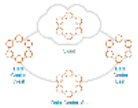


Why Cassandra (C*)?
- Always ON
  - No SPOF
- Data Automatically
  - Partitioned
  - Distributed
  - Replicated
- Linear Scale
- Ease of Admin

All part of Cassandra (C*) Architecture

Q: Cassandra is great for high performance, always-on data store. So, why DataStax?

A: DataStax provides*

Certified Cassandra Integration with:
- Spark
- Solr
- Hadoop

Security
- Kerberos, LDAP, in-flight and on-disk data encryption
- Audit Logging

Certified Drivers: java, C#, Node.js, Ruby, C/C++, Spark Connector, Spark ODBC, ODBC, Hive ODBC

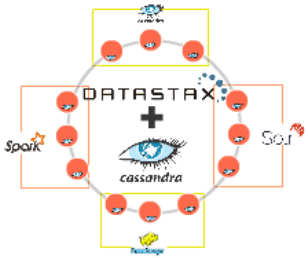
In-memory option

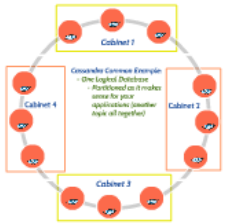
Tools: OpsCenter, DevCenter

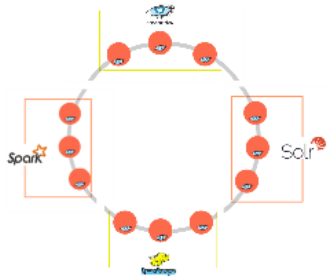
Support

Training*

* Download away (http://www.datastax.com/download) ...
... or build you're own (https://github.datastax.com : http://cassandra.apache.org ... or a igatrillion other C* related projects)
* https://academy.datastax.com







Back to Cassandra and DataStax Enterprise functionality...

Q: How does OSS C* Relate to DataStax Enterprise (DSE) and C*?

With C* 3.0 OSS will be "tick-tock" release cycle
- First month feature release, next bug fixes. Rinse. Repeat*.

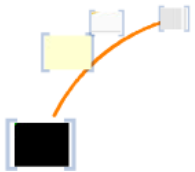
- Great for implementing new features and ideas / fast / agile
- Don't get bug fix patch only releases... support yourself.

DataStax Engineering selects an OSS C* version as a candidate for DSE production release*.
- Run through QA / production testing
- DSE integration built
- Bugs fixed and passed back to the OSS community

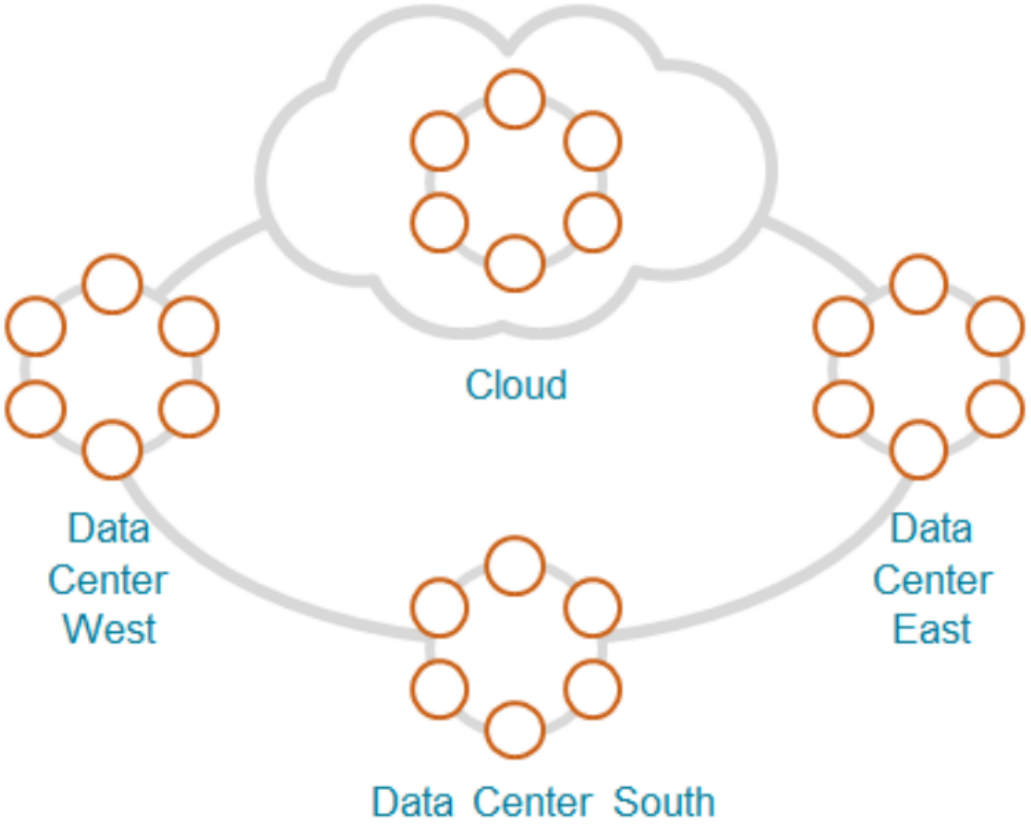
- When bug fixes are implemented in OSS is up to the community.
- OSS always has new features / fixes introduced.

* Note: Only one OSS source for Cassandra
* DSE Engineering is a major part of the OSS community / only one vendor (DataStax)

Where does DSE / Cassandra fit?
- C* delivers always-on, high performance transactional data store (i.e. transactions).
- Solr often used for search (i.e. catalog, metadata, or other information stored in C*... people get creative with Solr search.
- Hadoop for batch processing, integration with "data lakes", and ETL for C* and other data.
- Spark for advanced analytics, integrations with other data sources, data ETL / movement, and general queries on C* and other data.
- Storm, Kafka, Flume...



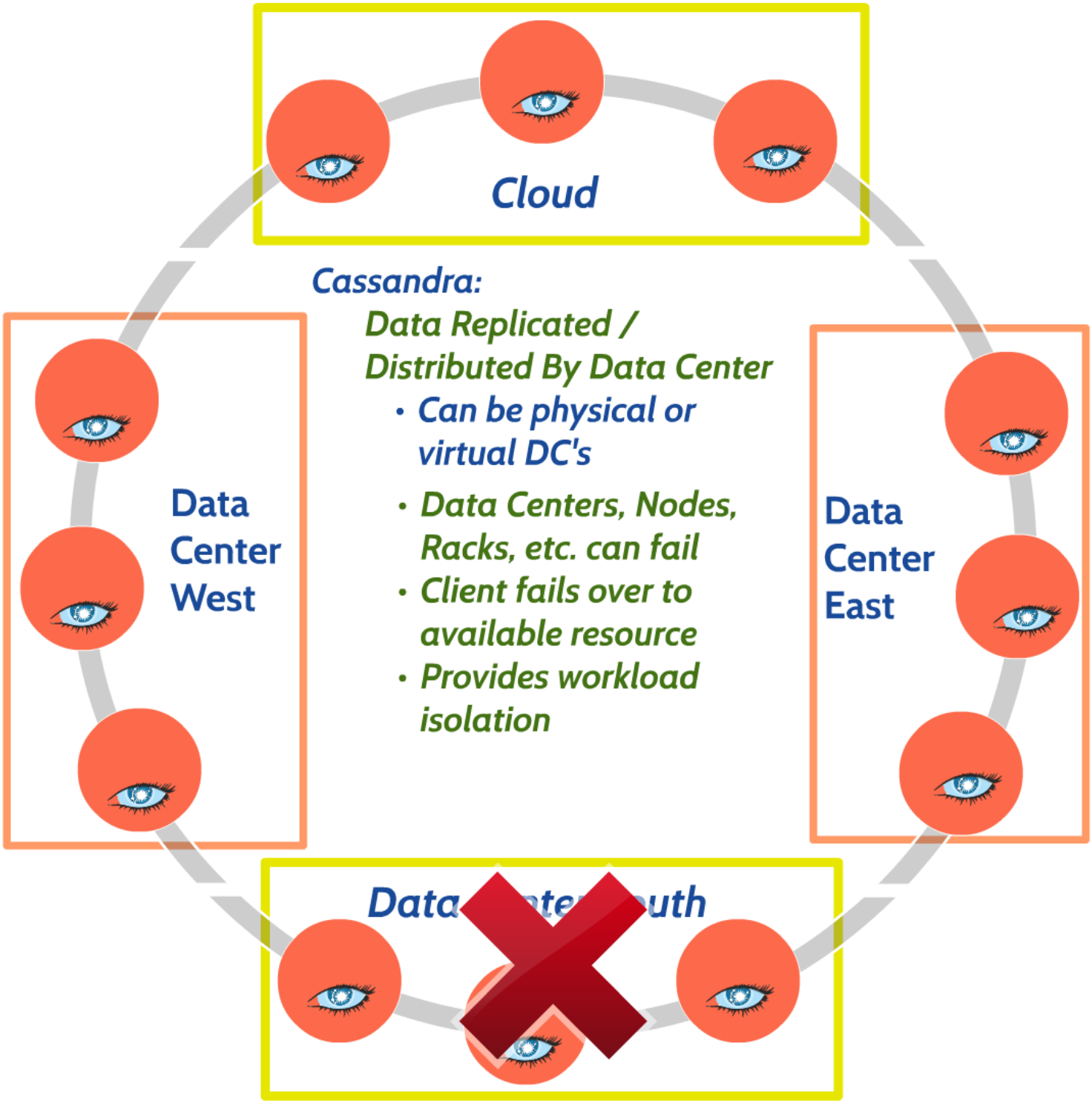

DATASTAX
cassandra



Why Cassandra (C*)?

- Always ON
  - No SPOF
- Data Automatically
  - Partitioned
  - Distributed
  - Replicated
- Linear Scale
- Ease of Admin

All part of Cassandra (C*) Architecture


Cloud
Cassandra:
Data Replicated /
Distributed By Data Center
• Can be physical or virtual DC's
• Data Centers, Nodes, Racks, etc. can fail
• Client fails over to available resource
• Provides workload isolation
Data Center West
Data Center East

Q: Cassandra is great for high performance, always-on data store. So, why DataStax?

A: DataStax provides*

Certified Cassandra

Integration with:

- Spark
- Solr
- Hadoop

Security

- Kerberos, LDAP, in-flight and on-disk data encryption
- Audit Logging

Certified Drivers:

java, C#, Node.js, Ruby, C/C++, Spark Connector, Spark ODBC, ODBC, Hive ODBC

In-memory option

Tools:

OpsCenter, DevCenter

Support

Training*

* Download away (http://www.datastax.com/download) ...

... or build you're own (https://github.datastax.com : http://cassandra.apache.org ... or a igatrillion other C* related projects)

* https://academy.datastax.com

Q: How does OSS C* Relate to DataStax Enterprise (DSE) and C*?

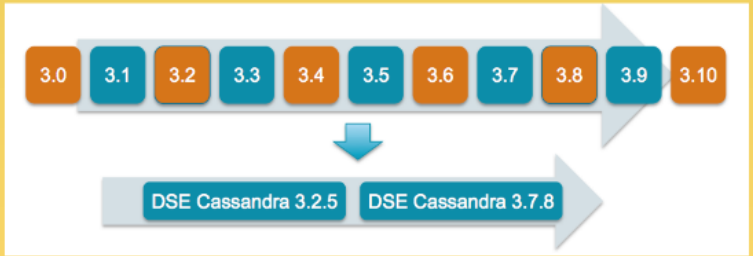
With C* 3.0 OSS will be "tick-tock" release cycle

- First month feature release, next bug fixes. Rinse. Repeat*.

- Great for implementing new features and ideas / fast / agile
- Don't get bug fix patch only releases... support yourself.

DataStax Engineering selects an OSS C* version as a candidate for DSE production release*.

- Run through QA / production testing
- DSE integration built
- Bugs fixed and passed back to the OSS community

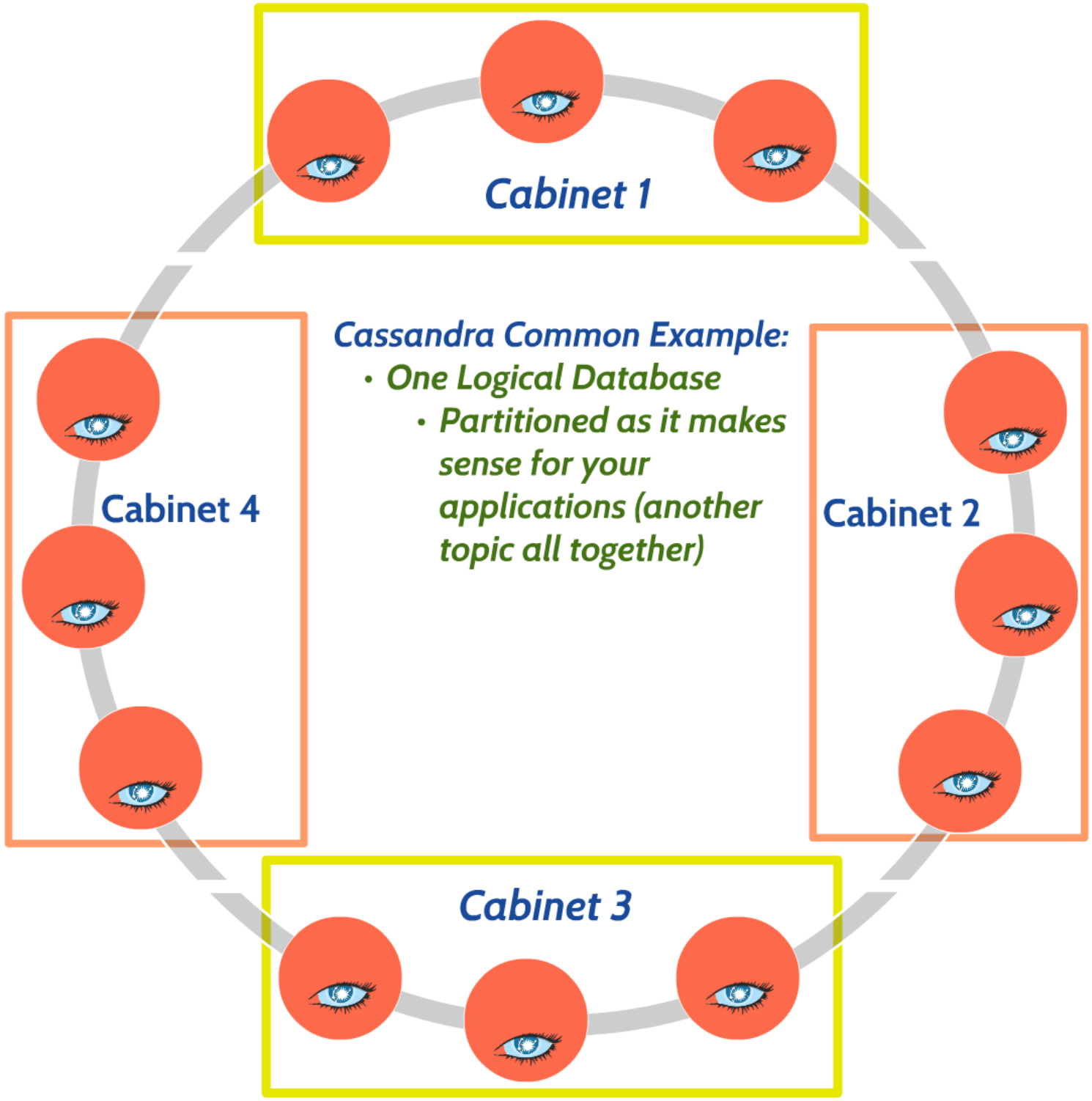


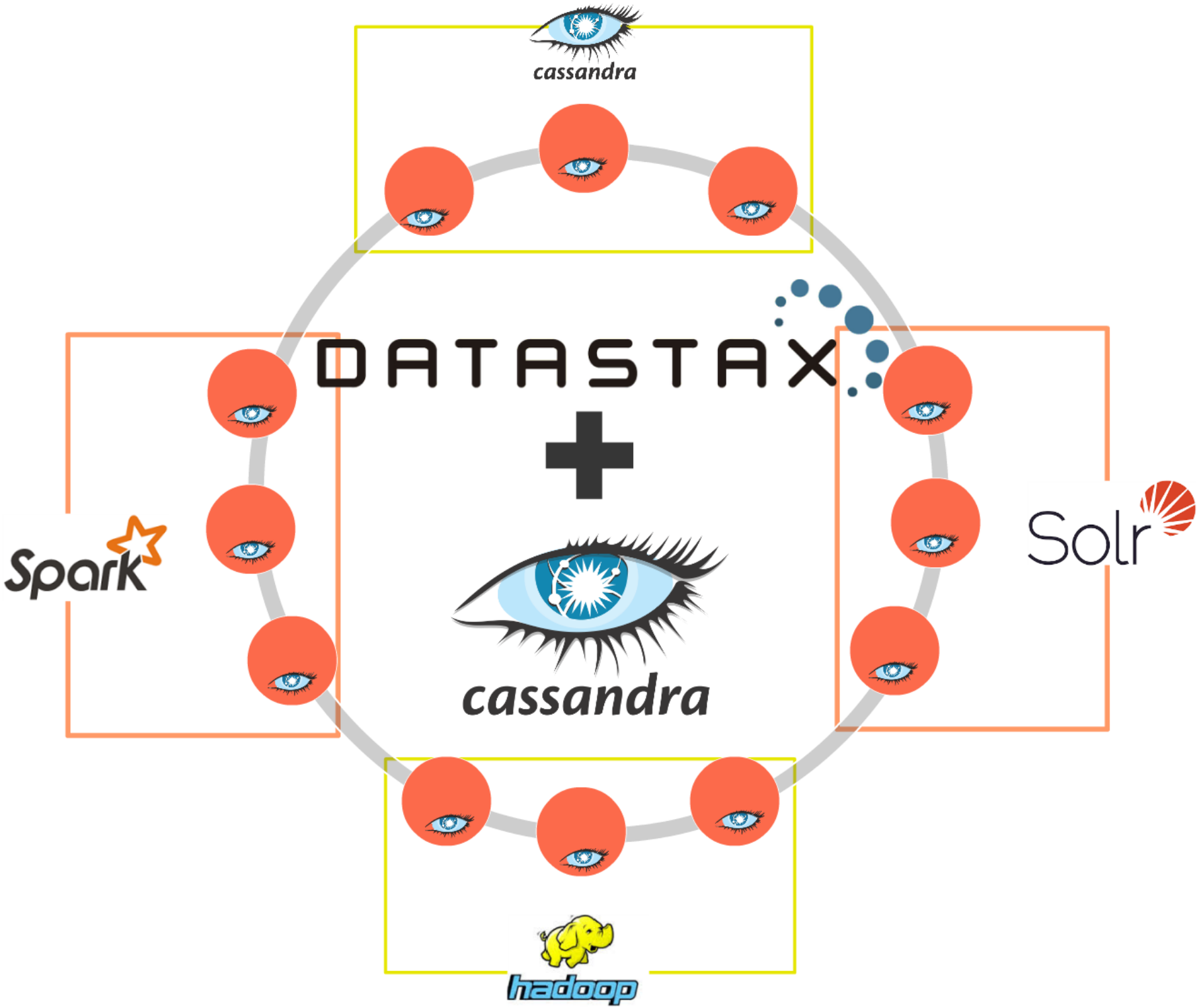
- When bug fixes are implemented in OSS is up to the community.
- OSS always has new features / fixes introduced.

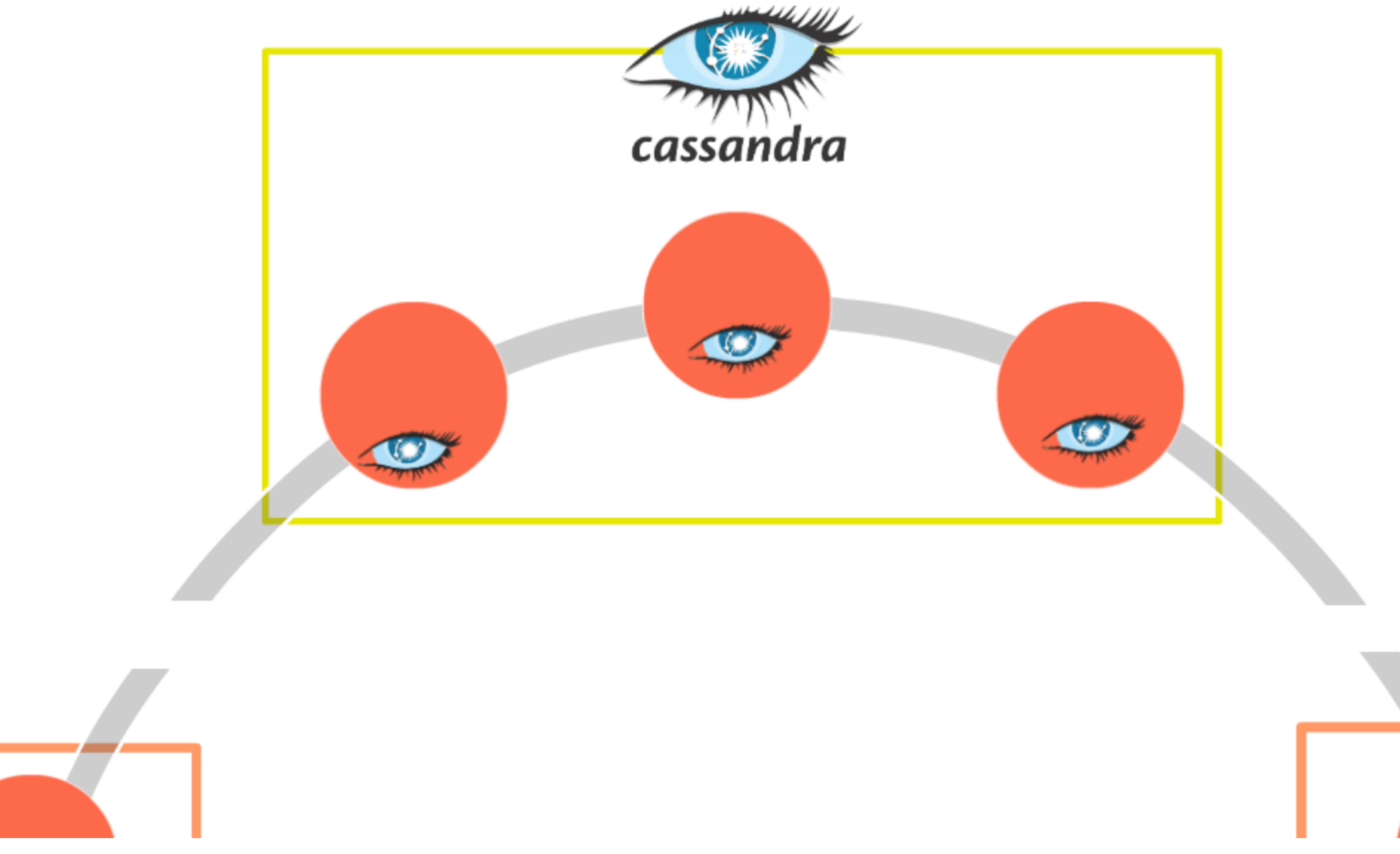
* Note: Only one OSS source for Cassandra

* DSE Engineering is a major part of the OSS community / only one vendor (DataStax)

Back to Cassandra and DataStax Enterprise functionality...


Cabinet 1
Cabinet 2
Cabinet 3
Cabinet 4
Cassandra Common Example:
• One Logical Database
• Partitioned as it makes sense for your applications (another topic all together)


cassandra
DATASTAX
+
cassandra
Spark
Solr
hadoop


cassandra

DataStax Enhancements For C* Environment

DSE Operational Enhancements

Automatic Management Services

- Capacity Planning Service
  - History used to forecast future resource needs
- Performance Service
  - Collects performance statistics by:
    - Slow queries by time threshold
    - Database, keyspace (schema), table
    - Cluster, data center, node
    - System (thread pool, etc.)
    - User activity
    - Histogram summaries
    - Solr statistics (as above, and Solr specific)
    - Data stored in tables for custom queries
- Backup / Restore Service
  - Slow queries by time threshold
  - Database, keyspace (schema), table
  - Cluster, data center, node
  - System (thread pool, etc.)
  - User activity
- Repair Service
- Best Practice Service

DSE Security

Automatic Management Services

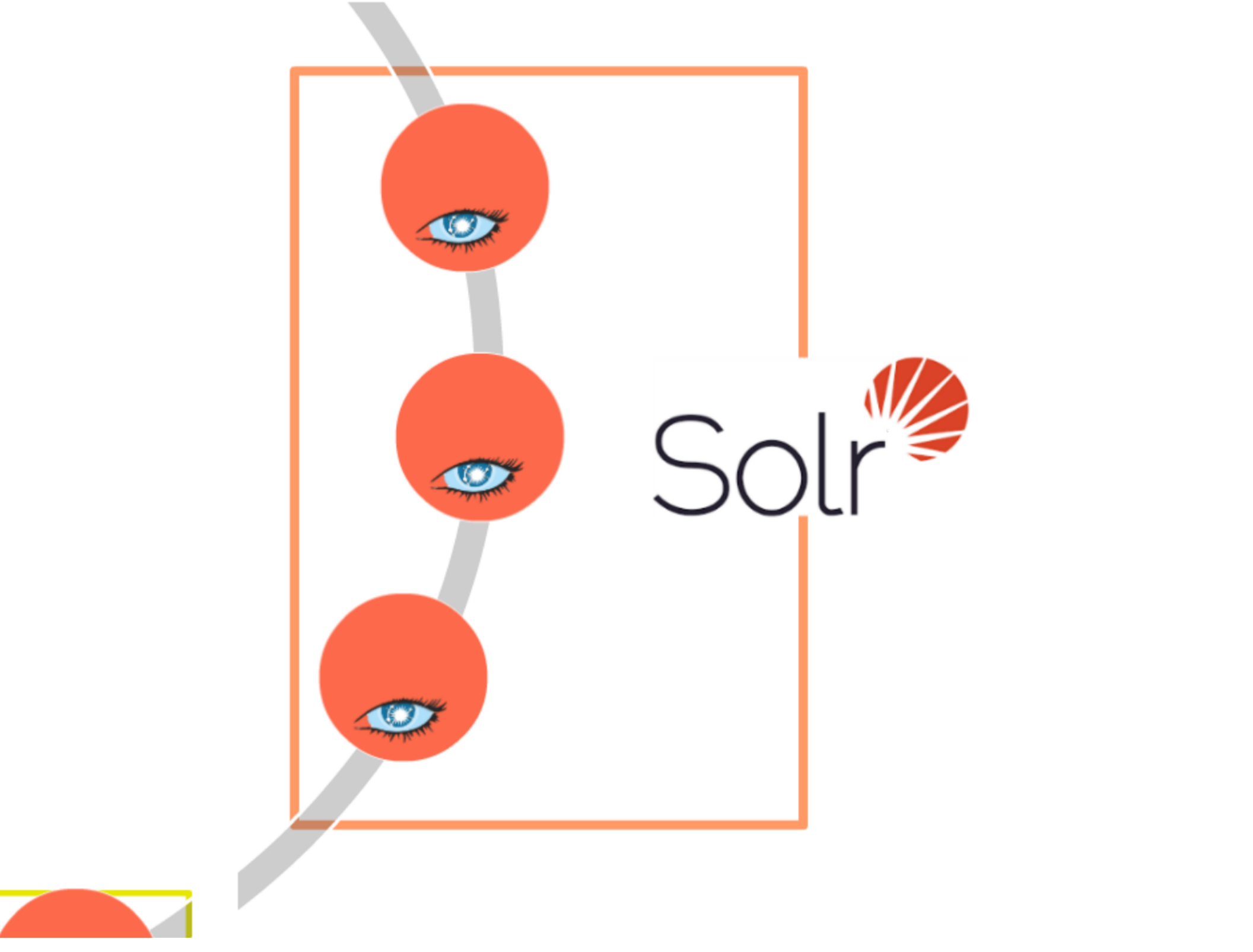
- Authentication
  - LDAP
  - Kerberos
- Authorization
- Encryption
  - Client-to-node
  - Node-to-node
  - At-rest
    - Data tables
    - System level (Vormetric)
- Audit Logging
  - CQL activity

DSE In-Memory Tables

Used for smaller, heavily trafficked tables
Commonly used for overwrite workloads

Sqoop For Import / Export

Not the only way to get data in / out


Solr

DataStax Integration with Solr

Solr Backed By C* Datastore

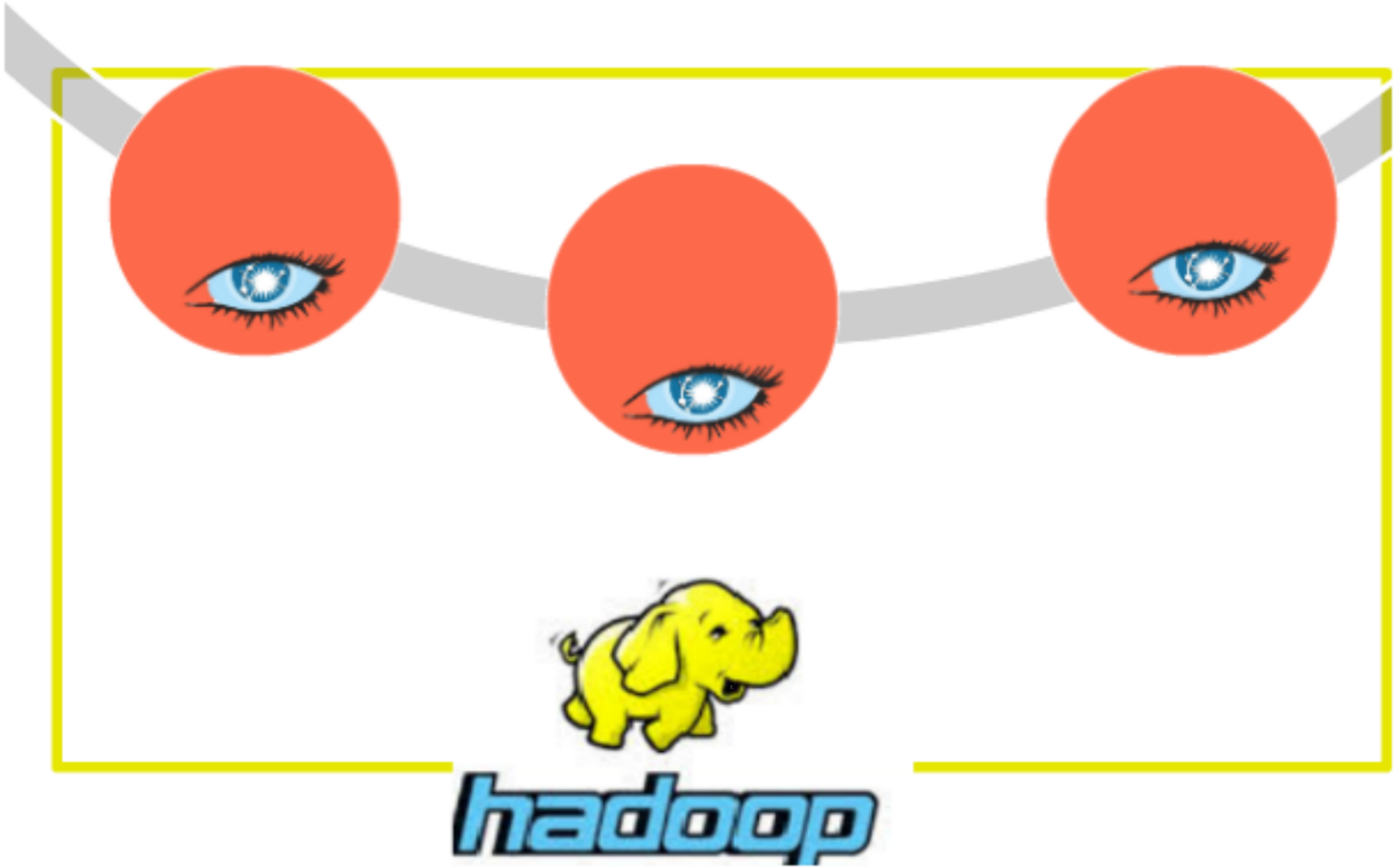
Automatic real-time Solr indexing

- No full re-index when data changes

Using Cassandra Allows For

- Scalability
  - Data part of C*
  - Cluster rebalancing
  - Linear scalability, no SPOF (no Zookeeper!)
- DSE Security
- Traditional Solr interface and CQL interface
  - inserts, updates, deletes apply to the *Cassandra Database* (*mutations* are durable)
  - Solr allows for queries to be run that are not part of the C* access path

It's Solr! Another way to interact with Cassandra stored data. All DSE integrations (Solr, Spark, Hadoop) work with the same Cassandra data.


hadoop

DataStax Integration with Hadoop(s)

C* Data Accessible With Hadoop

Map-Reduce Against C* Data

- No SPOF... and all the other C* goodness
- Run one or more job trackers across multiple data centers (workload isolation)

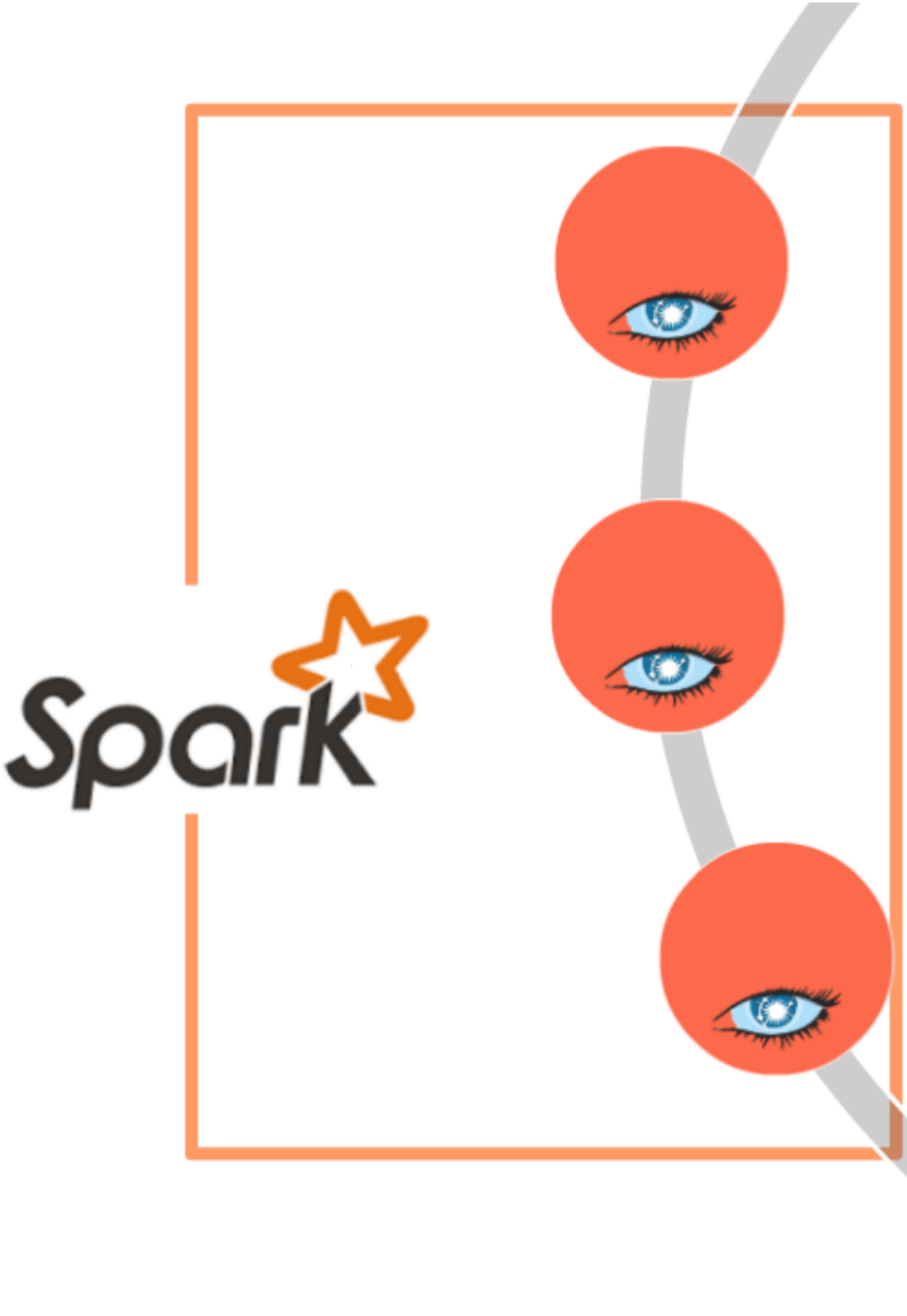
Two Integration Flavors

- Bring Your Own Hadoop (BYOH)
  - Integration with Cloudera and Hortonworks
- Apache Hadoop
  - Ships with DSE

Tools integration

- Hive / Hive ODBC
- Pig
- Mahout

Again, another way to interact with Cassandra stored data. All DSE integrations (Solr, Spark, Hadoop) work with the same Cassandra data.


Spark

DataStax Integration with Spark

Spark on C*

- Server-Side filters (where clauses)
- Cross-table operations (JOIN, UNION, etc.)
- Data transformation, aggregation, etc.

Spark Leverages C*

- Data location aware (speed)
- Spark Master leverages C* (no Zookeeper!)

Integration includes

- Spark streaming
- Scala support
- Spark Java API support
- Spark Python API (PySpark) support
- Spark SQL support

Databricks ODBC Driver

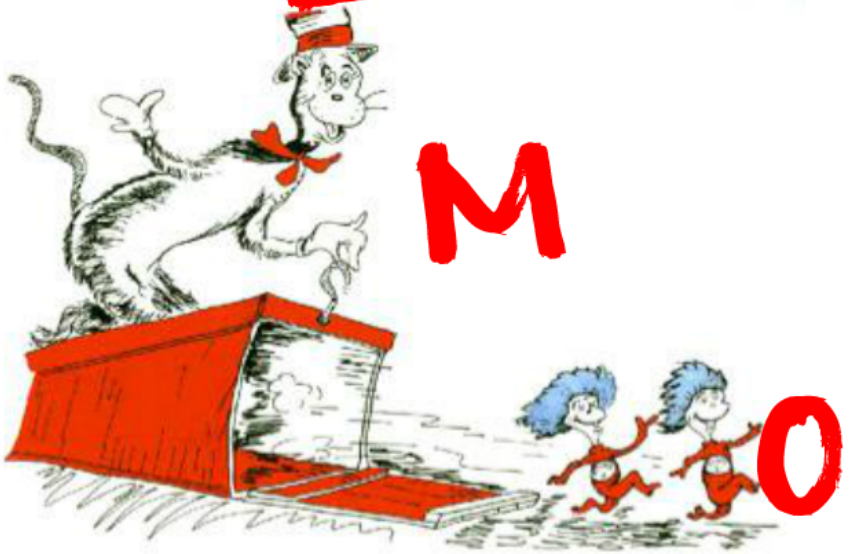
Much is OSS

... another way to interact with Cassandra stored data. All DSE integrations (Solr, Spark, Hadoop) work with the same Cassandra data.

Where does DSE / Cassandra fit?

- C* delivers always-on, high performance transactional data store (i.e. transactions).
- Solr often used for search (i.e. catalog, metadata, or other information stored in C*... people get creative with Solr search.
- Hadoop for batch processing, integration with "data lakes", and ETL for C* and other data.
- Spark for advanced analytics, integrations with other data sources, data ETL / movement, and general queries on C* and other data.
- Storm, Kafka, Flume...

DEMO



Demo Environment

8 Nodes w/ 7 Node Cluster!

- ✓ 6 DSE / C* nodes configured as two Data Centers, named **ORANGE** and **BLUE**.
- ✓ 1 Spark / Solr node configured as Data Center named **SPARK**
- ✓ 1 DataStax **OpsCenter** Node for monitoring

Dang. That sounds impressive!

No, not really impressive...

X-over Cable on USB Ethernet

Laptop

Windows 7 / 16 GB RAM

2x VMWare

- 8 GB RAM
  - Ubuntu
  - DSE 4.6 / Spark / Solr
  - Data Center **SPARK**
  - 2 GB RAM
- Ubuntu
  - DSE 4.6/ OpsCenter
  - Data Center **OpsCenter**

NUC

Ubuntu 16 GB RAM

6x VMWare

- 2 GB RAM
- Ubuntu
- DSE 4.6
- 2 Data Centers, 3 VM's each
  - **BLUE**
  - **ORANGE**

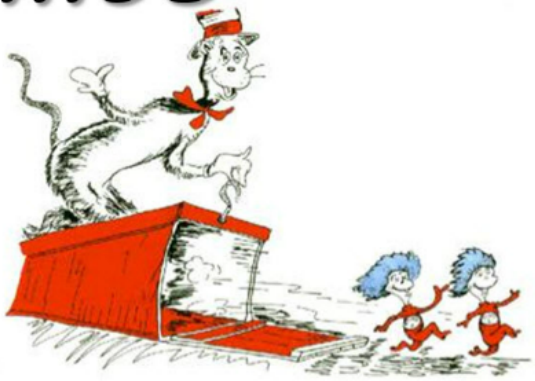
Light Bar

Blink when VM's are running

1st Orange = 1st **ORANGE** Data Center VM, etc.

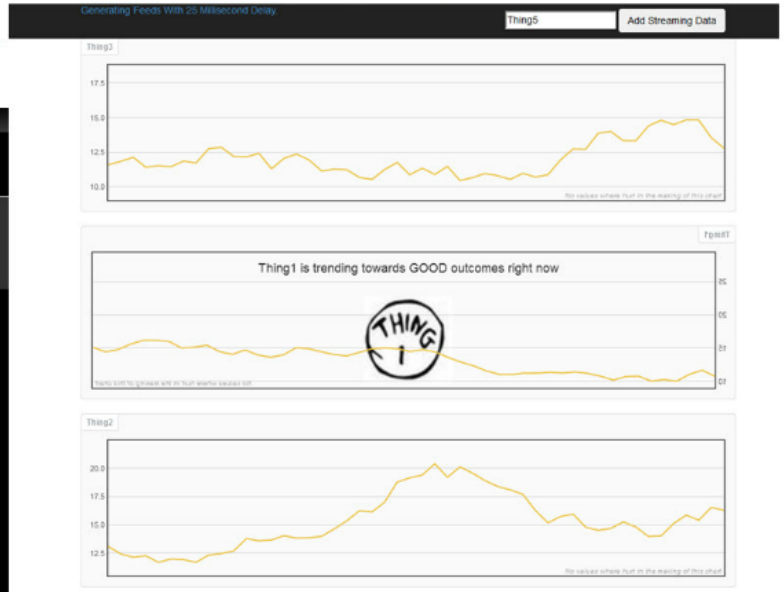
- **ORANGE = Orange DC**
- **BLUE = Blue DC**
- **Green = OpsCenter**

DemoS



Continuous Data Stream

1. Fail nodes
2. Run CQL Queries
3. Run Spark SQL Queries
4. Solr CQL Queries



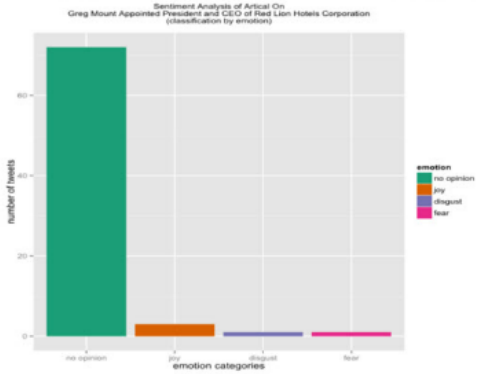
Application uses Play framework, Akka, Scala, java, C* java driver

DemoS

Other Interfaces

1. C* Solr Queries from R
   - R Package
   - Spark ODBC Tableau

```
url<-'http://192.168.73.128:8983/solr/bc1.xn/select?q=txt%3A%22execs+appointed%22~15
in.df<-solr_search(q='*:*', rows=2, fl='id,title,txt', url=url, key=key)
inText<-in.df[1,3]
```



Emotion

Polarity