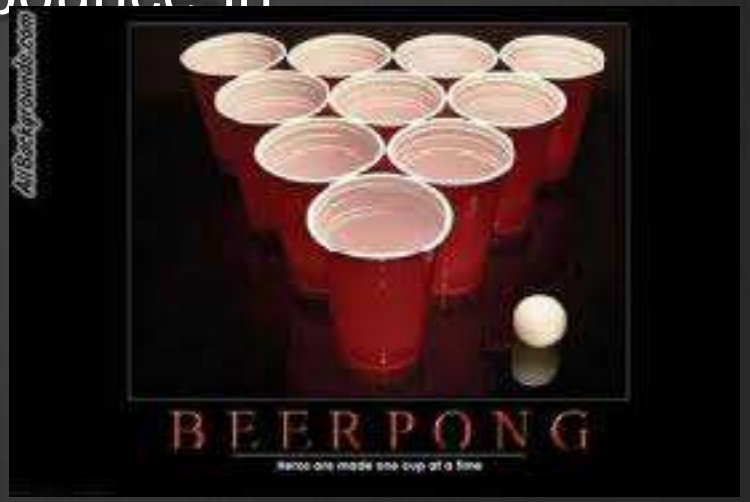
The Big Data Quadfecta

Brian O'Neill
Lead Architect, Health Market Science
@boneill42, bone@alumni.brown.edu

Taylor Goetz
Development Lead, Health Market Scienc
@ptgoetz, ptgoetz@gmail.com

Quadfecta?

1. Quadfecta
   - A legendary beirut/beer pong shot that lands on the tops of four cups simultaneously. Considered the rarest shot in the game, topping even the trifecta, 2-cup knockover-and-sink, and simultaneous 6-cup game-ending double bounce-in



- Kafka
- Storm
- Elastic Search
- Cassandra

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

3 V's

Volume

Variety

Velocity

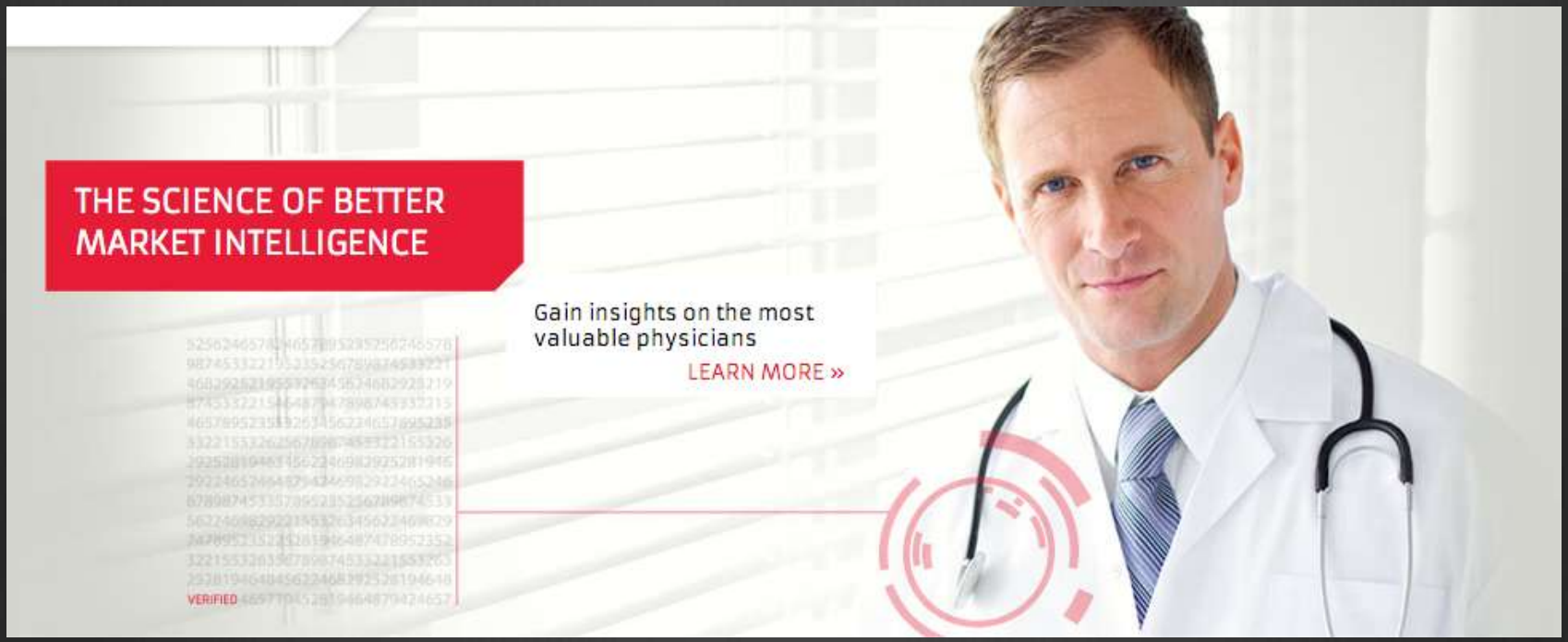
The Use Case

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

Our Mission



- Prescriber eligibility and remediation
- Eliminate fraud, waste and abuse
- Insights into the healthcare space

The Business

Master Data Solutions

Health Care Provider & Facilities

Variety/Velocity

- >l2000 of sources
- 6 Million unique HCPs
- 10+ years history

Data Challenges

- Constant change in real world data
- Conflicting & partial info
- Frequent changes to source structure
- Authoritative sources vs. crowdsource
- Predicting source quality

Business Solutions

CompleteView, Expense Manager, CompleteSpend

Prescriber Eligibility/Remdiation

Analtyics (Influencer Networks)

Medical Claims Data

Medical Procedures & Diagnosis

Volume/Velocity

- ~1B claims annually
- +5B records annually
- 5+ years history

Data Challenges

- Sources have incomplete capture
- Overlapping source data
- Statistical projections & biases
- Social media type relationships

1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

Our Solutions

1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

Business Needs


Sales & Marketing

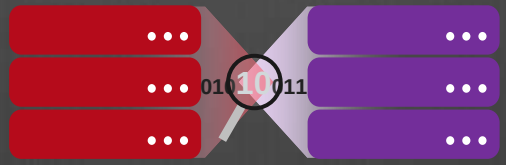

Compliance


Business Systems

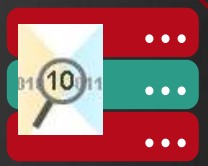

Finance & Legal

Solutions

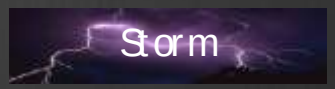

Provider Data

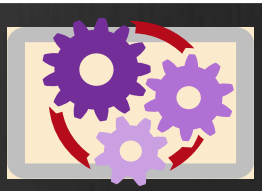

Data Assessment, Integration & Enrichment Services


Compliance


Market Intelligence

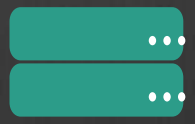
Advanced Technology

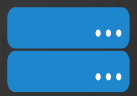


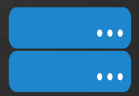

Master Data Management

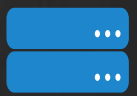
HMS Authoritative Sources

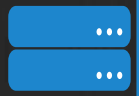

PDC


Medical Claims


Federal


State


Web


Derived

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

Datacenter

- ¾ Petabytes of raw storage
- Virtualized (VMware)
- On a SAN
- Should we go physical???

Under the Hood

Customer

Web

Government

Data Sources

Analytics

Match

Consolidate

Standardize

Validate

Distributed Processing

User Interface

Web Services

Interfacing

Dashboard / Reports

Visualization

Indexing

Relational

Structured Storage

NoSQL

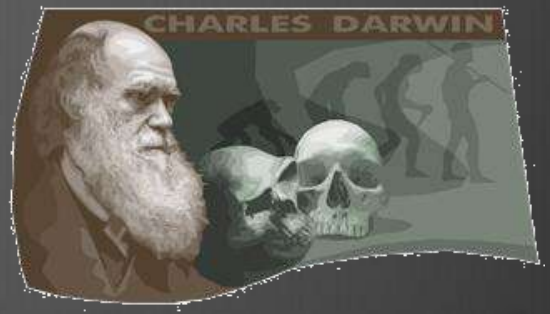
Graph(s)

Flexible Storage

1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

Master Data Management



Harvested

$f_{address} \in F@t_0$

Government

$f_{license} \in F@t_5$

Private

$f_{sanction} \in F@t_1$

$f_{sanction} \in F@t_4$

Schema Change!

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

The Design

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

System of Record

Flexibility (Variety)
Scalability (Velocity + Volume)

1• 800.593.4467 • info@healthmarketscience.com

Design Principles

- Patterns
  - Idempotent Operations
    - Elegantly handle replay
  - Immutable data
    - Assertions of facts over time
- Anti-Patterns
  - Transactions / Locking

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

State / Counting

- Exactly-once semantics for state
  - Create small batches
  - Order batches

Batch	Total
1	4
3	4 (wait)
2	10 (+6)
3	23 (+13)
3'	23 (+0)

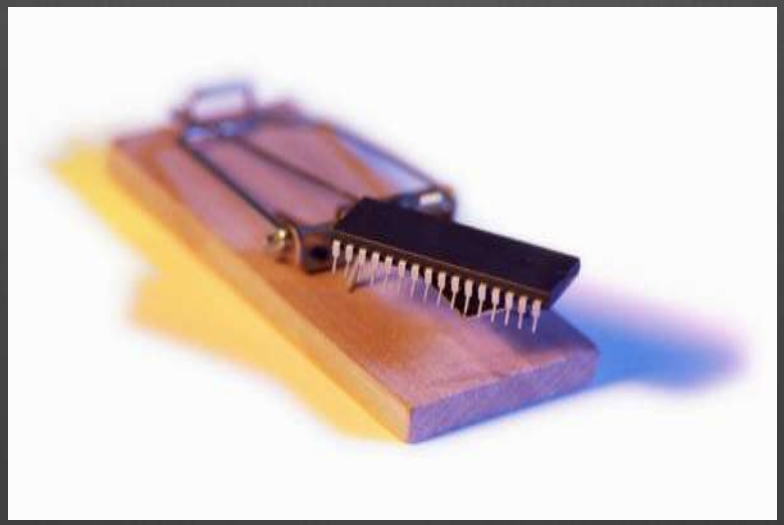
1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

What we did wrong…

- Could not react to transactional changes
- Needed extra logic to track what changed
- Took too long

What we did wrong… (II)



- AOP-based triggers
  - Worked well initially.
  - Business Processes captured as side-effects.

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

What we did right.

- REST APIs for Loose Coupling
- See Virgil:
  - https://github.com/hmsonline/virgil
- But really… watch out for Intravert
  - https://github.com/zznate/intravert-ug

1• 800.593.4467 • info@healthmarketscience.com
Health Market Science

1• 800.593.4467 • info@healthmarketscience.com

Health Market Science

Kafka

- Millions of Messages
- Replay Enabled
- No transactions / Lightning Fast

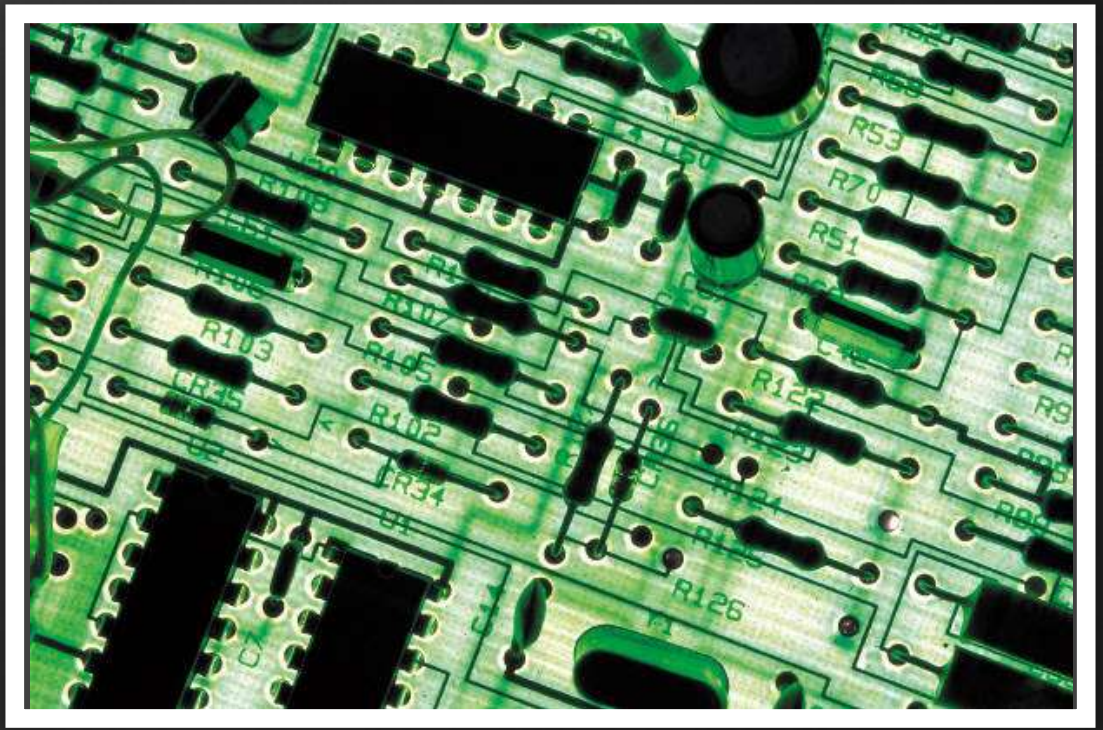
Elastic Search

- Edit Distance / Soundex
- Native Scalability
- Fuzzy Search
- Geospatial
- Facets

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

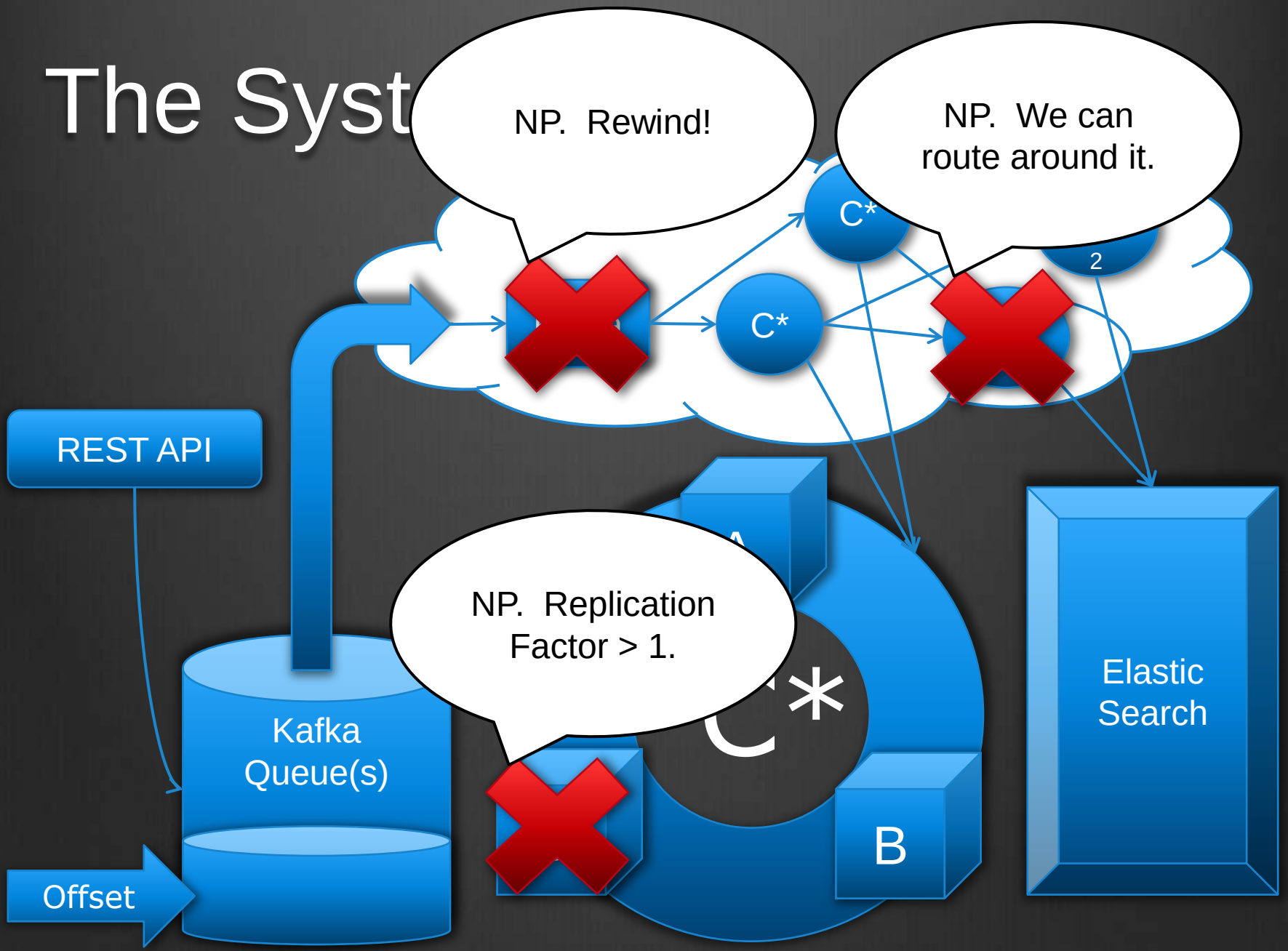
Storm

- Guaranteed once semantics
- Well-designed processing abstraction
- Beats BYODP
- Momentum



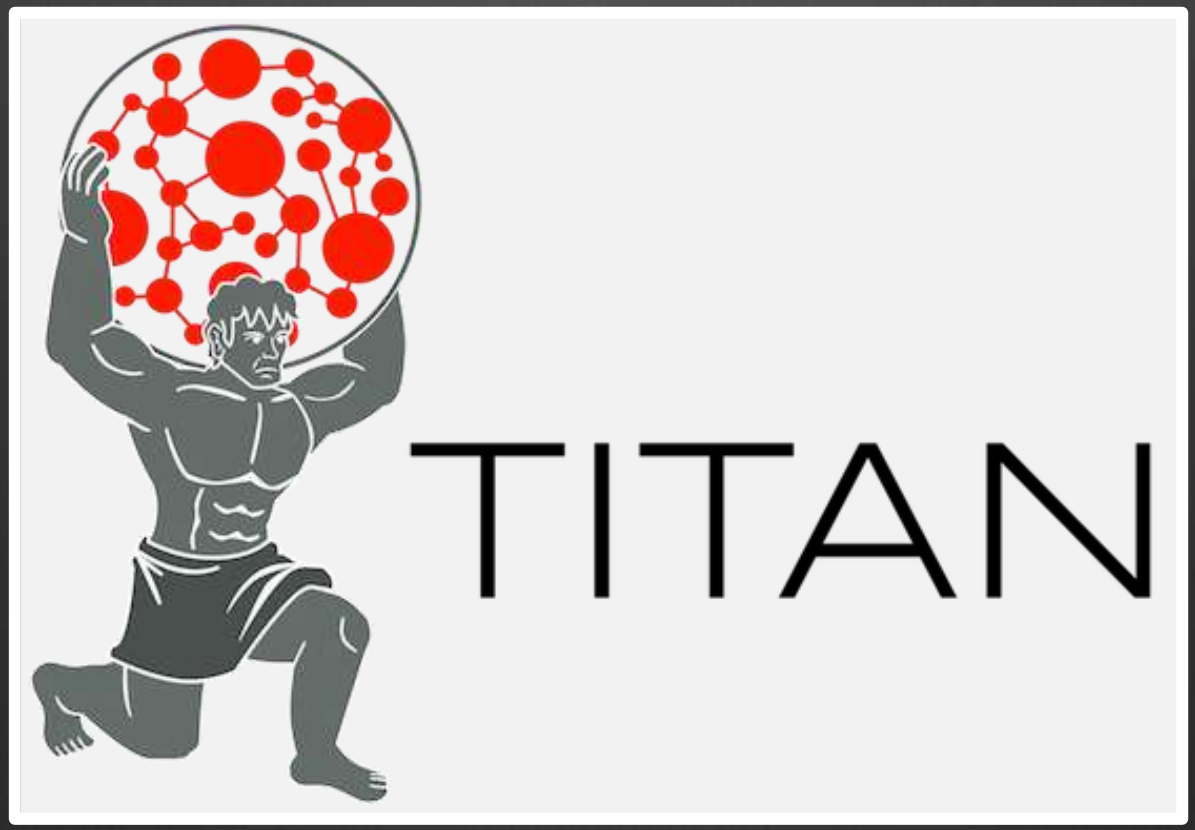
Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

The Syst



1• 800.593.4467 • info@healthmarketscience.com

What comes after Quadfecta?



?

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

Real-Time Integration

- Real-time CRUD via Web Services
  - DRPC
  - “Real-time” Queue

Not quite sure?

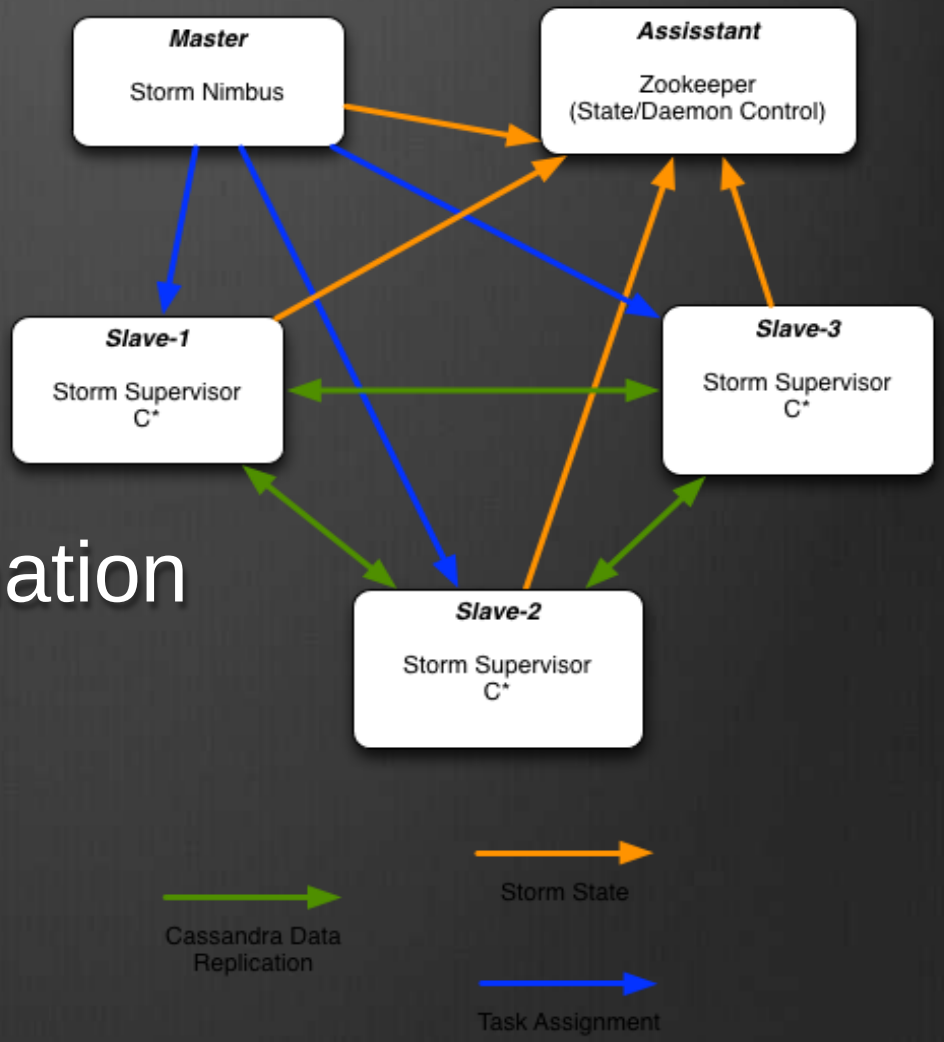
1• 800.593.4467 • info@healthmarketscience.com
Health Market Science

The Storm/C* Bridge

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

Anatomy of a Storm Cluster

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



Storm Primatives

- Streams
  - Unbounded sequence of tuples
- Spouts
  - Stream Sources
- Bolts
  - Unit of Computation
- Topologies
  - Combination of n Spouts and n 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

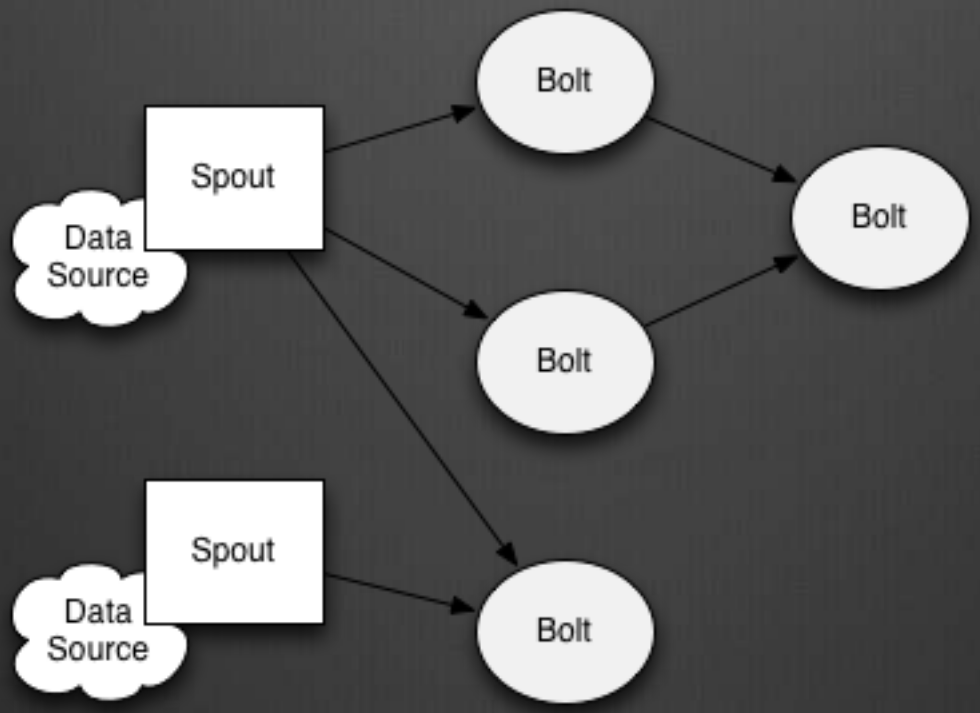
Storm Bolts

- 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



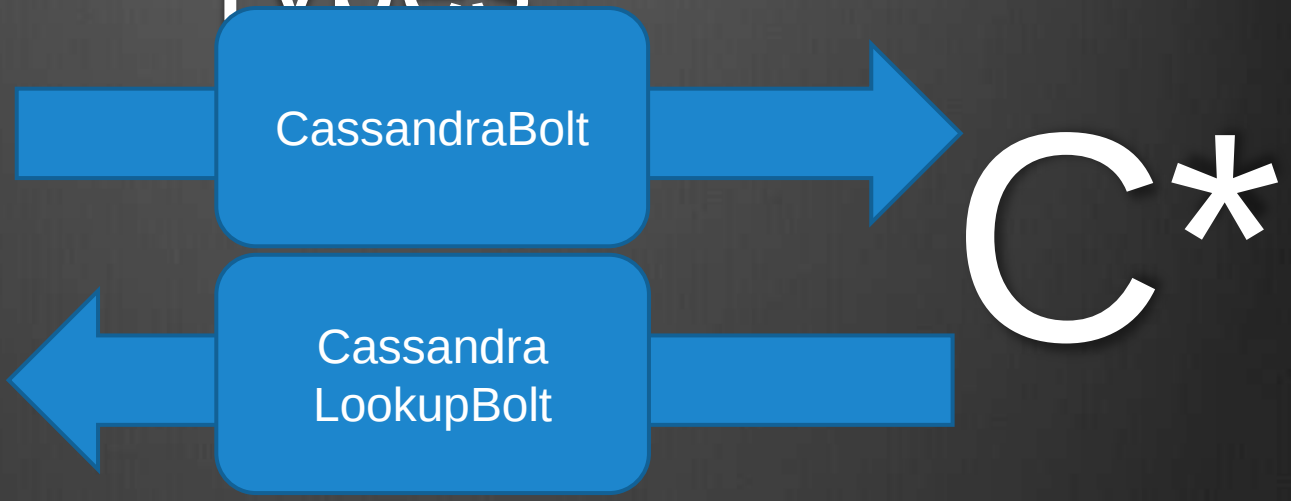
1• 800.593.4467 • info@healthmarketscience.com
Health Market Science

Storm and Cassandra

- Use Cases:
  - Write Storm Tuple data to C*
    - Computation Results
    - Pre-compute indices
  - Read data from C* and emit Storm Tuples
    - Dynamic Lookups

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

Storm Cassandra Bolt Types

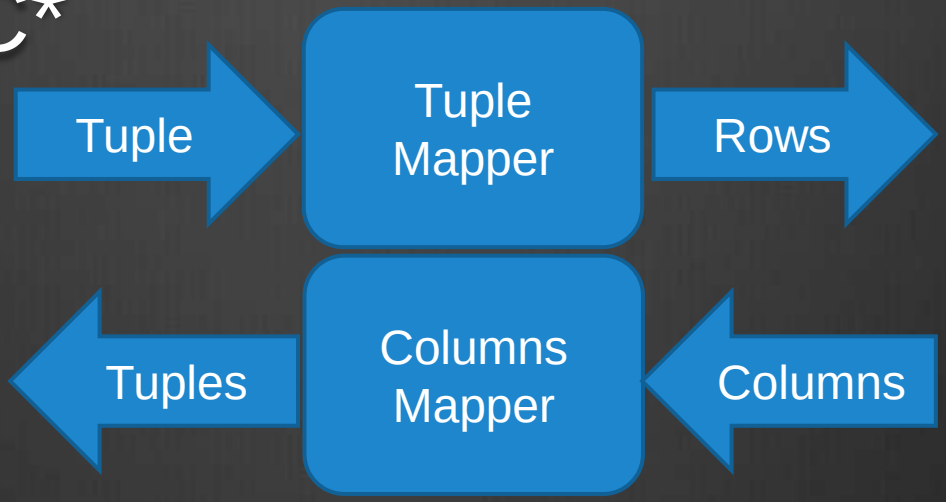


- CassandraBolt
  - Writes data to Cassandra
  - Available in Batching and Non-Batching
- CassandraLookupBolt
  - Reads data from Cassandra

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

Storm-Cassandra Project

- Provides generic Bolts for writing/reading Storm Tuples to/from C*



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

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

Storm-Cassandra Project

- TupleMapper Interface
  - Tells the CassandraBolt how to write a tuple to an arbitrary data model

- Given a Storm Tuple:
  - Map to Column Family
  - Map to Row Key
  - Map to Columns

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

Storm-Cassandra Project

- ColumnsMapper Interface
  - Tells the CassandraLookupBolt how to transform a C* row into a Storm Tuple
- Given a C* Row Key and list of Columns:
  - Return a list of Storm Tuples

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

1• 800.593.4467 • info@healthmarketscience.com
Health Market Science

Storm-Cassandra Project

- Current State:
  - Version 0.4.0
  - Uses Astyanax Client
  - Several out-of-the-box *Mapper Implementations:
    - Basic Key-Value Columns
    - Value-less Columns
    - Counter Columns
    - Lookup by row key
    - Lookup by range query
  - Composite Key/Column Support
  - Trident support

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

Storm-Cassandra Project

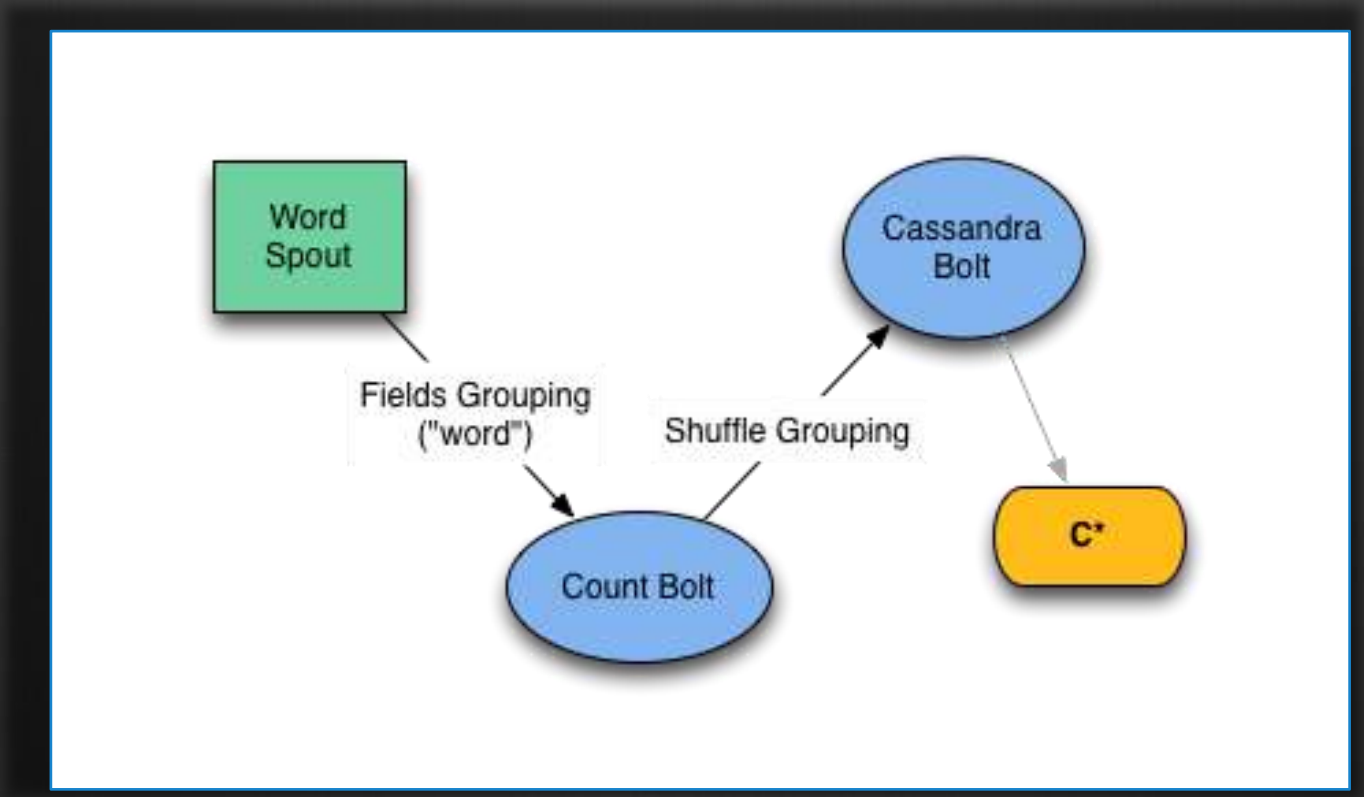
- Future Plans:
  - Switch to CQL
  - Enhanced Trident Support

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

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

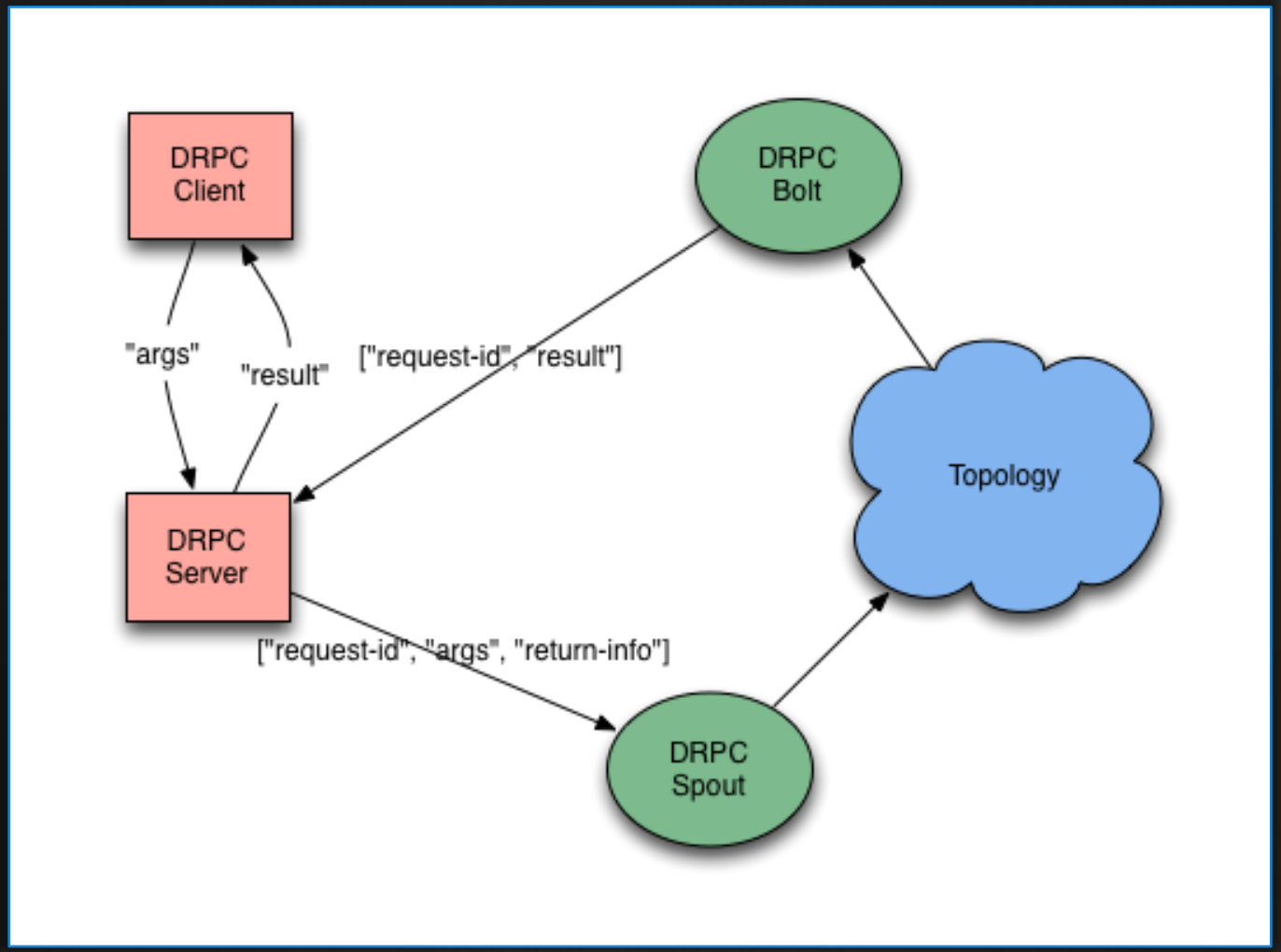
1• 800.593.4467 • info@healthmarketscience.com

Persistent Word Count

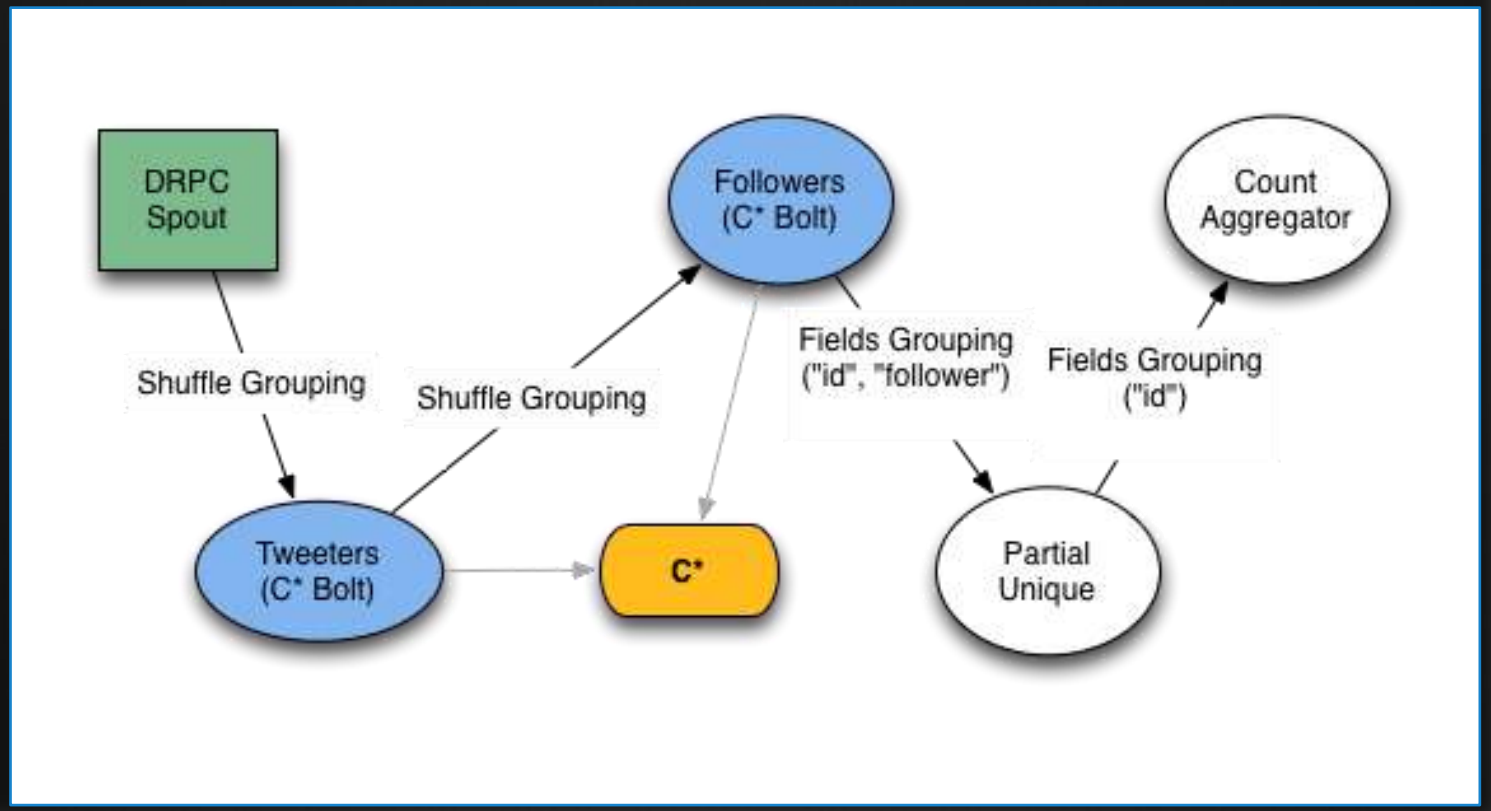


http://github.com/hmsonline/storm-cassandra

DRPC



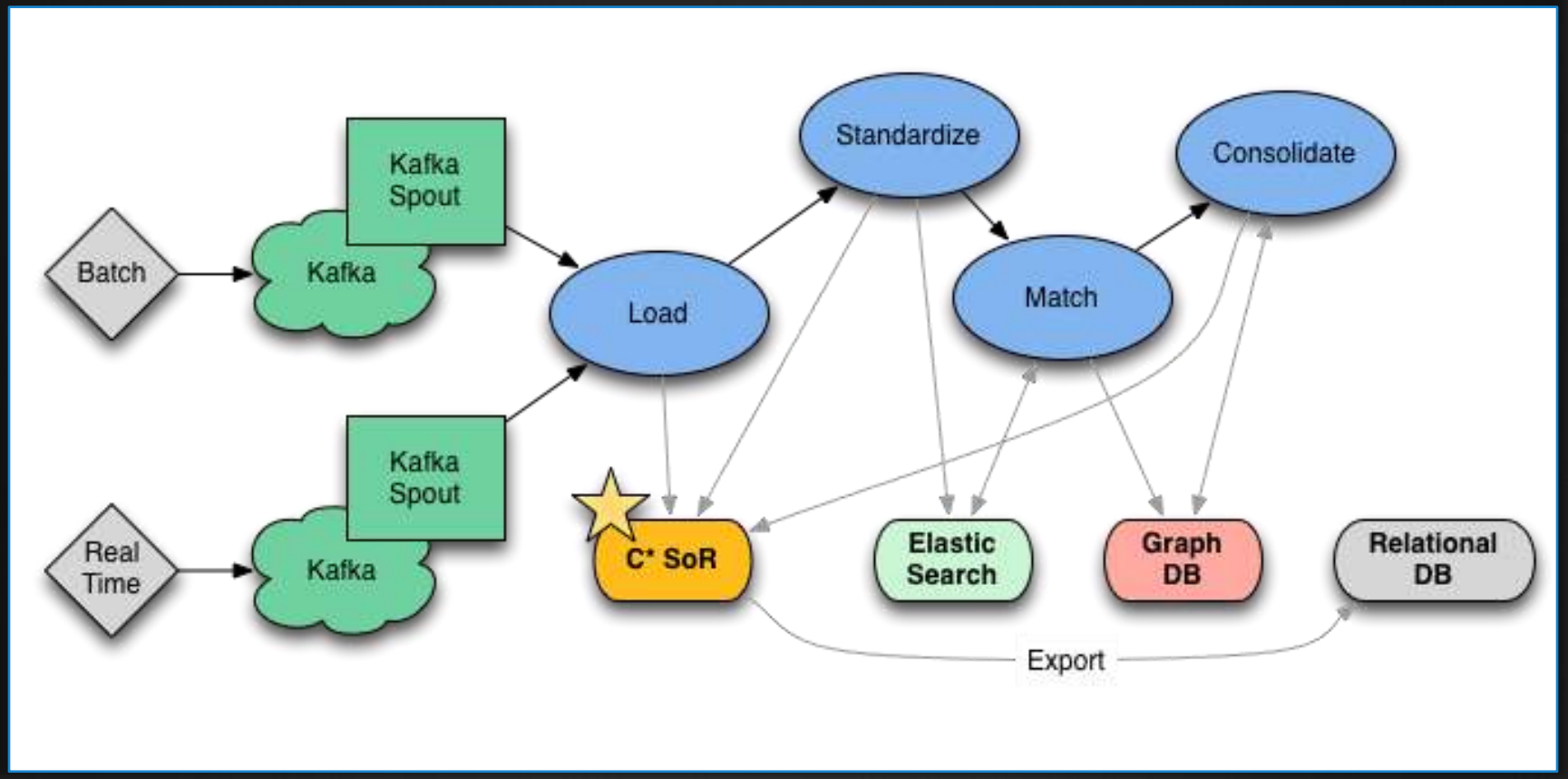
“Reach” Computation



1• 800.593.4467 • info@healthmarketscience.com

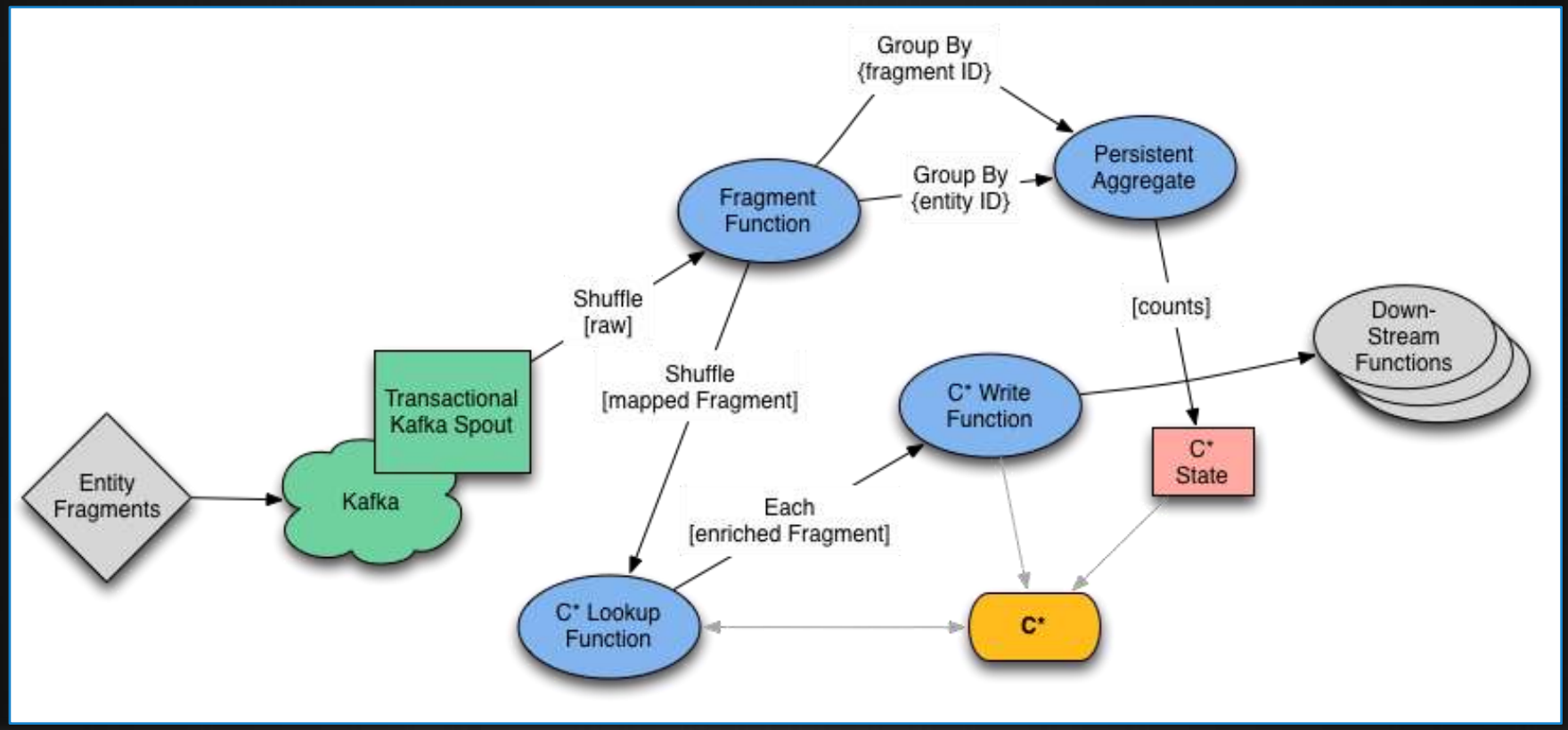
Health Market Science

MDM Topology*



**Notional*

Load Topology



Shameless Shoutouts

- HMS (https://github.com/hmsonline/)
  - storm-cassandra
  - storm-elastic-search
  - storm-jdbi (coming soon)
- ptgoetz (https://github.com/ptgoetz)
  - storm-jms
  - storm-signals

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

Next Level : Trident

Health Market Science
1• 800.593.4467 • info@healthmarketscience.com

Trident

- Provides a higher-level abstraction for stream processing
  - Constructs for state management and Batching
- Adds additional primitives that abstract away common topological patterns
- Deprecates transactional topologies
- Distributes with Storm

Sample Trident Operations

- Partition Local
  - Functions (execute(x) → x + y)
  - Filters (isKeep(x) → 0,x)
  - PartitionAggregate
    - Combiner (pairwise combining)
    - Reducer (iterative accumulation)
    - Aggregator (byoa)

A sample topology

```
TridentTopology topology = new TridentTopology();

TridentState wordCounts =

     topology.newStream("spout1", spout)

       .each(new Fields("sentence"),

                                          new Split(),

                                          new Fields("word"))

       .groupBy(new Fields("word"))

       .persistentAggregate(

                    MemcachedState.opaque(serverLocations),

                                         new Count(),

                                         new Fields("count"))

       .parallelismHint(6);
```

https://github.com/nathanmarz/storm/wiki/Trident-state

Trident State

Sequenced writes by batch/transaction id.

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