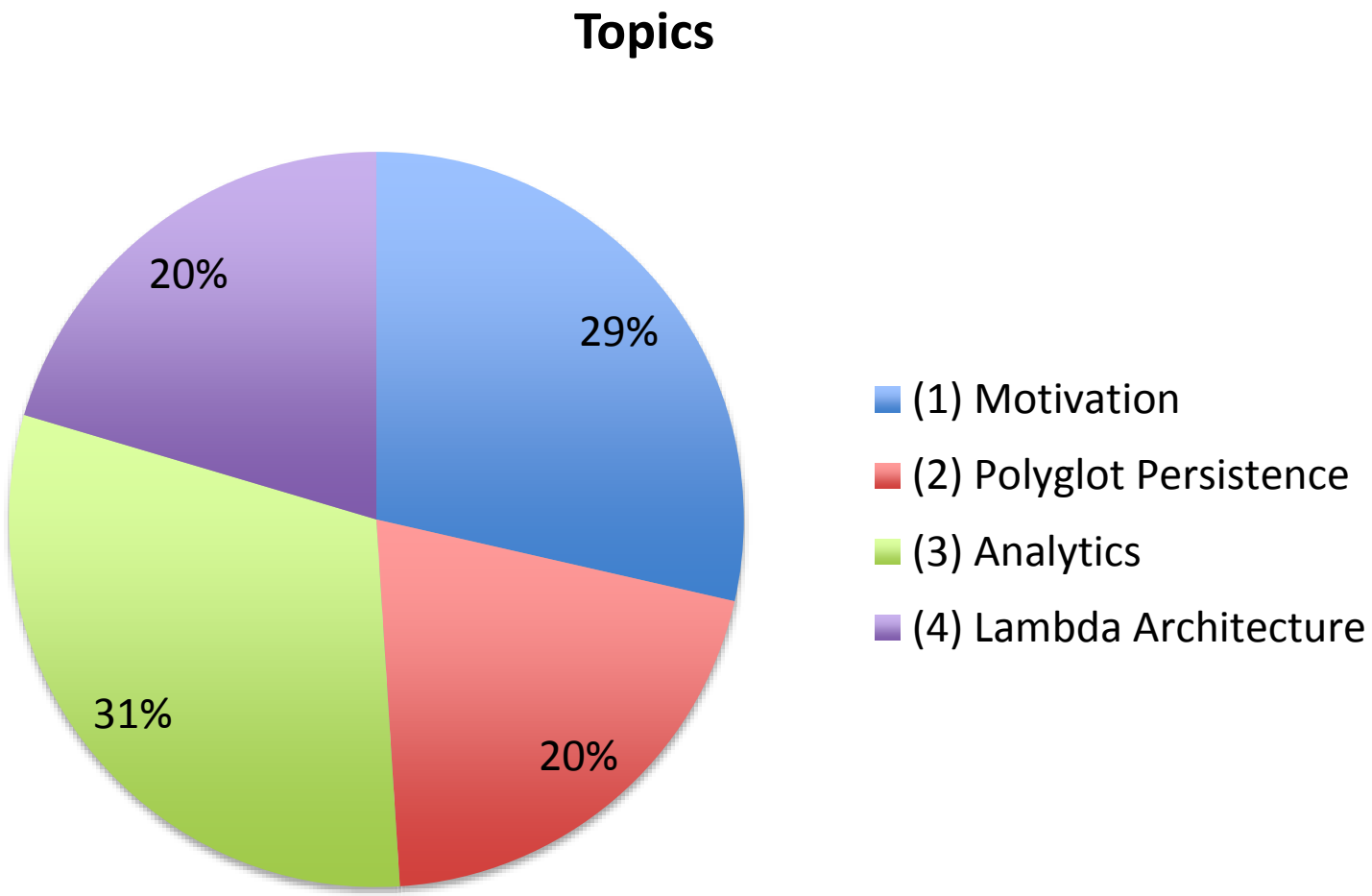
Brian O'Neill, CTO
boneill@healthmarketscience.com
@boneill42

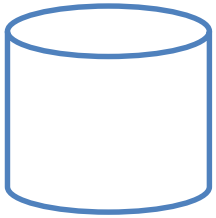
Health Market Science
The Science of Better Results

Re-envisioning the Lambda Architecture: Web Services & Real-time Analytics w/ Storm and Cassandra

Talk Breakdown

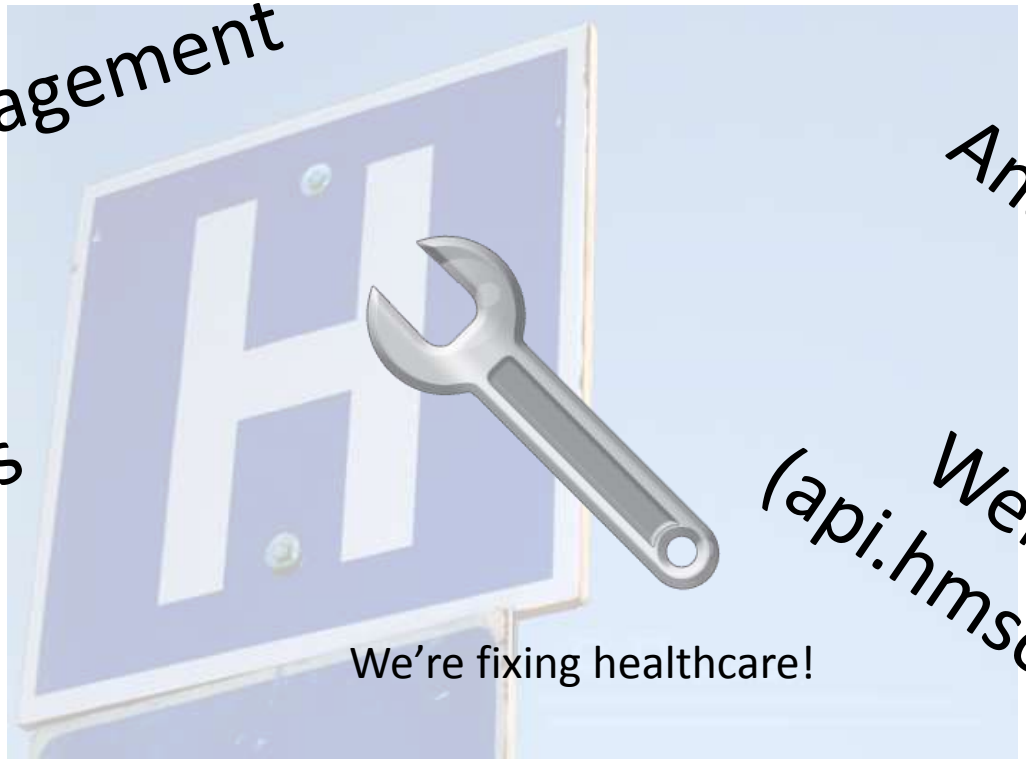


Health Market Science - Then



What we were.

Health Market Science - Now



Data Management

Analytics

Dashboards

Web Services
(api.hmsonline.com)

Intersecting Big Data w/ Healthcare

Data Pipelines

I/O

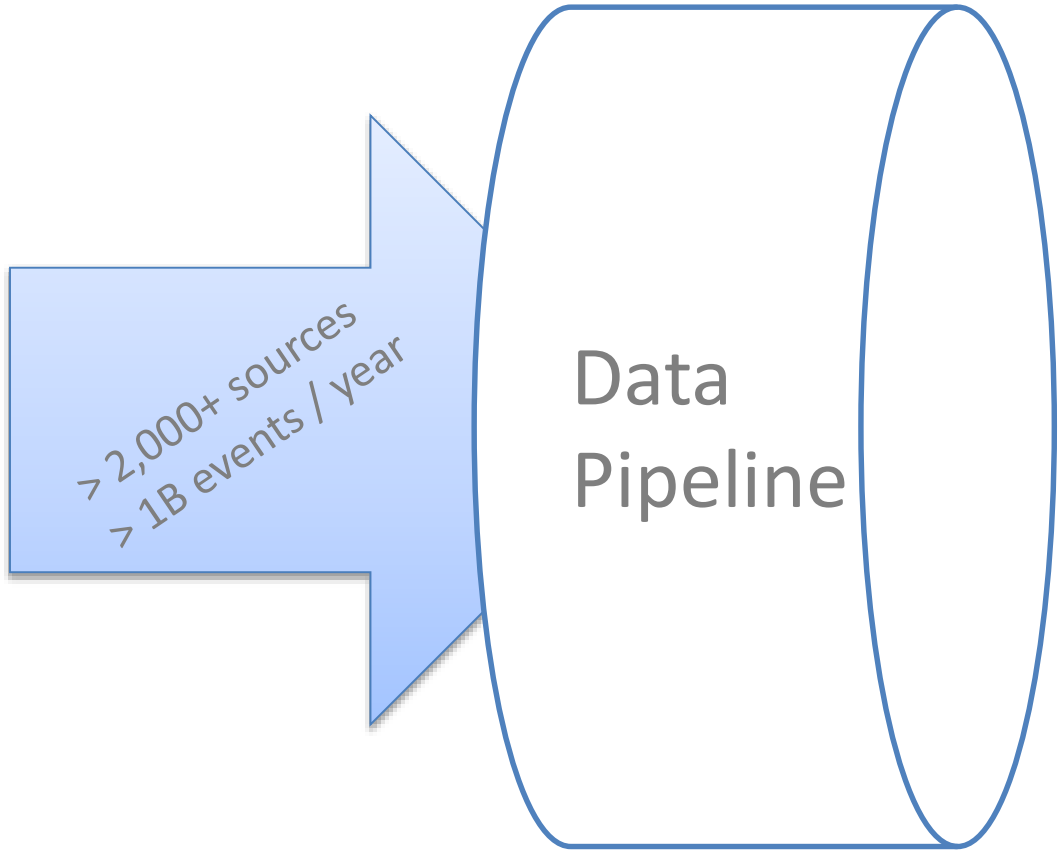
The Input

From government,
state boards, etc.

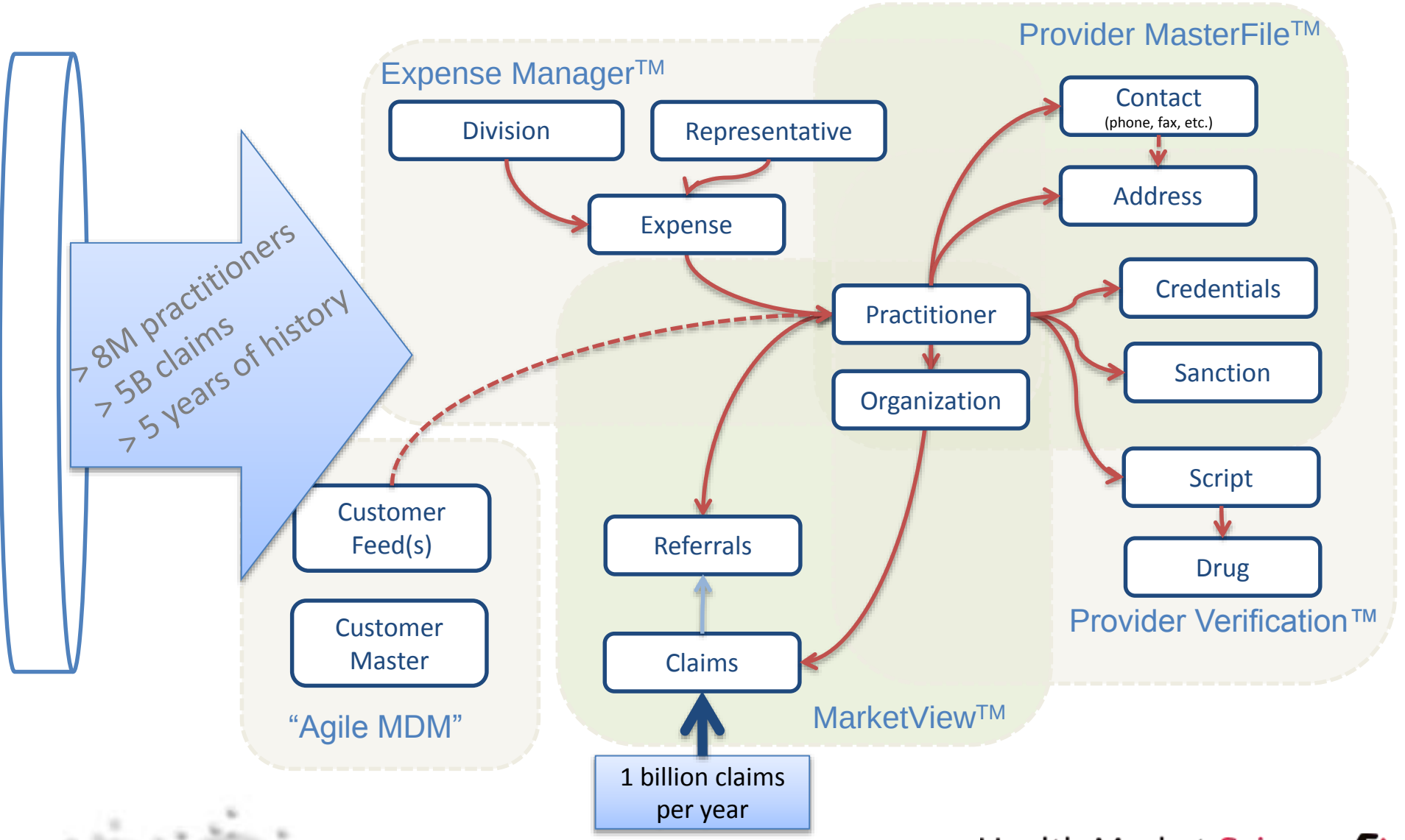
From the internet,
social data,
networks / graphs

From third-parties,
medical claims

From customers,
expenses,
sales data,
beneficiary information,
quality scores



The Output



Health Market Science
The Science of Better Results

Sounds easy

Except...

Incomplete Capture

No foreign keys

Differing schemas

Changing schemas

Conflicting information

Ad-hoc Analysis (is hard)

Point-In-Time Retrieval

Why?

?'s

Compliance & Safety

Is this expense,
Legal?
Compliant?
Reported?

Transparency

Is this doctor,
Licensed?
Sanctioned?
Influential?

Sales Operations

Is this market,
Saturated?
Penetrable?

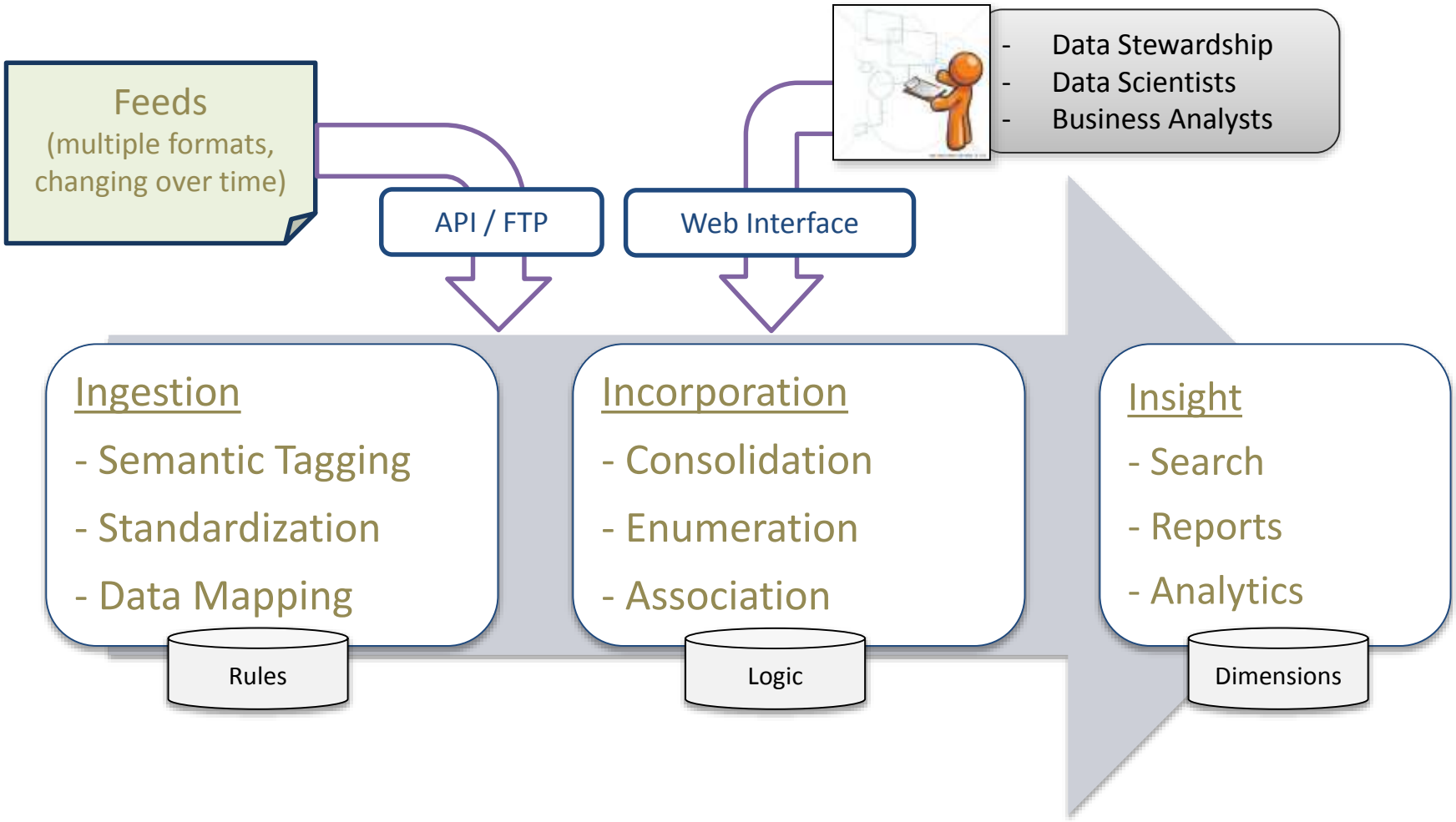
Marketing Optimization

Is this claim,
Fraudulent?
Wasteful?
Abusive?

Cost Control

Health Market Science
The Science of Better Results

Our MDM Pipeline



Our first “Pipeline”

+

Sweet!

Dirt Simple

Lightning Fast

Highly Available

Scalable

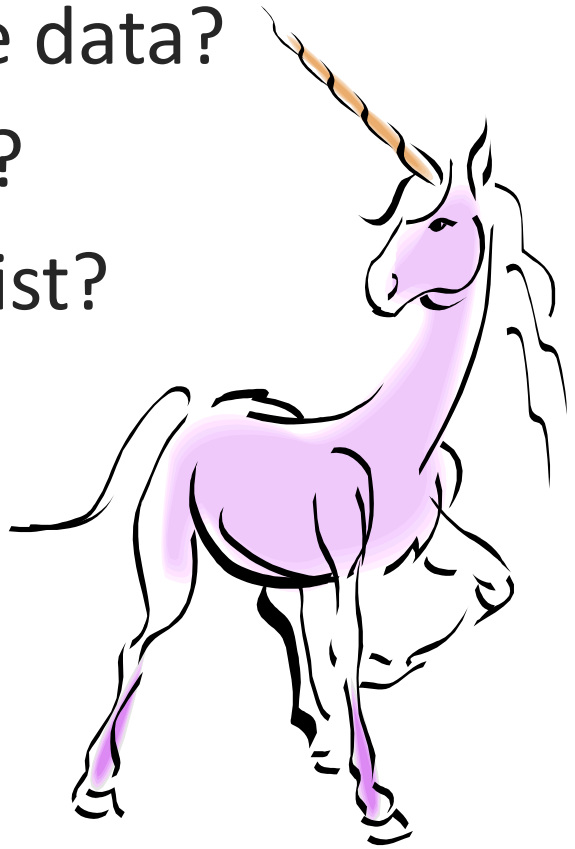
Multi-Datacenter (DR)

Not Sweet.

How do we query the data?

NoSQL Indexes?

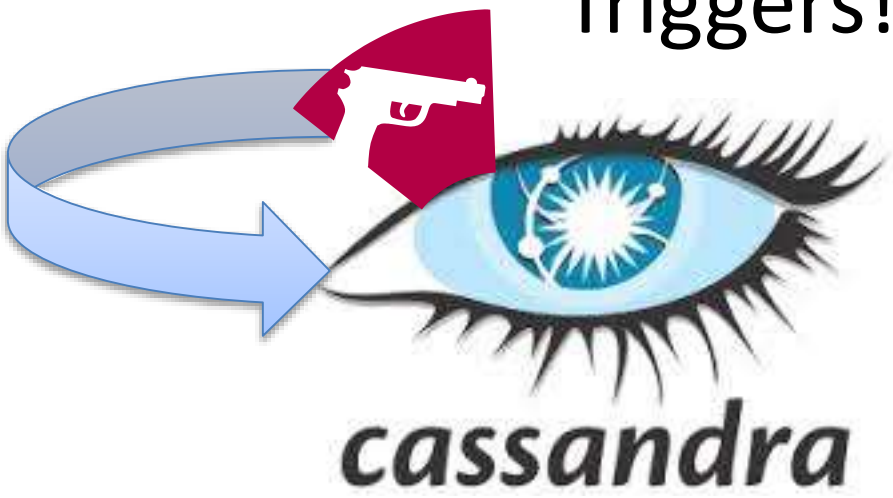
Do such things exist?



Rev. 1 – Wide Rows!

Data model to support your queries.

AOP Triggers!

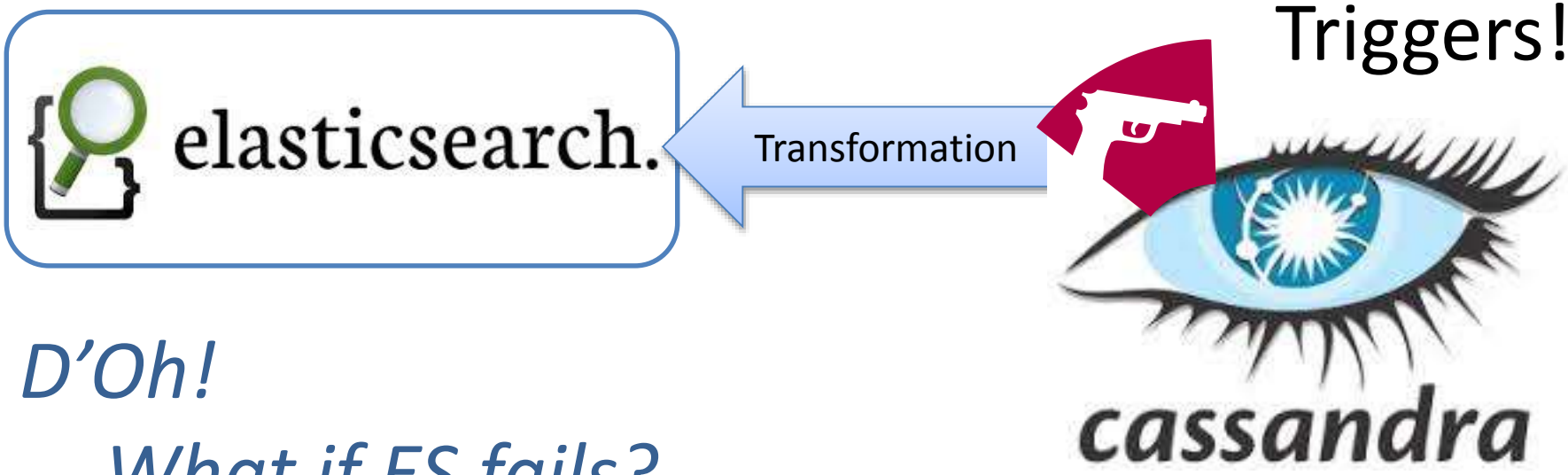


ONC : PA : 19460						
9	7	32	74	99	12	42
$3.50	$7.00	$8.75	$1.00	$4.20	$3.17	$8.88

D'Oh! What about ad hoc?

Rev 2 – Elastic Search!

AOP
Triggers!



D'Oh!

What if ES fails?

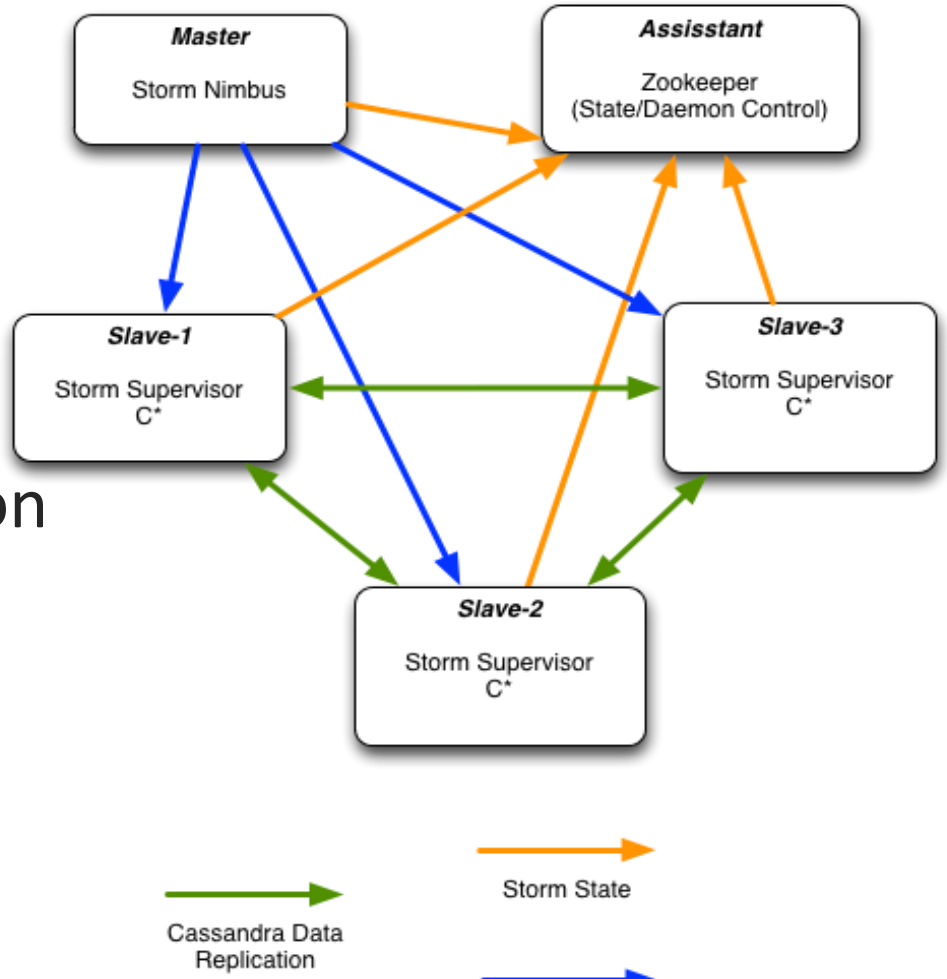
What about schema / type information?

Rev 3 - Apache Storm!

ApacheStorm

Anatomy of a Storm Cluster

- Nimbus
  - Master Node
- Zookeeper
  - Cluster Coordination
- Supervisors
  - Worker Nodes



Storm Primitives

- Streams
  - Unbounded sequence of tuples
- Spouts
  - Stream Sources
- Bolts
  - Unit of Computation
- Topologies
  - Combination of *n* Spouts and *m* Bolts
  - Defines the overall “Computation”

Storm Spouts

- Represents a source (stream) of data
  - Queues (JMS, Kafka, Kestrel, etc.)
  - Twitter Firehose
  - Sensor Data
- Emits “Tuples” (Events) based on source
  - Primary Storm data structure
  - Set of Key-Value pairs

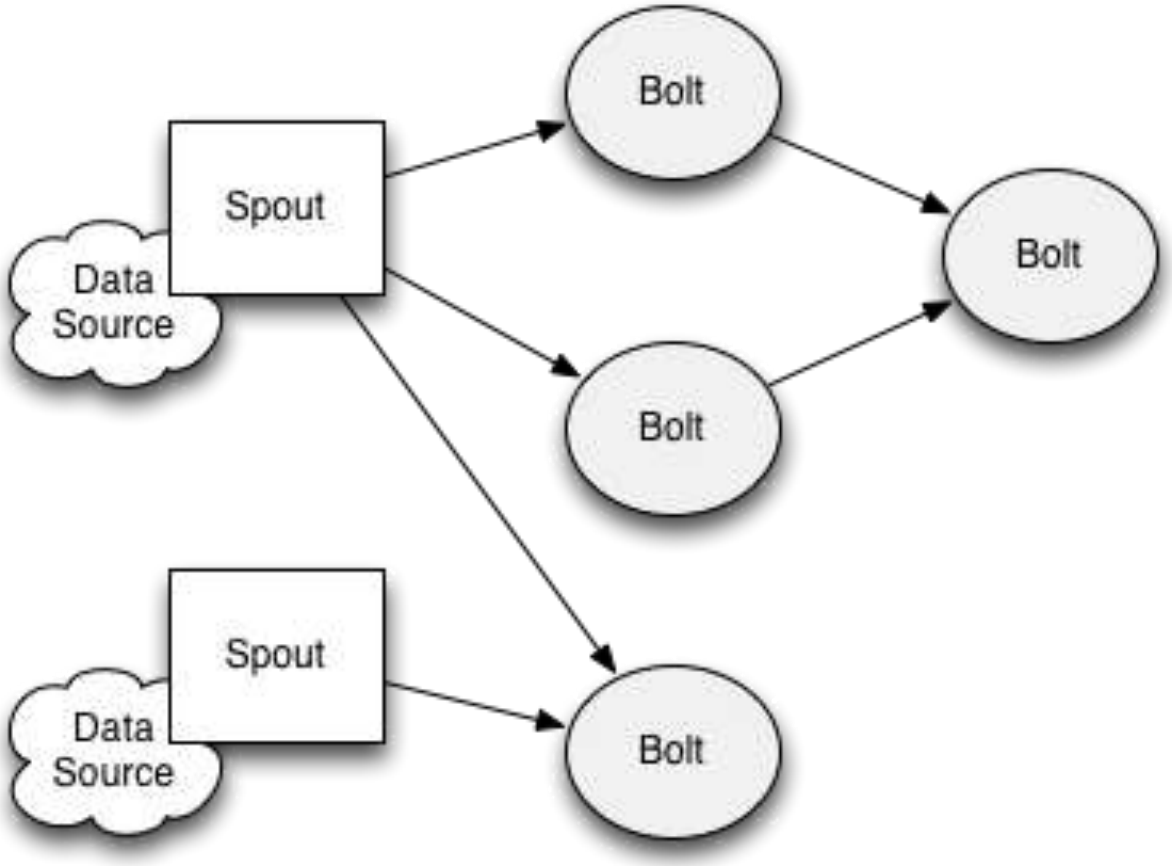
Storm Bolts

- Receive Tuples from Spouts or other Bolts
- Operate on, or React to Data
  - Functions/Filters/Joins/Aggregations
  - Database writes/lookups
- Optionally emit additional Tuples

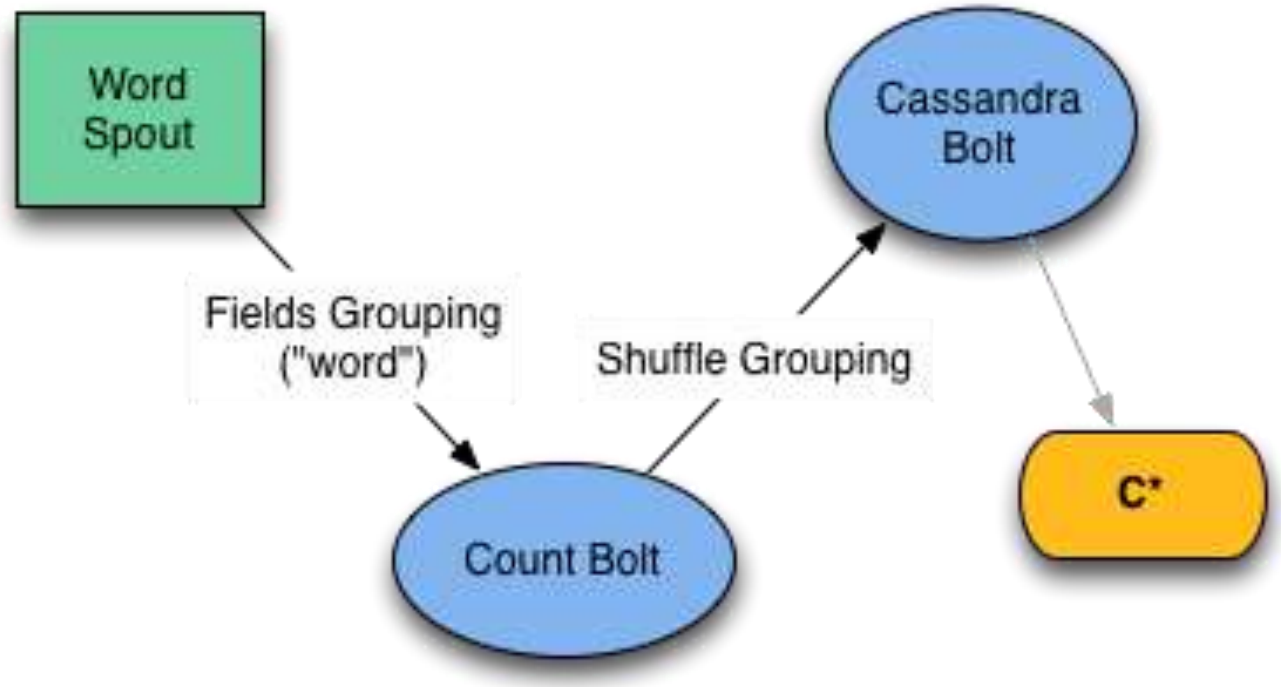
Storm Topologies

- Data flow between spouts and bolts
- Routing of Tuples between spouts/bolts
  - Stream “Groupings”
- Parallelism of Components
- Long-Lived

Storm Topologies



Persistent Word Count



http://github.com/hmsonline/storm-cassandra

NEXT LEVEL : TRIDENT

Health Market Science
The Science of Better Results

Trident

- Part of Storm
- Provides a higher-level abstraction for stream processing
  - Constructs for state management and batching
- Adds additional primitives that abstract away common topological patterns

Trident State

Sequences writes by batch

- Spouts
  - Transactional
    - Batch contents never change
  - Opaque
    - Batch contents can change
- State
  - Transactional
    - Store batch number with counts to maintain sequencing of writes
  - Opaque
    - Store previous value in order to overwrite the current value when contents of a batch change

State Management

Transactional

Last Batch	Value
15	1000

(+59)

Last Batch	Value
16	1059

replay == incorporated already?
(because batch composition is the same)

Opaque

Last Batch	Previous	Current
15	980	1000

(+59)

Last Batch	Previous	Current
16	1000	1059

Batch composition changes! (not guaranteed)

Last Batch	Previous	Current
15	980	1000

(+72)

Last Batch	Previous	Current
16	1000	1072

replay == re-incorporate

BACK TO OUR REGULARLY SCHEDULED TALK

Polyglot Persistence

“The Right Tool for the Job”

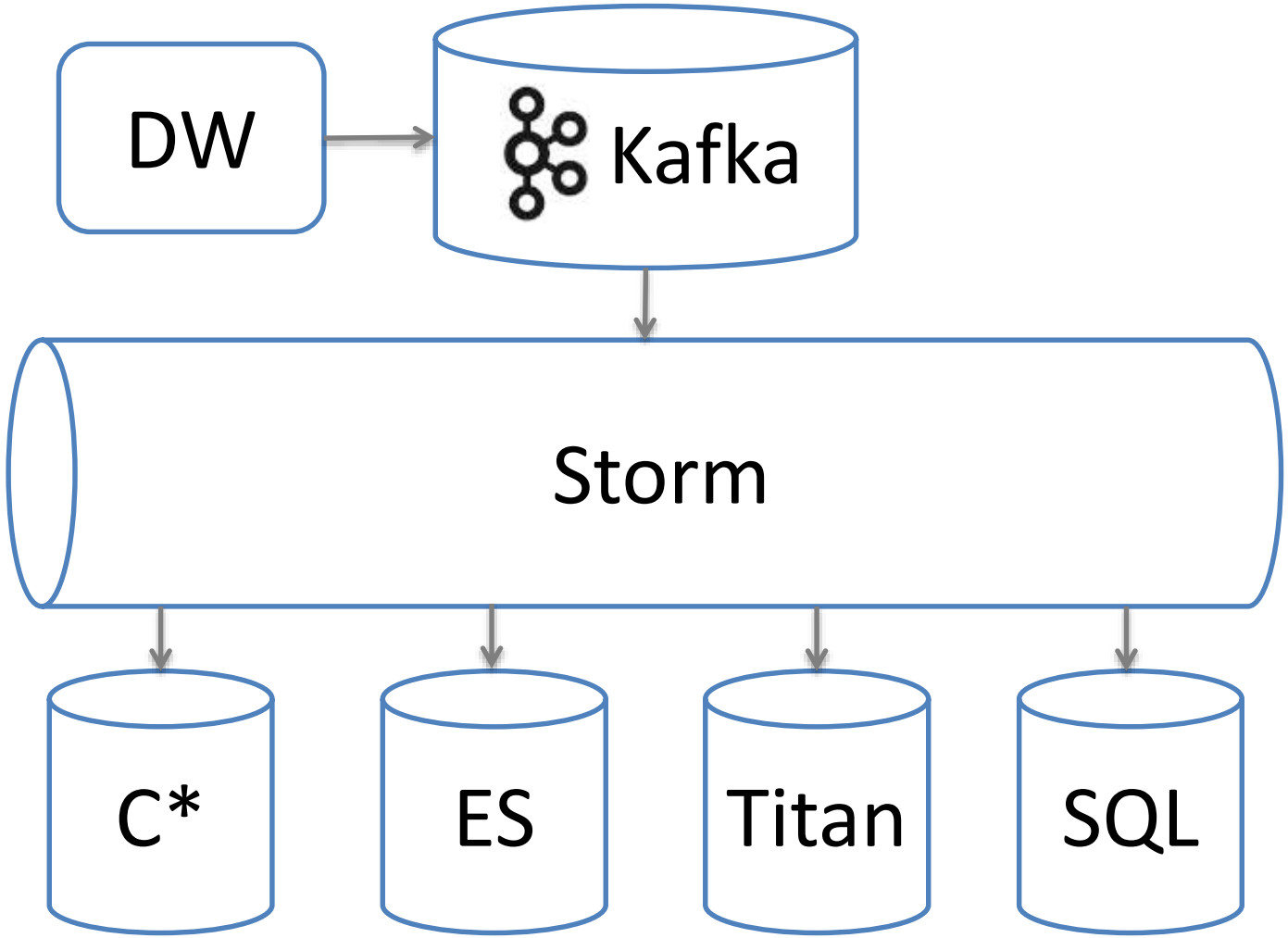
elasticsearch.

ORACLE

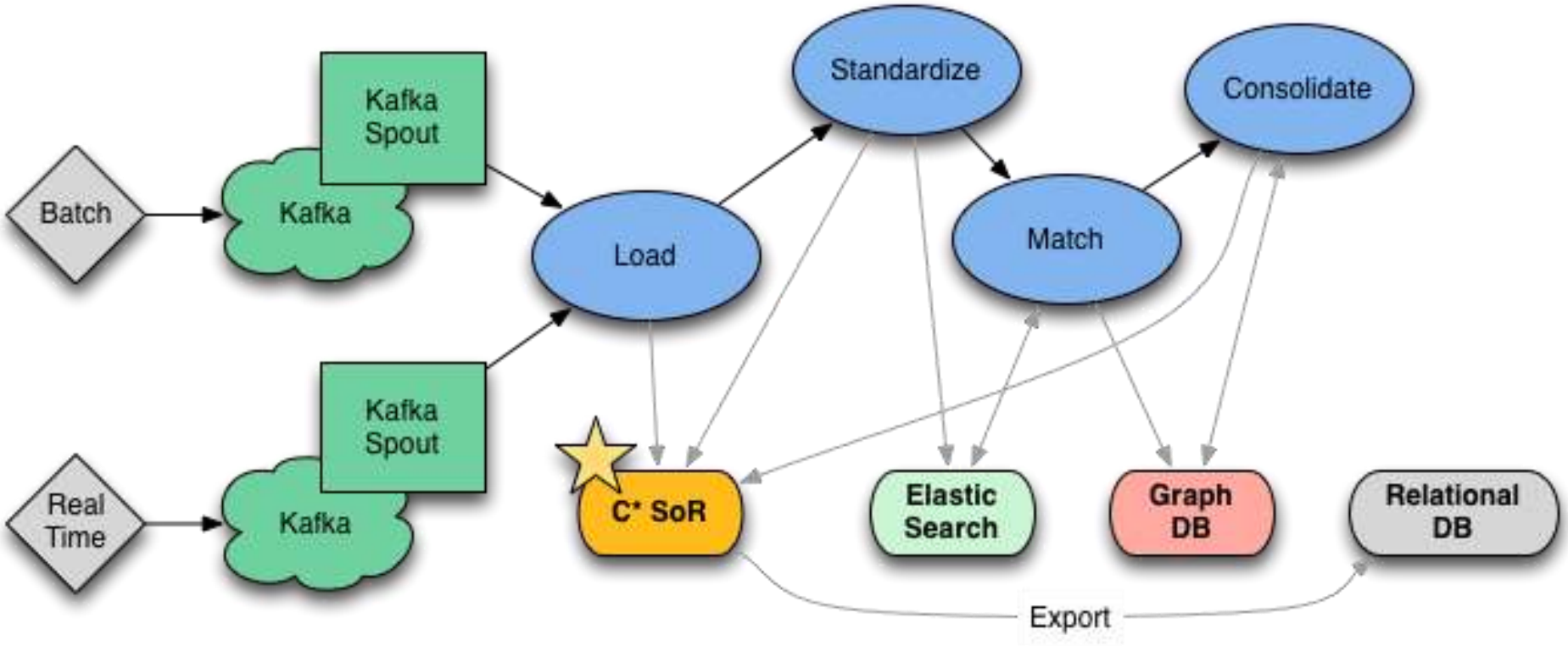
Health Market Science

The Science of Better Results

Back to the Pipeline



MDM Topology*



**Notional*

Health Market Science
The Science of Better Results

Design Principles

- What we got:
  - At-least-once processing
  - Simple data flows
- What we needed to account for:
  - Replays

Idempotent Operations!

Immutable Data!

FTW!

Cassandra State (v0.4.0)

git@github.com:hmsonline/storm-cassandra.git

Storm {tuple} ← <mapper> → Cassandra (ks, cf, row, k:v[])

Trident Elastic Search (v0.3.1)

git@github.com:hmsonline/trident-elasticsearch.git

Storm

{tuple} ← <mapper> → (idx, docid, k:v[])

Elastic Search

Storm Graph (v0.1.2)

Coming soon to...

git@github.com:hmsonline/storm-graph.git

```
for (tuple : batch)
    <processor> (graph, tuple)
```

Storm JDBI (v0.1.14)

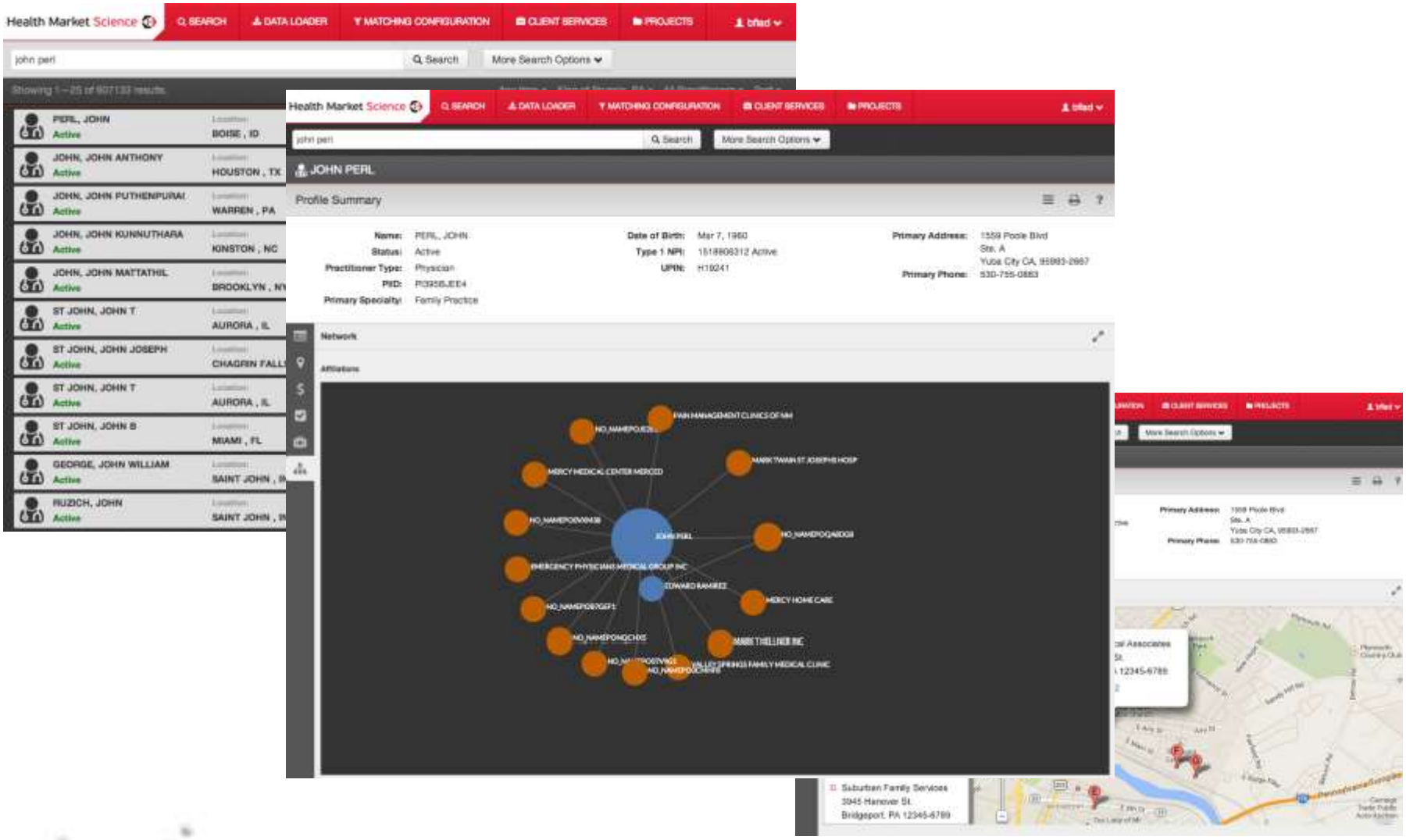
INTERNAL ONLY (so far)

Worth releasing?

{tuple} ← <mapper> → (JDBC Statement)

Health Market Science
The Science of Better Results

All good!

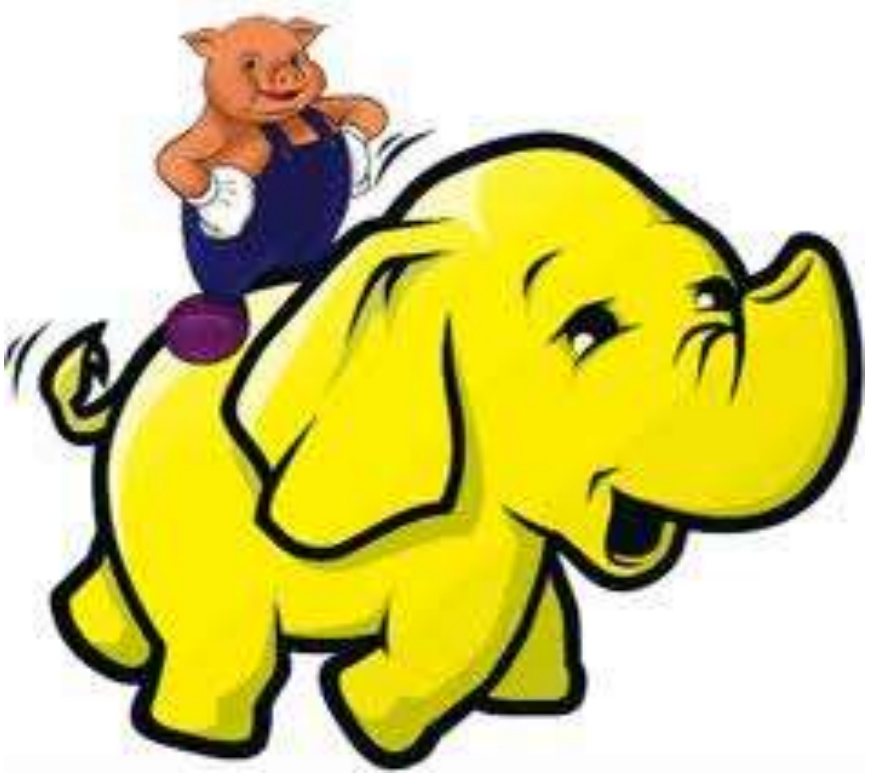


Health Market Science
The Science of Better Results

But...

What was the average amount paid for a medical claim associated with procedure X by zip code over the last five years?

Hadoop (<2)? Batch?



http://www.slideshare.net/prash1784/introduction-to-hadoop-and-pig-15036186

Yuck. ‘Nuff Said.

Alternatives?

Let's Pre-Compute It!

```
stream
    .groupBy(new Field("procedure"))
    .groupBy(new Field("zip"))
    .aggregate(new Field("amount"),
          new Average())
```

D'Oh!

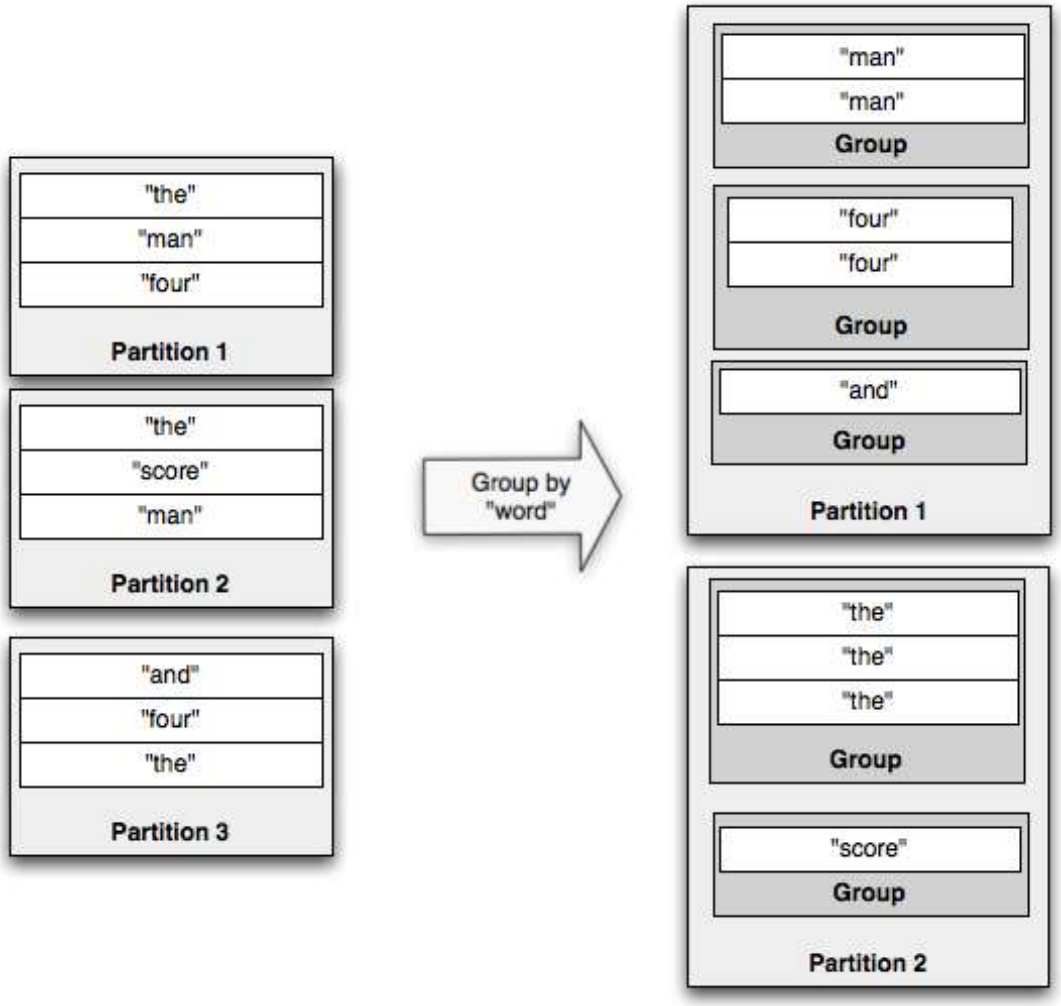
GroupBy's.

They set data in motion!

Lesson Learned

If possible, avoid re-partitioning operations!

(e.g. LOG.error!)



https://github.com/nathanmarz/storm/wiki/Trident-API-Overview

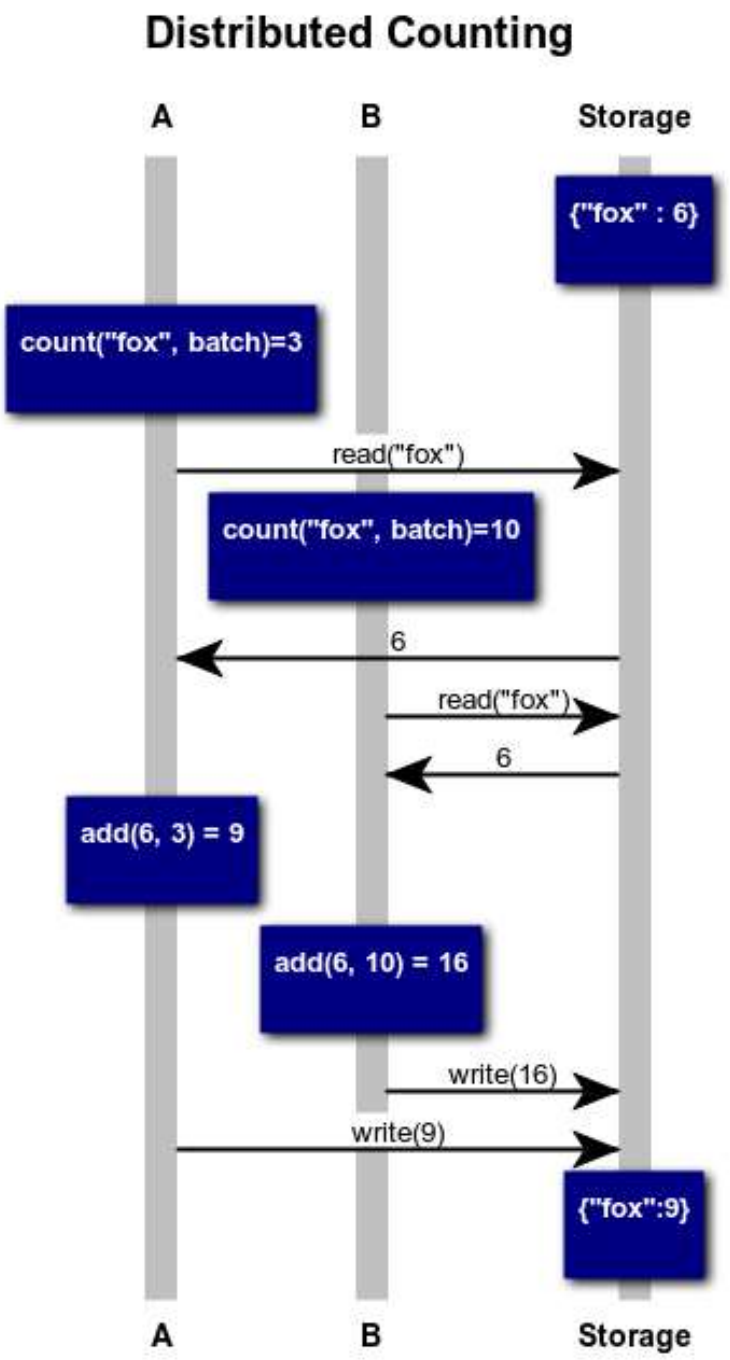
Why so hard?

D'Oh!

What we don't want:
LOCKS!

What's the alternative?
CONSENSUS!

19 != 9



Cassandra 2.0!

http://www.slideshare.net/planetcassandra/nyc-jonathan-ellis-keynote-cassandra-12-20
http://www.cs.cornell.edu/courses/CS6452/2012sp/papers/paxos-complex.pdf

Conditional Updates

"The alert reader will notice here that Paxos gives us the ability to agree on exactly one proposal. After one has been accepted, it will be returned to future leaders in the promise, and the new leader will have to re-propose it again."

http://www.datastax.com/dev/blog/lightweight-transactions-in-cassandra-2-0

```
UPDATE value=9 WHERE word="fox" IF value=6
```

Love CQL

Conditional Updates

+

Batch Statements

+

Collections

=

BADASS DATA MODELS

Announcing : Storm Cassandra CQL!

git@github.com:hmsonline/storm-cassandra-cql.git

{tuple} ← <mapper> → (CQL Statement)

Trident Batching =? CQL Batching

Incremental State!

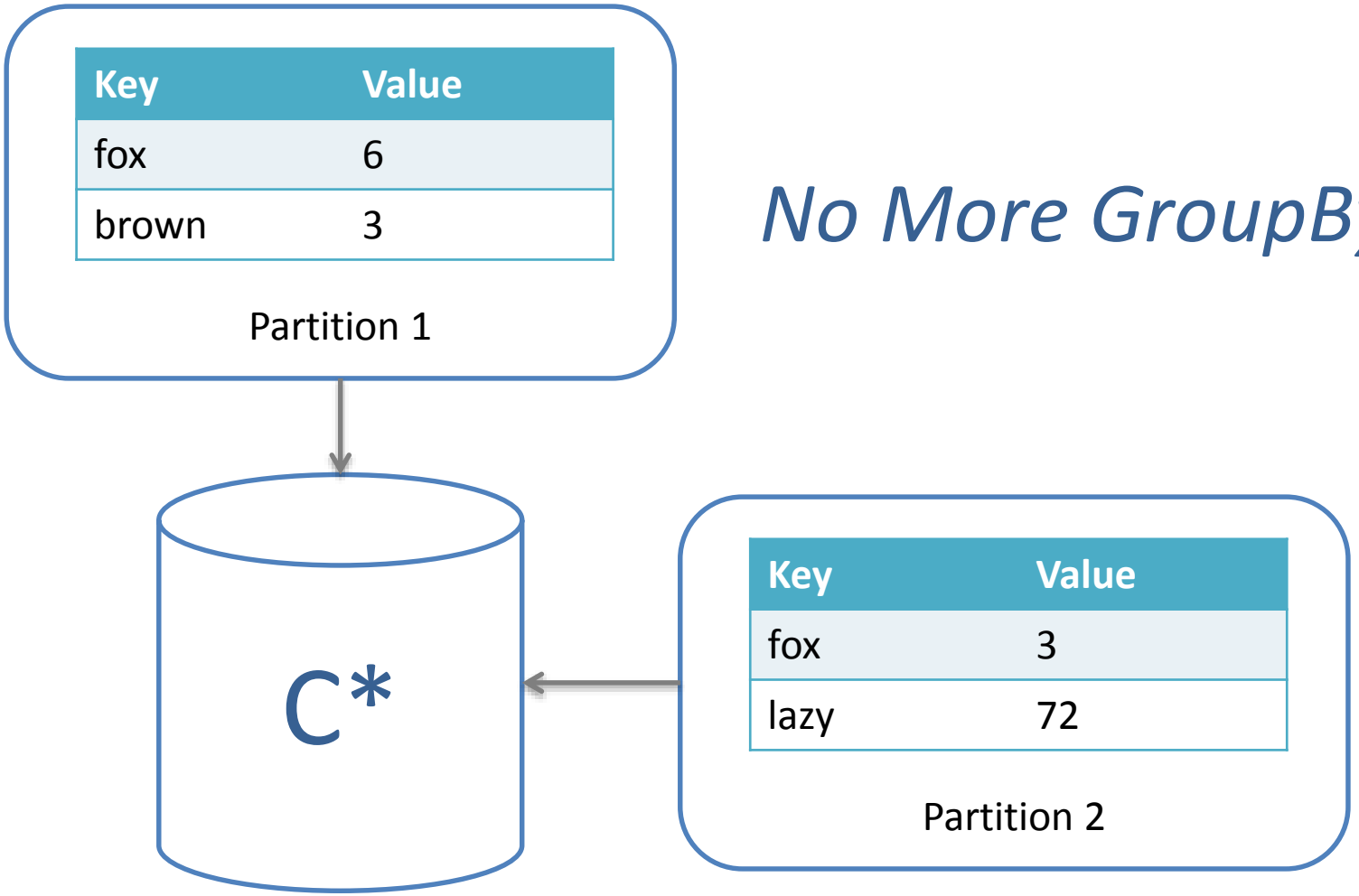
- Collapse aggregation into the state object.
  - This allows the state object to aggregate with current state in a loop until success.

- Uses Trident Batching to perform in-memory aggregation for the batch.

```
for (tuple : batch)
    state.aggregate(tuple);

while (failed?) {
    persisted_state = read(state)
    aggregate(in_memory_state, persisted_state)
    failed? = conditionally_update(state)
}
```

In-Memory Aggregation by Key!



No More GroupBy!

To protect against replays

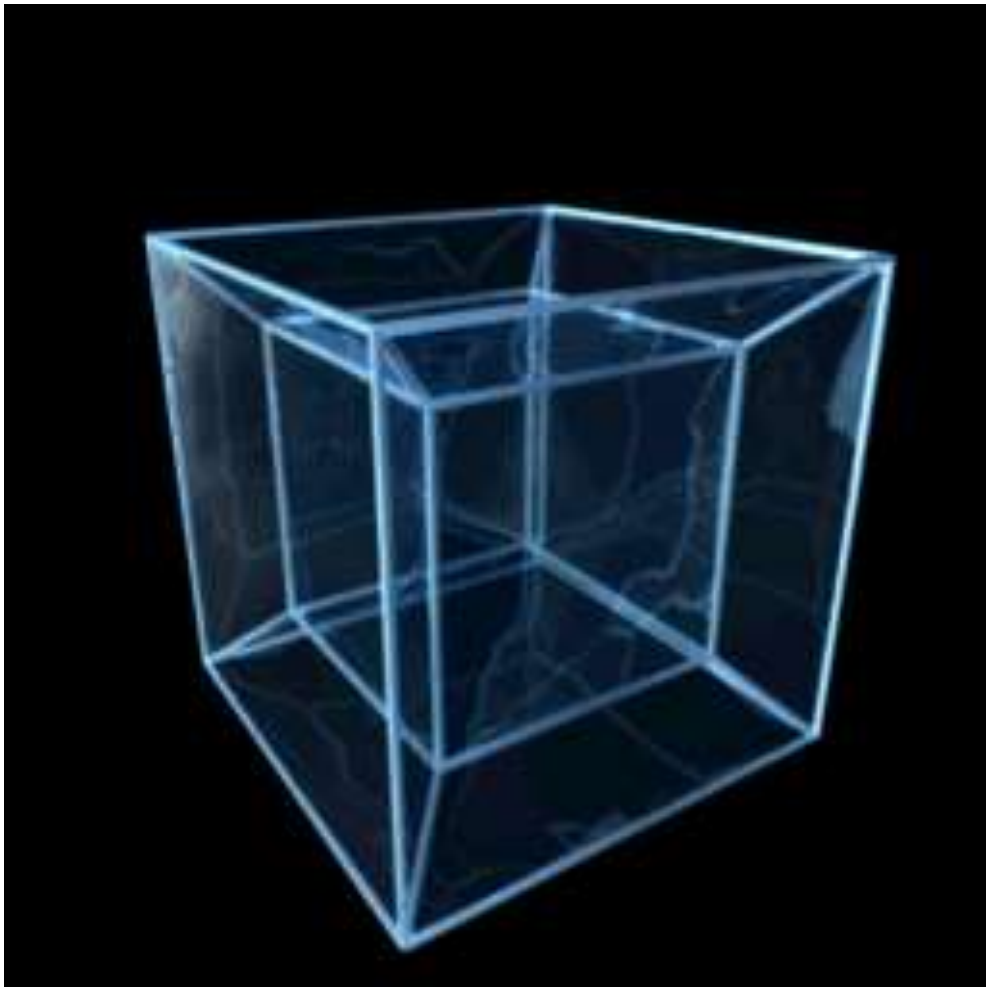
Use partition + batch identifier(s) in your conditional update!

"BatchId + partitionIndex consistently represents the same data as long as:

1. Any repartitioning you do is deterministic (so partitionBy is, but shuffle is not)
2. You're using a spout that replays the exact same batch each time (which is true of transactional spouts but not of opaque transactional spouts)"

- Nathan Marz

Hyper-Cubes!



Our Terminology

- A *cube* comprises:
  - Dimensions (e.g. procedure, zip, time slice)
  - A function (e.g. count, sum)
  - Function fields (e.g. amount paid)
  - Granularity

- A *metric* comprises:
  - Coordinates (e.g. vasectomy, 19460, 879123 – hour since epoch)
  - A value (e.g. 500 procedures)

- A *perspective* comprises:
  - A range of coordinates (*, 19460, January)
  - An interval (day)

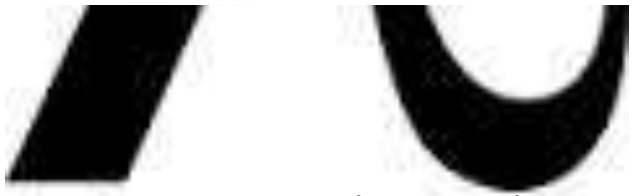
Complex Event Processing

- For each event,
  - Find relevant cubes
  - Adjust metrics for event
  - Group and aggregate metrics
  - Use conditional updates to incorporate metric into persistent state

The Lambda Architecture

Computing arbitrary functions on an arbitrary dataset in real time is a daunting problem. There is no single tool that provides a complete solution. Instead, you have to use a variety of tools and techniques to build a complete Big Data system.

The lambda architecture solves the problem of computing arbitrary functions on arbitrary data in real time by decomposing the problem into three layers: the batch layer, the serving layer, and the speed layer.



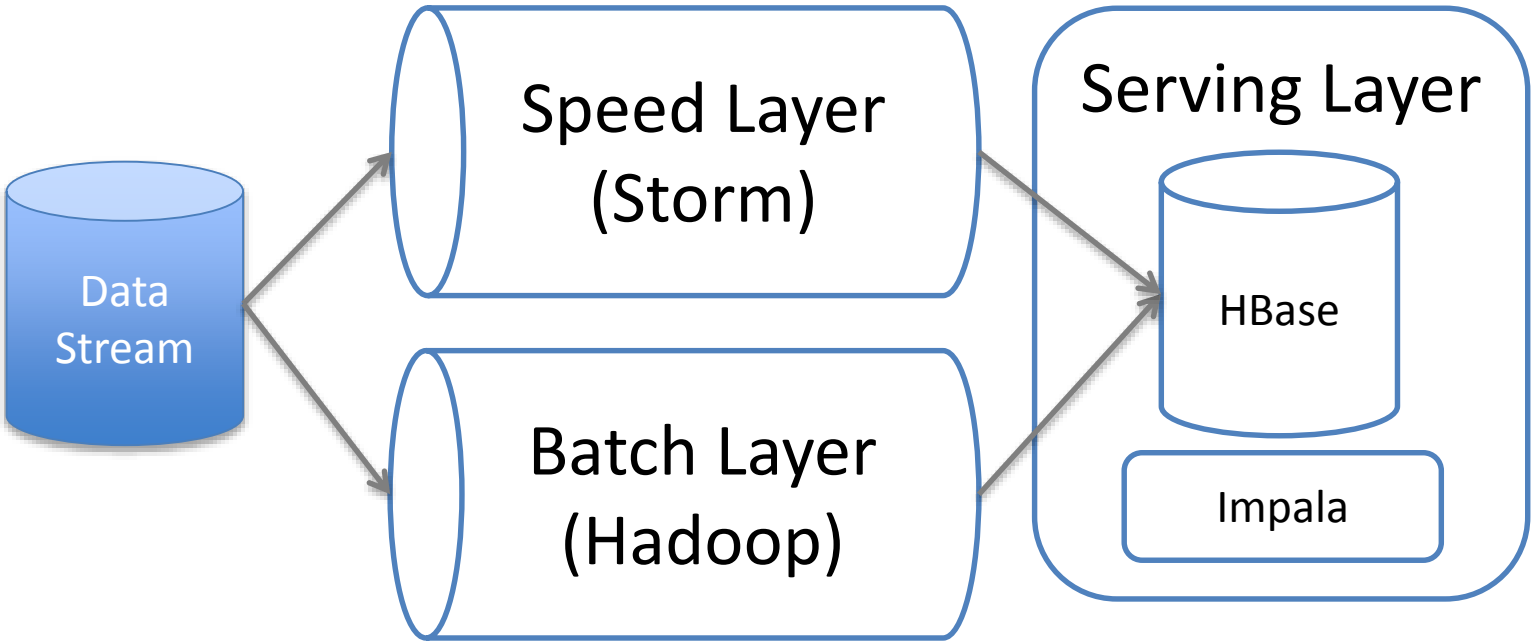
http://architects.dzone.com/articles/nathan-marzs-lamda

Let's Challenge This a Bit

because "additional tools and techniques" cost money and time.

- Questions:
  - Can we solve the problem with a single tool and a single approach?
  - Can we re-use logic across layers?
  - Or better yet, can we collapse layers?

A Traditional Interpretation



D'Oh! Two pipelines!

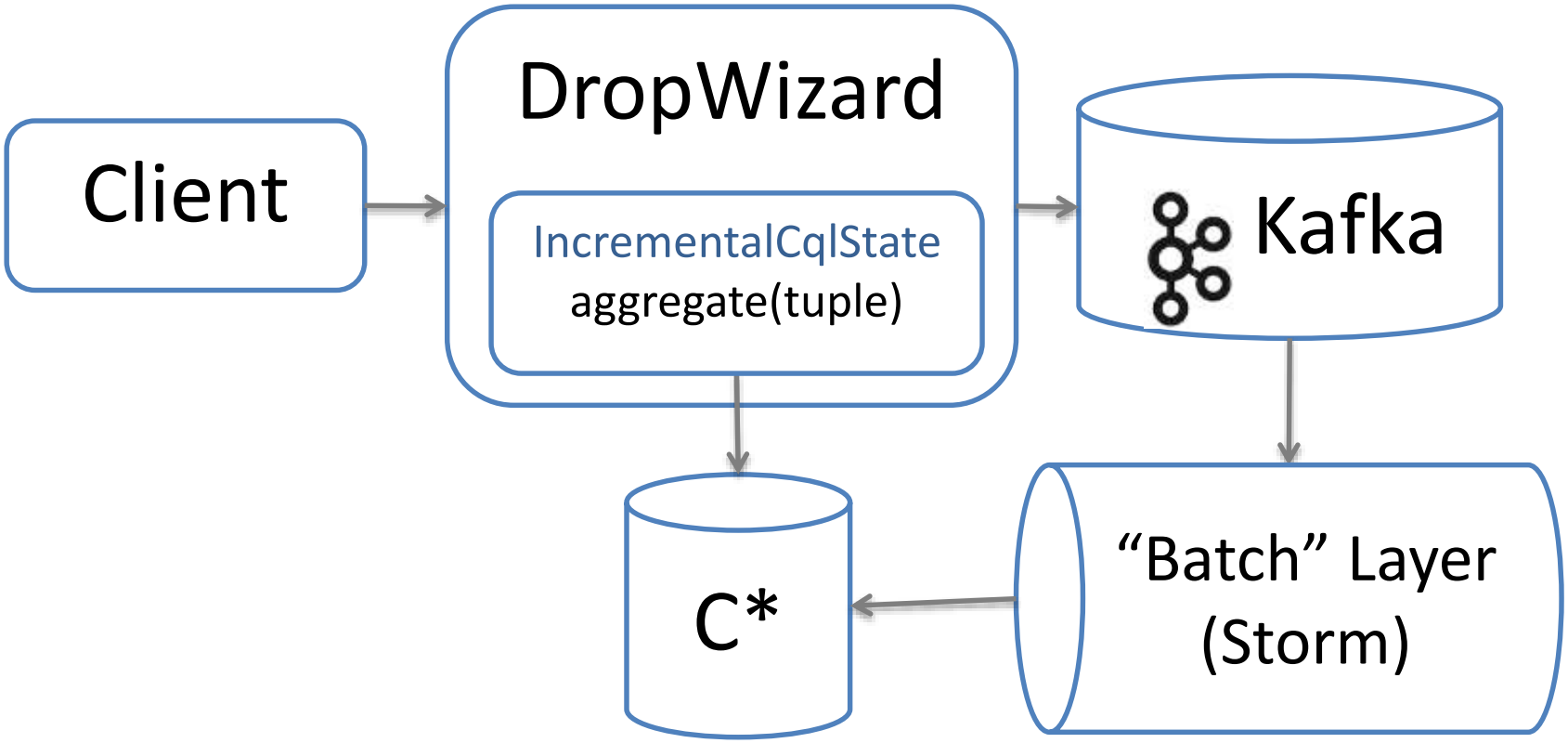
Integrating Web Services

- We need a web service that receives an event and provides,
  - an *immediate acknowledgement*
  - a *high likelihood* that the data is integrated *very soon*
  - a *guarantee* that the data *will be integrated* eventually

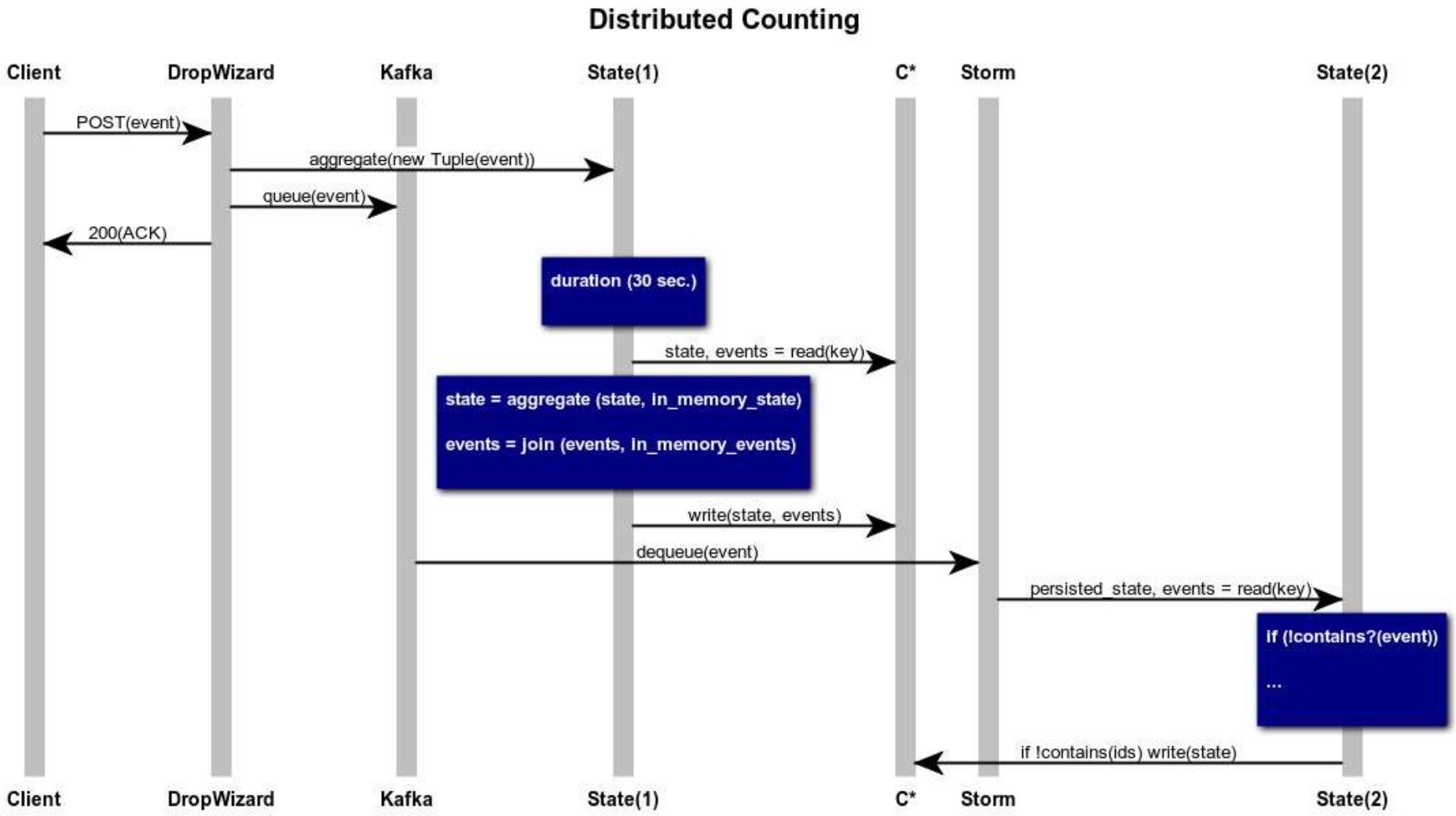
- We need an architecture that provides for,
  - Code / Logic and approach re-use
  - Fault-Tolerance

Grand Finale

The Idea : Embedding State!



The Sequence of Events



www.websequencediagrams.com

The Wins

- Reuse Aggregations and State Code!
- To re-compute (or backfill) a dimension, simply re-queue!
- Storm is the “safety” net
  - If a DW host fails during aggregation, Storm will fill in the gaps for all ACK’d events.

- Is there an opportunity to reuse more?
  - BatchingStrategy & PartitionStrategy?

In the end, all good. =)

Health Market Science
The Science of Better Results

Plug

[The Book]

Shout out:
Taylor Goetz

Thanks!

Brought to you by

12 years together

Brian O'Neill, CTO
boneill@healthmarketscience.com
@boneill42

Health Market Science
The Science of Better Results

APPENDIX

CassandraCqlState

```
public void commit(Long txid) {
    BatchStatement batch = new BatchStatement(Type.LOGGED);
    batch.addAll(this.statements);
    clientFactory.getSession().execute(batch);
}

public void addStatement(Statement statement) {
    this.statements.add(statement);
}

public ResultSet execute(Statement statement){
    return clientFactory.getSession().execute(statement);
}
```

CassandraCqlStateUpdater

```
public void updateState(CassandraCqlState state,
    List<TridentTuple> tuples,
    TridentCollector collector) {
      for (TridentTuple tuple : tuples) {
          Statement statement = this.mapper.map(tuple);
          state.addStatement(statement);
      }
 }
```

ExampleMapper

```
public Statement map(List<String> keys, Number value) {
    Insert statement =
    QueryBuilder.insertInto(KEYSPACE_NAME, TABLE_NAME);
    statement.value(KEY_NAME, keys.get(0));
    statement.value(VALUE_NAME, value);
    return statement;
}

public Statement retrieve(List<String> keys) {
    Select statement = QueryBuilder.select()
        .column(KEY_NAME).column(VALUE_NAME)
        .from(KEYSPACE_NAME, TABLE_NAME)
        .where(QueryBuilder.eq(KEY_NAME, keys.get(0)));
     return statement;
}
```