

DATA PLATFORMS
2018

Real-Time Decision Engine using Structured Streaming & ML

Garren Staubli

Solution Architect
Blueprint Technologies

Real-Time Decision Engine using Structured Streaming & Machine Learning

Garren Staubli
Solution Architect
@gstaubli



Blueprint™

bpcs.com

Presentation + Resources: garrens.com/streamingml

Am I qualified?



Presentation + Resources: garrens.com/streamingml

Am I qualified?

Maybe?

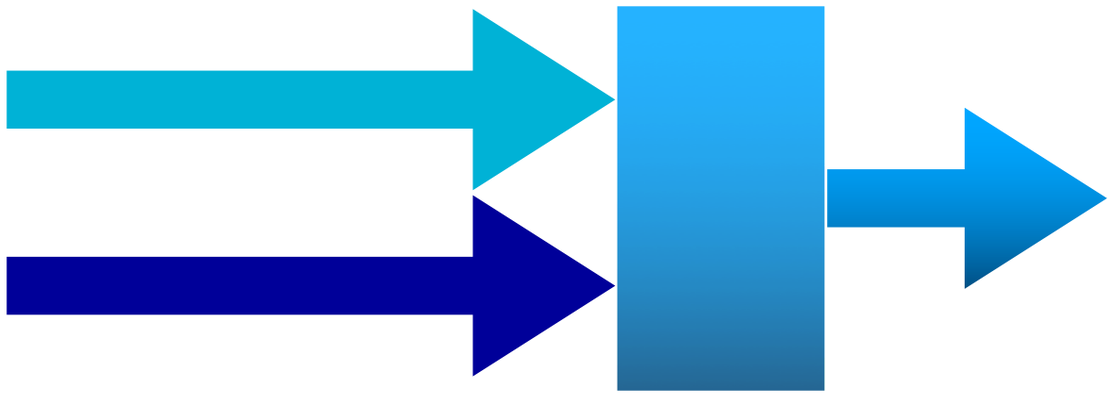
- Working with Spark since 2015
- Hive, Pig and MapReduce (2012+)
- Real-time big data reporting using Spark/Impala/CDH
- Real-time Decision Engine using Spark Structured Streaming + ML

50+ answers on stackoverflow for Spark



Presentation + Resources: garrens.com/streamingml

SPOALL | Objectives



Spark 2.3 Streaming Features

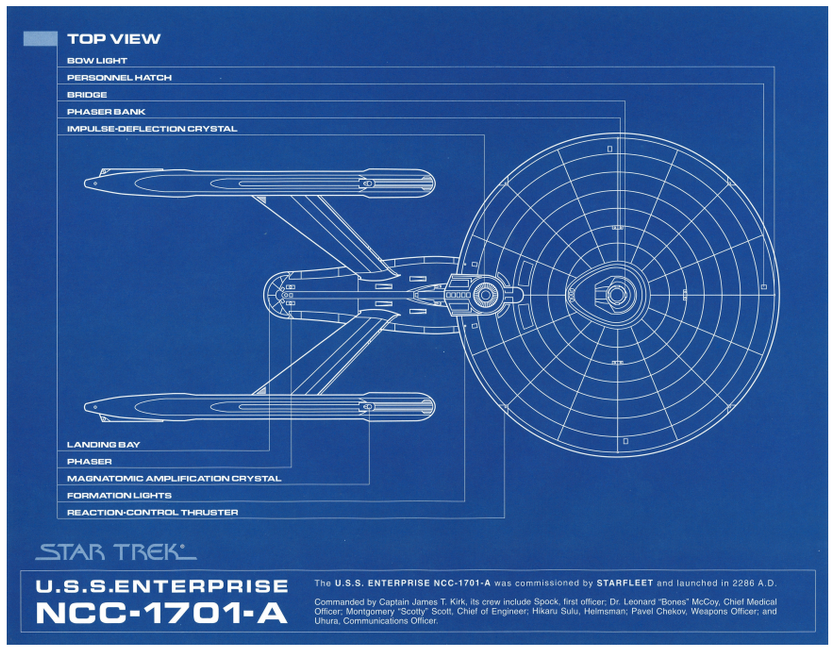
Problem Definition



Options for Solutions

Architecture Overview

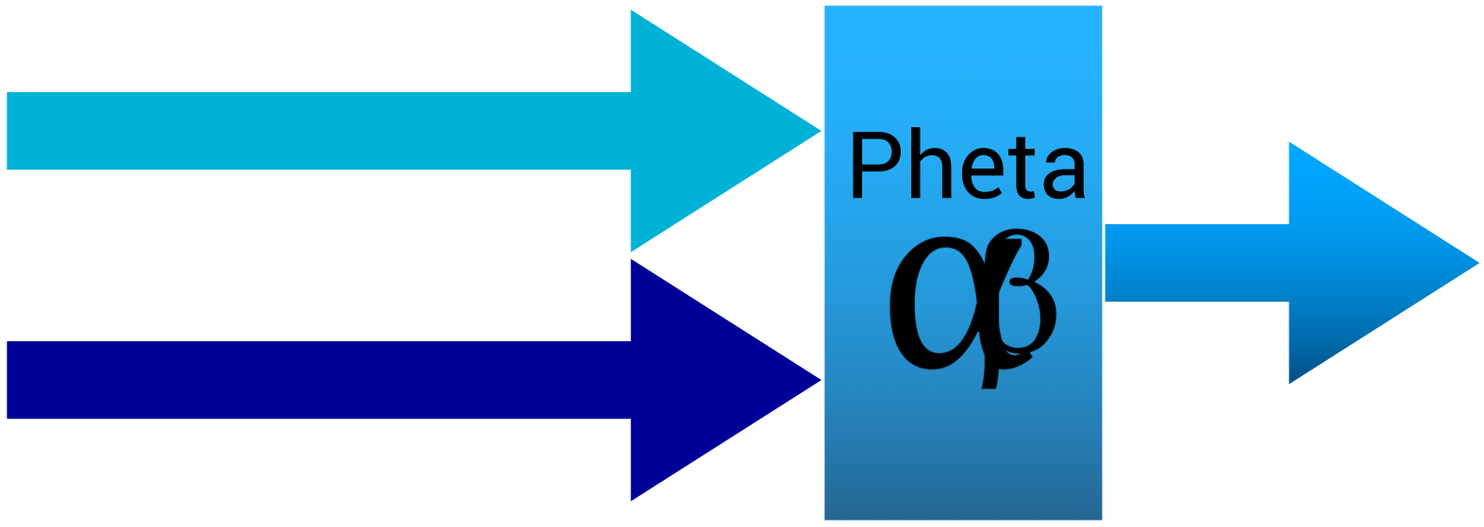
Lessons Learned



Presentation + Resources: garrens.com/streamingml

SPOALL | Spark 2.3+ Streaming Features

Stream-Stream Joins
- Buffered
- Late Events
- Same Join Semantics



Continuous Processing



Before CP
(micro-batch)

After CP
(ms latency)



Presentation + Resources: garrens.com/streamingml

SPOALL | Problem Definition

User



55%

30%

15%

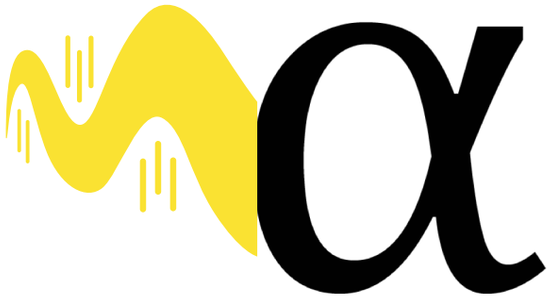
Search Results

OPTION 1

OPTION 2

OPTION 3

Streaming Logs





Presentation + Resources: garrens.com/streamingml

SPOALL | Options for Solutions

Batch Rule Generation

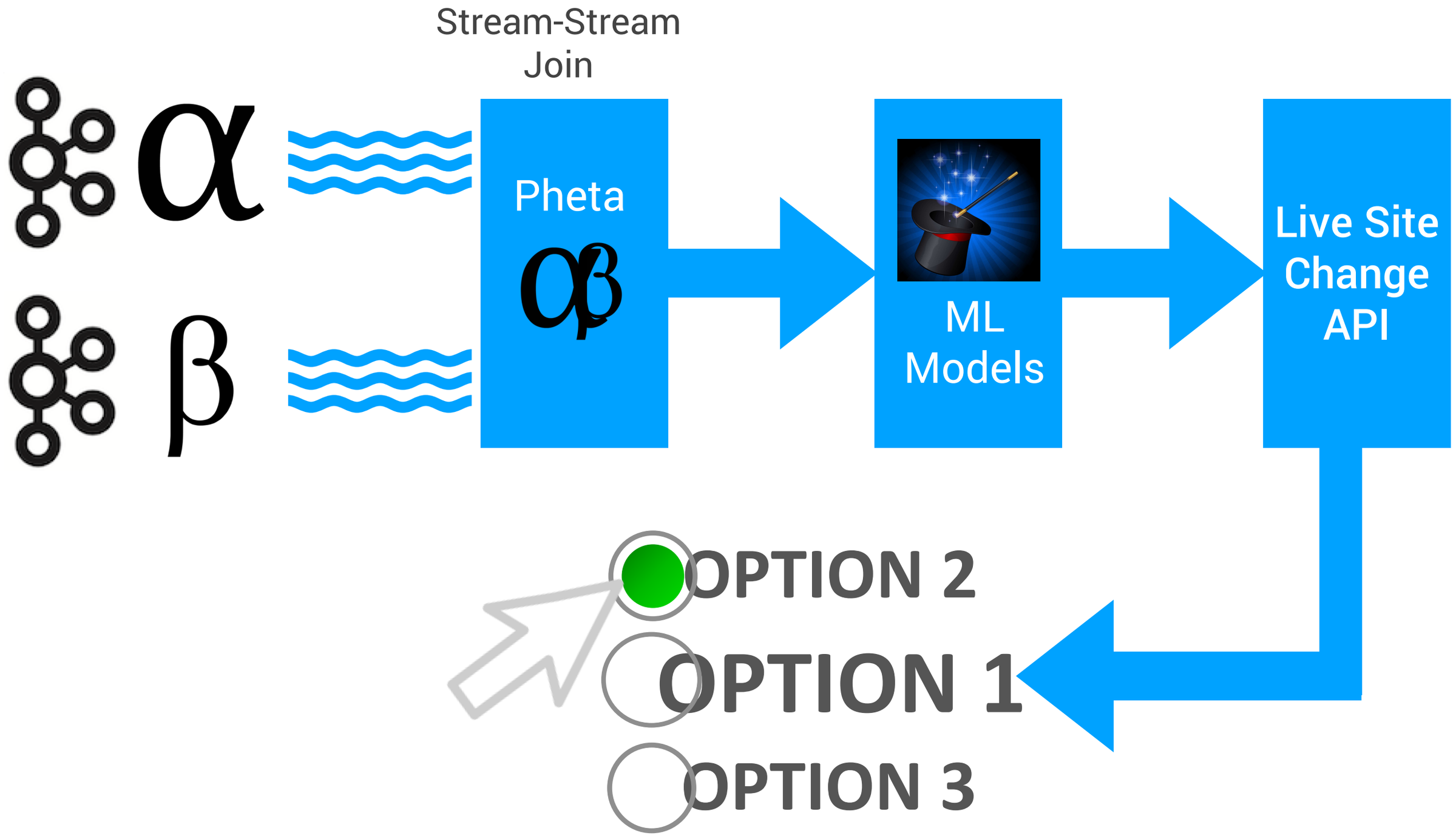
Pros	Cons
Straightforward schedule-based deployment	Delayed Response
Simple Architecture	Reactive
Low cost	Lesser Impact

Streaming Decision Engine

Pros	Cons
Real-time adjustments	High durability required
Proactive	More Complex Architecture
Impactful	24/7 monitoring

Presentation + Resources: garrens.com/streamingml

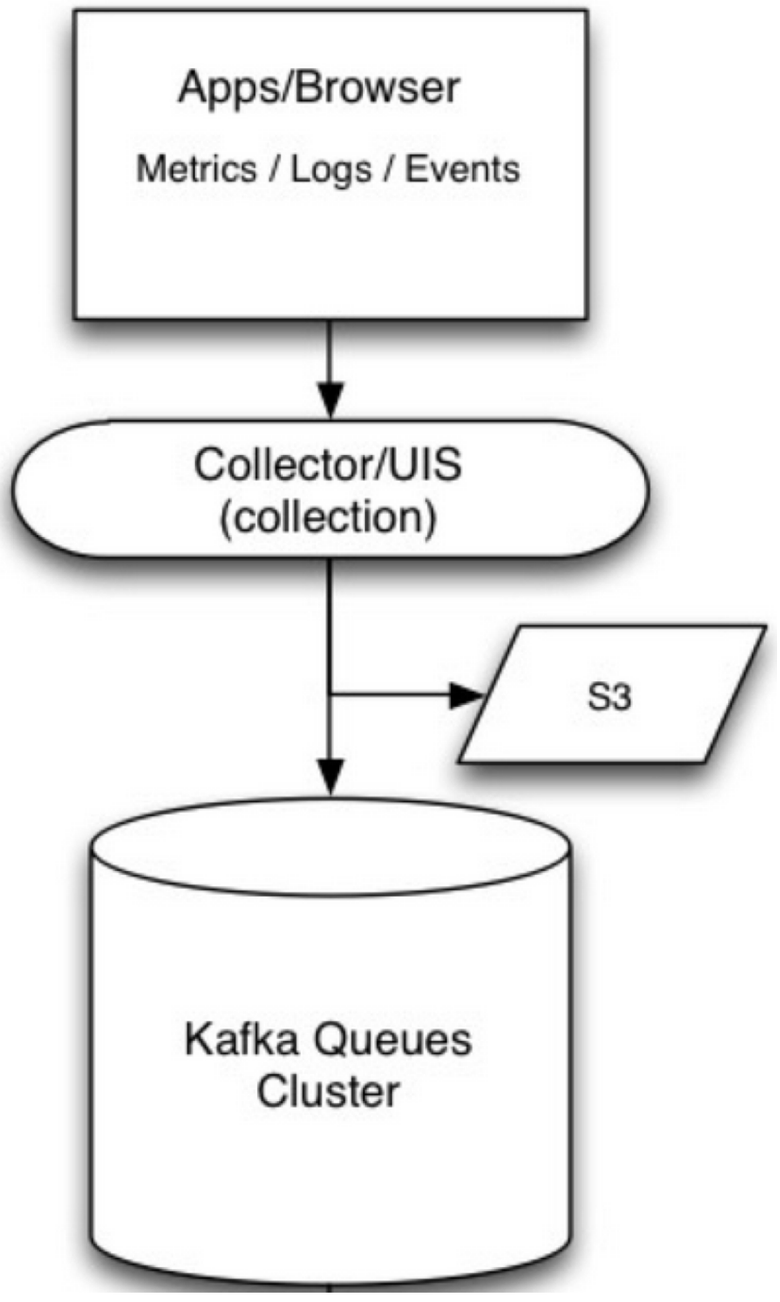
SPOALL | Architecture



Presentation + Resources: garrens.com/streamingml

SPOALL | Architecture

EXTENDED



S3	Kafka
Partitioned by Date + time	α 48 partitions / 2 replicas β 24 partitions / 2 replicas
<50KB gzipped JSON Files 1M+ files/day	Streaming JSON

Presentation + Resources: garrens.com/streamingml

SPOALL | Architecture






```
{
  "id": "0001",
  "type": "donut",
  "name": "Cake",
  "ppu": 0.55,
  "batters": {
    "batter": [
      {
        "id": "1001",
        "type": "Regular"
      },
      {
        "id": "1002",
        "type": "Chocolate"
      },
      {
        "id": "1003",
        "type": "Blueberry"
      },
      {
        "id": "1004",
        "type": "Devil's Food"
      }
    ]
  }
}
```

Flatten

Type	Name	PPU	Batter_id	Batter_type
Donut	Cake	0.55	1001	Regular
Donut	Cake	0.55	1002	Chocolate
Donut	Cake	0.55	1003	Blueberry
Donut	Cake	0.55	1004	Devil’s Food

Presentation + Resources: garrens.com/streamingml







Scala Code Coming…

Presentation + Resources: garrens.com/streamingml

SPOALL | Architecture





Scala Code!

```scala
case class Batter(
    id: String,
    type: String
)

case class PastryRecord(
    id: String,
    type: String,
    name: String,
    ppu: Double,
    batters: Seq[Batter]
)
```

Nested Case classes

```scala
import org.apache.spark.sql.Encoders
import org.apache.spark.sql.{functions => F}
1 val prSchema = Encoders.product[PastryRecord].schema
val prDataset = df.selectExpr("CAST(value AS STRING) AS json_str")
2   .select(F.from_json(F.col("json_str"), prSchema).alias("json_parsed"))
3   .select("json_parsed.*")
4   .as[PastryRecord]
```

Make top-level

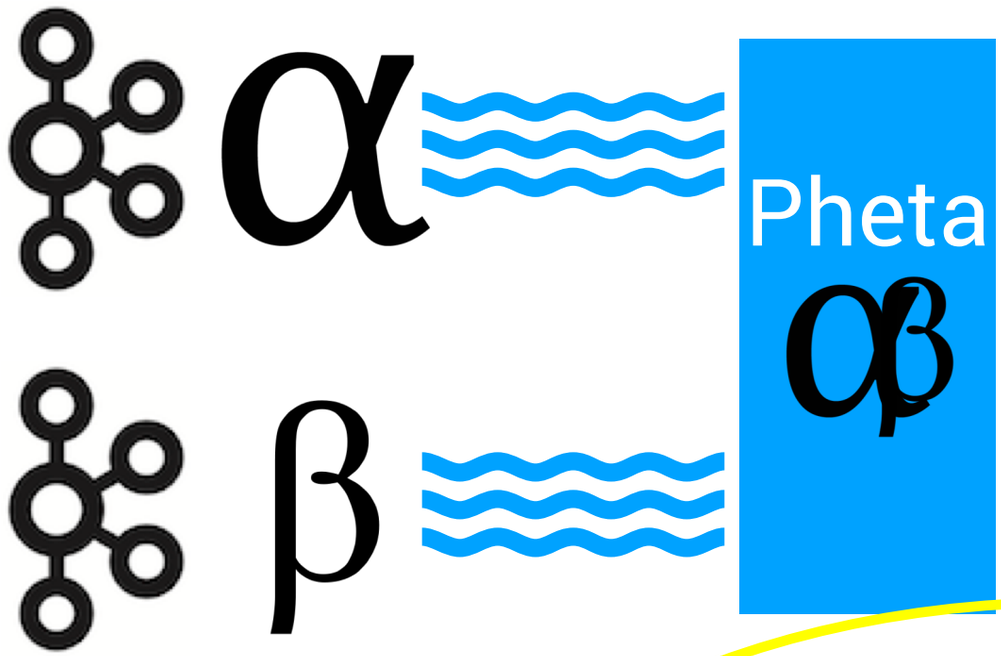
Convert to Dataset[T]

```json
{
  "id": "0001",
  "type": "donut",
  "name": "Cake",
  "ppu": 0.55,
  "batters": {
    "batter": [
      {
        "id": "1001",
        "type": "Regular"
      },
      {
        "id": "1002",
        "type": "Chocolate"
      },
      {
        "id": "1003",
        "type": "Blueberry"
      },
      {
        "id": "1004",
        "type": "Devil's Food"
      }
    ]
  }
}
```

Presentation + Resources: garrens.com/streamingml

EXTEN PART IV TQUEL X

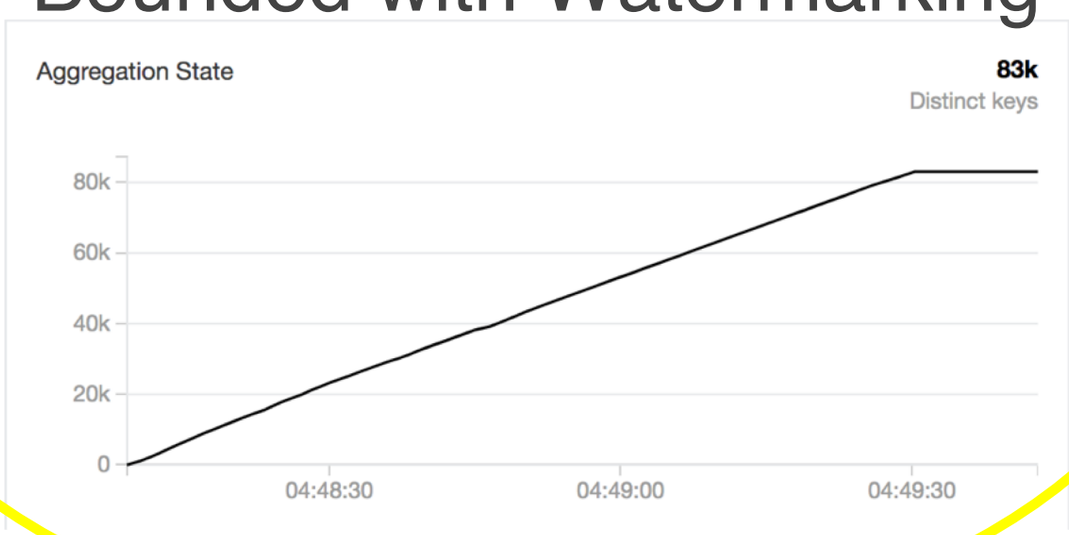
Stream-Stream Join



Unbounded



Bounded with Watermarking



Presentation + Resources: garrens.com/streamingml



Machine Learning

- Spark.ML | DataFrame/Dataset-based
- XGBoost &/or RF Regression
- Trained offline
- Served directly in stream
- Predicting future value



Presentation + Resources: garrens.com/streamingml



Taking Action

```
datasetOfString.writeStream.foreach(new ForeachWriter[String] {

      def open(partitionId: Long, version: Long): Boolean = {
        // open connection
      }

      def process(record: String) = {
        // write string to connection
      }

      def close(errorOrNull: Throwable): Unit = {
        // close the connection
      }
    })
}}}
```

Presentation + Resources: garrens.com/streamingml

SPOALL | Lessons Learned

Infamous Small Files Problem





Parsing JSON

- Nested Case classes
- Using Encoders.product
- Converting to Dataset[T]

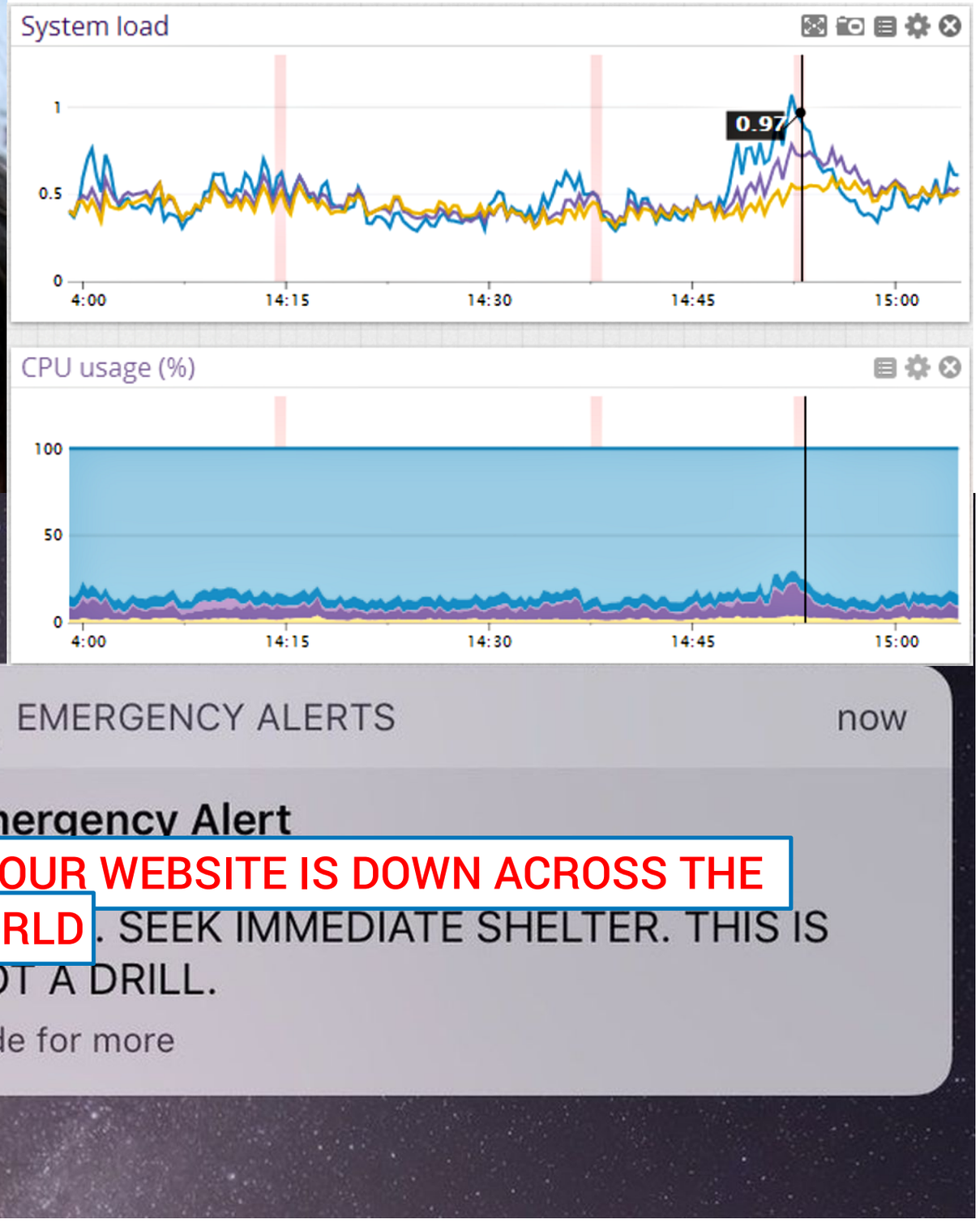
Presentation + Resources: garrens.com/streamingml

SPOALL | Lessons Learned

ENCORE

Ops





Always ON

✓Durable

✓Monitored

✓Tested

Presentation + Resources: garrens.com/streamingml

Blueprint™



- 2015 #1
- 2016 #1

FASTEST-GROWING PRIVATE COMPANIES

Seattle Business 100 WASHINGTON'S BEST COMPANIES TO WORK FOR

- 2014
- 2015
- 2016

BEST WASHINGTON'S BEST WORKPLACES

- 2016

PUGET SOUND BUSINESS JOURNAL 50 EASTSIDE FASTEST-GROWING PRIVATE COMPANIES

- 2014
- 2015 #
- 2016 #
- 2017 #

Inc. 500

- 2015 #37
- 2016 #16
- 2017 #16

19

Blueprint™