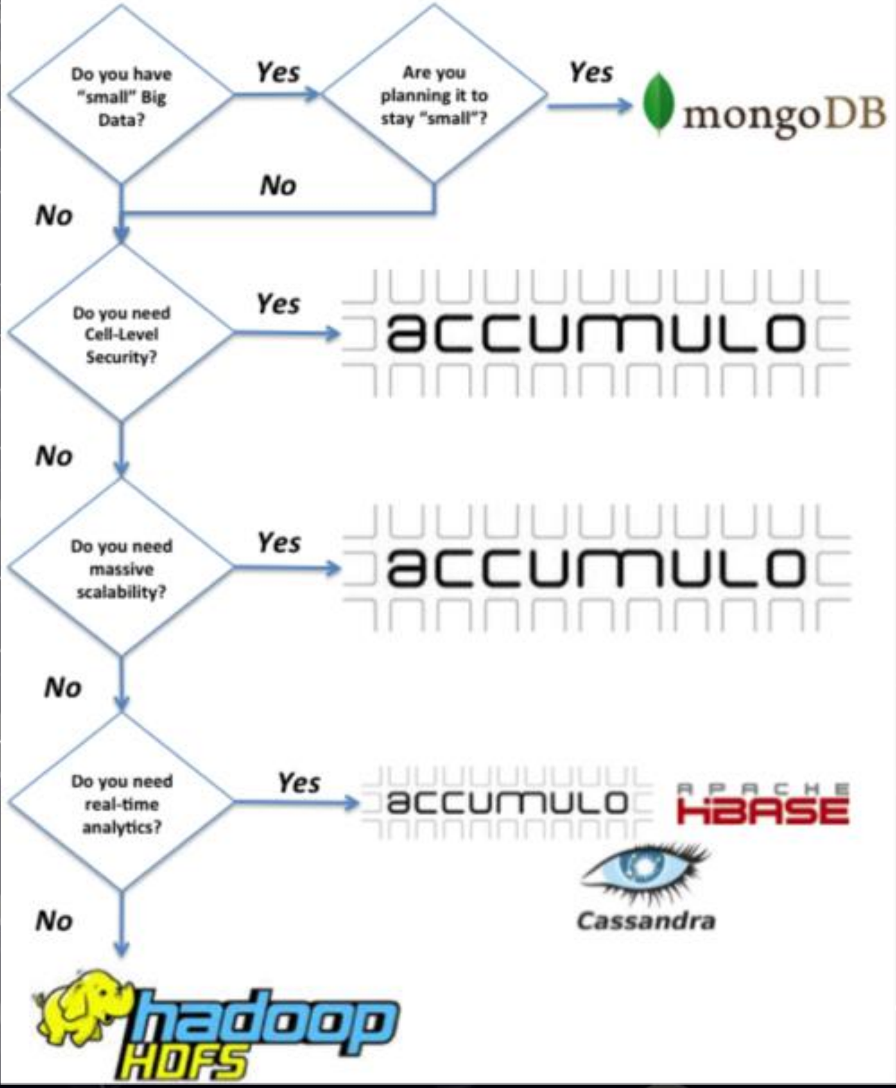
sqrrl and Accumulo

Presented by: John Dougherty, CIO
5/21/2013

Which NoSQL solution?

Simplified NoSQL Decision Tree



Do you have "small" Big Data?

Yes → Are you planning it to stay "small"? Yes → mongoDB

No

Do you need Cell-Level Security? Yes → accumulo

No

Do you need massive scalability? Yes → accumulo

No

Do you need real-time analytics? Yes → accumulo, Apache HBase, Cassandra

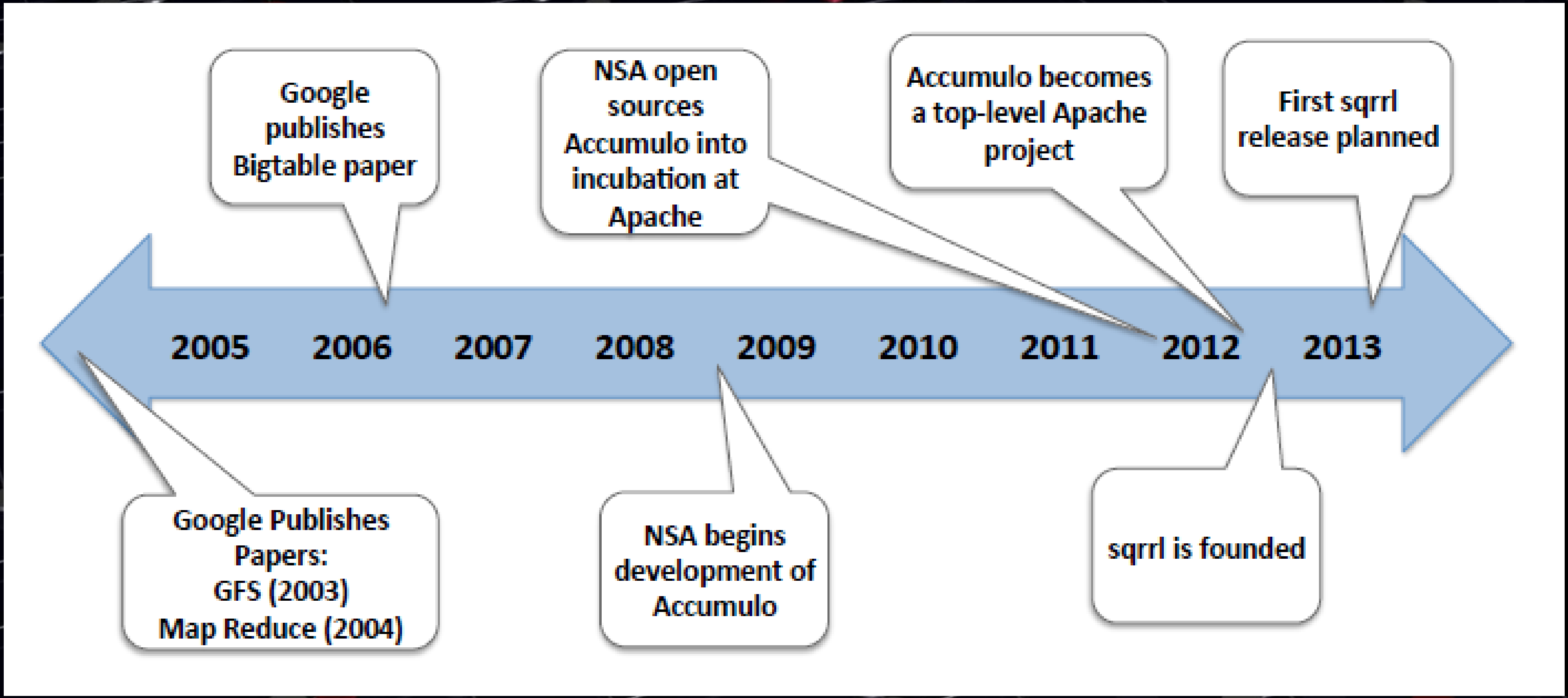
No → hadoop HDFS

There are a lot of places to fit sqrrl, and Accumulo.

What is sqrrl?

- Based on Accumulo
- A proven, secure, multi tenant, data platform for building real-time applications
- Scales elastically to tens of petabytes of data and enables organizations to eliminate their internal data silos
- Seamless integration with Hadoop, and most of its variants
- Providing a supply for a much needed security demand (ground-up security)
- Already deployed and utilized by defense and government industries

A history of sqrrl



Google Publishes Papers:
GFS (2003)
Map Reduce (2004)

Google publishes Bigtable paper

NSA begins development of Accumulo

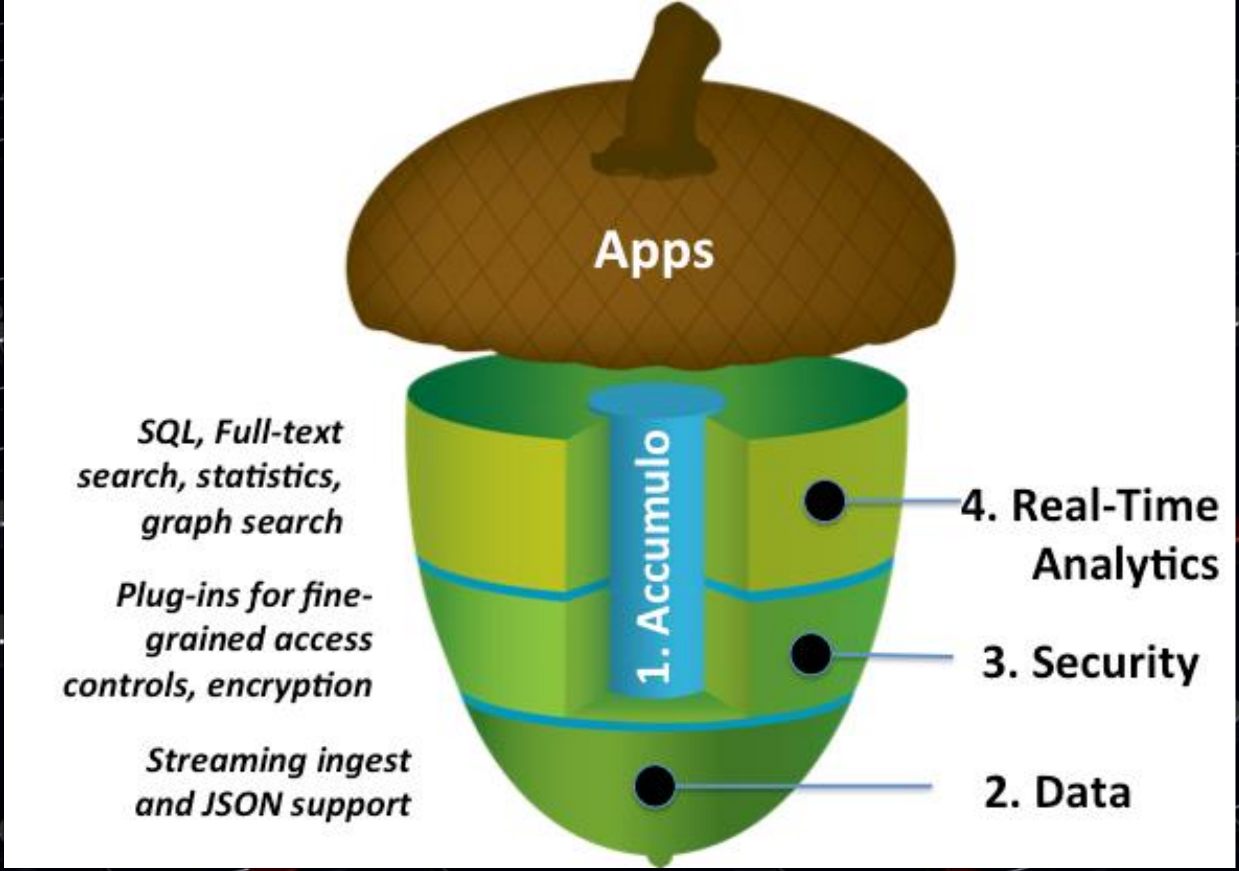
NSA open sources Accumulo into incubation at Apache

Accumulo becomes a top-level Apache project

sqrrl is founded

First sqrrl release planned

A sqrrl's architecture

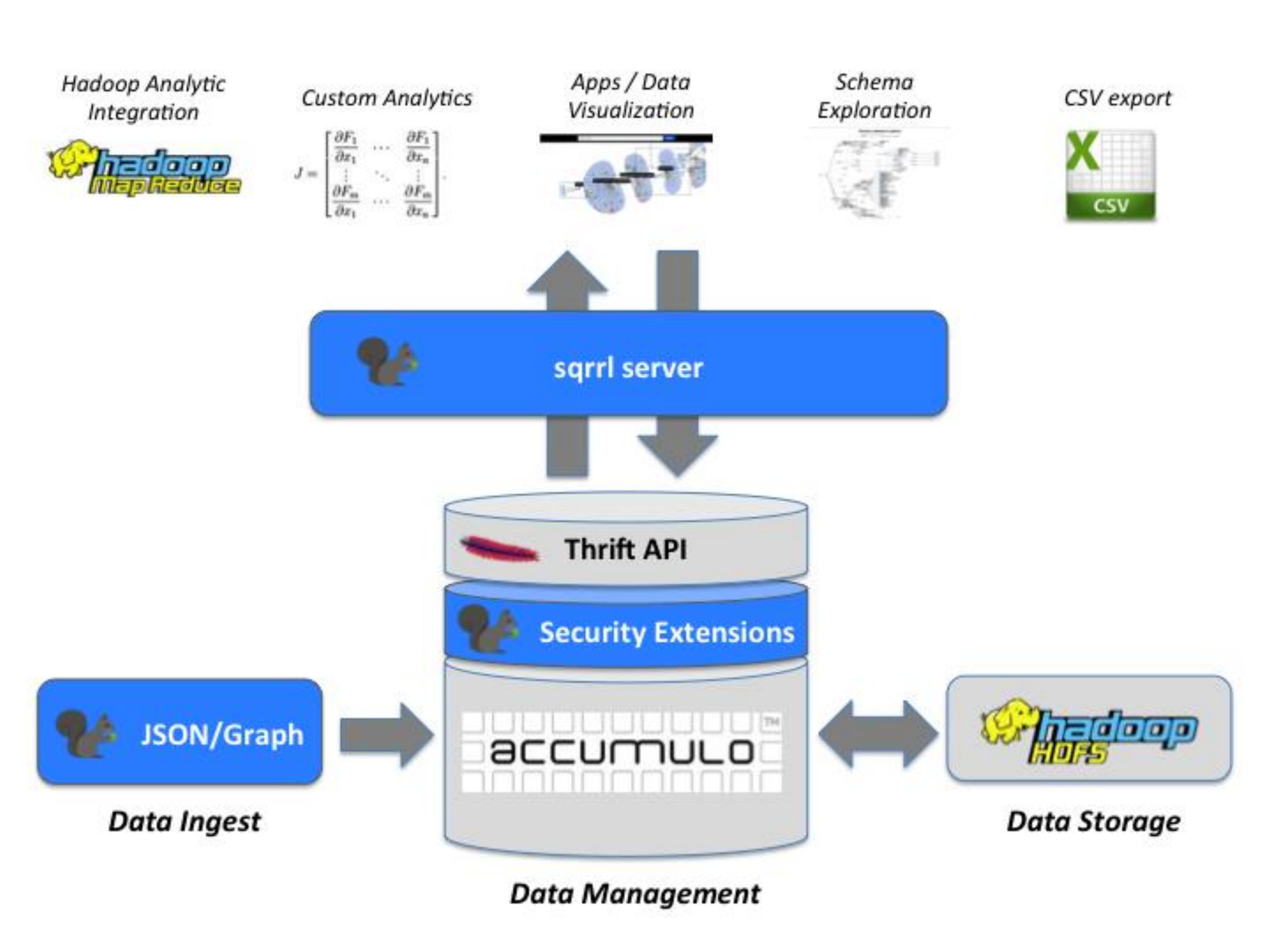

Apps
SQL, Full-text search, statistics, graph search
Plug-ins for fine-grained access controls, encryption
Streaming ingest and JSON support
1. Accumulo
4. Real-Time Analytics
3. Security
2. Data

© sqrrl, Inc.

What is Accumulo?

- Development began at the NSA in 2008
- Base foundation for sqrrl
- Cell-level security reduces the cost of app development, circumnavigating complex, sometimes impossible, legal or policy restrictions
- Provides the ability to scale to >PB levels
- Highly adaptive schema and sorted key/value paradigm
- Stores key/value pairs in parsed, sorted, secure controls

Where does Accumulo fit?



© sqrrl, Inc.

How does Accumulo provide security?

- Security Labels are applied to keys
- Cell-level security is implemented to allow for security policy enforcement, using data labeler tags
- These policies are applied when data is ingested
- Tablets contain data, are controlled using security policies
- Stores key/value pairs in parsed, sorted, secure controls, a 5-tuple key system

Accumulo Security (cont.)

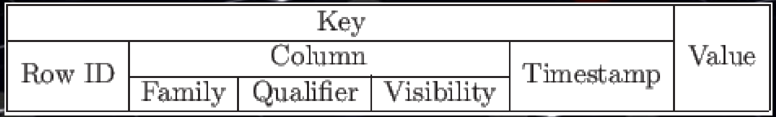
Why Cell-Level Security Is Important:

Many databases insufficiently implement security through row-and column-level restrictions. Column-level security is only sufficient when the data schema is static, well known, and aligned with security concerns. Row-level security breaks down when a single record conveys multiple levels of information. The flexible, fine-grained cell-level security within Sqrrl Enterprise (or its root Accumulo) supports flexible schemas, new indexing patterns, and greater analytic adaptability at scale.

Accumulo Security (cont.)

An Accumulo key is a 5-tuple key, consisting of:

- **Row:** Controls Atomicity
- **Column Family:** Controls Locality
- **Column Qualifier:** Controls Uniqueness
- **Visibility Label:** Controls Access
- **Timestamp:** Controls Versioning



Key					Value
Row ID	Column			Timestamp	
	Family	Qualifier	Visibility		

(Values are byte arrays)

Keys are sorted:

- **Hierarchically:** Row first, then column family, and so on
- **Lexicographically:** Compare first byte, then second, and so on

Accumulo Security (cont.)

An example of column usage

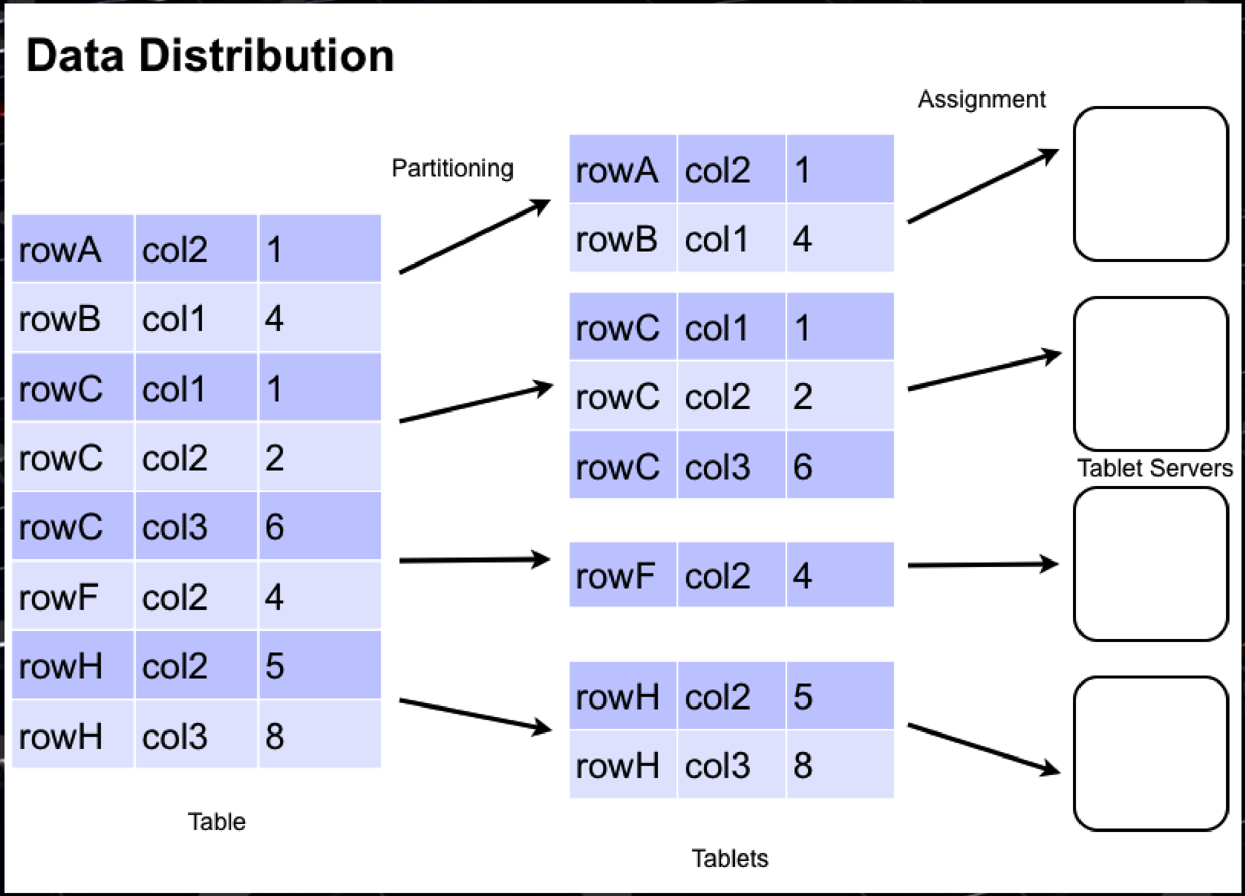
Row	Col. Fam.	Col. Qual.	Visibility	Timestamp	Value
Jane Doe	Friends	John Doe	JD	20121130	
Jane Doe	Phone Number	555-1212		20090115	
John Doe	Friends	Jane Doe	JD	20121201	
John Doe	Notes	PCP	PCP_JD	20120912	Patient suffers from an acute …
John Doe	Test Results	Cholesterol	JD|PCP_JD	20120912	183
John Doe	Test Results	Mental Health	JD|PSYCH_JD	20120801	Pass
John Doe	Test Results	Mental Health	PSYCH_JD	20120801	Crazy!
John Doe	Test Results	X-Ray	JD|PHYS_JD	20120513	1010110110100…

Accumulo Architecture

Accumulo servers (tablets) utilize a multitude of big data technologies, but their layout is different than Map/Reduce, HDFS, MongoDB, Cassandra, etc. used alone.

- Data is stored in HDFS
- Zookeeper is utilized for configuration management
- SSH, password-less, node configuration
- An emphasis, more of an imperative, on data model and data model design

Accumulo Architecture (cont.)



Data Distribution

Partitioning

Assignment

rowA	col2	1
rowB	col1	4
rowC	col1	1
rowC	col2	2
rowC	col3	6
rowF	col2	4
rowH	col2	5
rowH	col3	8

Table

rowA	col2	1
rowB	col1	4

rowC	col1	1
rowC	col2	2
rowC	col3	6

rowF	col2	4

rowH	col2	5
rowH	col3	8

Tablets

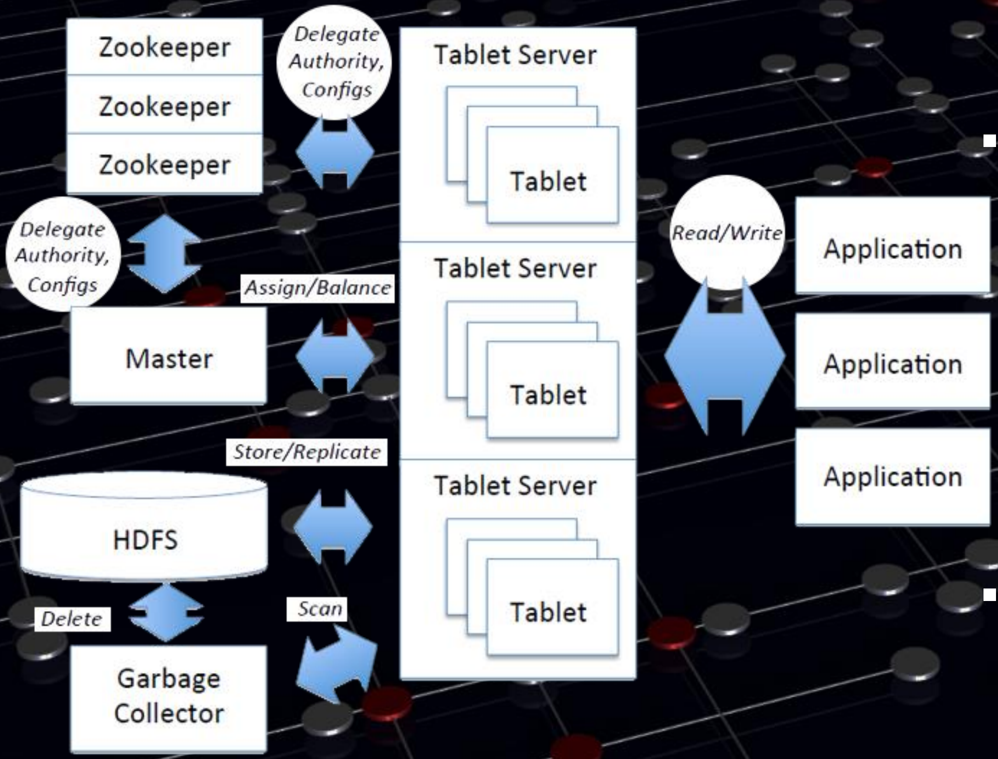
Tablet Servers

Tablets

- Partitions of tables, collections of sorted key/value pairs
- Held and managed by Tablet Servers

Accumulo Architecture (cont.)

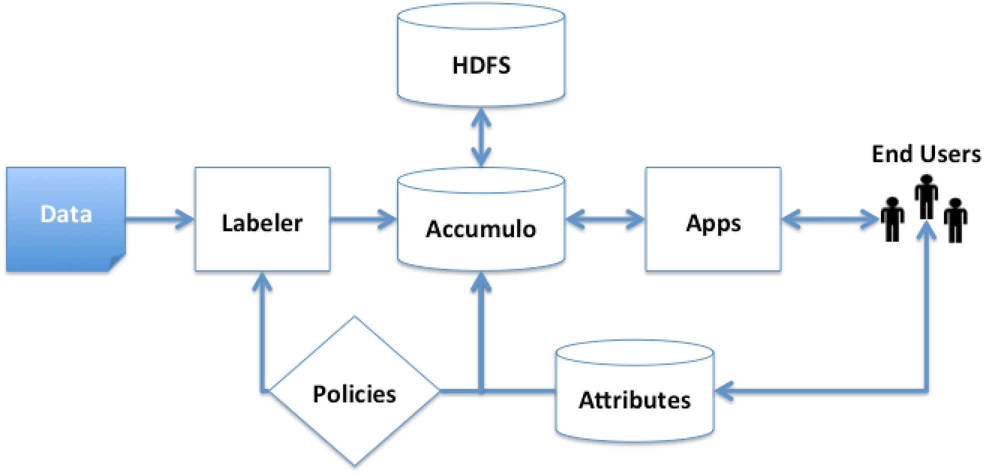
Tablet Servers



- Receive writes, responds to reads, from clients
- Writes to a write-ahead log, sorting new key/value pairs in memory, while periodically flushing sorted key/value pairs to new files in HDFS
- Managed by Master
  - Responsible for detecting and responding to Tablet Server failure, load balancing
  - Coordinates startup, graceful shutdown, and recovery of write-ahead logs
- Zookeeper
  - An apache project, open source
  - Utilized for distributed locking mechanism, with no single point of failure

Integration with users/access

The visibility labels are a feature that is unique to Accumulo. No other database can apply access controls at such a fine-grained level.



- Labels are generated by translating an organization's existing data security and information sharing policies into Boolean expressions

1. Gather an organization's information security policies and dissecting them into data--centric and user--centric components
2. As data is ingested into Accumulo, a data labeler tags individual key/value pairs with the appropriate data--centric visibility labels based on these policies.
3. Data is then stored in Accumulo where it is available for real--time queries by operational applications. End users are authenticated through these applications and authorized to access underlying data
4. As an end user performs an operation via the app (e.g., performs a search request), the visibility label on each candidate key/value pair is checked against his or her attributes, and only the data that he or she is authorized to see is returned.

Making sqrrl work

sqrrl’s extensibility of Accumulo allows it to process millions of records per second, as either static or streaming objects

These records are converted into hierarchical JSON documents, giving document store capabilities

Passing this data to the analytics layer is designed to make integration and development of real-time analytics possible, and accessible

Combining access at the cell level, with Accumulo, sqrrl integrates Identity and Access Management (IAM) systems (LDAP, RADIUS, etc.)

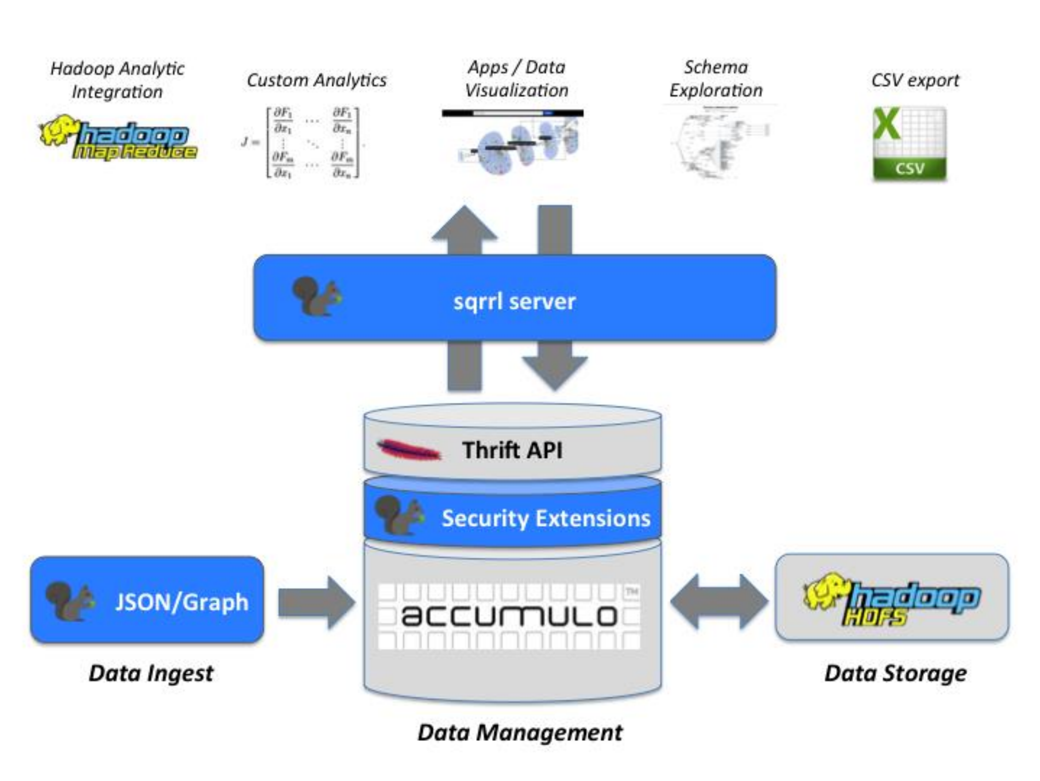
Making sqrrl work (cont.)

Sqrrl process

Apache thrift:

Enables development in diverse language choices

Apache Lucene: Custom iterators, providing developers with real-time capabilities, such as full-text search, graph analysis, and statistics, for analytical applications and dashboards.



Data Ingest:

JSON, or graph format

HDFS:

File Storage system, compatible with both open source (OSS) and commercial versions

Apache Accumulo:

The core of transactional and online analytical data processing in sqrrl

Who is sqrrl for?

CTOs/CIOs: Unlock the value in fractured and unstructured datasets across your organization

Developers: More easily create apps on top of Big Data and distributed databases

Infrastructure Managers: Simplify administration of Big Data through highly scalable and multitenant distributed systems

Data Analysts: Dig deeper into your data using advanced analytical techniques, such as graph analysis

Business Users: Use Big Data seamlessly via apps developed on top of *sqrrl enterprise*

sqrrl/Accumulo wrap-up

Accumulo bridges the gap for security perspectives that restrict a large swath of industries

sqrrl combines the best of available technologies, develops and contributes their own, and designs big apps for big data.

Accumulo Setup:

1. Installation of HDFS and ZooKeeper must installed and configured
2. Password-less SSH should be configured between all nodes (emphasized master <> tablet)
3. Installation of Accumulo (from http://accumulo.apache.org/downloads/ using http://accumulo.apache.org/1.4/user_manual/Administration.html#Installation

Or get started using their AMI (http://www.sqrrl.com/downloads#getting-started)