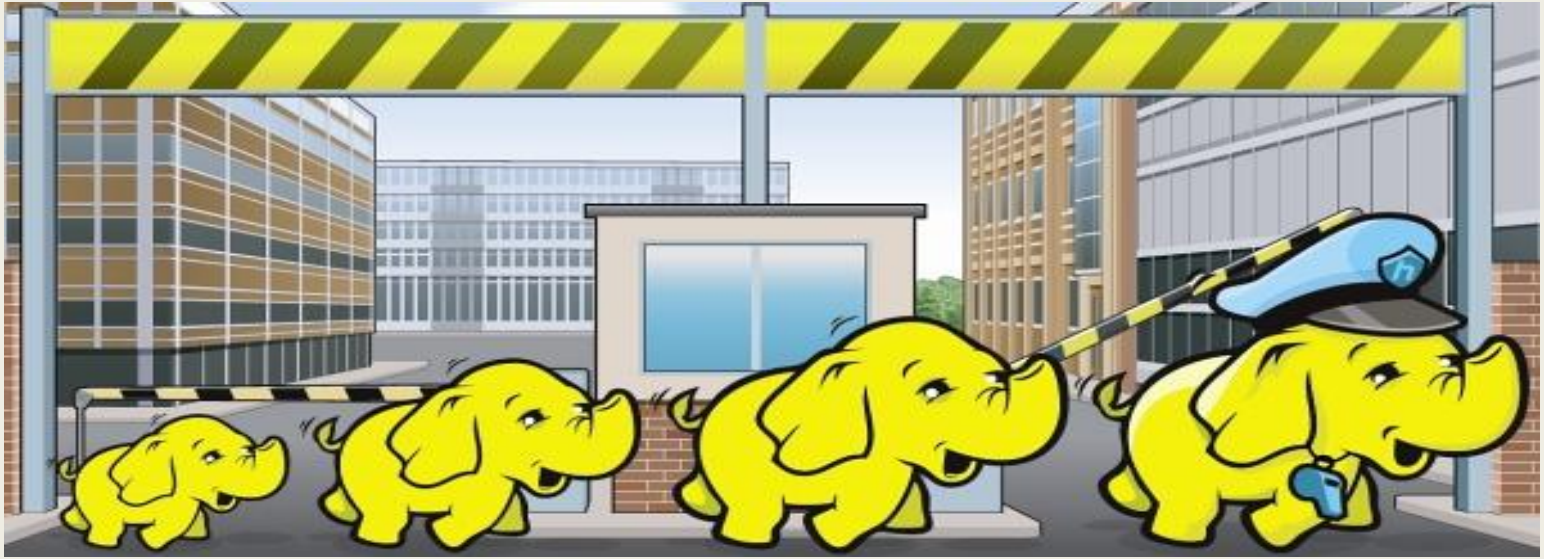
Big Data Inference and Security Concerns



Presented by Group 1

Credit: Kevin T. Smith [Smith]

Data in the Modern World

Some statistics

More data has been created in the past two years than in the entire course of human history

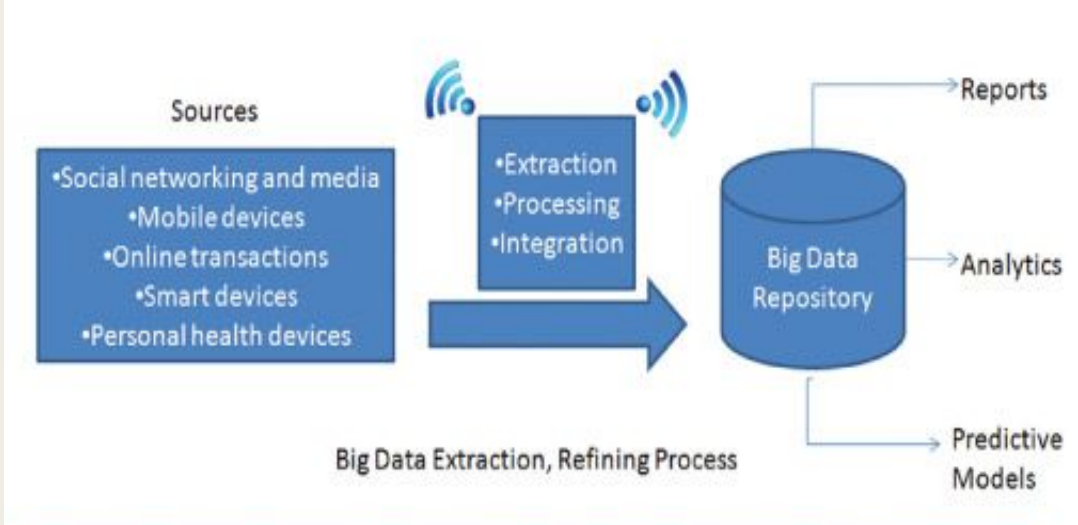
1.2 Trillion searches on Google per year

Facebook users send 31.25 million messages and 2.77 million videos per day

Users on Facebook's mobile app: 1.44 billion. The population of the Earth is estimated to be 7.4 billion…~15% of Earth

By 2020 ⅓ of all data will pass through cloud-based hardware

[Marr][Facebook]



Credit: PCQuest [PCQuest]

History of Hadoop

- Began with papers by Google in 2003 and 2004 on a distributed file system (Hadoop File System) and processing of data (MapReduce)
- Doug Cutting, open source web crawler, problem with computations - non-trivial scaling characteristics [T. White]
- Yahoo! And Dreadnaught and widespread adoption

How Hadoop Works: An Overview I

Hadoop is composed of two parts:

-A distributed file system Hadoop File System (HDFS)

-A data processing framework MapReduce

HDFS splits data up into 128MB blocks and distributes them among storage clusters, replicating for failover. A *namenode* keeps a tree structure where all the *datanodes* are in the

How Hadoop Works: An Overview II

MapReduce is a batch query processor [T. White]

MapReduce maintains a JobTracker on the central, master node and TaskTracker that is in the location where the data is actually stored

The JobTracker partitions the job and maps the data to the data cluster and the TaskTracker where it is actually “reduced”

The reductions are returned and combined together with other reductions using shuffle and sort functions on the key for each (K,V)

How Hadoop Works: Overview III

Files are passed in from HDFS to MapReduce usually line by line

1. The mapping phase runs on unsorted (K,V)
2. Combiners combine together items based on equivalent keys
3. Data from all the mappers, grouped together by key, are passed to the reducers and sorted by key
4. The reducer reduces the sorted, like-key value pairs and outputs the results where they are combined

Why Hadoop?

Social media companies have become very interested in mining user data for insights and targeted marketing [Lohr]

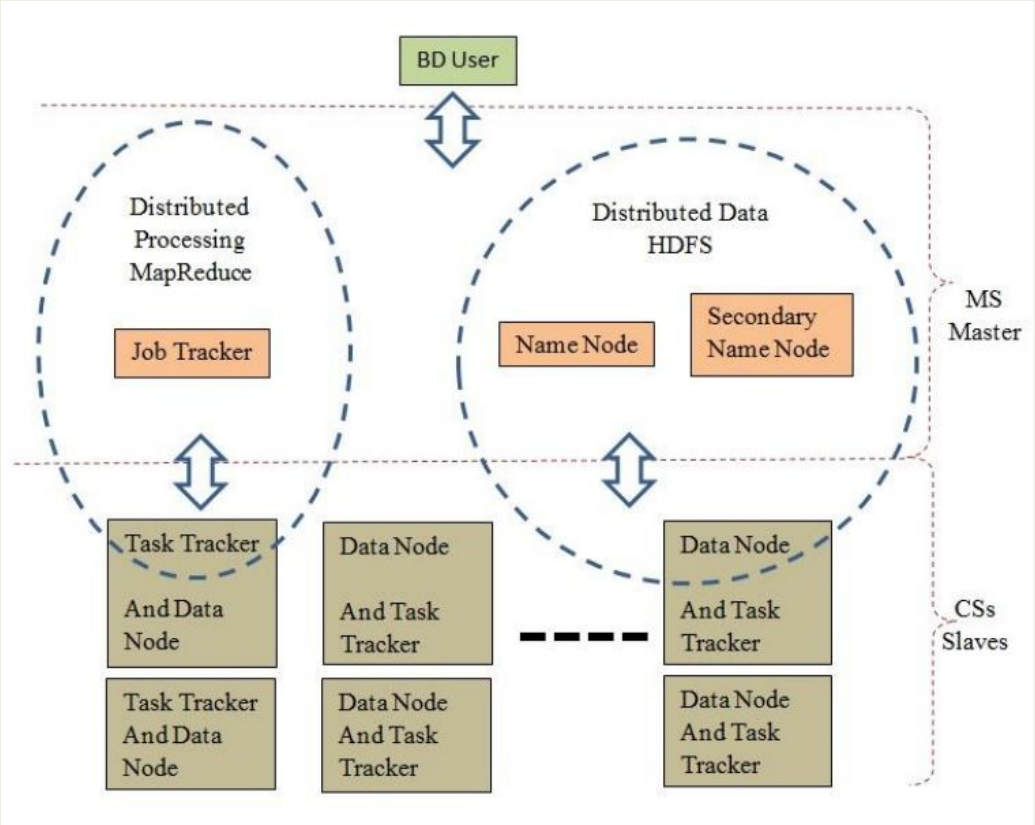
Lots of unstructured and semi-structured data

Computations that require processing large portions of the sample data sets

Expedient scaling and use of currently available hardware [admin-magazine]

Information gained from big data processing can enhance profit by as much as 60% [Forbes]

Hadoop Model

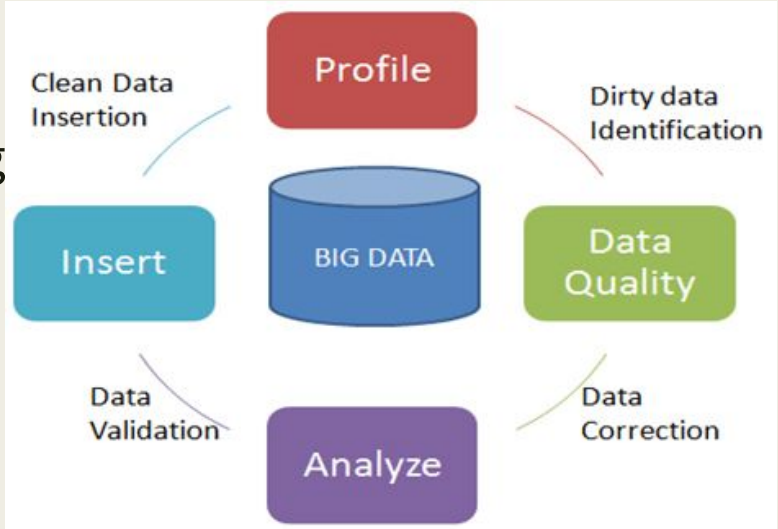


Credit: Vincent Hu, et al. [Hu, et al.]

Credit: PCQuest [PCQuest]

Cycle of data ingestion and processing

Data Ingestion: Process of Obtaining, Importing & processing data for later use



Credit: Sudhakara St. [St]

Data Ingestion in Hadoop

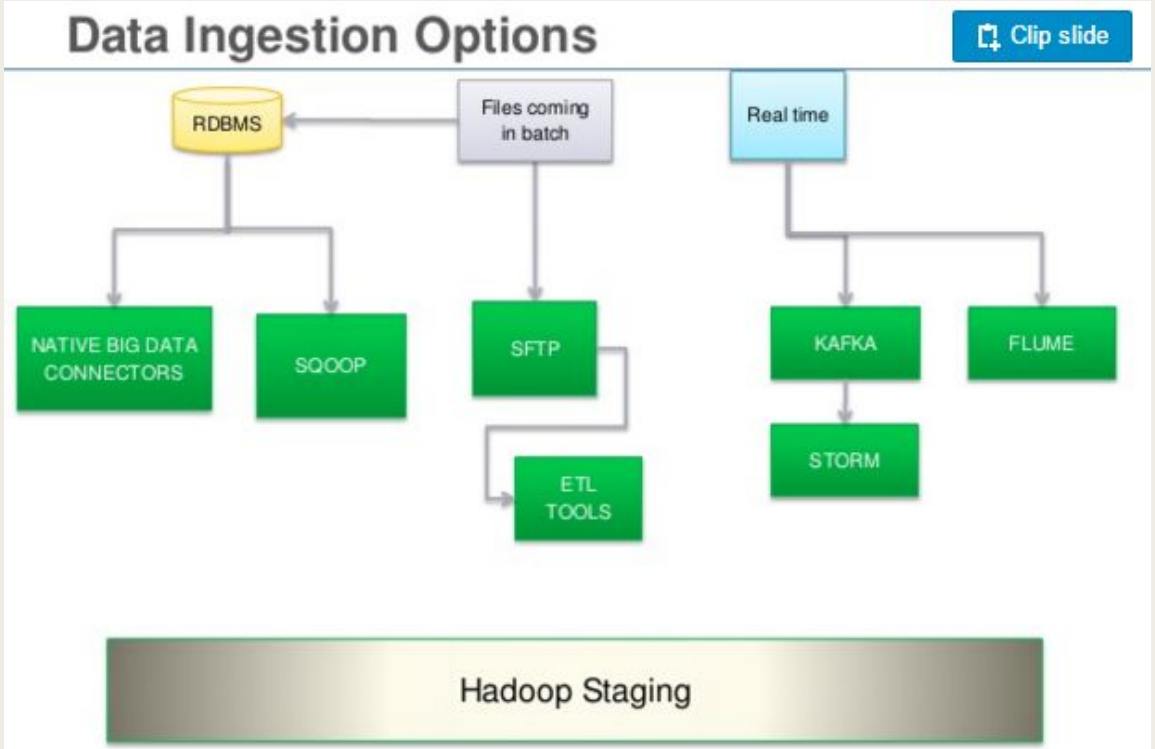


Fig: Data Ingestion techniques

Credit: Vinod Nayal [Nayal]

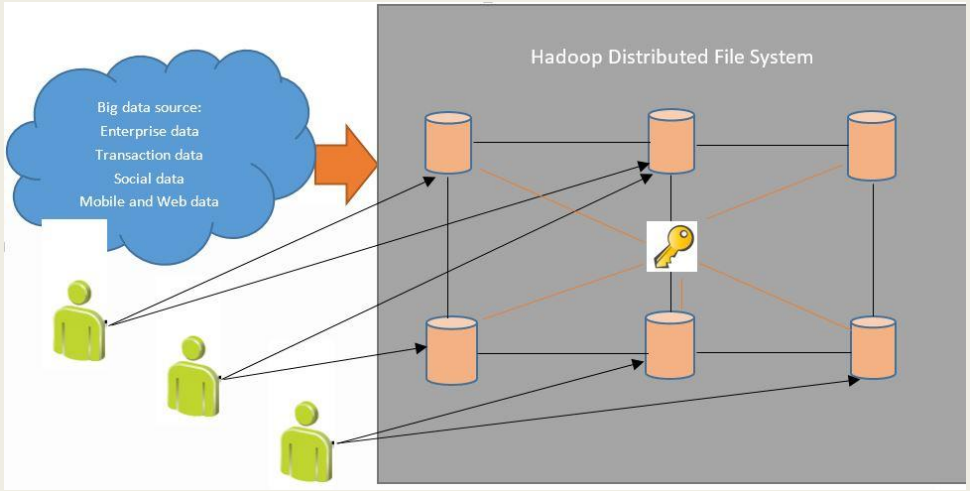
Gobblin’- Framework to solve Data Ingestion problem

- Problem : Large amount of data for ingestion and integration at high velocity and high quality
- [Lin Qiao]Gobblin’s Solution:
    - Source Integration
    - Processing Paradigm
    - Data quality Assurance
    - Extensibility
    - Self Service

Credit: [NTRecycling]

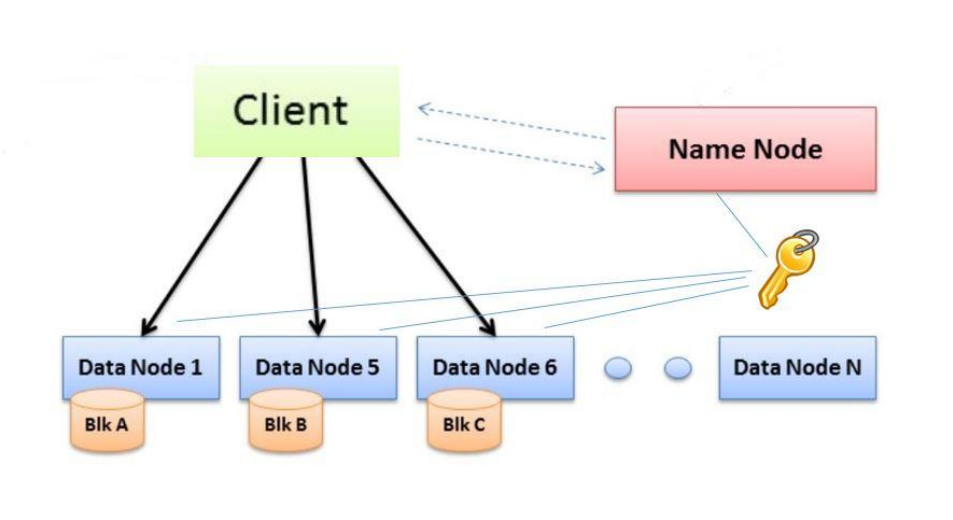
Hadoop Distribution Problem

- Hadoop, as a powerful open source framework, is deployed by many organizations due to its cost effective and scalable characteristics
- Hadoop distributes the processing of large data sets across clusters of computers to deliver high-availability and good performance.



What happened across these distribution environment?

Hadoop Authentication and key management



- Name node and data nodes share the same symmetric authentication key.
- Key management is also critical in encryption process
- Key compromise could lead to failure to data decryption and affect availability of data

Hadoop data storage and masking

- Sensitive data stores distributedly in data nodes in Hadoop clusters
- Original Hadoop does not provide data encryption features
- Different users may have different permissions to different nodes
- Unauthorized users may attack nodes in clusters

Hadoop data transmitting security

- Data transmitting during process:Map Reduce process from job tracker to task tracker
- Data transmitting from the namenode to data nodes
- Data transmitting through the network: HTTP, RPC, DTP and JDBC

Hadoop access control

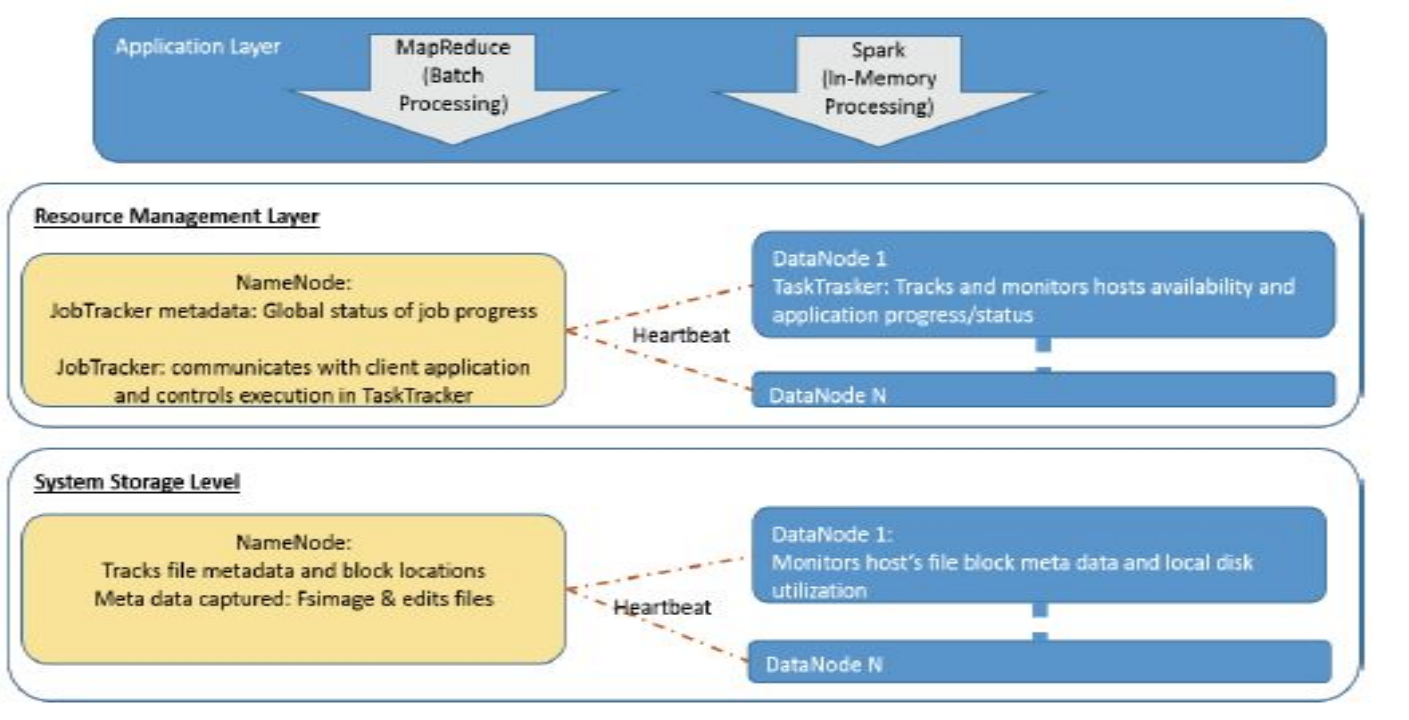
- **HDFS Permission Guide** allow administrators set permissions for directories and files in the HFDS for different groups and individuals levels.
- **Service Level Authorization** verify that the clients are authorized to access to the service which Hadoop provides
- **Sentry** works with Hive, HDFS data tables and other components to provide more granular permissions management for data and metadata in Hadoop clusters
- However, organizations also contract with some third parties or use third party tools, the issue is most service vendors and providers are more powerful than organizations in terms of technology. How to manage their access to our data?

Credit: Dreamstime.com/T.L. Furrer [Furrer]

Data Provenance and Distributed System

- One of the recent trends is to use data provenance to detect the data spillage in the big data system
- Name node provide commands that help to collect data about the location of the file blocks and IDs of each blocks based on their log files.
- The “Forensics Tool Kit” have been used to analyze and discover the data spillage in the system

Data Provenance and Distributed System



Data Provenance in Hadoop Ecosystem

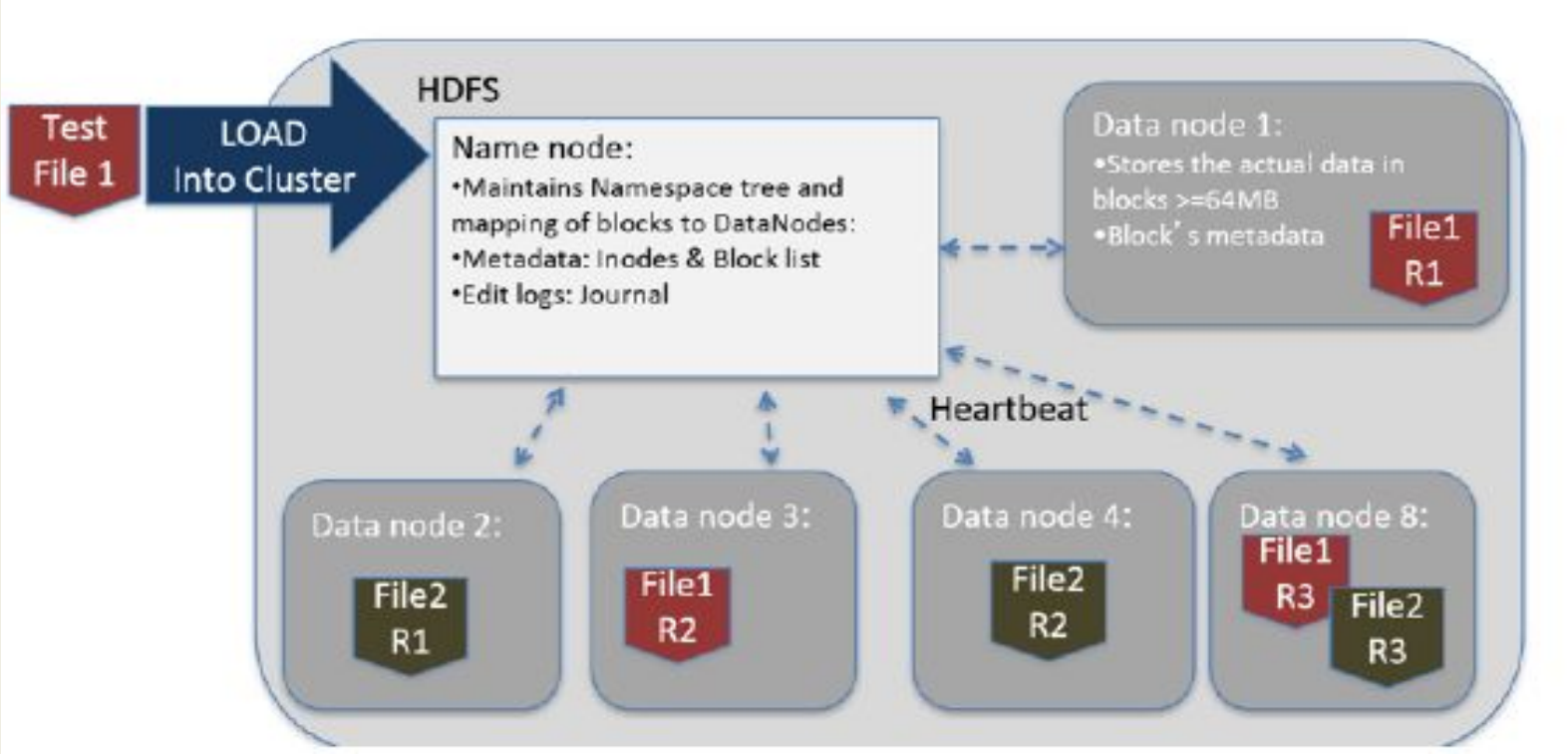
Credit: [Alabi, et al.]

Data Spillage with Approach Scenario

- Inject a tagged file and track it using the data provenance

There are two main accomplish based on that:

- Test the data provenance by knowing the location of the tagged file location
- Using Forensics Tool to track the tagged file after it was deleted from the hadoop cluster



User induced data spillage event into Hadoop Cluster

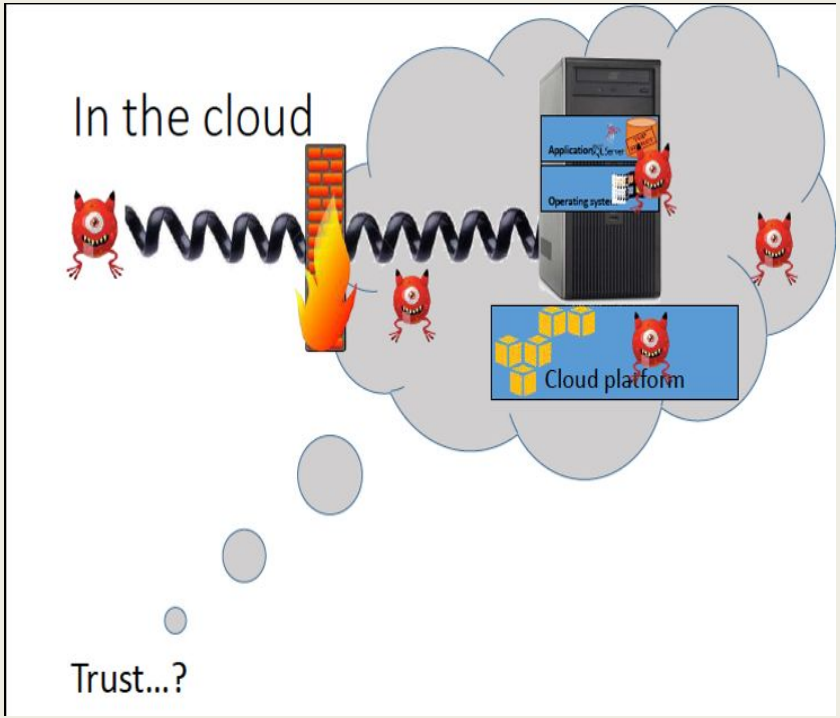
Credit: [Alabi, et al.]

Credit: Phil Schwarzmann [Schwarzmann]

Security in Cloud - Haven

- Malicious applications in the cloud.
- Goal of Haven:

  Secure, private execution Of unmodified applications (bugs and all) in untrusted Cloud on commodity hardware(Intel SGX)



Credit: [Baumann, et al]

Shielded Execution:

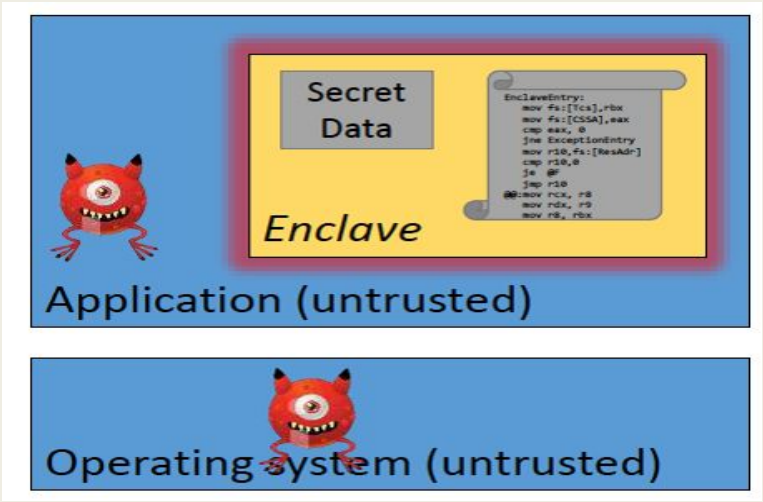
- Protection of specific program from rest of system(sandboxing, process isolation). New term but old concept.
- Confidentiality and integrity of the program, its intermediate state, control flow, etc (Input and output may be encrypted)
- Intel SGX (Software Guard Extension)
- Hardware isolation for an enclave
- New instructions to establish,

  Protect. Call gate to enter.

- Remote attestation: processor

  Manufacturer is the root of the

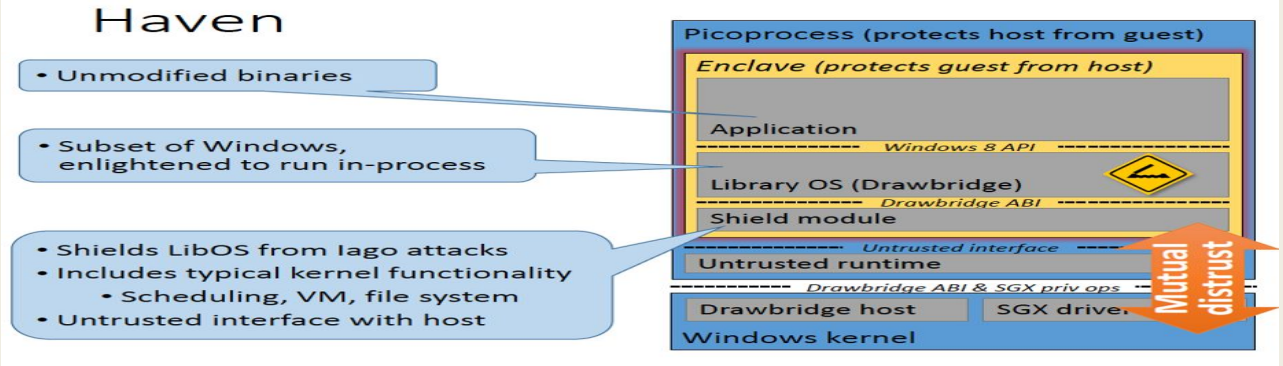
  trust.



Credit: [Baumann, et al]

SGX vs Haven

- SGX was designed to enable new trustworthy applications to protect specific secrets by placing portions of their code and data inside enclaves.
  - Self-contained code sequence
  - V2.0 supports dynamic memory allocation.
- Haven aims to shield entire unmodified legacy applications written without any knowledge of SGX
  - Challenge 1: execute legacy binary code
  - Challenge 2: Interaction with untrusted os and hardware(Iago attack)



Credit: [Baumann, et al]

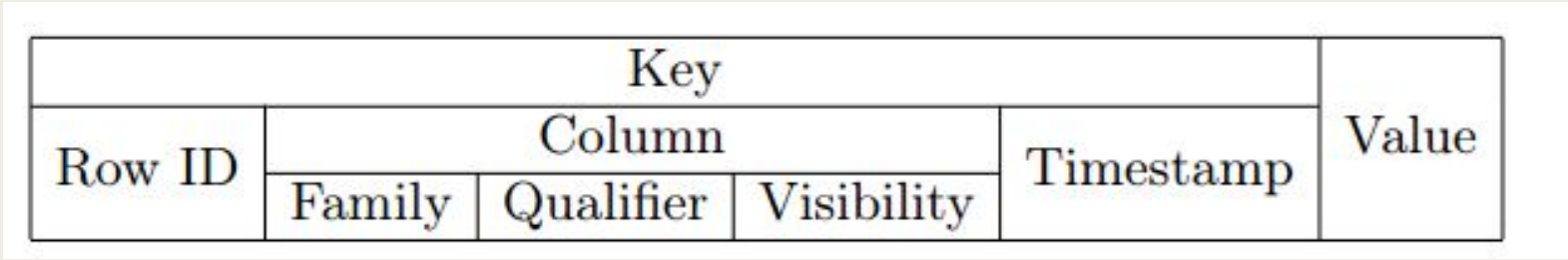
Credit: [sqrr]

Accumulo and Big Data System

- Apache Accumulo is a highly scalable structured store based on Google's BigTable.
- Accumulo is written in Java and operates over the Hadoop Distributed File System (HDFS)

Accumulo supported with two main features

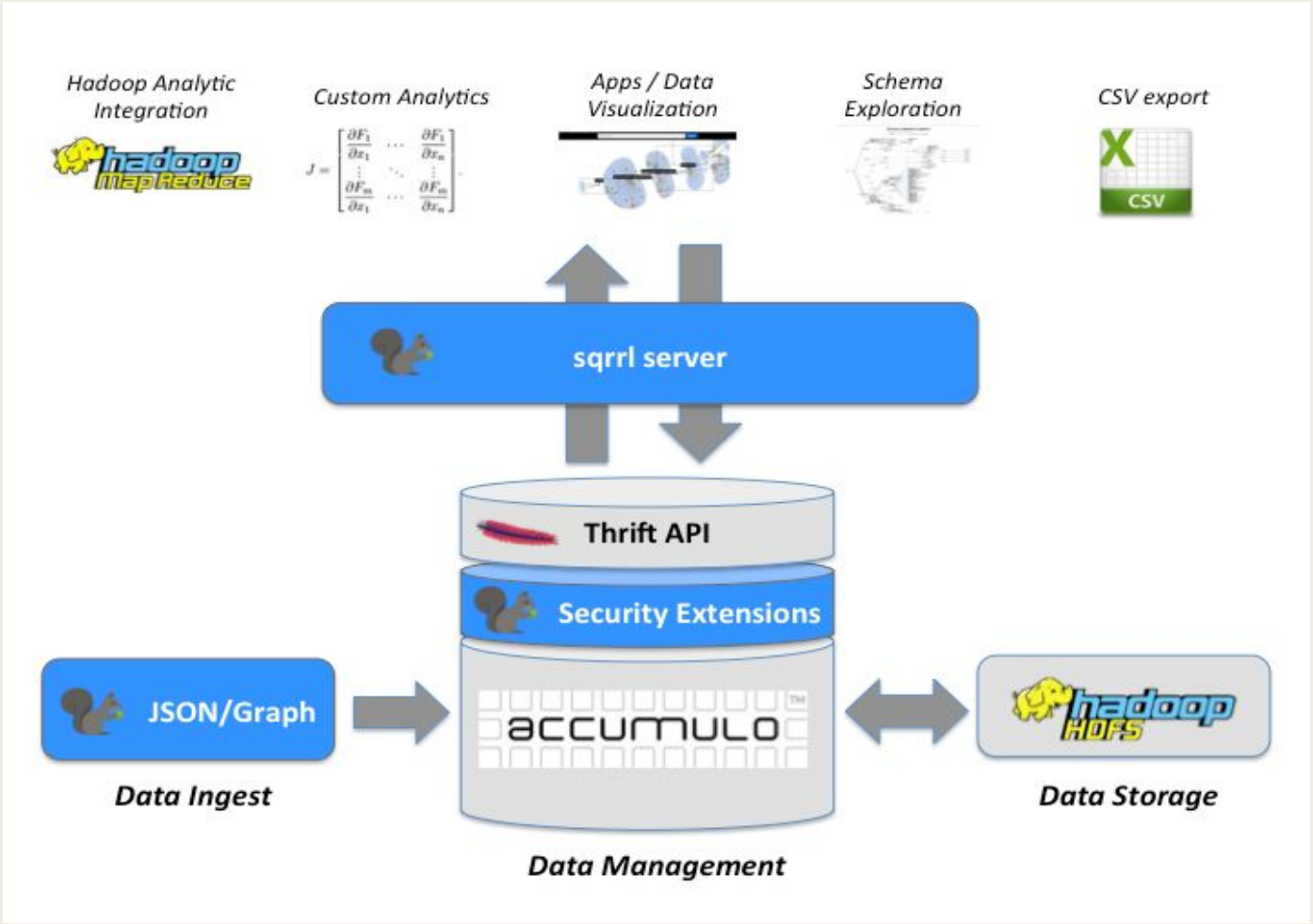
- Iterator framework that support user-programmed.
- key-value store that supports cell-based access control rules



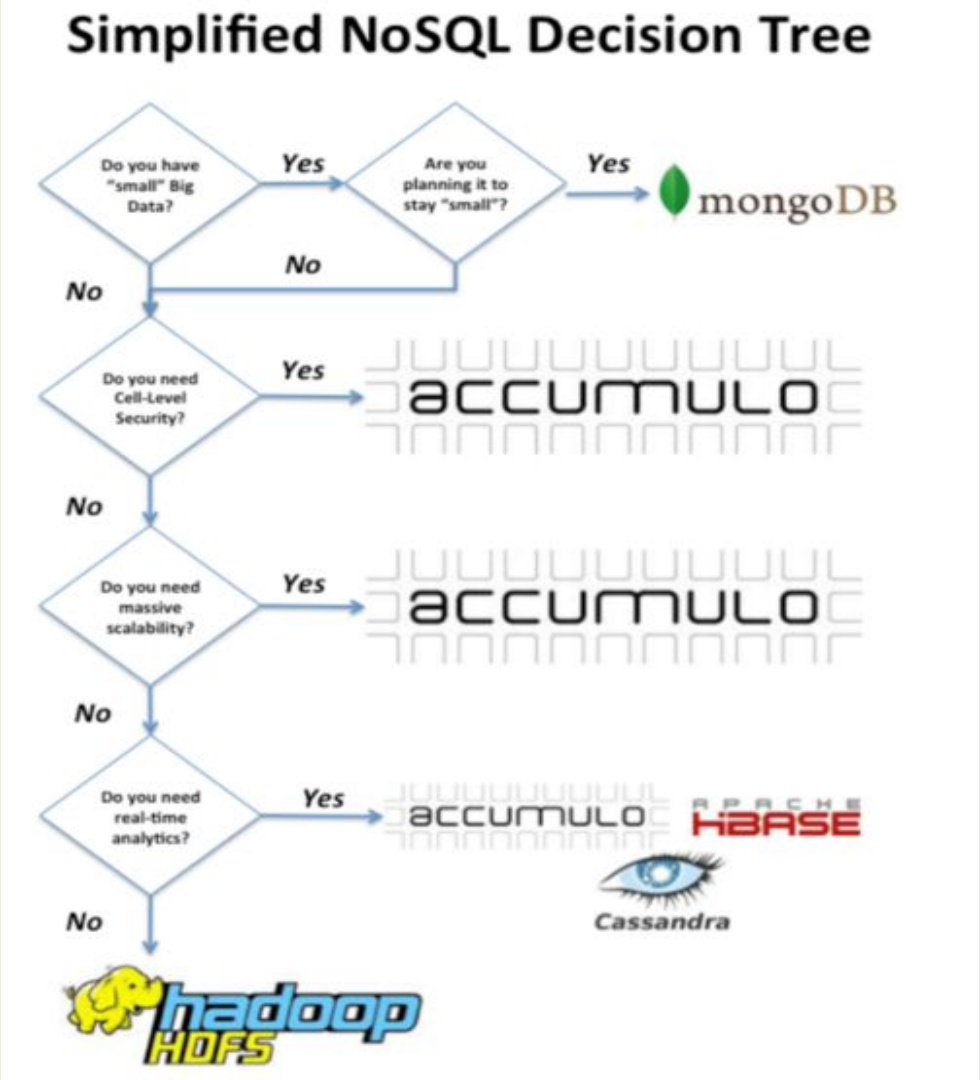
Column Visibility Field

- Assigning “authorizations” to various cells in a table
- Issuing the same query with different authorizations and get different results
- Classifying sensitive data from users who may not have access

❖ **However** , this may not be enough for all, because many organizations use flexible and dynamic AC policies.



Credit:
Andrei Macsin
[hadoop360]



Credit: [sqrrl]

Analysis: Limitations and extensions of using Accumulo

- The query is bound by communication overhead at the client
- Communication in Accumulo is handled by Apache Thrift, which does not saturate network bandwidth
- With low acceptance rates, the query is bound by I/O rates.
- Disk read rates are limited by the performance of the distributed filesystem and the storage hardware
- Multiple clients randomly accessing rows from the tablet servers resulted in contention that seriously degraded scan rate performance.
-

Recommendations:

- Row keys should be selected to accelerate the application's queries, by including a timestamp in the event
- table row keys improved query performance
- index table should only be used when the scan acceptance rate is expected to be very low
- Optimizing the Accumulo client.

Summary and Q&A

- Big data processing will become even more important as the availability of information grows
- Security is an ongoing concern for a platform that was not originally developed to be secure
- Accumolo provides some important rigor for securing enterprise jobs but at the expense of performance

References

- S. Lohr, “Sizing Up Big Data, Broadening Beyond the Internet,” *New York Times*, 19-Aug-2013.
- https://hadoopecosystemtable.github.io/ - Ecosystem table
- B. Marr, “Big Data: 20 Mind-Boggling Facts Everyone Must Read,” *Forbes*, 30-Sep-2015.
- Facebook.com,” *Facebook.com*, Dec-2015. [Online]. Available at: http://newsroom.fb.com/company-info/.
- T. White, Hadoop: the definitive guide: storage and analysis at internet scale. Beijing: O'Reilly Media, 2015.
- https://accumulo.apache.org/1.5/accumulo_user_manual.pdf
- https://accumulo.apache.org/papers/accumulo-benchmarking-2.1.pdf
- Sen, Rahul, Andrew Farris, and Peter Guerra. "Benchmarking apache accumulo bigdata distributed table store using its continuous test suite." Big Data (BigData Congress), 2013 IEEE International Congress on. IEEE, 2013.
- Devdatta Kulkarni. 2013. A fine-grained access control model for key-value systems. In Proceedings of the third ACM conference on Data and application security and privacy (CODASPY '13). ACM, New York, NY, USA, 161-164.
- Sawyer, Scott M., et al. "Understanding query performance in Accumulo." High Performance Extreme Computing Conference (HPEC), 2013 IEEE. IEEE, 2013.
- https://qconsf.com/sf2014/presentation/gobblin-framework-solving-big-data-ingestion-problem.html
- V. C. Hu, T. Grance, D. F. Ferraiolo and D. R. Kuhn, "An Access Control scheme for Big Data processing," *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*, Miami, FL, 2014, pp. 1-7.

IMAGE REFERENCES

- "Handling Privacy and Security Concerns with Big Data," *PCQuest*, 27-Mar-2015.
- S. St, *Slideshare*. SlideShare, 2014.
- V. Nayal, *Slideshare*. SlideShare, 2014.
- "How to Choose a NoSQL Database," sqrrl, 08-May-2013.
- K. T. Smith, "Big Data Security: The Evolution of Hadoop's Security Model," *InfoQ*, 14-Aug-2013.
- T. L. Furrer, *37754249*. Dreamstime.com.
- P. Schwarzmann, "Meeting Cloud Security Challenges – Insights From Equinix Partner, Alert Logic," *Interconnections*, 19-Feb-2016. .
- F. Wang, "Haven: Shielding applications from an untrusted cloud," 27-Feb-2015. .
- A. Baumann, M. Peinado, and G. Hunt, "Shielding Applications from an Untrusted Cloud with Haven," *ACM Trans. Comput. Syst. TOCS ACM Transactions on Computer Systems*, vol. 33, no. 3, pp. 1–26, 2015.
- *Data Security of Discarded Electronics*. NTRecycling, 2015.
- http://www.hadoop360.com/blog/accumulo-sqrrl-nosql-secure-database
- O. Alabi, J. Beckman, M. Dark, and J. Springer, "Toward a Data Spillage Prevention Process in Hadoop using Data Provenance," *Proceedings of the 2015 Workshop on Changing Landscapes in HPC Security - CLHS '15*, 2015.