Data Fidelity with Distributed Warehousing Strategies

Comparing RDBMS and Hadoop DFS

Dan Thomas III

MSCS 6060

Out of scope

- Data types
- Data structure
- Data flow and architecture strategies
- Specific RDBMS ETL brands (ie DataStage, Informatica)
- Specific Hadoop daemons (ie Mahout, Flume)

ETL (RDBMS)

- **E**xtract, **T**ranslate, and **L**oad
- Plumbing work of data warehousing
- Move from target source to target databases
- Usually proprietary and enterprise restrictive

- A very costly, time consuming bidirectional endeavor

Data Fidelity

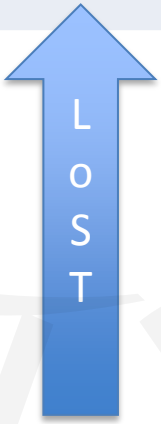
- Mandatory Translation phase
- Cleanse “dirty data”
  - Parsing
  - Correcting
  - Standardizing
  - Matching
  - Consolidating
- Symmetrical scheduling
- Changes story of data

Translation cases

- Dummy values
- Absence of Data
- Multipurpose fields
- Cryptic data
- Contradicting data
- Violations of business logic
- Non-unique identifiers

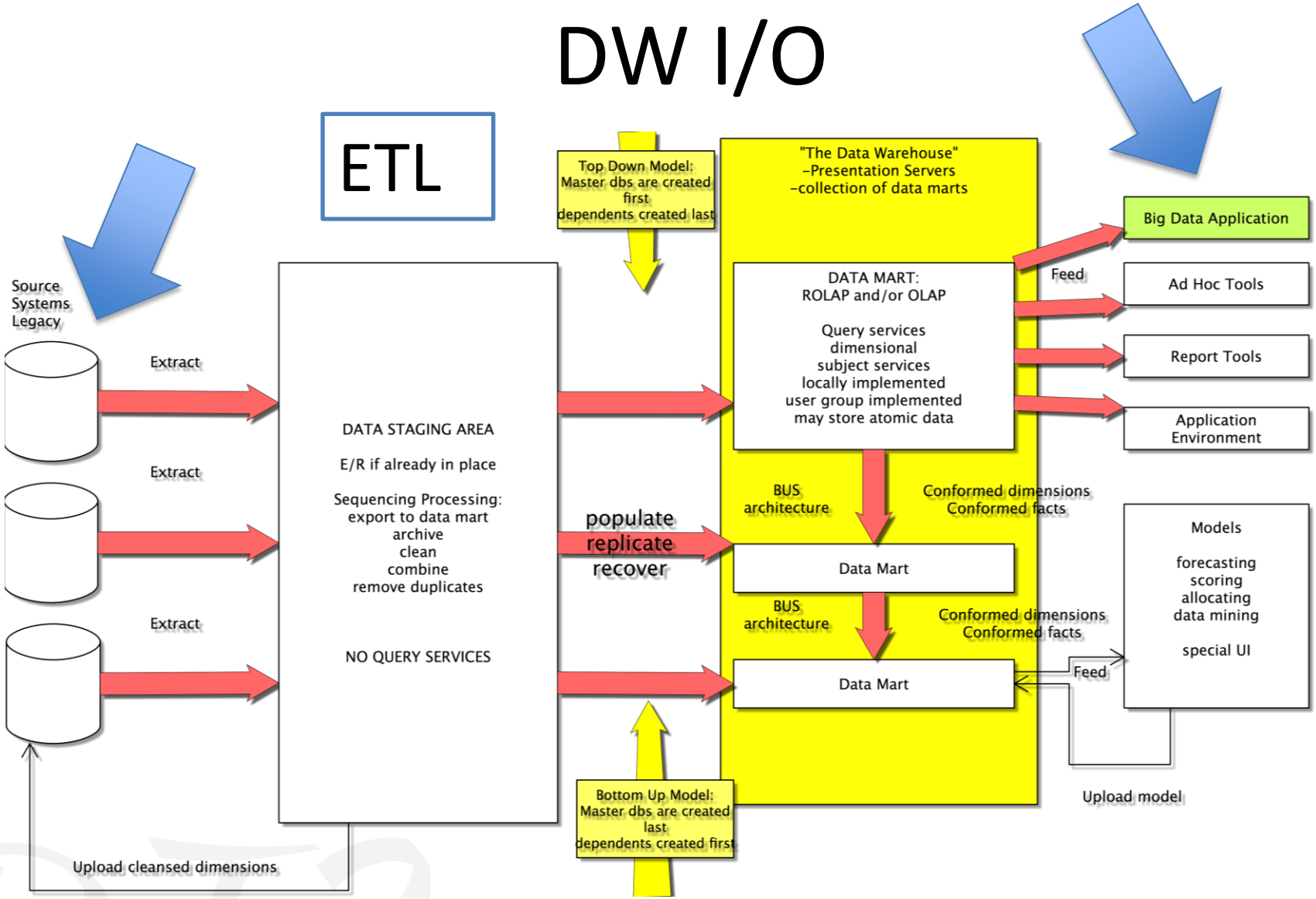
Interchangeable Examples

Old	New
53202	53202-4175 (*Correcting)
Male	M (*Standardizing)
J.	J (*Correcting, Standardizing)
414-555-1234	4145551234 (*Correcting, Standardizing)
John J Doe	John J. Doe (*Correcting, Standardizing, Matching, Consolidating)
John Doe	



Cleanse “dirty data”
-Parsing
-Correcting
-Standardizing
-Matching
-Consolidating

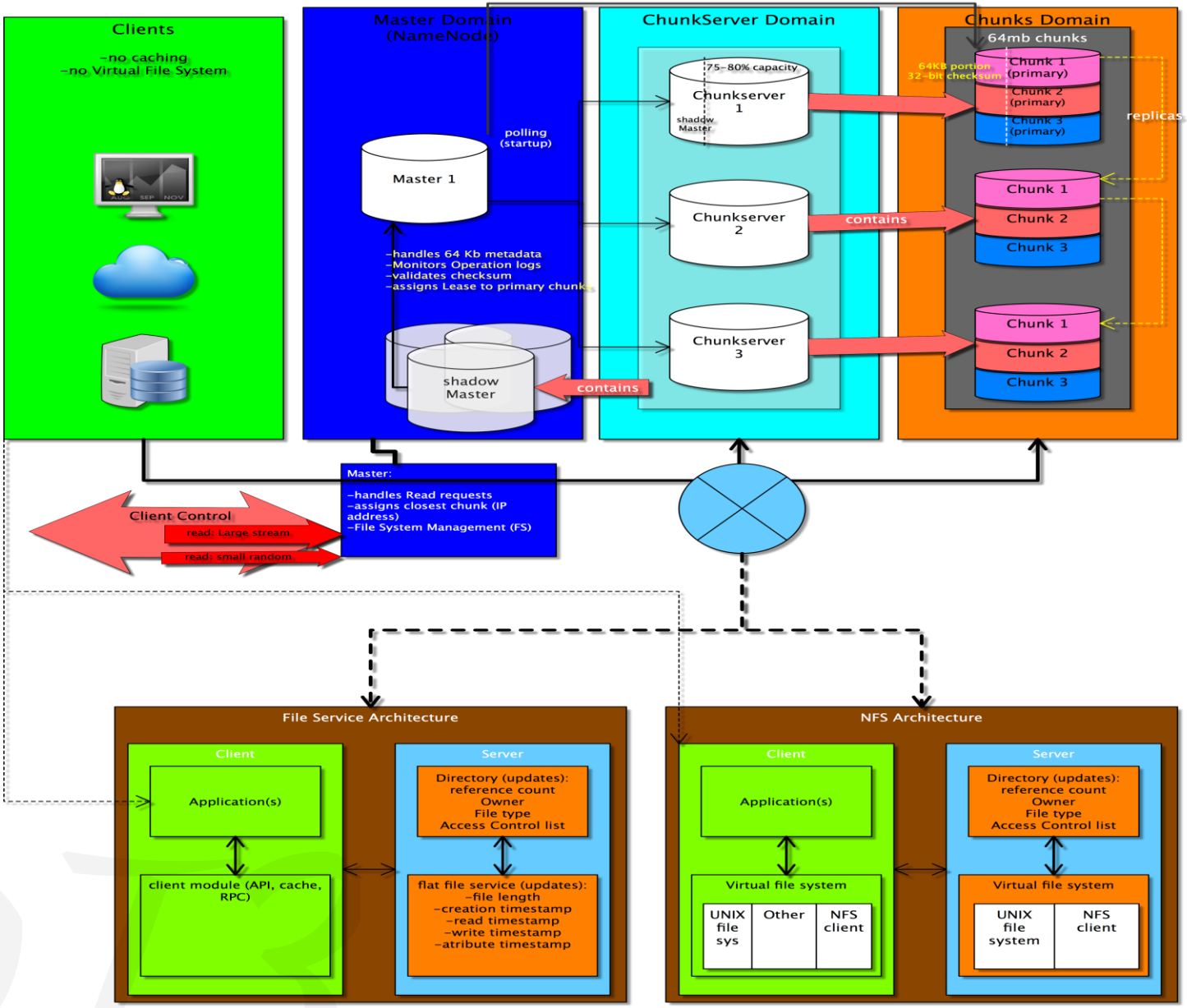
DW I/O


ETL
Source Systems Legacy
Extract
Extract
Extract
DATA STAGING AREA
E/R if already in place
Sequencing Processing:
export to data mart
archive
clean
combine
remove duplicates
NO QUERY SERVICES
Upload cleansed dimensions
Top Down Model:
Master dbs are created first
dependents created last
populate
replicate
recover
Bottom Up Model:
Master dbs are created last
dependents created first
"The Data Warehouse"
-Presentation Servers
-collection of data marts
DATA MART:
ROLAP and/or OLAP
Query services
dimensional
subject services
locally implemented
user group implemented
may store atomic data
BUS architecture
Conformed dimensions
Conformed facts
Data Mart
BUS architecture
Conformed dimensions
Conformed facts
Data Mart
Feed
Feed
Big Data Application
Ad Hoc Tools
Report Tools
Application Environment
Models
forecasting
scoring
allocating
data mining
special UI
Upload model

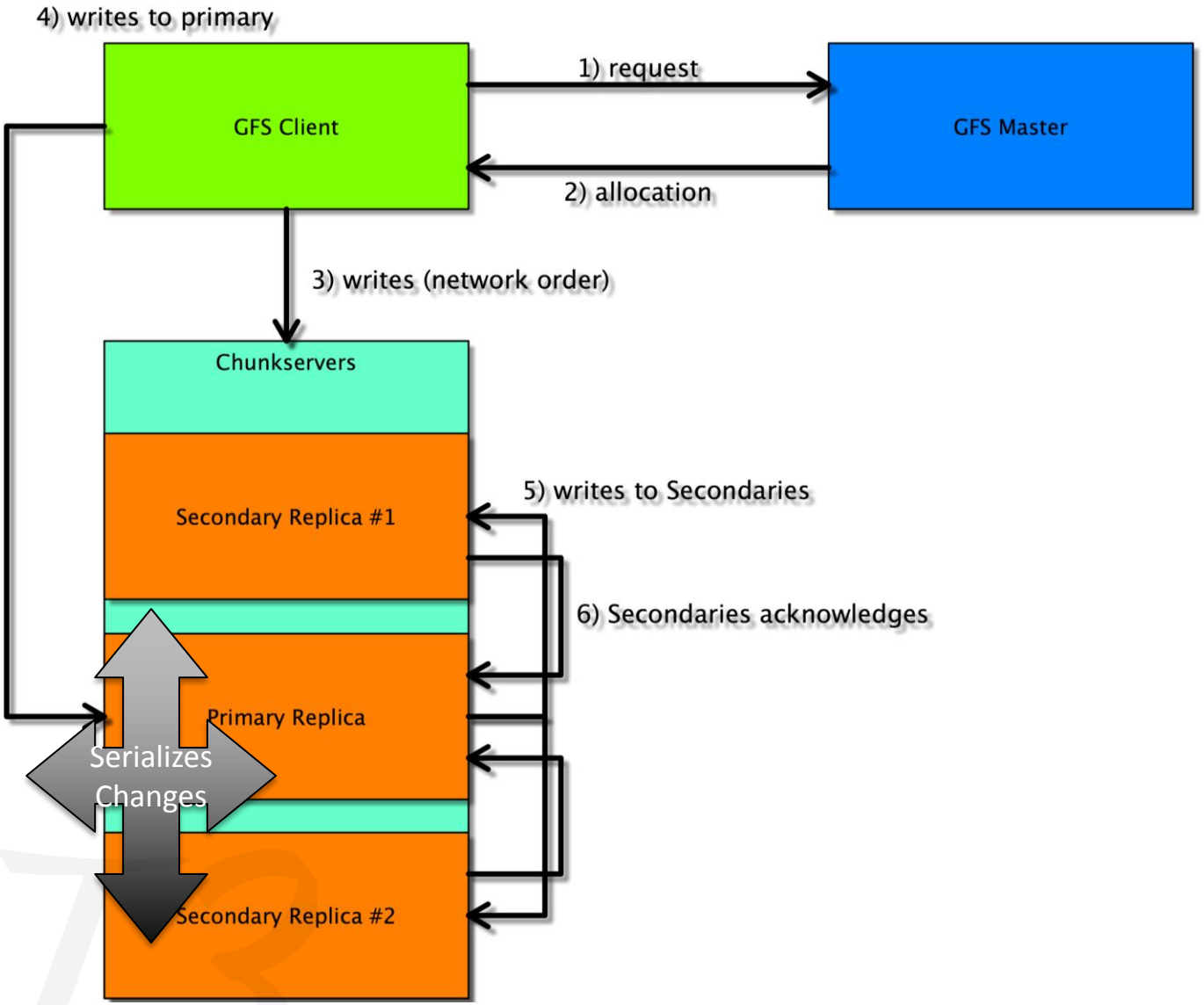
Hadoop (HDFS)

- Need for unstructured data exploding
  - No emphasis on translation
- Scribe and Hive daemons mentioned later

Hadoop (GFS) Output



Writes

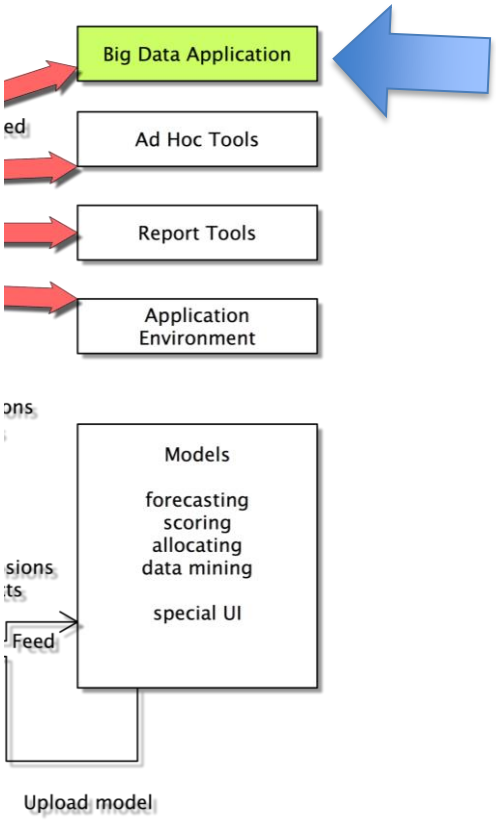

4) writes to primary
GFS Client
1) request
GFS Master
2) allocation
3) writes (network order)
Chunkservers
Secondary Replica #1
5) writes to Secondaries
6) Secondaries acknowledges
Primary Replica
Serializes
Changes
Secondary Replica #2

Data Fidelity and Integrity

- Output (from HDFS (and RDBMS))
  - It is possible that a block of data fetched from a DataNode (chunk) arrives corrupted
  - This corruption can occur because of faults in a storage device, network faults, or buggy software
- Input (to HDFS)
  - Designed for write one, read many cardinalities

http://hadoop.apache.org/docs/stable/hdfs_design.html#Data+Integrity

Cleanse Big Data Application



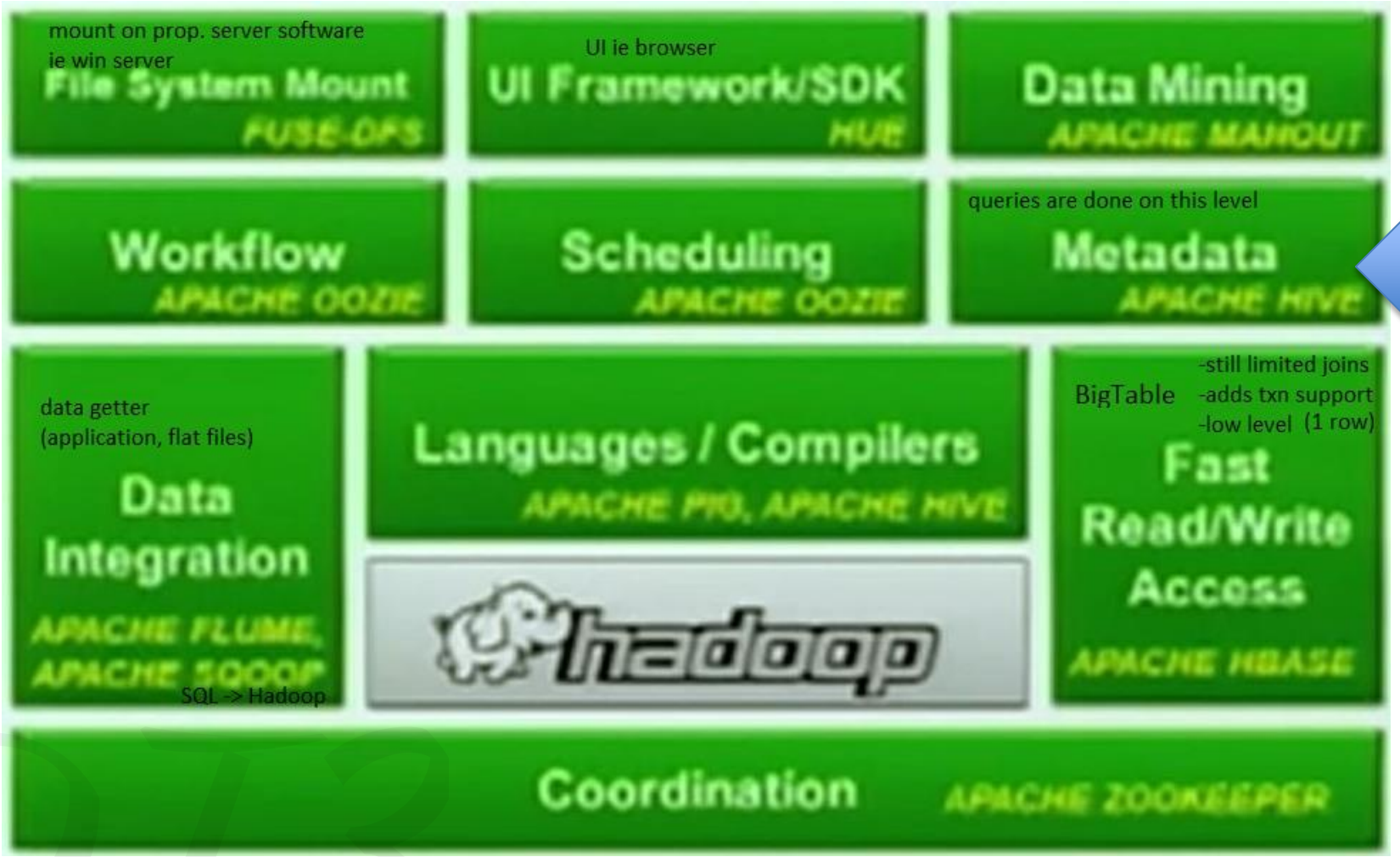
Middleware (java)

```
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(WordCount.class);
    conf.setJobName("wordcount");

    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);

    conf.setMapperClass(WordCountMapper.class);
    conf.setReducerClass(WordCountReducer.class);

    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);

    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));

    JobClient.runJob(conf);
```

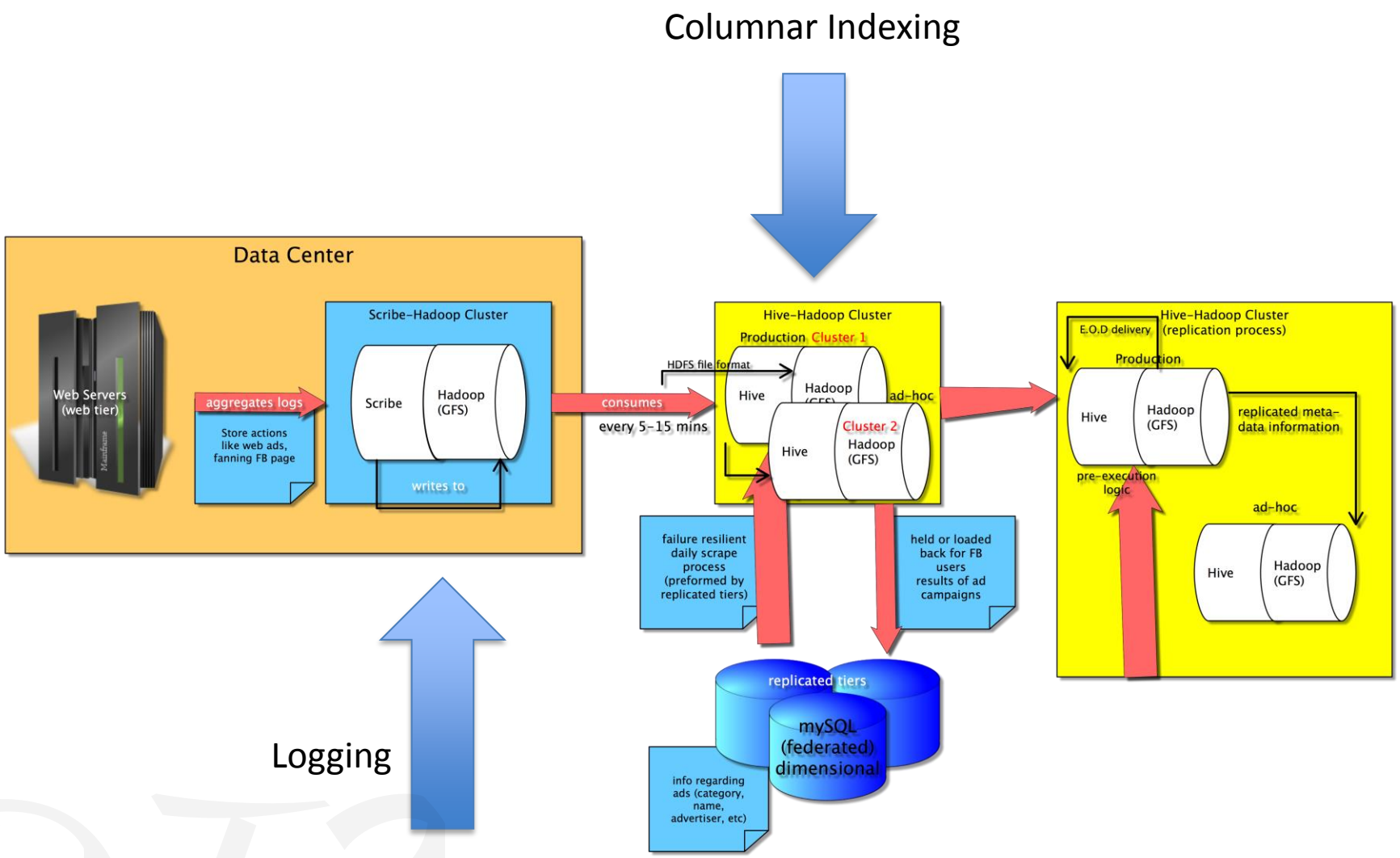
Sample I/O (US Patent office)

Input (with columns)	Output (count)
"CITING","CITED"	1 2
3858241,956203	10000 1
3858241,1324234	100000 1
3858241,3398406	1000006 1
3858241,3557384	1000007 1
3858241,3634889	1000011 1
3858242,1515701	1000017 1
3858242,3319261	1000026 1
3858242,3668705	1000033 2
3858242,3707004	1000043 1
	1000044 2
	1000045 1
	1000046 2
	1000049 1
	1000051 1
	1000054 1
	1000067 3

Daemons (wrt Data Fidelity)



Facebook // Use case



Right Tool?

Relational Databases:

+Latency

Use when:

- Interactive OLAP Analytics (<1sec)
- Multistep ACID Transactions
  +transactions
- 100% SQL Compliance

Hadoop: +Throughput

Use when: +bigger data pipelines

- Structured or Not (Flexibility)
- Scalability of Storage/Compute
  NoSQL+OOP langs
- Complex Data Processing