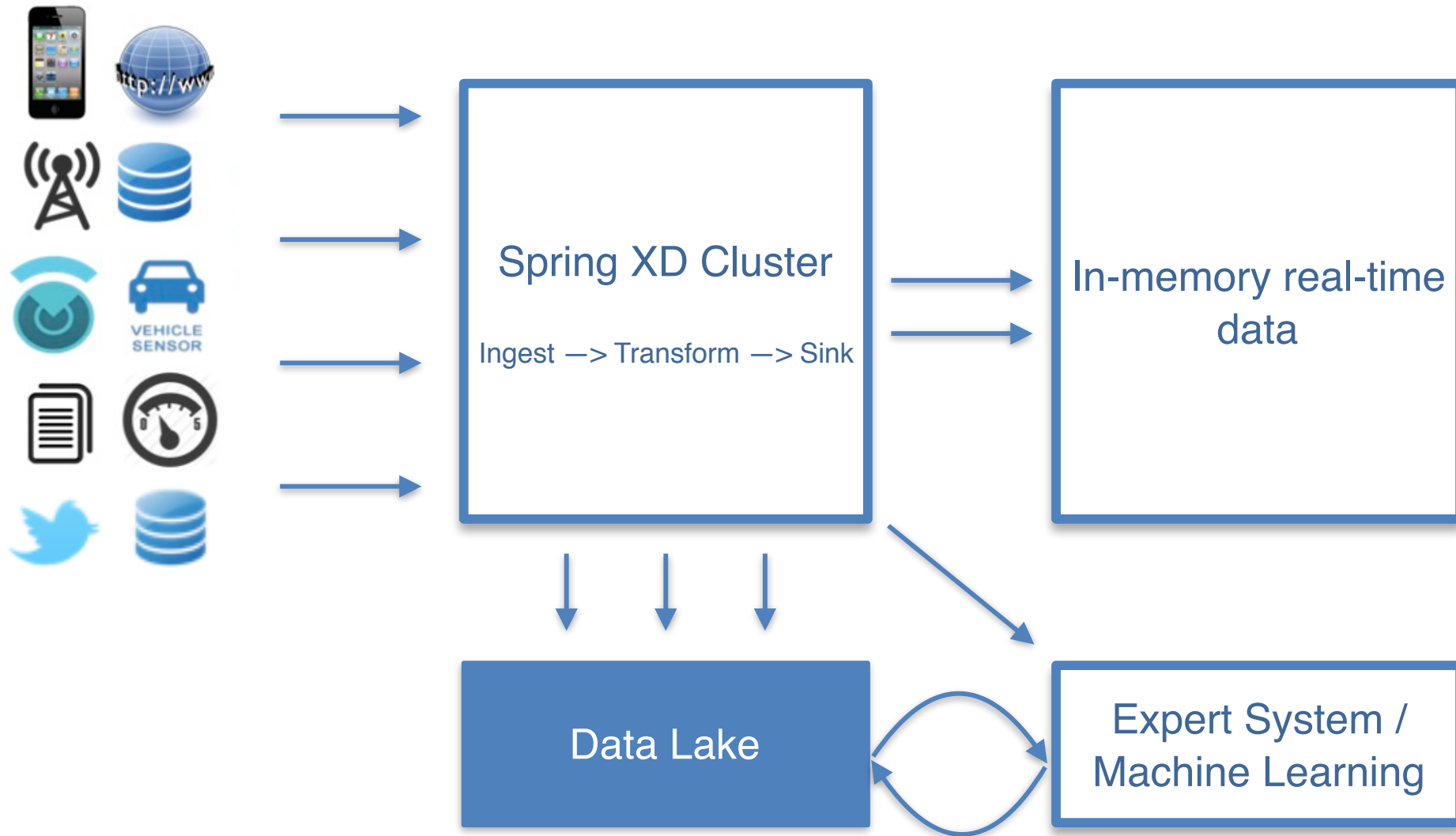


”Distributed data pipelines for real-time and batch processing”

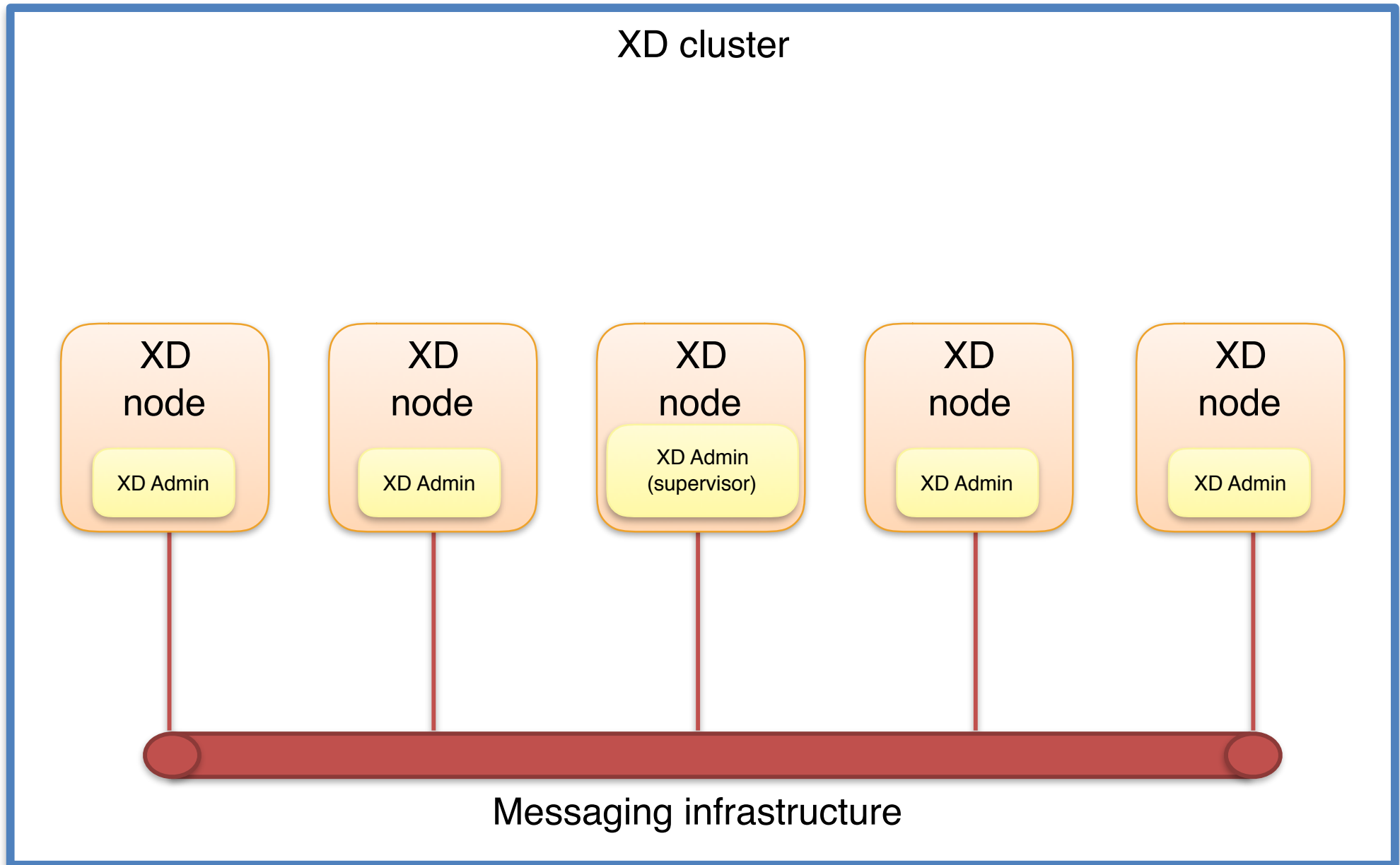
SPRING EXTREME DATA

- Which boils down to:
 - Framework for performing Data Ingestion
 - Distributed streaming of real-time data
 - Real-time analysis of data streams at ingestion time
 - Batch Jobs

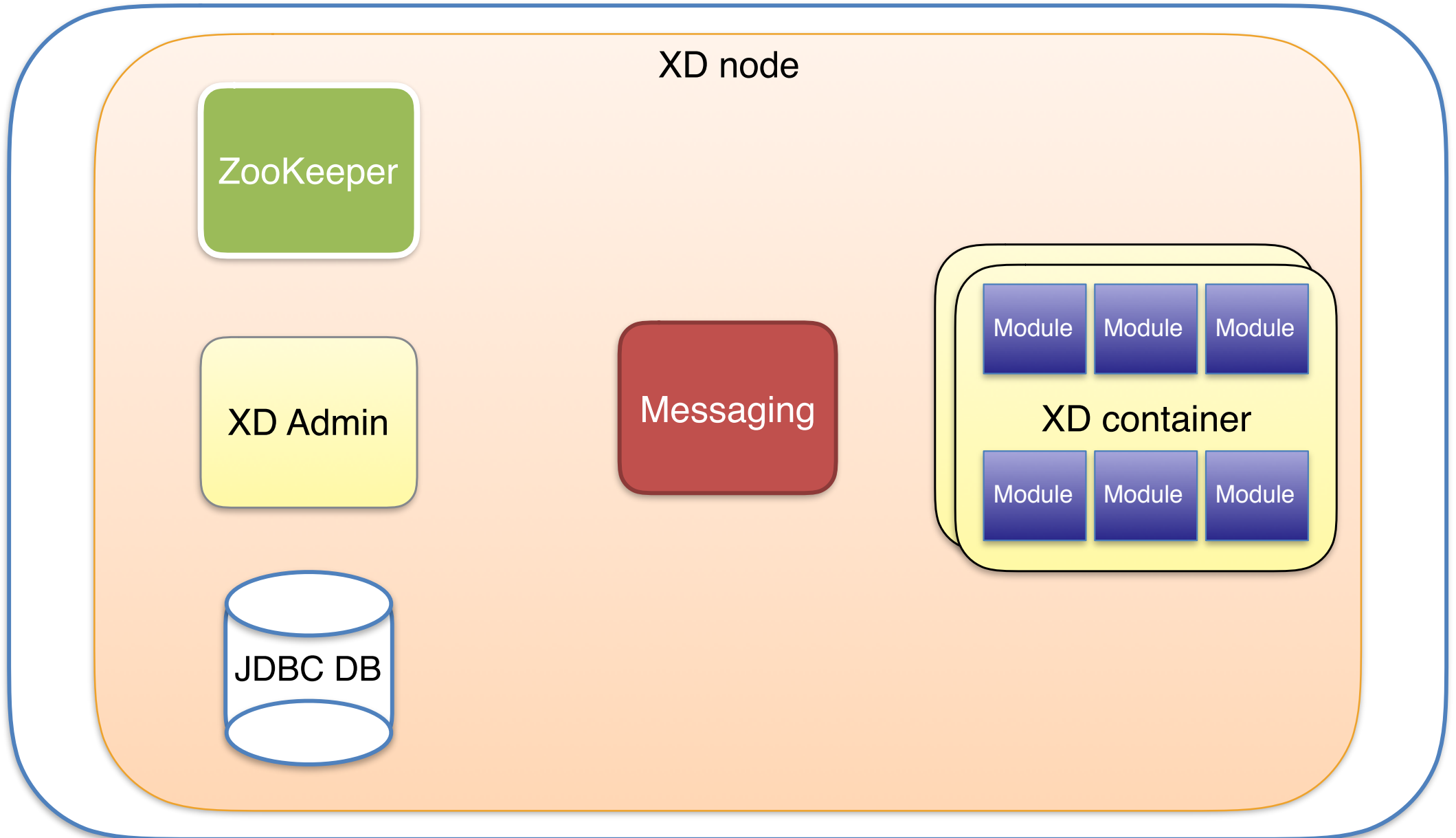
SPRING XD IN THE BIG DATA CONTEXT



SPRING XD — DISTRIBUTED SYSTEM LANDSCAPE



SPRING XD — A DISTRIBUTED XD NODE



THE XD CONTAINER

XD container

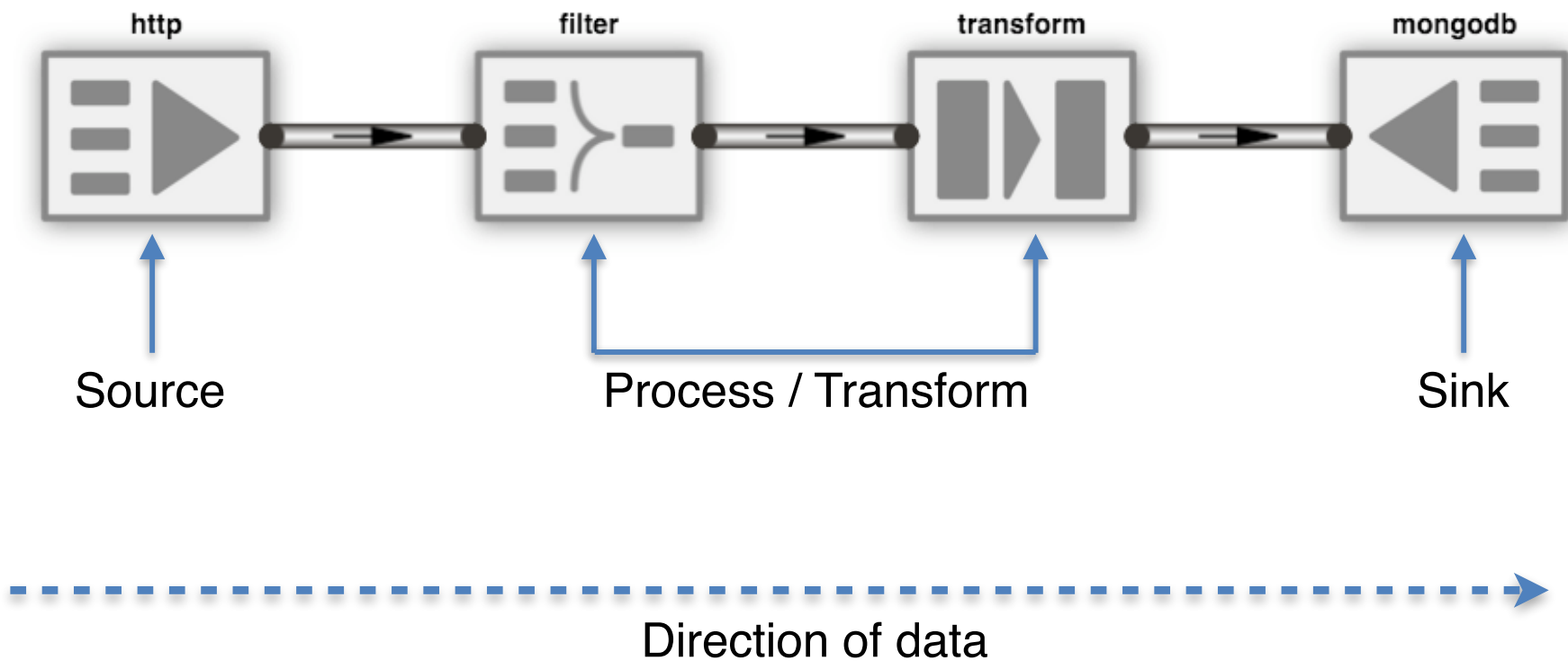


SPRING XD OPERATING MODES

- Distributed
 - QA, production etc.
 - Semi-complex infrastructure, more configuration
- Single-node
 - Development purposes

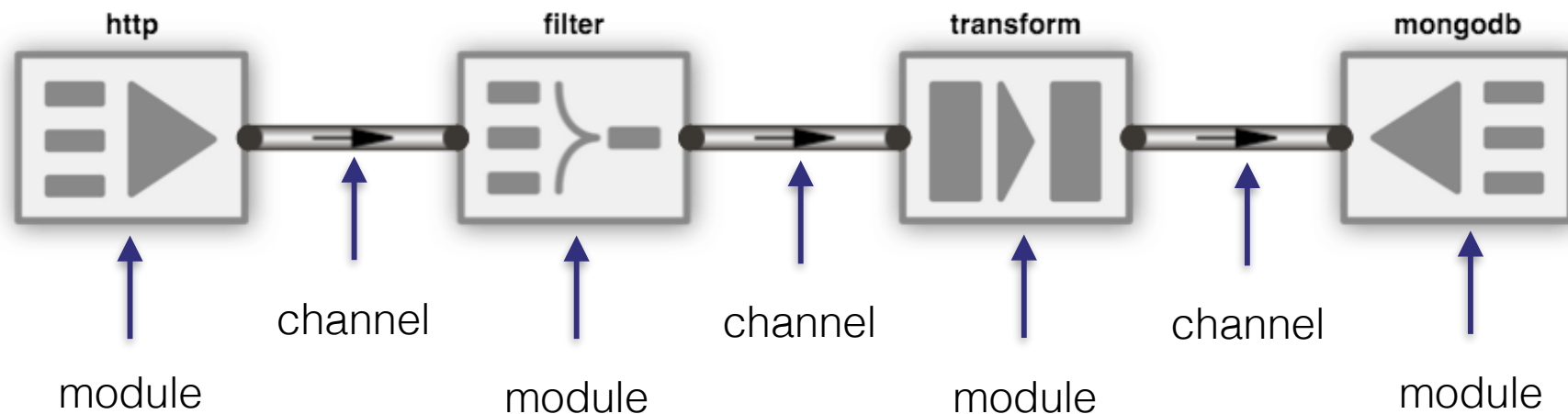
THE SPRING XD STREAM

- Ingest \rightarrow Transform \rightarrow Sink pattern
- A stream is defined by at least one source and one sink



STREAMS & MODULES

- Each part of a Stream is an XD module
- Each module is a "Spring Integration message flow" component
- Each module connects to the next one in the stream through a "channel"



CHANNELS

- Channels connects the XD modules of a Stream
- Based on distributed message passing
 - RabbitMQ, Redis or Kafka
- Modules in the same Stream can live in totally different XD containers and nodes of the cluster



DISTRIBUTION OF MODULES

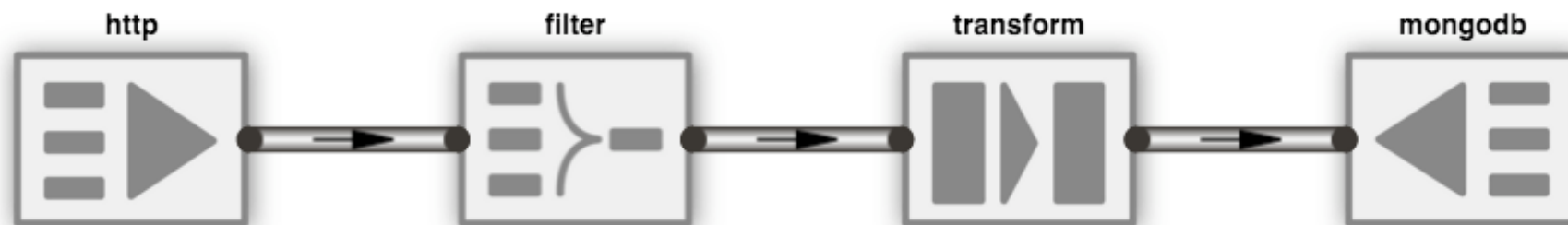
Containers				
Quick filter				
Container Id	Groups	Deployed Modules		Action
03da11e3-ceb0-470f-bd10-76f1b3b37d6e	N/A	httpStream	http.1	↓N/A ↑N/A
		httpStream	file.1	↓N/A ↑N/A
13aad62d-4ccc-4e33-8c33-af7b37848c81	N/A	httpStream	transform.1	↓N/A ↑N/A
2d6652d7-4d7b-4d16-91a3-74e0e63d5bb5	N/A	httpStream	filter.1	↓N/A ↑N/A

THE SPRING XD MODULE

- The XD module is the core building component
- Three major types:
 - Sources - Data inputs such as HTTP, TCP, JDBC, JMS, FTP
 - Processors - filters, transformers, aggregators, splitters, routers...
 - E.g. Good ol' Integration Patterns
 - Sinks - Data outputs. HDFS, JDBC, JMS, HTTP, TCP, FTP, Logs, Files, Null ...
- One special type: Taps
 - Comparable to a wiretap, branches off at a defined position in a Stream with a copy of the payload
- XD comes with ~ 60 ready-to-use modules
- Custom XD modules very easy to develop
 - Groovy scripts, Java, ...

INGEST -> TRANSFORM -> SINK

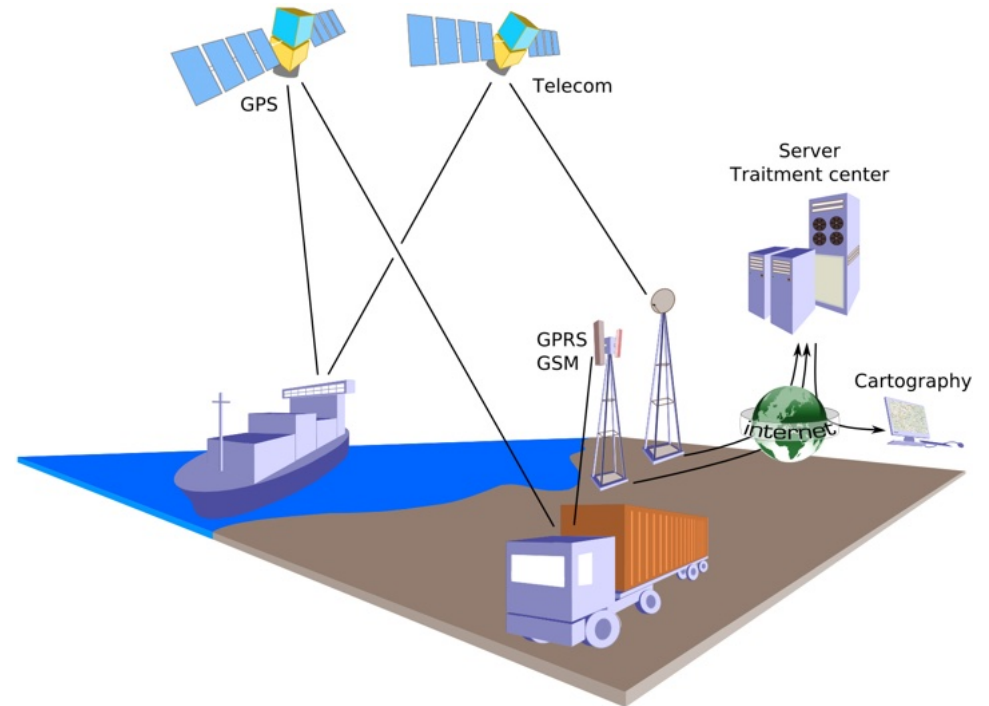
DSL demo



```
stream create --name myStream --definition
    "http --port=1717 |
filter --expression=payload.contains(':')
    | log"
```


THE DEMO

- Make-believe "Fleet Management"
- Simulates 2000 vehicles
- Each submitting one TCP and one HTTP message every two seconds
 - Geolocation & Engine data
- Process and dump TCP into MongoDB
- Log HTTP and TCP
- Realtime analytics



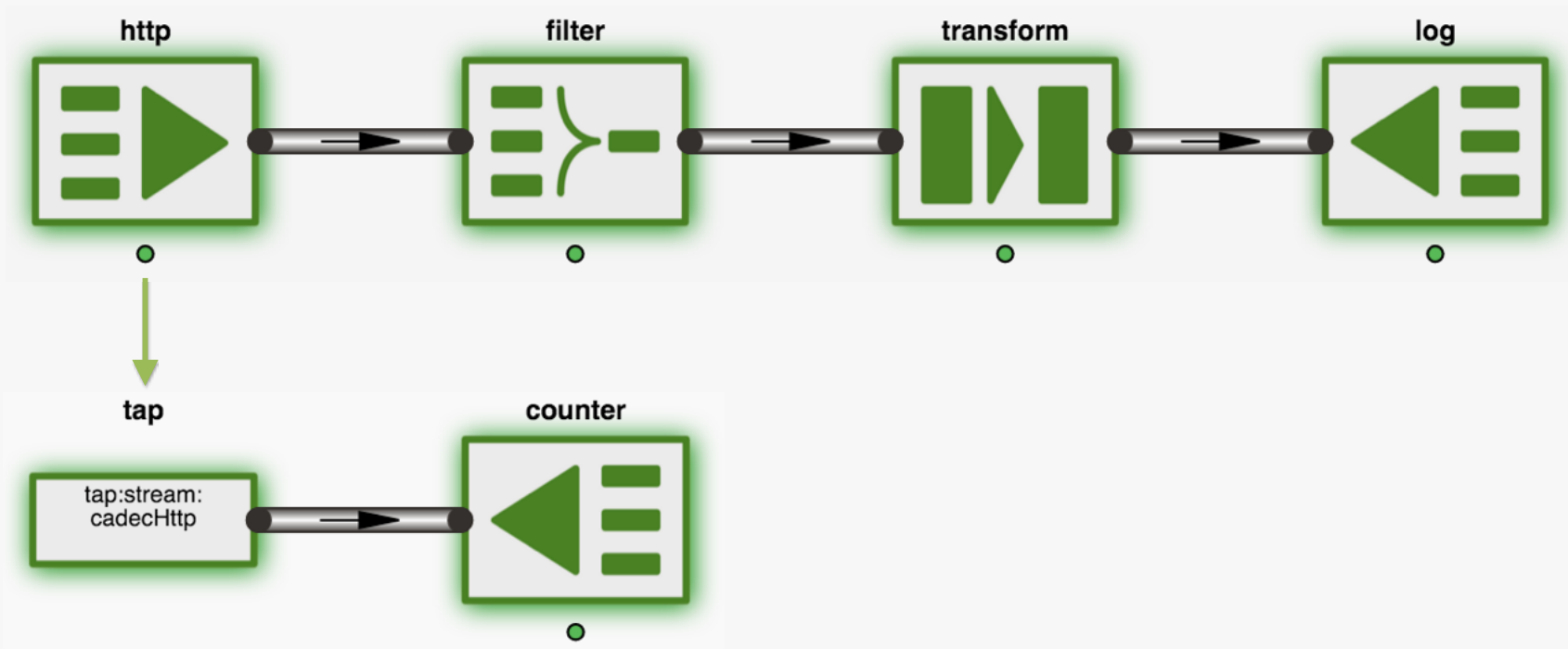
DEMO - LOAD TEST TOOL

- Uses "Gotling" load-test framework
 - Inspired by Gatling, but written i Go.
 - Support for high volumes of HTTP requests and TCP packets
- See <http://callistaenterprise.se/blogg/teknik/2015/11/22/gotling>

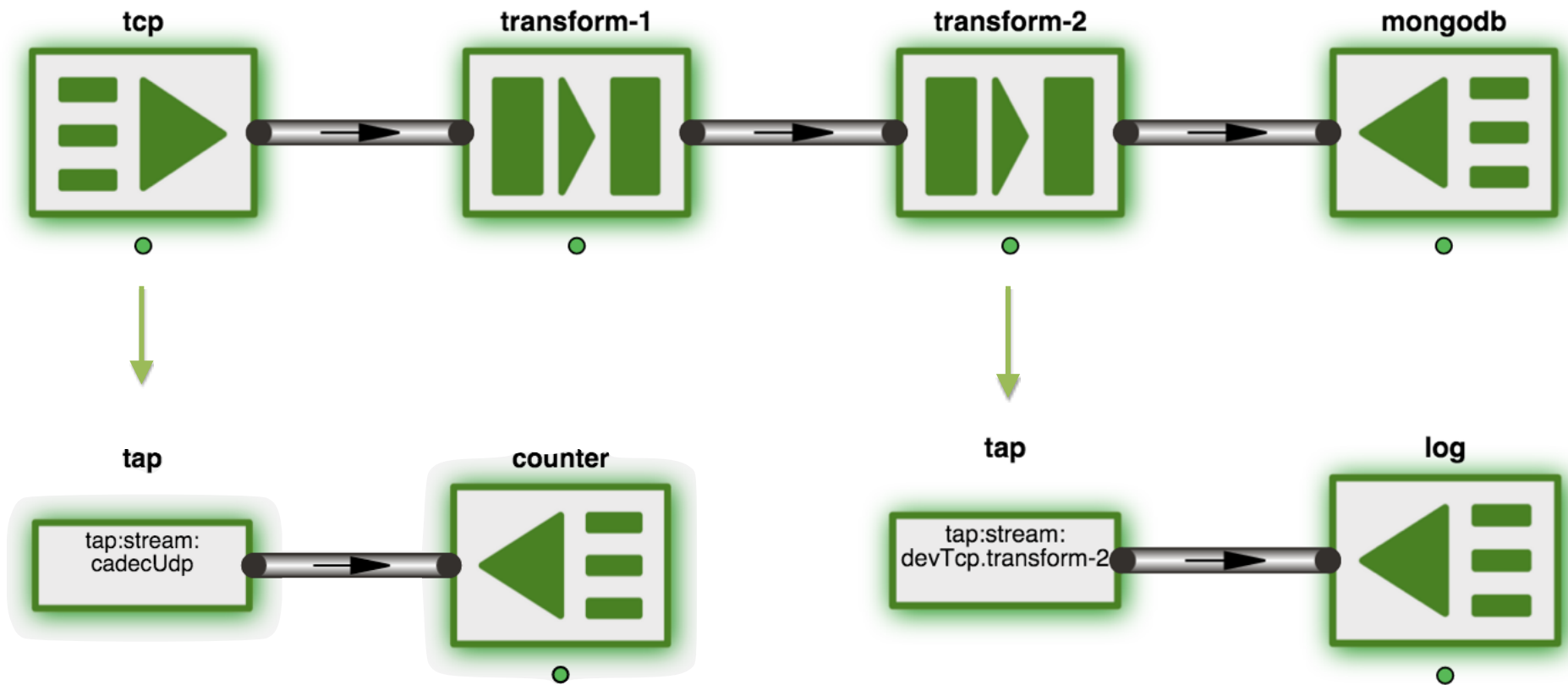
GOTLING SIMULATION SPECIFICATION

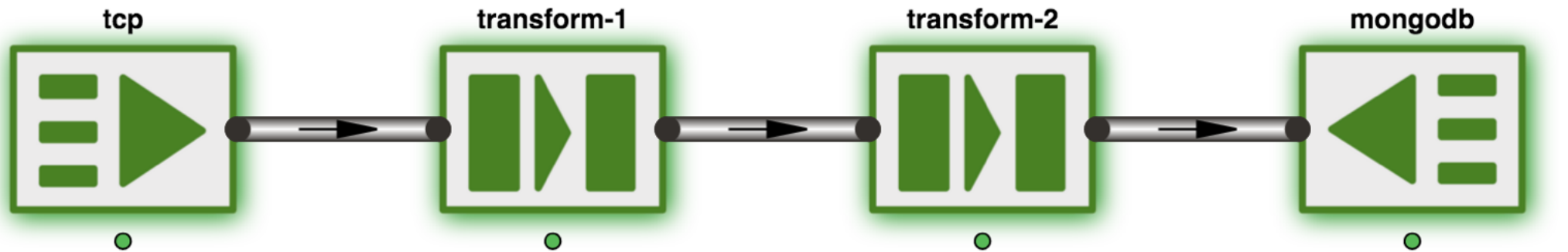
```
---
iterations: 500
users: 2000
rampup: 60
feeder:
  type: csv
  filename: fleetdata.csv
actions:
  - sleep:
    duration: 1
  - http:
    title: Submit Geolocation data
    method: POST
    url: http://localhost:10000
    accept: json
    body: '{"vehicleid":${UID},"lat":${lat},"lon":${lon}}'
  - sleep:
    duration: 1
  - tcp:
    title: TCP engine data packet
    address: 127.0.0.1:8081
    payload: ${UID}|${dp1}|${dp2}|${dp3}|${dp4}|${dp5}
```

DEMO STREAMS - HTTP GEOLOCATION



DEMO STREAMS – TCP ENGINE DATA





```
31303030337c313
32e337c2034332e
357c3334
```

```
10003|13.3|43.5|34
```

```
[
  "10003",
  "13.3",
  "43.5",
  "34"
]
```

```
{
  "UID" : "10003",
  "nox" : "13.3",
  "co2" : "43.5",
  "fuel" : "34"
}
```

byte[]



String



String[]



Tuple/JSON

CUSTOM PROCESSOR USING GROOVY

DSL SNIPPET

```
... | transform --script=totuple.groovy | ...
```

totuple.groovy

```
return org.springframework.xd.tuple.TupleBuilder.tuple()  
    .put("UID", payload[0])  
    .put("nox", payload[1])  
    .put("co2", payload[2])  
    .put("fuel", payload[3])  
    .build()
```

DEMO

SPRING CLOUD DATA FLOW VS SPRING XD

- Spring Cloud Data Flow
 - "Cloud-native Data Ingestion"
- Redesign of Spring XD for the Cloud
 - Uses the same DSL, admin shell, GUI etc.
- Geared for cloud-based (PaaS) runtime environments
 - Pivotal Cloud Foundry
 - YARN
 - Apache Mesos
 - Kubernetes



SIMILAR PROJECTS / PRODUCTS

- Apache Spark Streaming (Berkeley)
- Apache Storm (Twitter)
- Gobblin (LinkedIn)

SUMMARY

- Big Data & Data Ingestion:
 - Huge subject with many use cases and perspectives
 - Data Ingestion is important:
 - Many Data Scientists spend > 50% time on data setup
 - Data is diverse
 - There are a lot of data out there!
- Spring XD:
 - Provides a scalable framework for ingesting data
 - Simple DSL and decent admin tools
 - Easy to extend and customize
 - Slightly complex runtime environment
 - Spring Cloud Data Flow may be an option
 - Is NOT a traditional Integration Framework

Thanks for listening!

Questions?

SOME URLS

Dundee Precious Metals:

<http://www.wsj.com/articles/mining-sensor-data-to-run-a-better-gold-mine-1424226463>

GE Jet engine:

<http://www.bloomberg.com/bw/articles/2012-12-06/ge-tries-to-make-its-machines-cool-and-connected>

Connected car:

<https://www.hds.com/assets/pdf/hitachi-point-of-view-internet-on-wheels-and-hitachi-ltd.pdf>

F1 teams

<http://www.forbes.com/sites/frankbi/2014/11/13/how-formula-one-teams-are-using-big-data-to-get-the-inside-edge/>

Oil Industry

<https://blog.pivotal.io/pivotal/case-studies/data-as-the-new-oil-producing-value-for-the-oil-gas-industry>