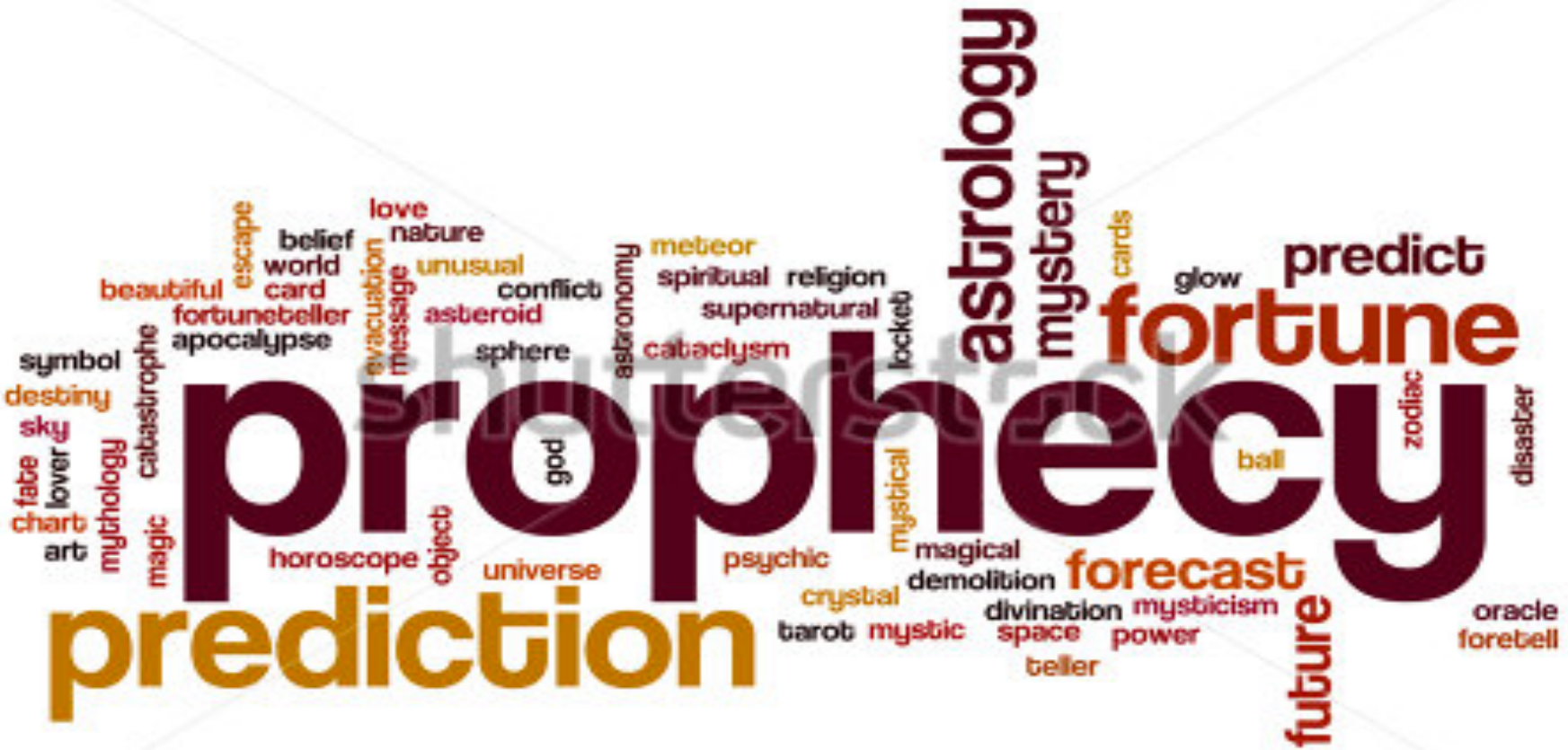
A Stock Prediction System using open-source software

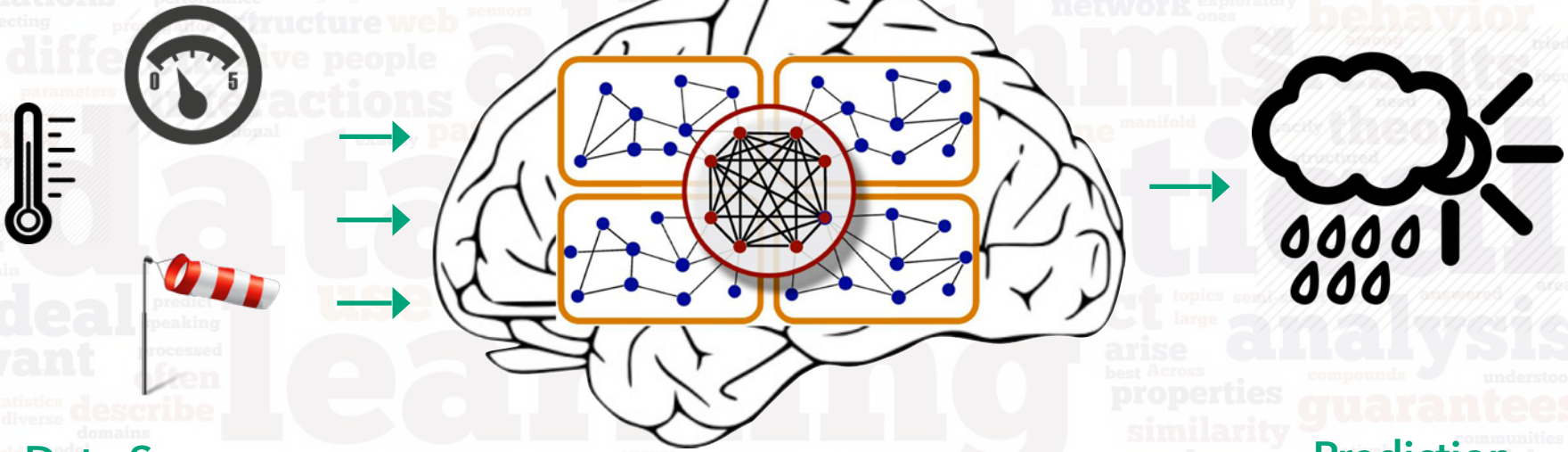
Fred Melo
fmelo@pivotal.io
@fredmelo_br

William Markito
wmarkito@pivotal.io
@william_markito

Pivotal



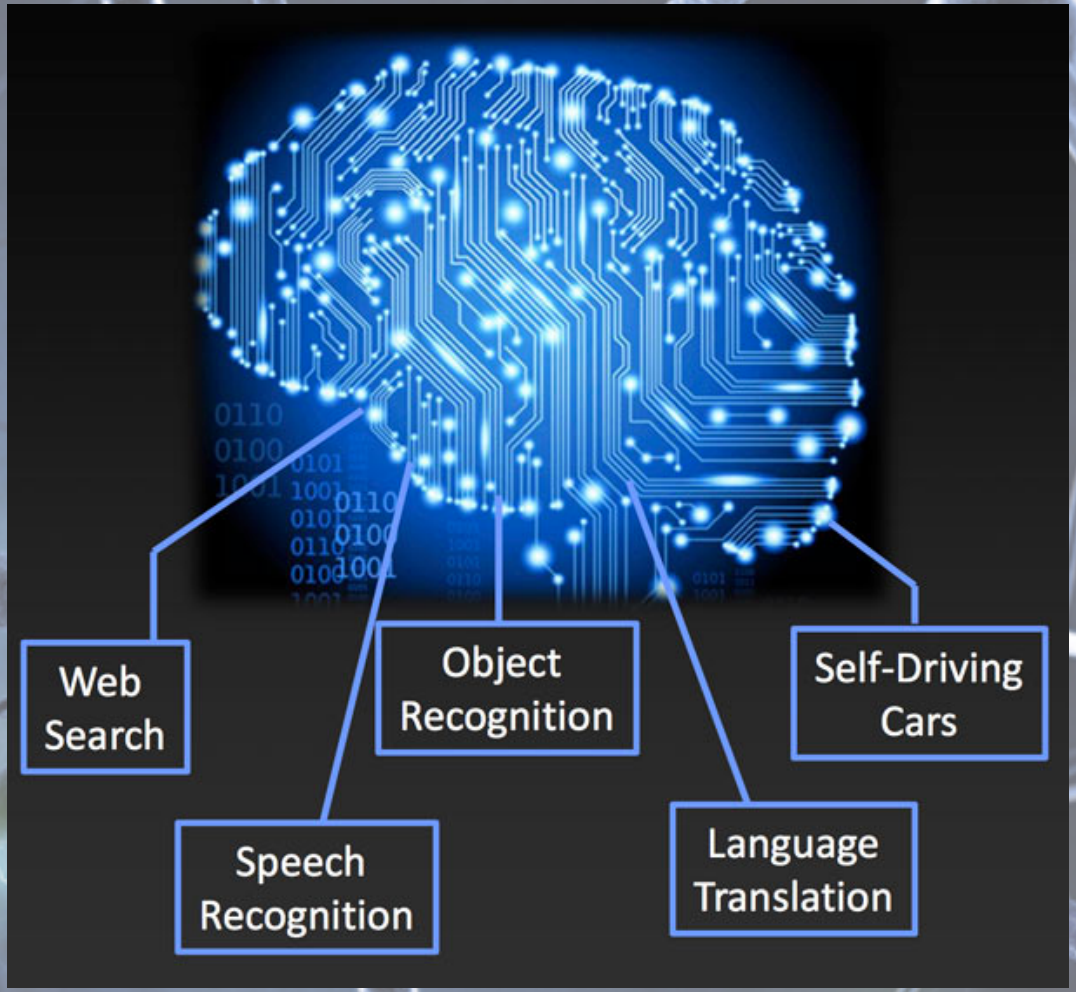
It's all about **DATA**



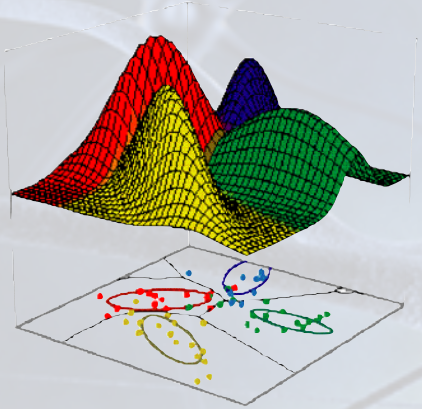
Data Sources

Look for patterns

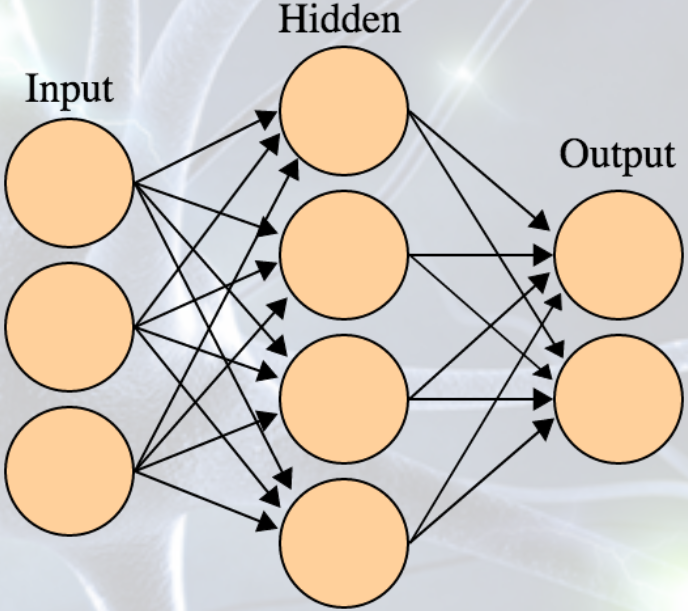
Prediction


Web Search
Object Recognition
Self-Driving Cars
Speech Recognition
Language Translation

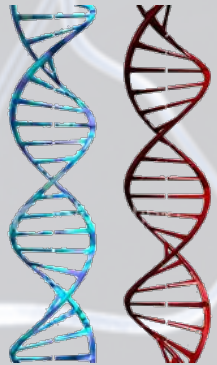
Machine Learning is the answer



Clustering

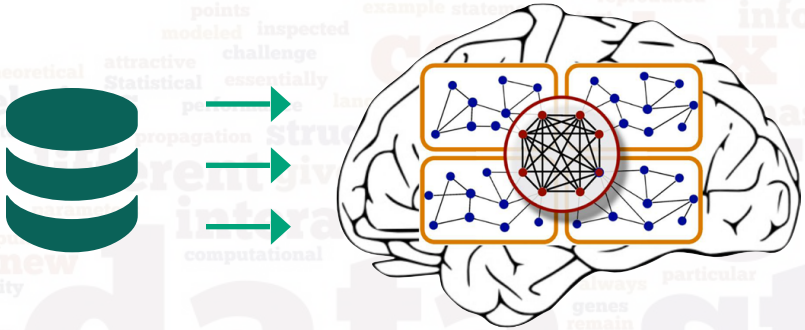


Neural Networks

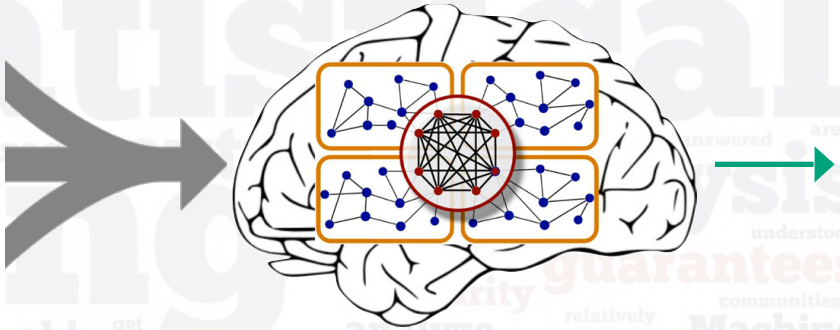


Genetic Algorithms

Applying Machine Learning



Train with historical dataset



Apply model to the new input

Why so hard?

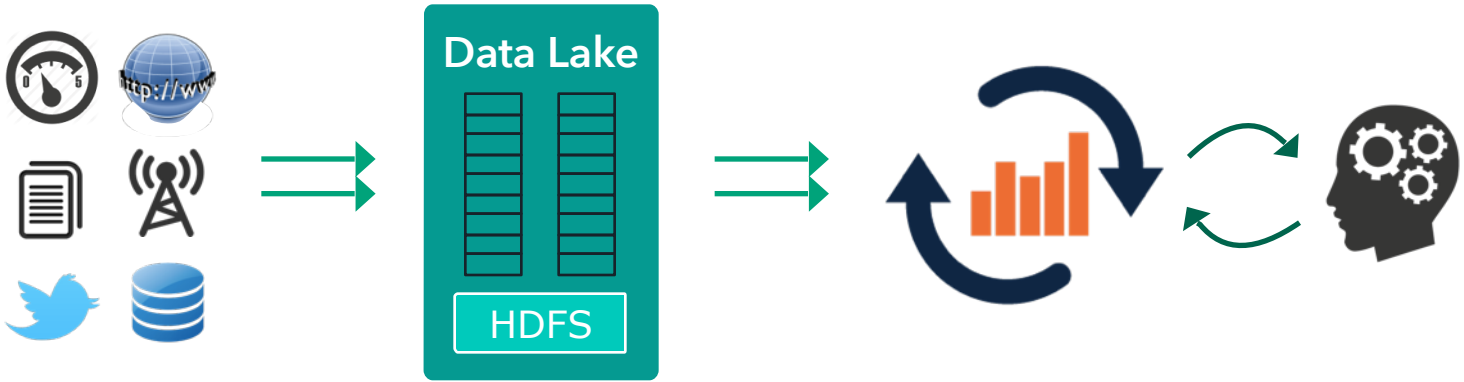
Hard to scale

Hard to make it real-time

Hard to add new data sources

Why?

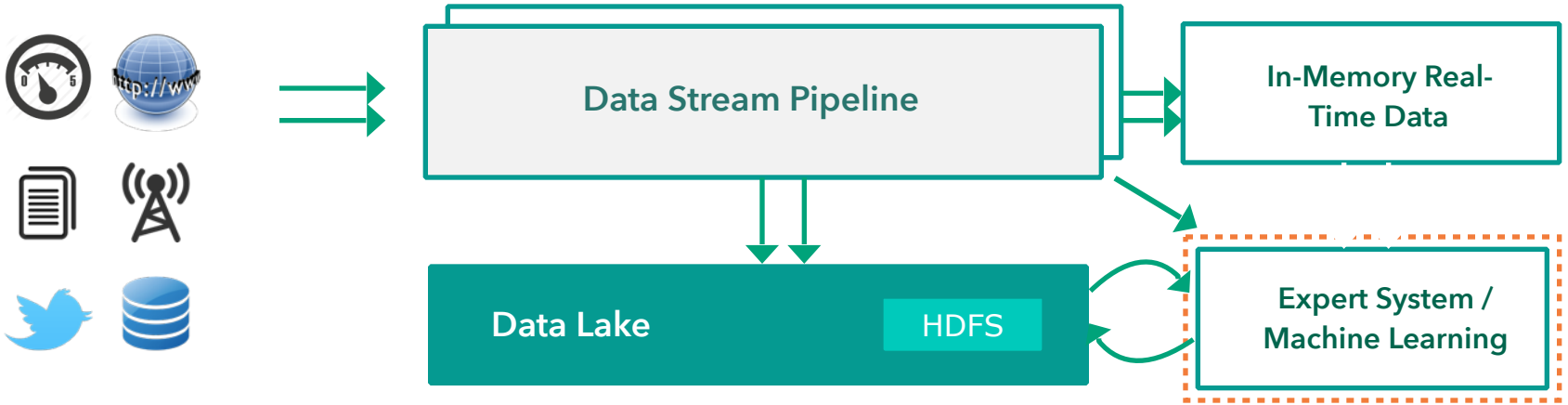
Traditional models are reactive and static



No real-time information
ETL based
Data-source specific

Hard to change
Labor intensive
Inefficient

Stream-based, real-time closed-loop analytics are needed



Multiple Data Sources
Real-Time Processing
Store Everything

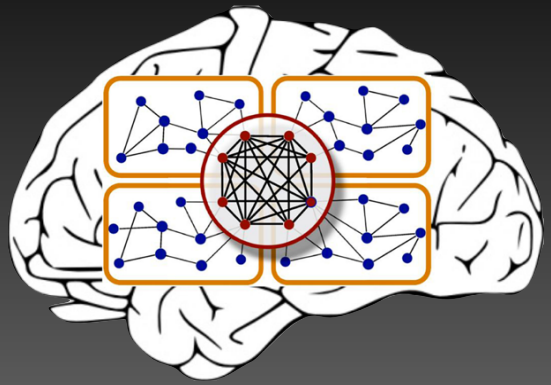
Continuous Learning
Continuous Improvement
Continuous Adapting

How can it be addressed?

Info

Analysis

Neural Network



Look at past trends
(for similar input)

Evaluate current input

Score / Predict

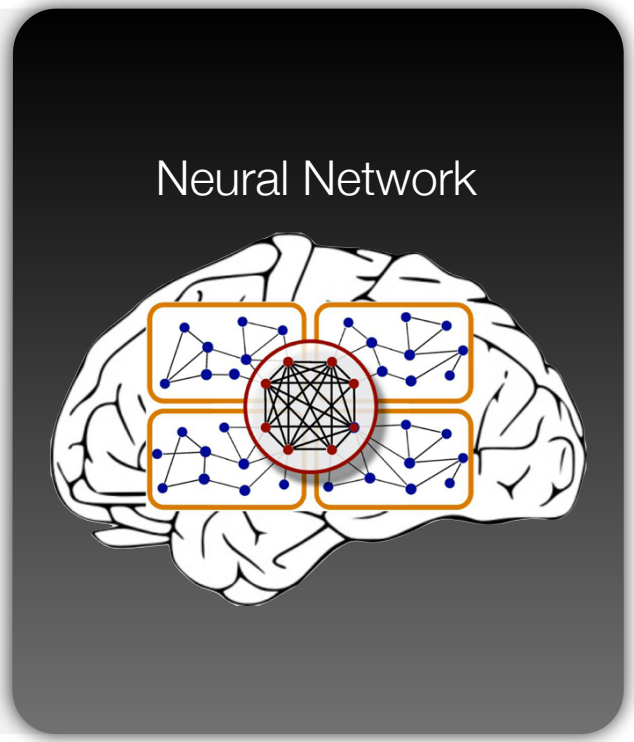
How can it be addressed?

Info

Analysis

Filter

[json]



How can it be addressed?

Info

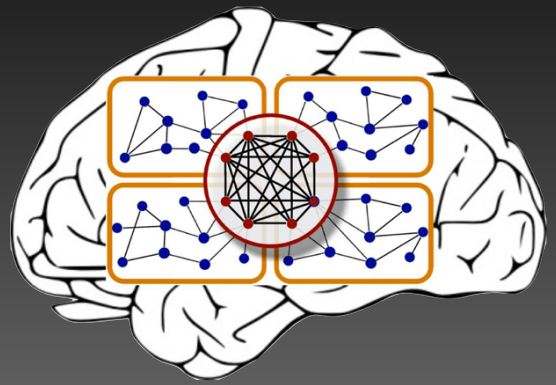
Analysis

Filter → Enrich

Spark

GEODE

Neural Network

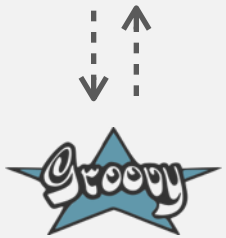


How can it be addressed?

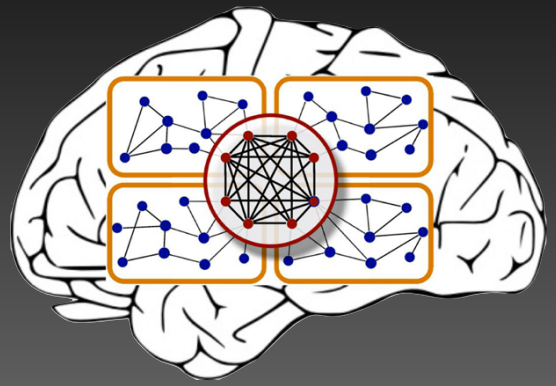
Info

Analysis

Filter → Enrich → Transform



Neural Network



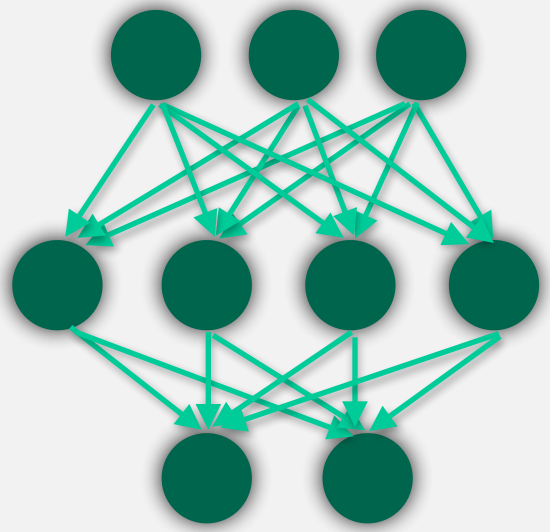
How can it be addressed?

Info

Analysis

Filter → Enrich → Transform →

Neural Network

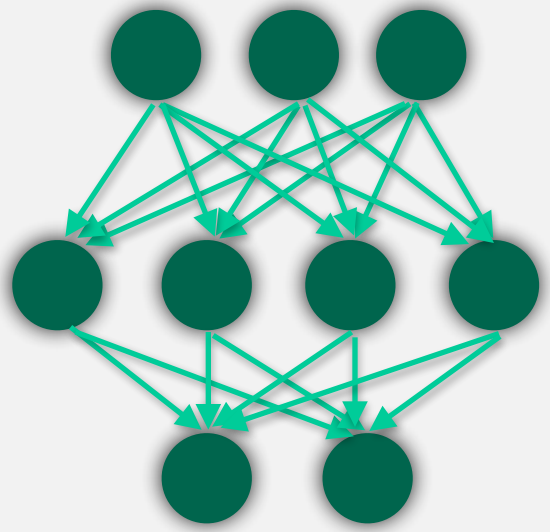


How can it be addressed?

Info

Filter → Enrich → Transform →

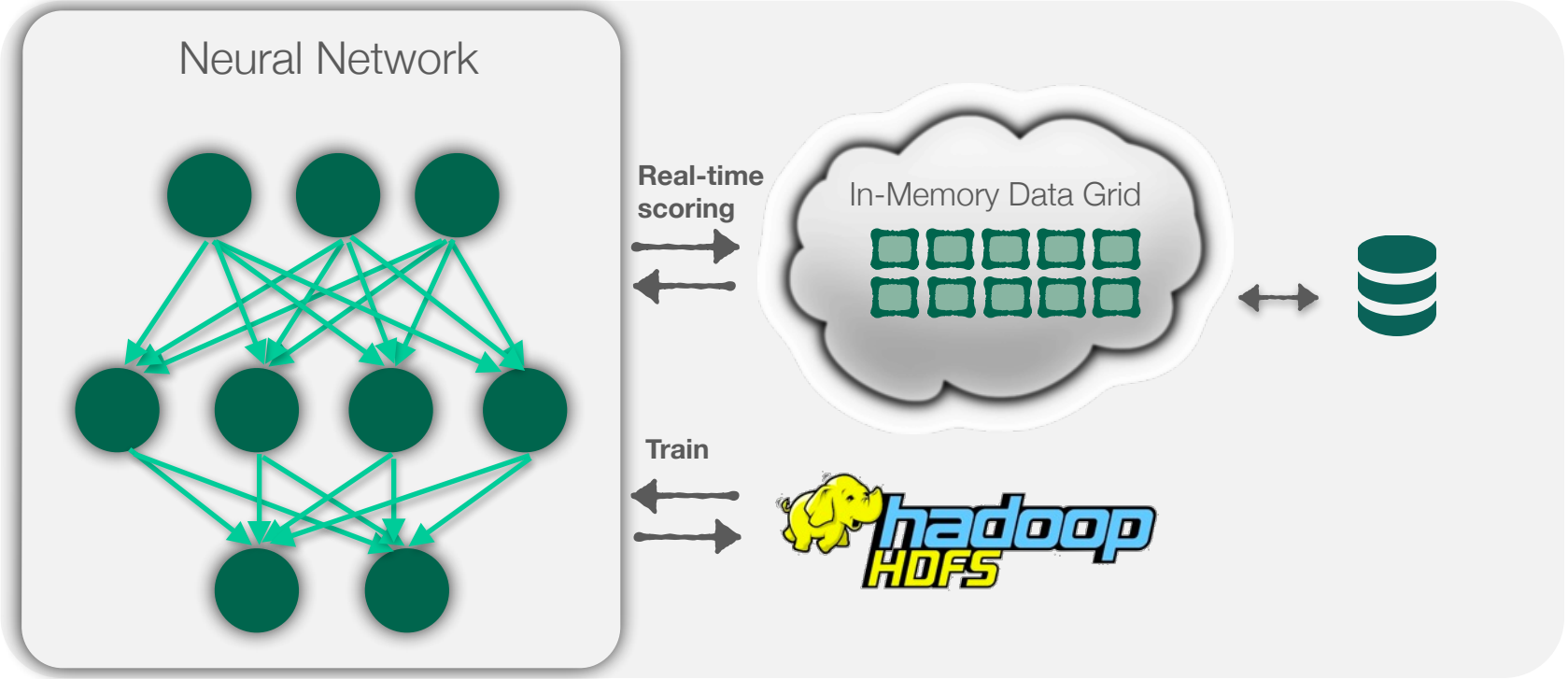
Neural Network



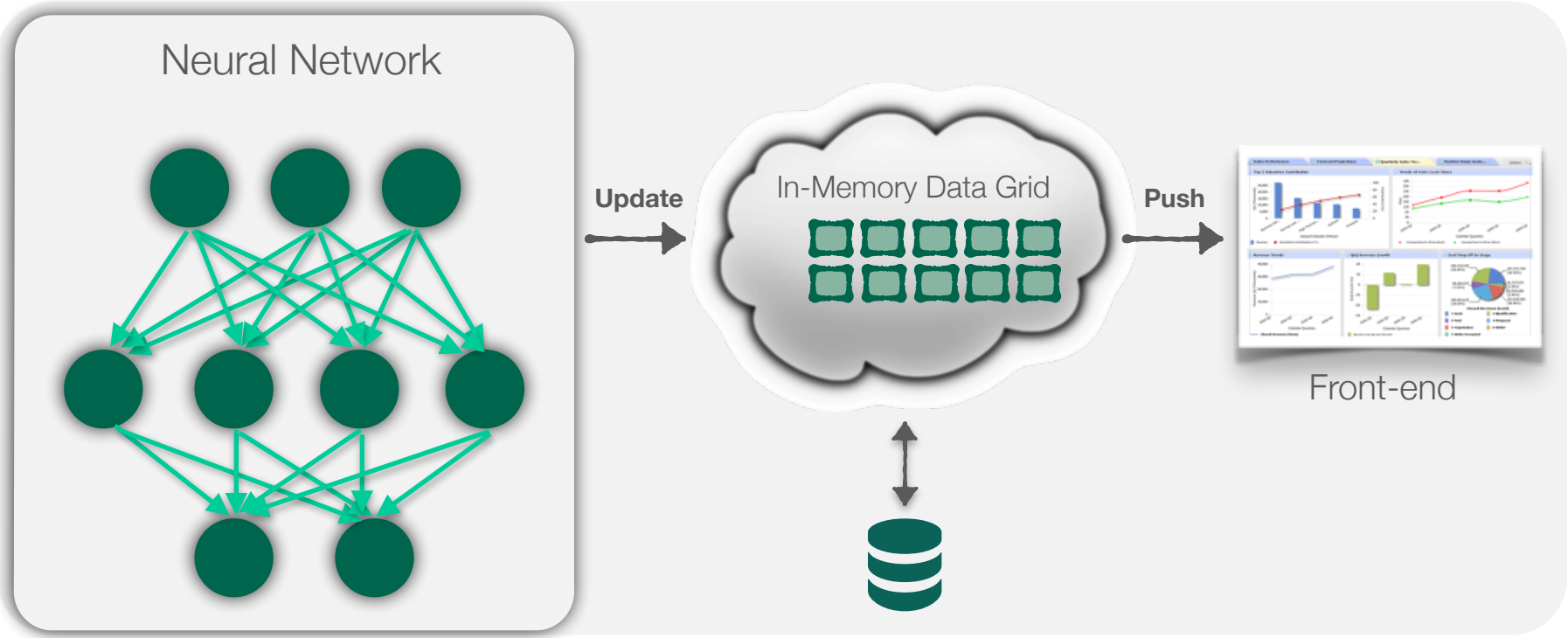
Analysis

← Transform ←

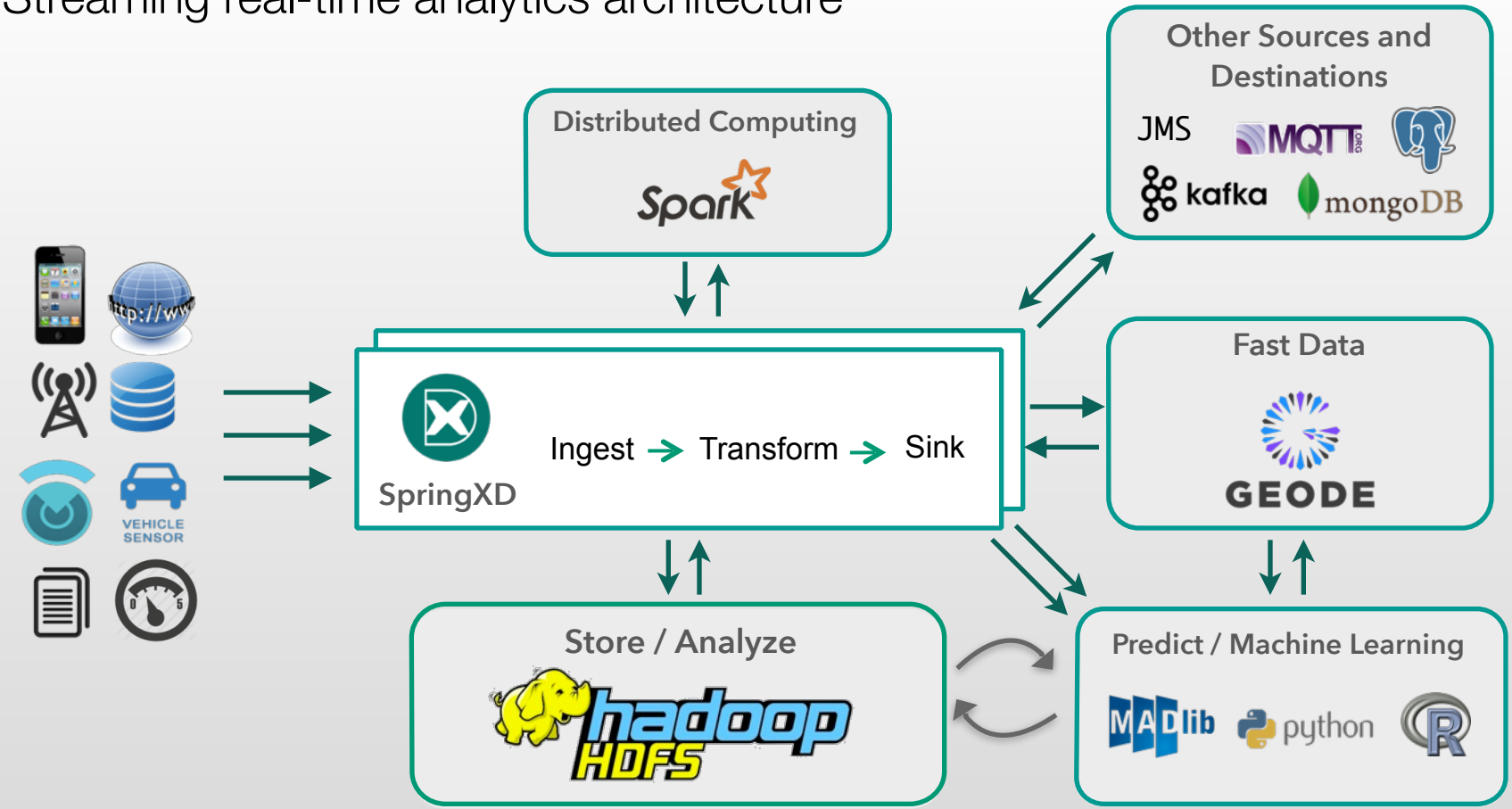
How can it be addressed?



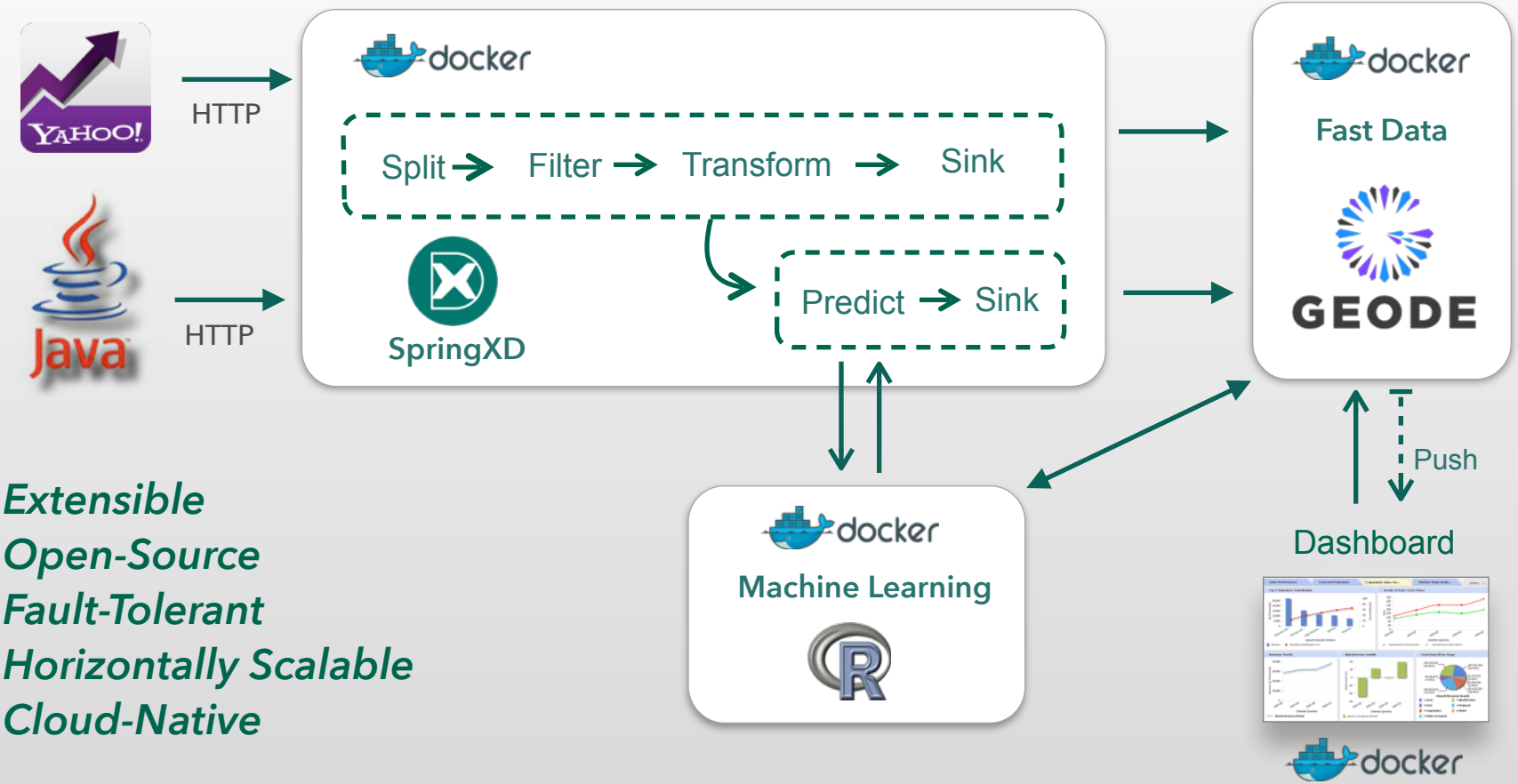
How can it be addressed?



Streaming real-time analytics architecture



Demo Architecture



Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable
Cloud-Native

SpringXD

Data Stream Pipelining

INGEST / SINK

- Little or no coding required
- Dozens of built-in connectors
- Seamless integration with Kafka, Sqoop
- Create new connectors easily using Spring

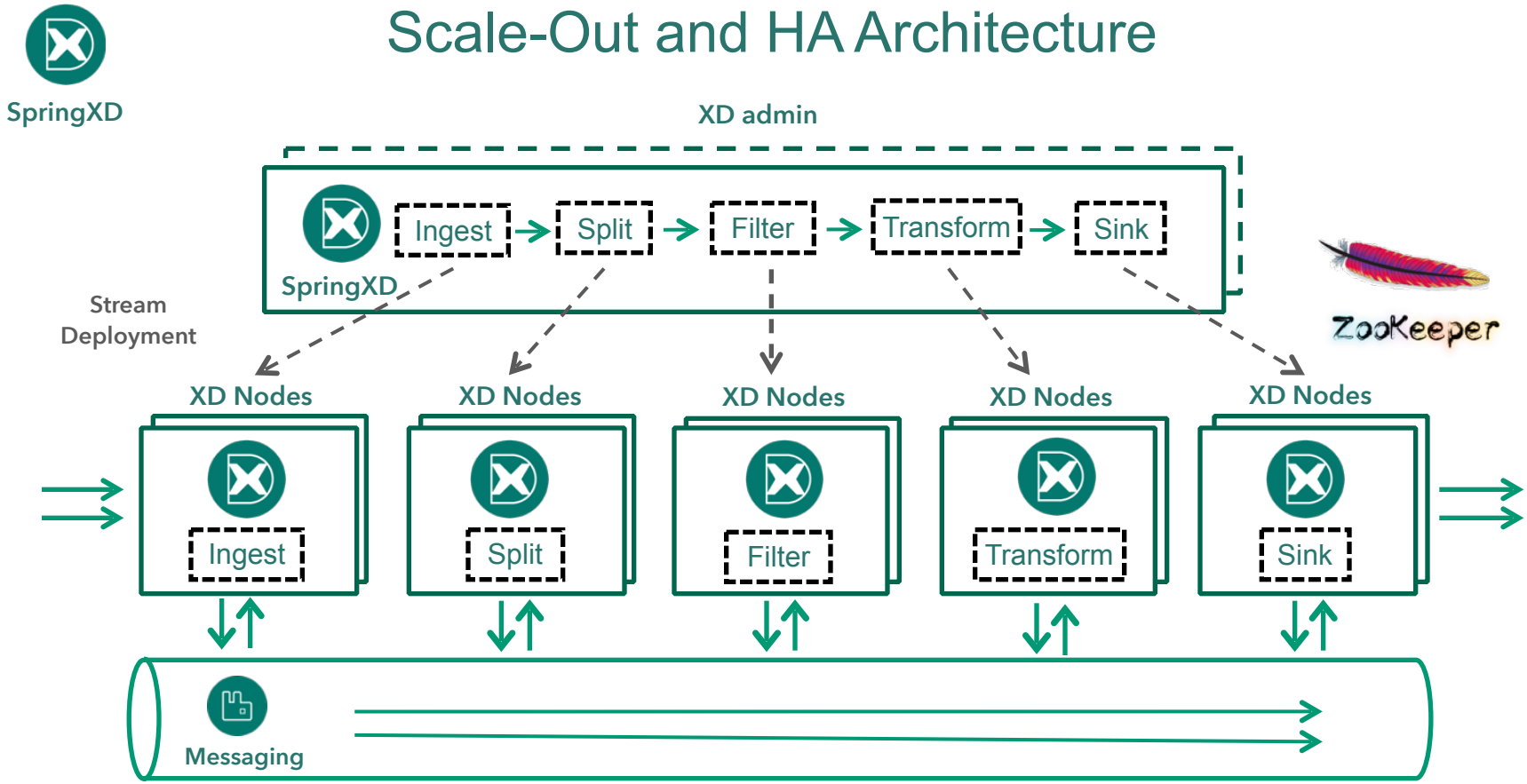
PROCESS

- Call Spark, Reactor or RxJava
- Built-in configurable filtering, splitting and transformation
- Out-of-box configurable jobs for batch processing

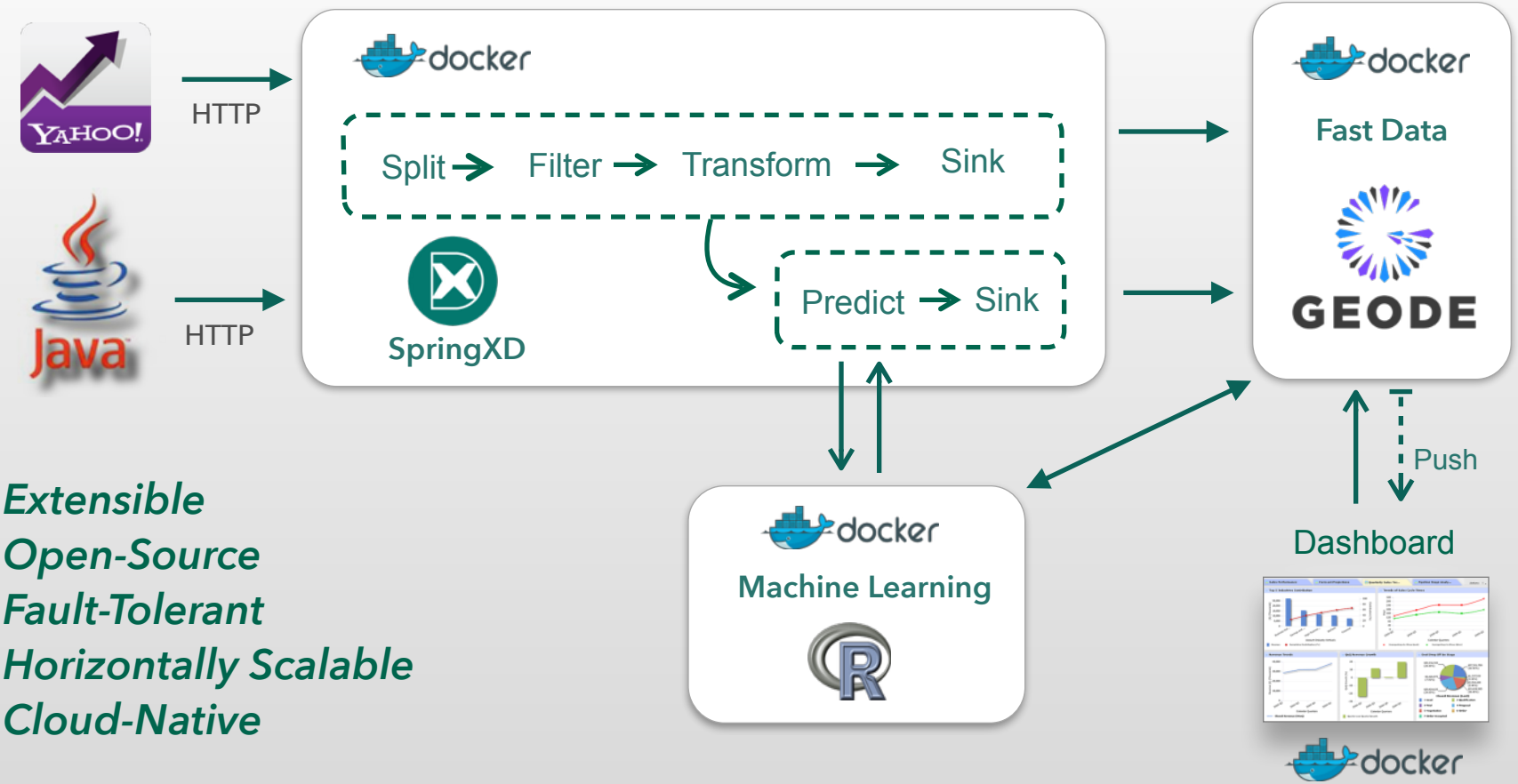
ANALYZE

- Import and invoke PMML jobs easily
- Call Python, R, Madlib and other tools
- Built-in configurable counters and gauges

Scale-Out and HA Architecture

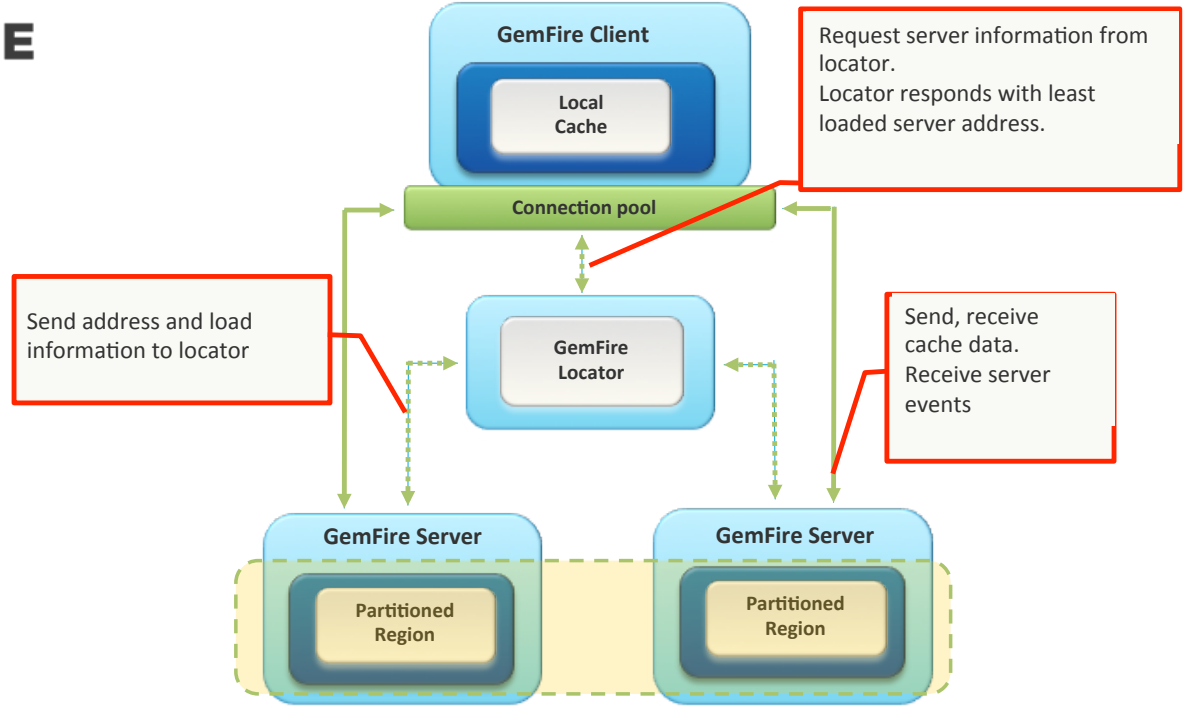


Demo Architecture

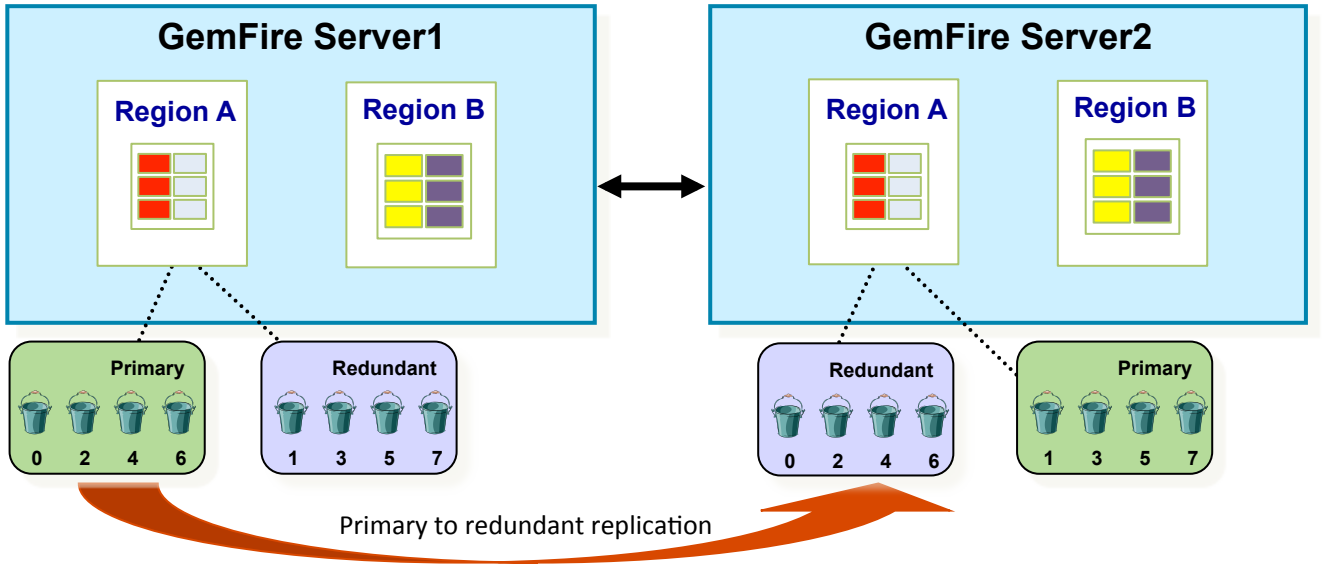


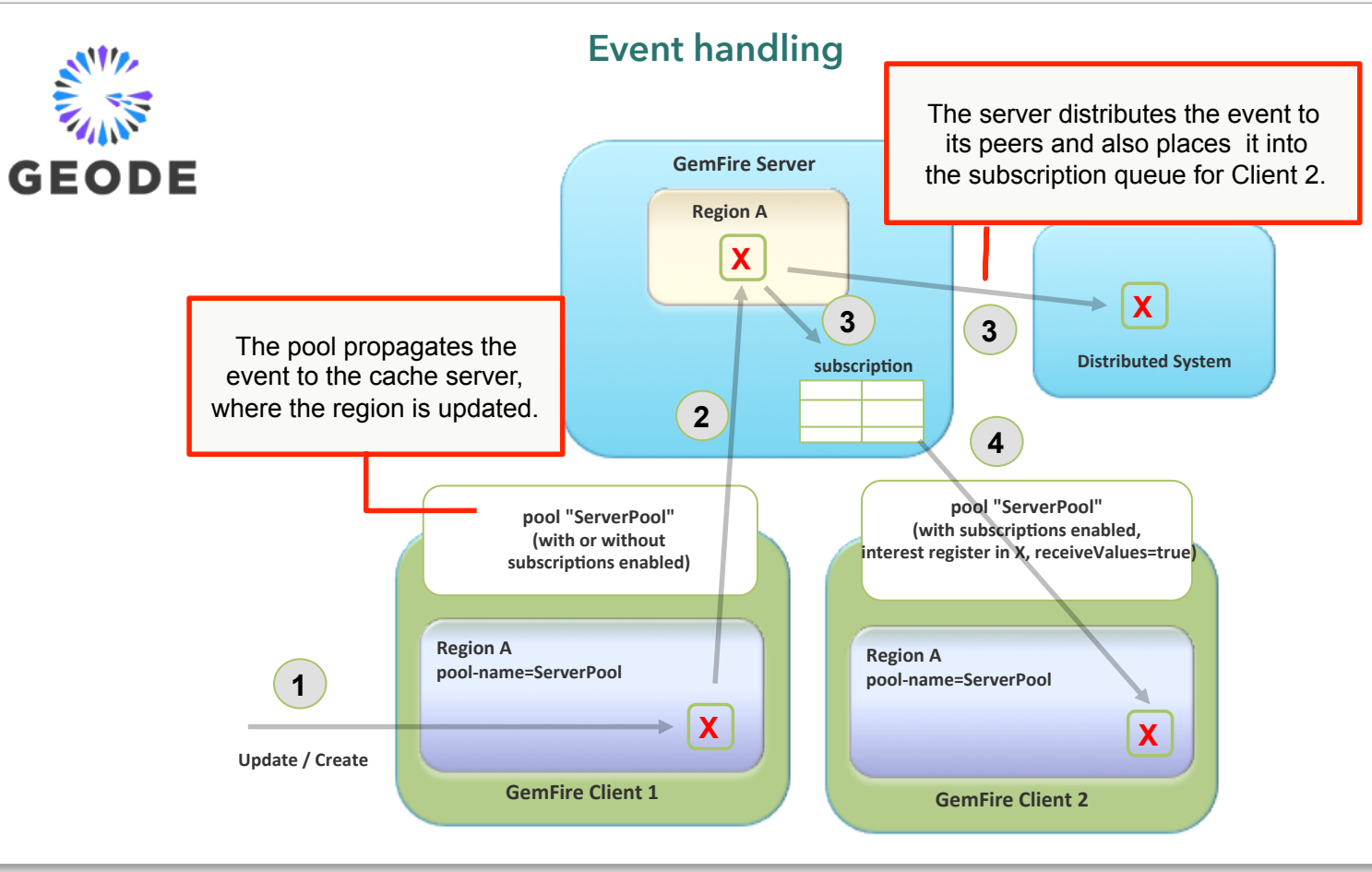
Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable
Cloud-Native

Geode client-server architecture

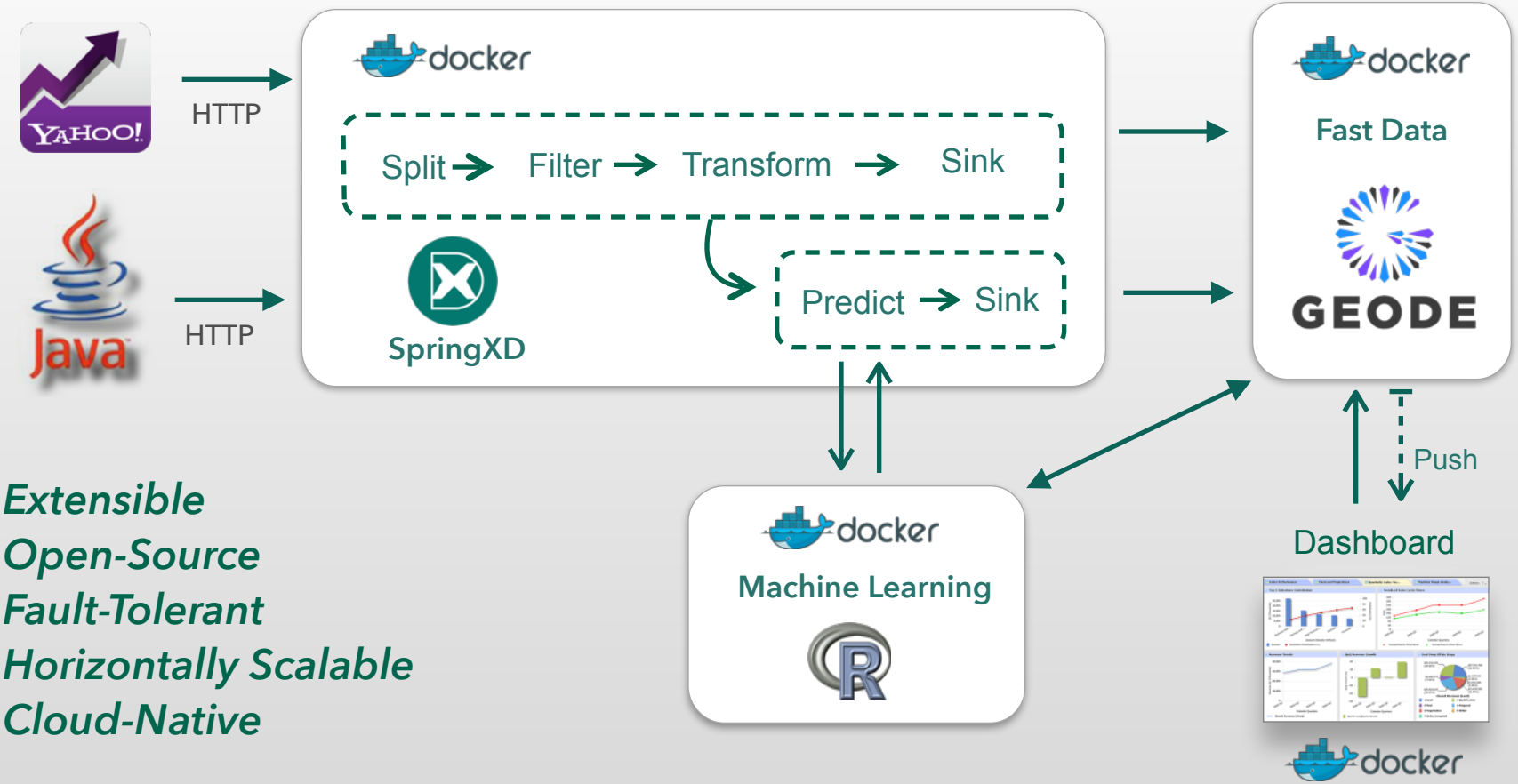


Partitioned Regions



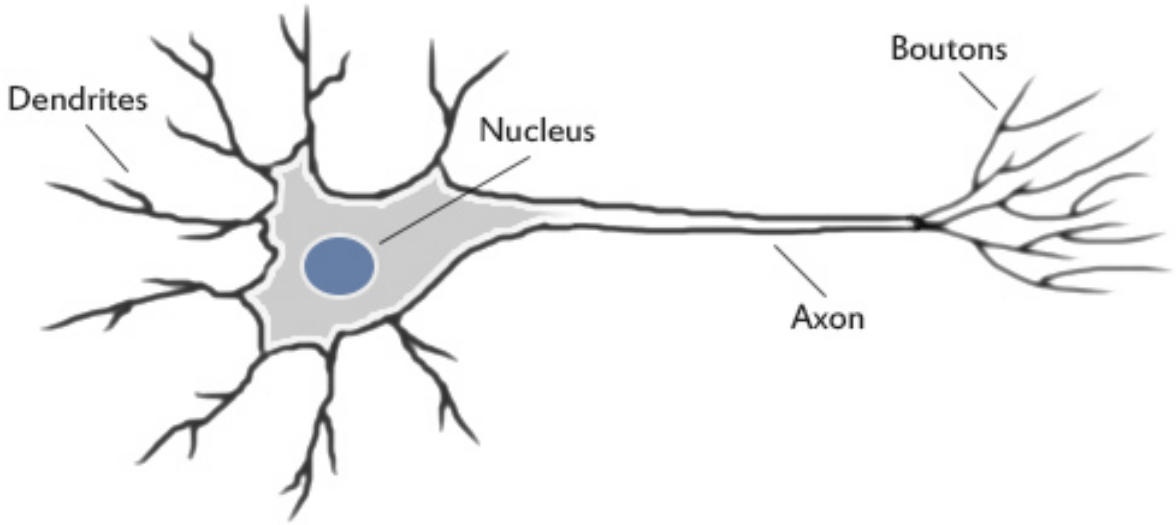

Event handling
GEODE
The server distributes the event to its peers and also places it into the subscription queue for Client 2.
GemFire Server
Region A
X
3
subscription
2
3
X
Distributed System
4
The pool propagates the event to the cache server, where the region is updated.
pool "ServerPool"
(with or without subscriptions enabled)
Region A
pool-name=ServerPool
1
Update / Create
X
GemFire Client 1
pool "ServerPool"
(with subscriptions enabled,
interest register in X, receiveValues=true)
Region A
pool-name=ServerPool
X
GemFire Client 2

Demo Architecture

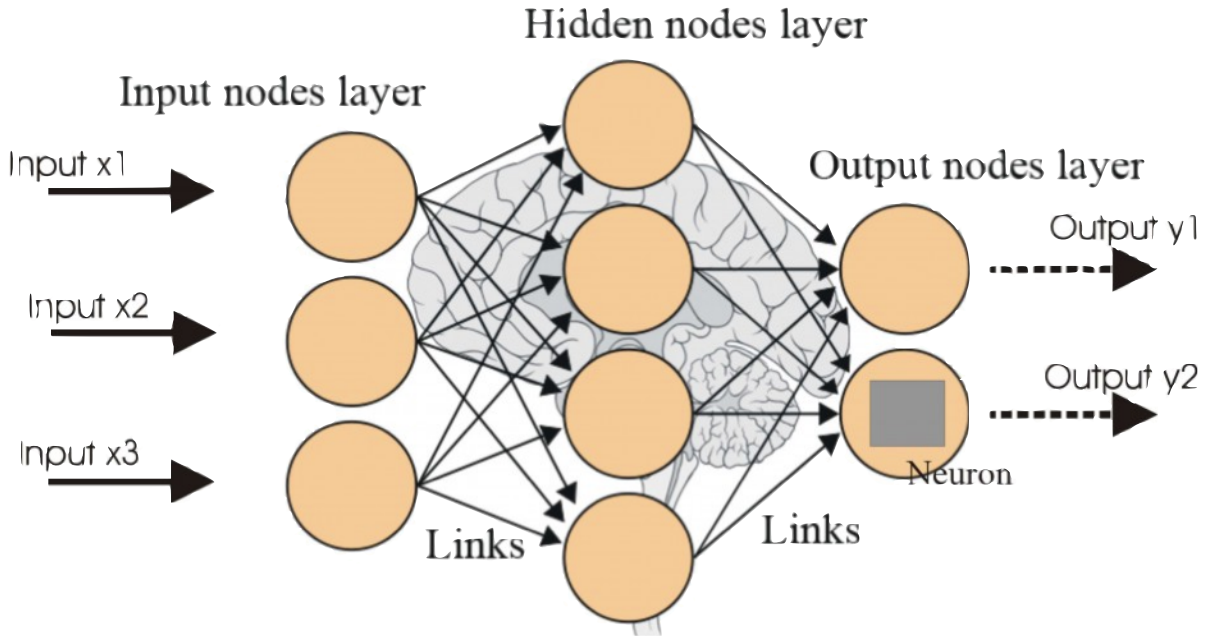


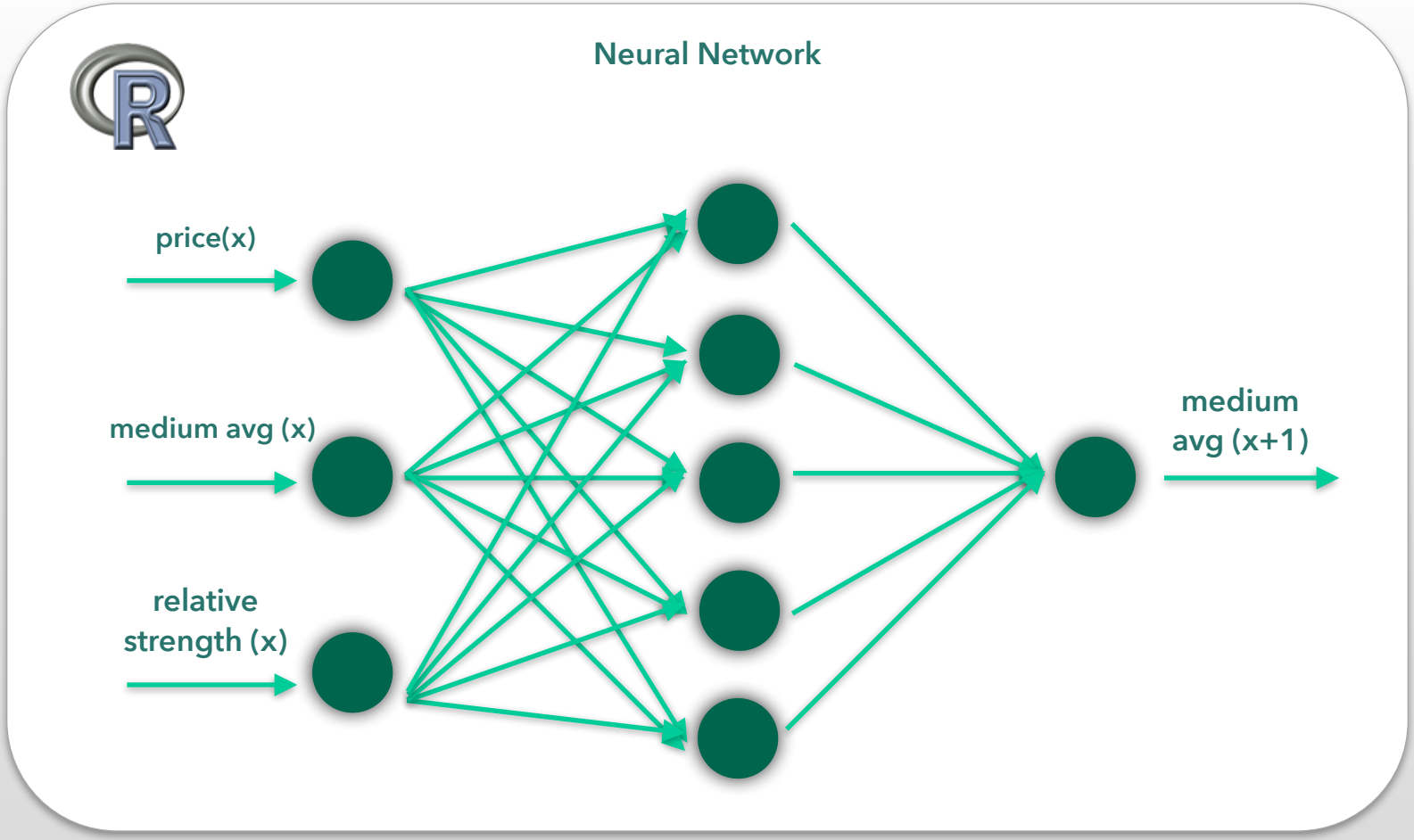
Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable
Cloud-Native

Neural Networks



Neural Networks

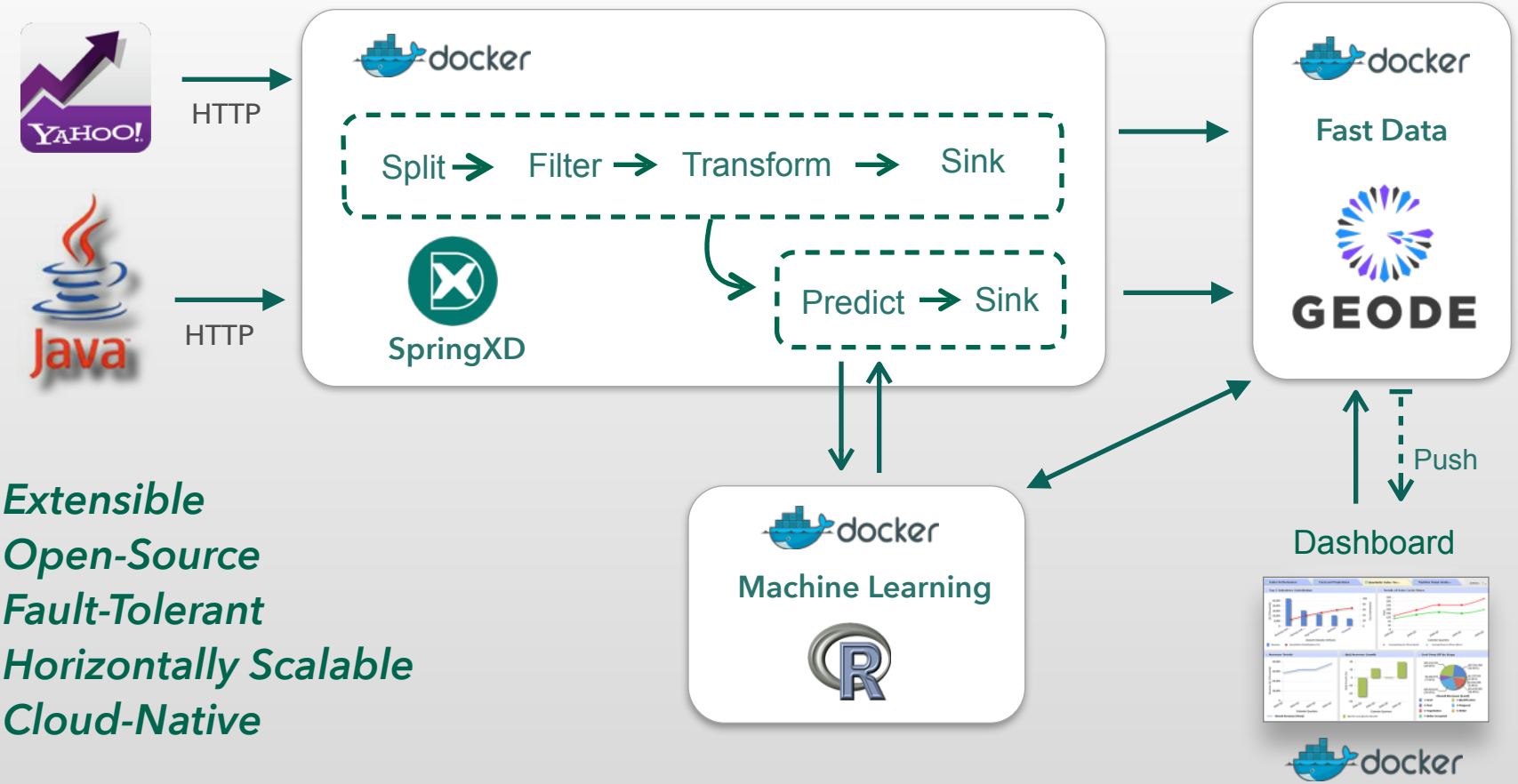



Neural Network
price(x)
medium avg (x)
relative strength (x)
medium avg (x+1)

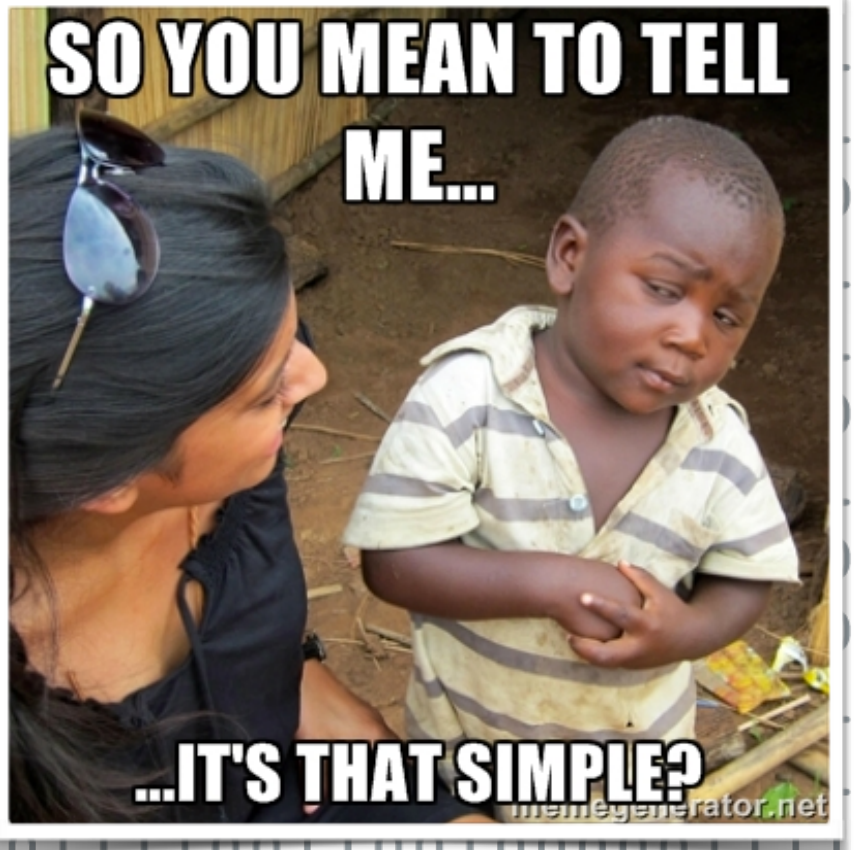
Neural Network

```
# R sample source code for NN using RSNNS project
# http://cran.r-project.org/web/packages/RSNNS/
# (…)

inputs <- techIndicators[,inputColumns(techIndicators)]
targets <- techIndicators[,outputColumns(techIndicators)]
patterns <- splitForTrainingAndTest(inputs, targets, ratio = 0.15)

model <- jordan(patterns$inputsTrain, patterns$targetsTrain,
+     size = c(8), learnFuncParams = c(0.1), maxit = 5000,
+     inputsTest = patterns$inputsTest, targetsTest = patterns$targetsTest,
+     linOut = FALSE)
```

Demo Architecture



Extensible
Open-Source
Fault-Tolerant
Horizontally Scalable
Cloud-Native



Pivotal

EURUSD,M5 1.16118 1.16118 1.16098 1.16114

WWW.ALTREDO.COM

BINARY OPTIONS TRADING SIGNALS

CALL

PUT

Start Time: 2015.01.19 08:10:00

Expire Time: 2015.01.19 08:15:00

Last Close: 1.16121

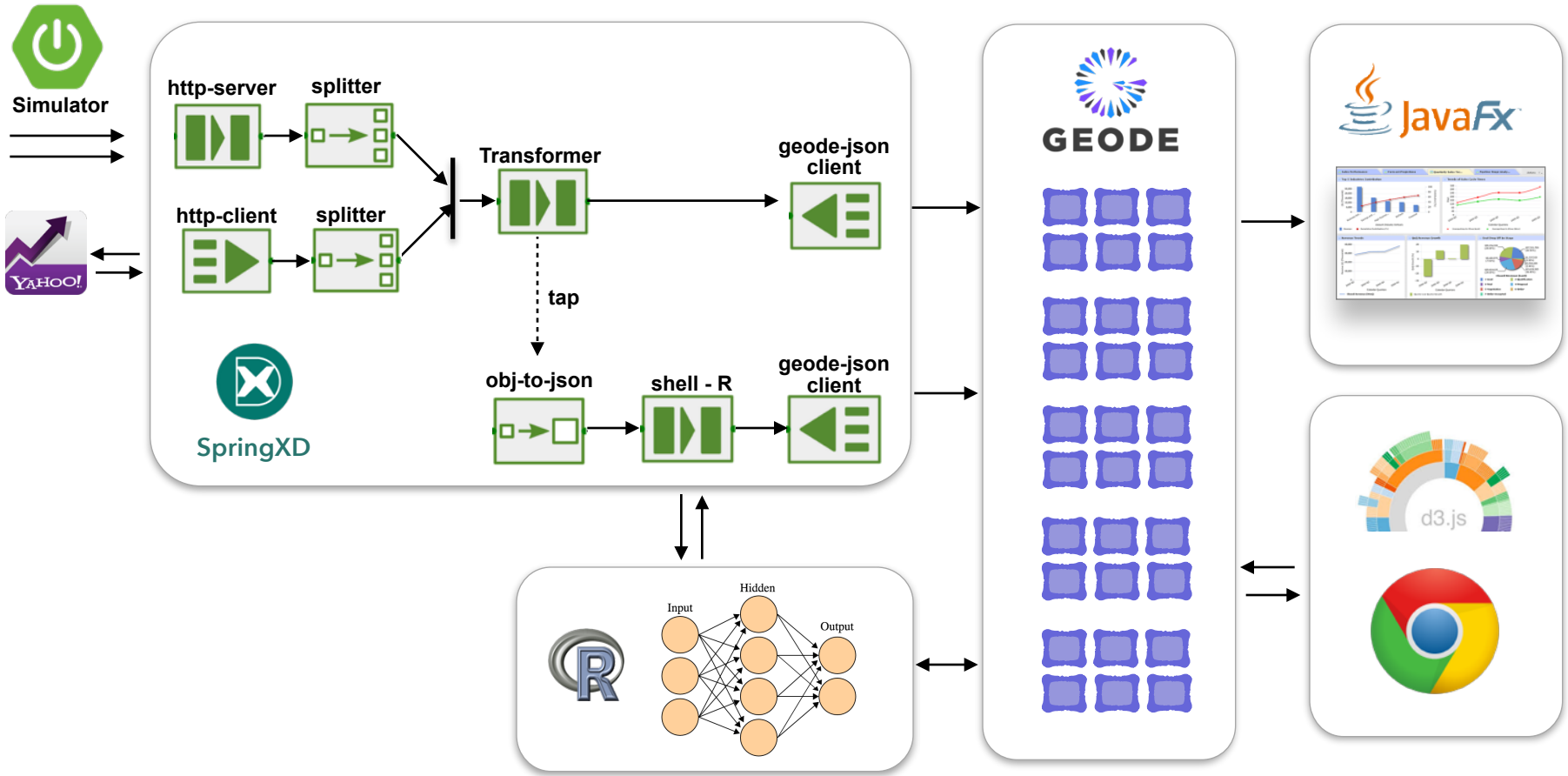
Predicted Close Direction: Down

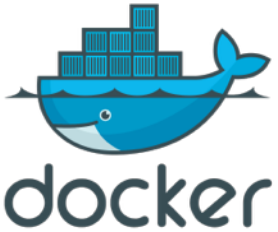
Accuracy: 70.7%

Demo Time

19 Jan 2015 | 19 Jan 05:40 | 19 Jan 05:50 | 19 Jan 06:00 | 19 Jan 06:10 | 19 Jan 06:20 | 19 Jan 06:30 | 19 Jan 06:40 | 19 Jan 06:50 | 19 Jan 07:00 | 19 Jan 07:10 | 19 Jan 07:20 | 19 Jan 07:30 | 19 Jan 07:40 | 19 Jan 07:50 | 19 Jan 08:00 | 19 Jan 08:10

GBPUSD,M1 | EURUSD,M5


Simulator
YAHOO!
http-server
splitter
http-client
splitter
Transformer
geode-json client
tap
obj-to-json
shell - R
geode-json client
SpringXD
GEODE
JavaFx
d3.js
Input
Hidden
Output
Pivotal



http://projectgeode.org

http://projects.spring.io/spring-xd

https://registry.hub.docker.com/

http://www.r-project.org


ANY
questions?

A NEW PLATFORM FOR A NEW ERA