SPARK
SUMMIT
2017

Next Generation Archiving with Hadoop

A look at archiving, e-discovery, and supervision on the Hadoop platform

Jordan Volz

Systems Engineer @ cloudera®

What are we talking about?

Archiving

- Long-term storage of data
  - Compliance (WORM/NENR) vs non-compliance
  - Ingestion + Enrichment
  - Retention
  - Active vs passive
  - Reconciliation
  - Auditing
  - Search

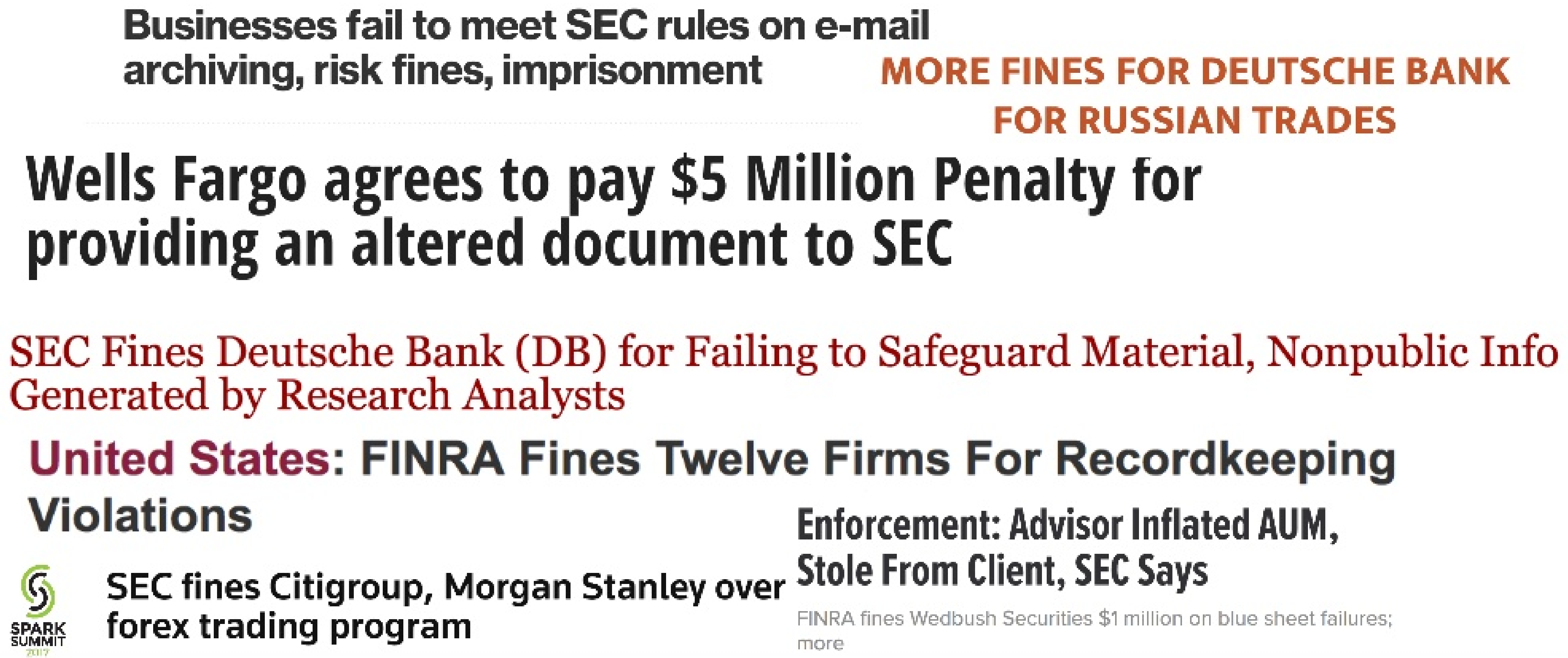
e-Discovery

- Review of electronic data to assess its relevance in legal proceedings
  - ECA
  - Legal Hold
  - TAR
  - Production
  - Case Management
  - Metadata management
  - EDRM

Supervision

- Review of electronic communication to detect unethical conduct
  - Risk-based policies
  - Random Sampling
  - Surveillance
  - Auditing
  - Analytics + reporting
  - CO workflow
  - Lexicon management

Why do we care about this?



Businesses fail to meet SEC rules on e-mail archiving, risk fines, imprisonment

MORE FINES FOR DEUTSCHE BANK FOR RUSSIAN TRADES

Wells Fargo agrees to pay $5 Million Penalty for providing an altered document to SEC

SEC Fines Deutsche Bank (DB) for Failing to Safeguard Material, Nonpublic Info Generated by Research Analysts

United States: FINRA Fines Twelve Firms For Recordkeeping Violations

Enforcement: Advisor Inflated AUM, Stole From Client, SEC Says

SEC fines Citigroup, Morgan Stanley over forex trading program

FINRA fines Wedbush Securities $1 million on blue sheet failures; more

Shortcomings of Traditional Systems

Archiving

- Dated architecture
- Scalability
- Can't handle all data types
- Inflexible deployment
- Proprietary technology
- End-to-end vs piecemeal + siloed solutions
- Security/Encryption at Rest

e-Discovery

- Best tools are not native to archive
- ECA scalability
- Lack of support for advanced media (audio/video)
- Lack of machine learning/advanced analytics
- Demanding SLAs for data export/production – hard to scale up for large cases/demand

Supervision

- Rule-based
- Too many false positives
- Lack of machine learning/advanced analytics
- Focused on batch processing
- Difficult to customize
- Difficult integrations to archive for surveillance

Today's Wisdom

Spark is awesome, but for complicated workflows you often need **more than Spark**.

Luckily, it has great integrations in the Hadoop ecosystem.

Strengths of Hadoop/CDH

Fast

- Fast SQL with **Impala**
- Advanced Analytics + ML via **Spark ML**
- **Kafka** for streaming data ingestion
- In-flight data processing via **Spark Streaming**
- Distributed Search with **Solr**
- **Kudu** for tracking status/updates

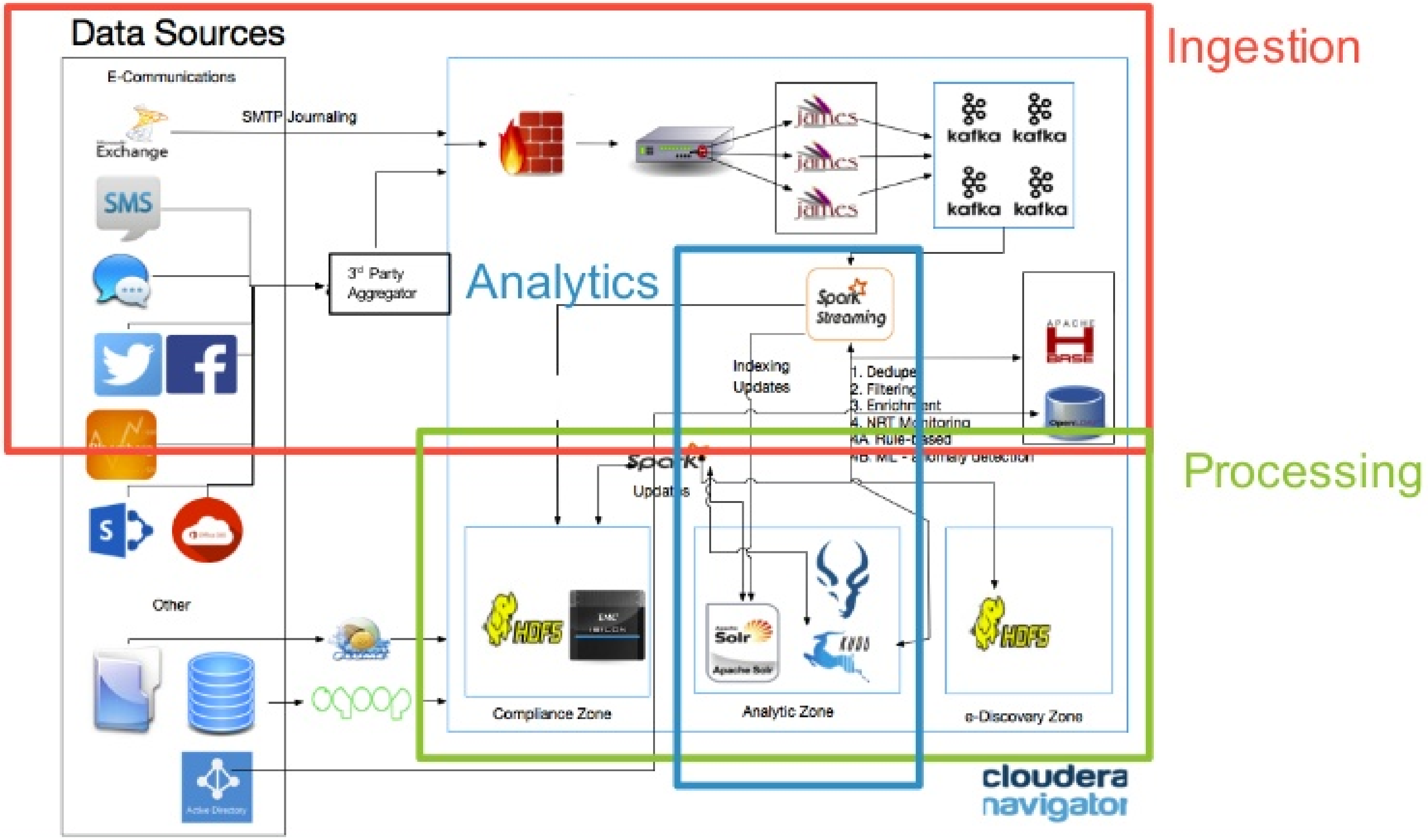
Easy

- Tested scalability to PBs
- Handles all data types
- Single platform for all solutions
- Easy integrations
- Driven by Open Source innovation
- Cloud or on-premise
- Integrated Data Science Platform (**Cloudera Data Science Workbench**)

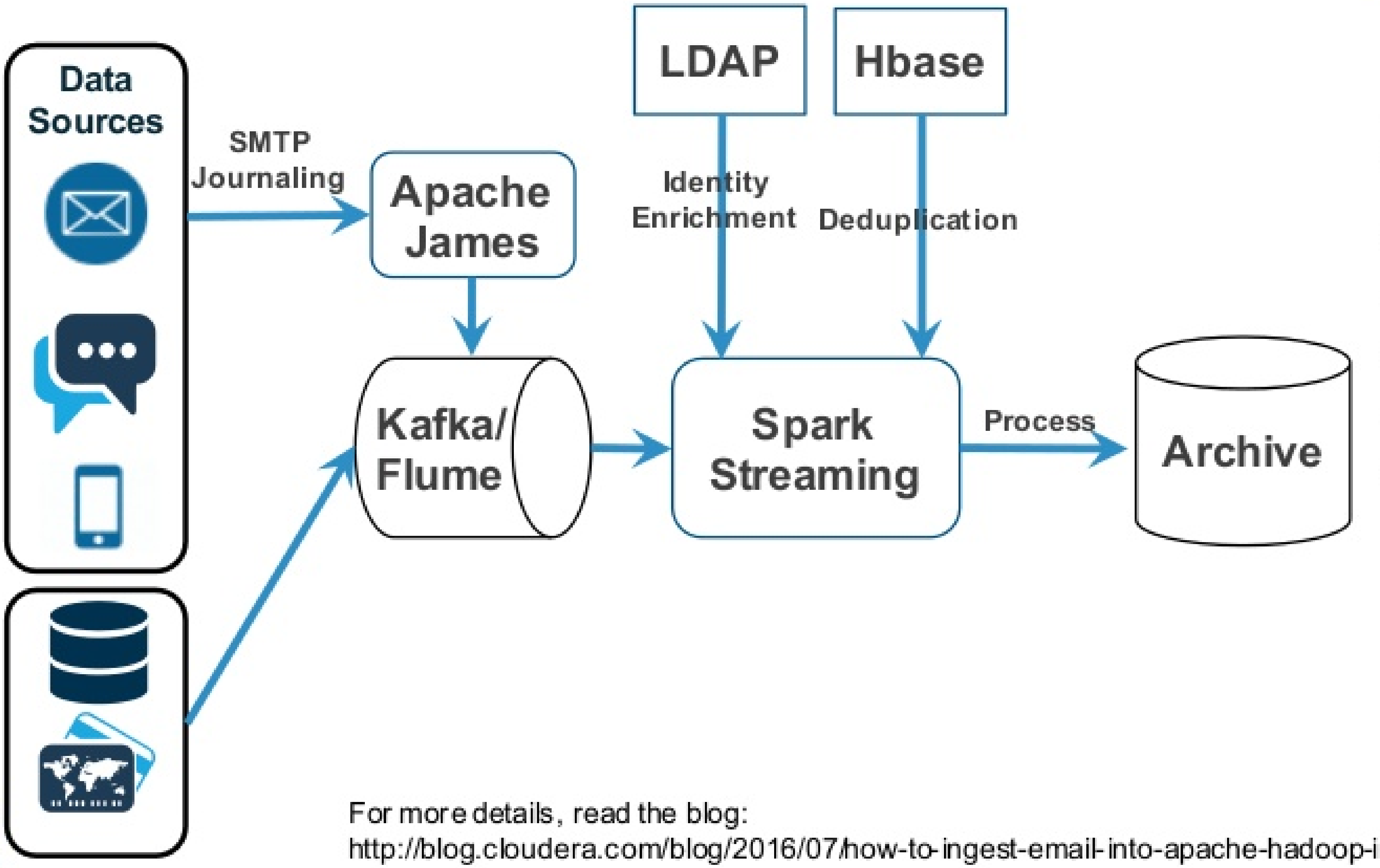
Secure

- End-to-end Security on a common platform
- Full system Data Encryption at Rest (**KMS/KTS**)
- Full system auditing with **Navigator**
- RBAC with **Sentry**
- Multi-tenancy
- **BDR** built-in (geo replication)

Architectural Overview



Architectural Overview - Ingestion



Data Sources

SMTP Journaling

Apache James

Kafka/ Flume

LDAP

Identity Enrichment

Hbase

Deduplication

Spark Streaming

Process

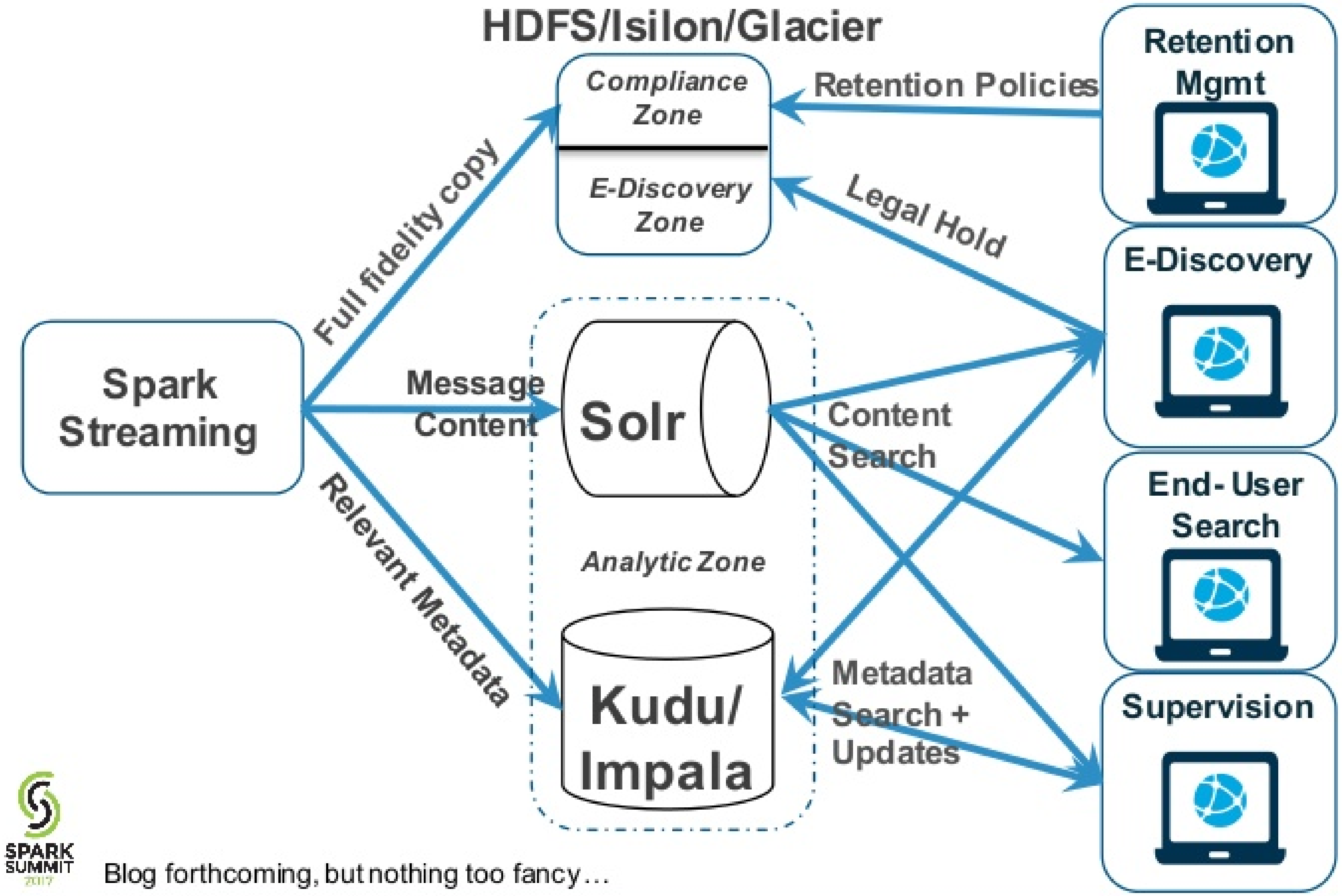
Archive

Uses of Spark

- Deduplication (Hashing)
- Message Enrichment
- Reconciliation
- Message Filtering
- Policy Filtering

For more details, read the blog:
http://blog.cloudera.com/blog/2016/07/how-to-ingest-email-into-apache-hadoop-in-real-time-for-analysis/

Architectural Overview - Processing

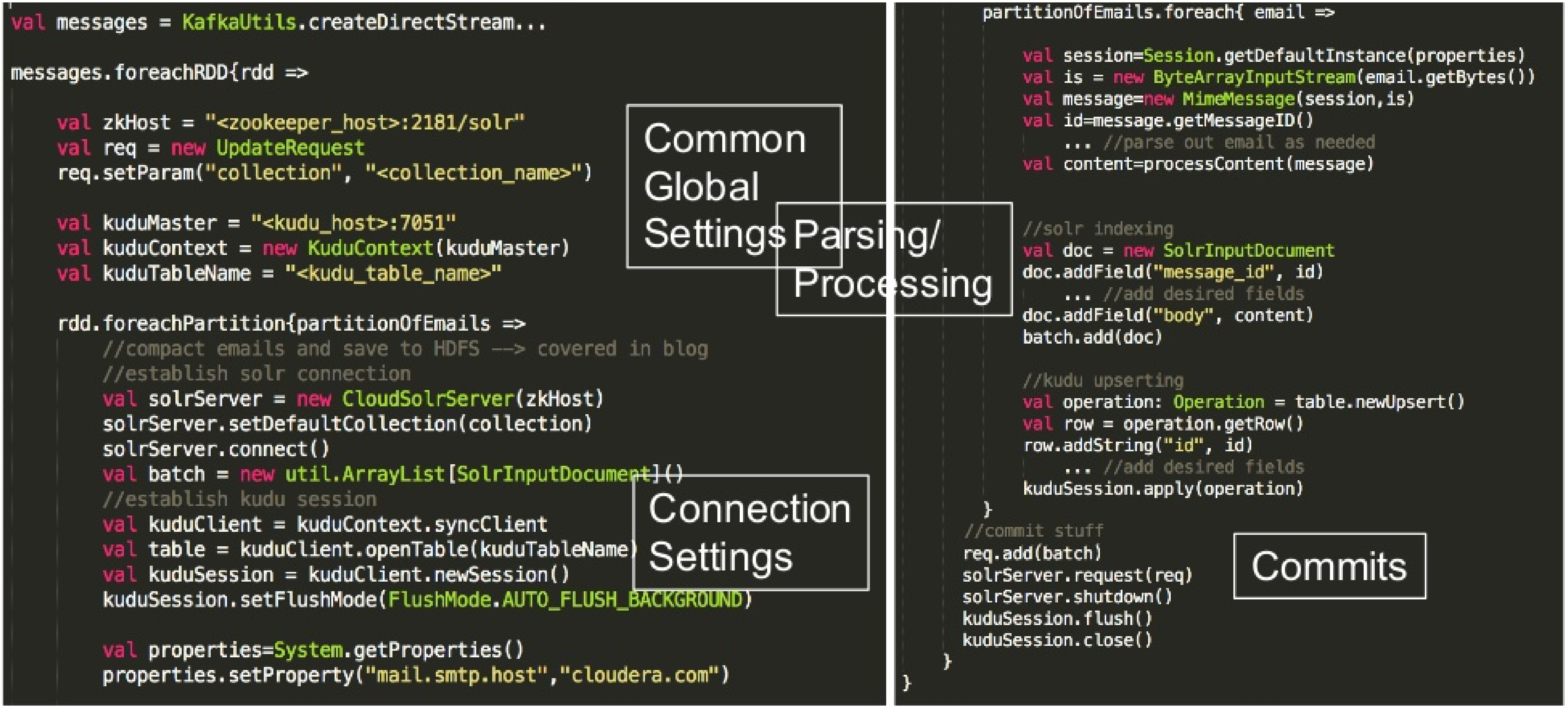


Uses of Spark

- Message compaction
- Message parsing and indexing
- Metadata extraction and storage
- Metadata updates (optional)
- Content re-indexing (optional)
- Content tokenization (optional)

Blog forthcoming, but nothing too fancy…

Quick Sample Code



```
val messages = KafkaUtils.createDirectStream...

messages.foreachRDD{rdd =>

    val zkHost = "<zookeeper_host>:2181/solr"
    val req = new UpdateRequest
    req.setParam("collection", "<collection_name>")

    val kuduMaster = "<kudu_host>:7051"
    val kuduContext = new KuduContext(kuduMaster)
    val kuduTableName = "<kudu_table_name>"

    rdd.foreachPartition{partitionOfEmails =>
        //compact emails and save to HDFS --> covered in blog
        //establish solr connection
        val solrServer = new CloudSolrServer(zkHost)
        solrServer.setDefaultCollection(collection)
        solrServer.connect()
        val batch = new util.ArrayList[SolrInputDocument]()
        //establish kudu session
        val kuduClient = kuduContext.syncClient
        val table = kuduClient.openTable(kuduTableName)
        val kuduSession = kuduClient.newSession()
        kuduSession.setFlushMode(FlushMode.AUTO_FLUSH_BACKGROUND)

        val properties=System.getProperties()
        properties.setProperty("mail.smtp.host","cloudera.com")
```

```
        partitionOfEmails.foreach{ email =>

            val session=Session.getDefaultInstance(properties)
            val is = new ByteArrayInputStream(email.getBytes())
            val message=new MimeMessage(session,is)
            val id=message.getMessageID()
                ... //parse out email as needed
            val content=processContent(message)


            //solr indexing
            val doc = new SolrInputDocument
            doc.addField("message_id", id)
                ... //add desired fields
            doc.addField("body", content)
            batch.add(doc)

            //kudu upserting
            val operation: Operation = table.newUpsert()
            val row = operation.getRow()
            row.addString("id", id)
                ... //add desired fields
            kuduSession.apply(operation)
        }
      //commit stuff
      req.add(batch)
      solrServer.request(req)
      solrServer.shutdown()
      kuduSession.flush()
      kuduSession.close()
    }
}
```

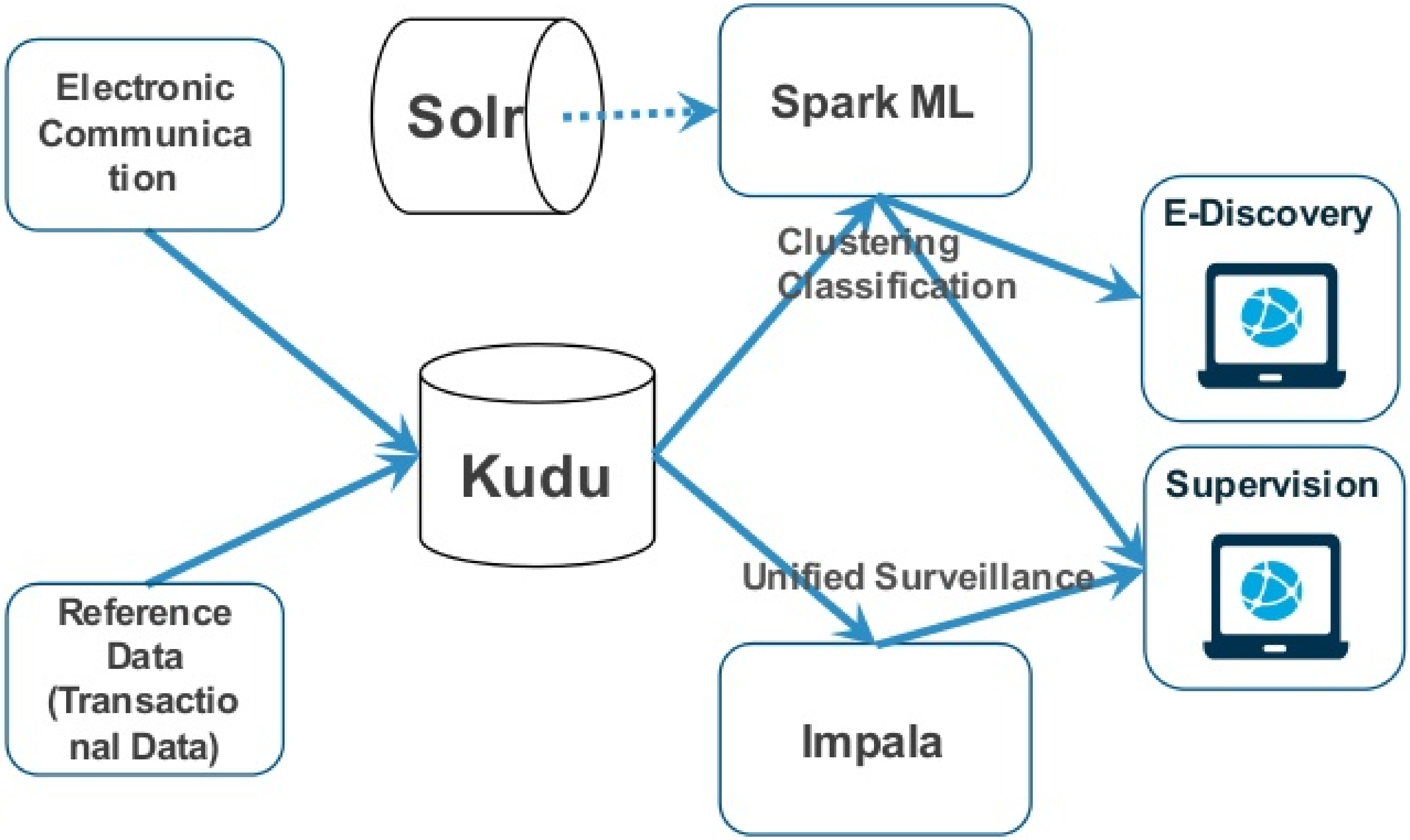
Common Global Settings

Parsing/ Processing

Connection Settings

Commits

Architectural Overview – Analysis + ML



Uses of Spark

- Machine Learning – Clustering for real-time anomaly detection in e-comm
- Model Permanence + Updates
- TAR for e-discovery (automated or recommended tagging + classification)
- Access to unified surveillance measures

Let's look at a quick example

Surveillance Examples

- With only two tables, ecomm metadata and transactional data, but we can still start to answer some pretty interesting questions like…
  - What communication is occurring after a large transaction?
  - What trades are made after a flagged message?
  - Etc…

Sample Schemas

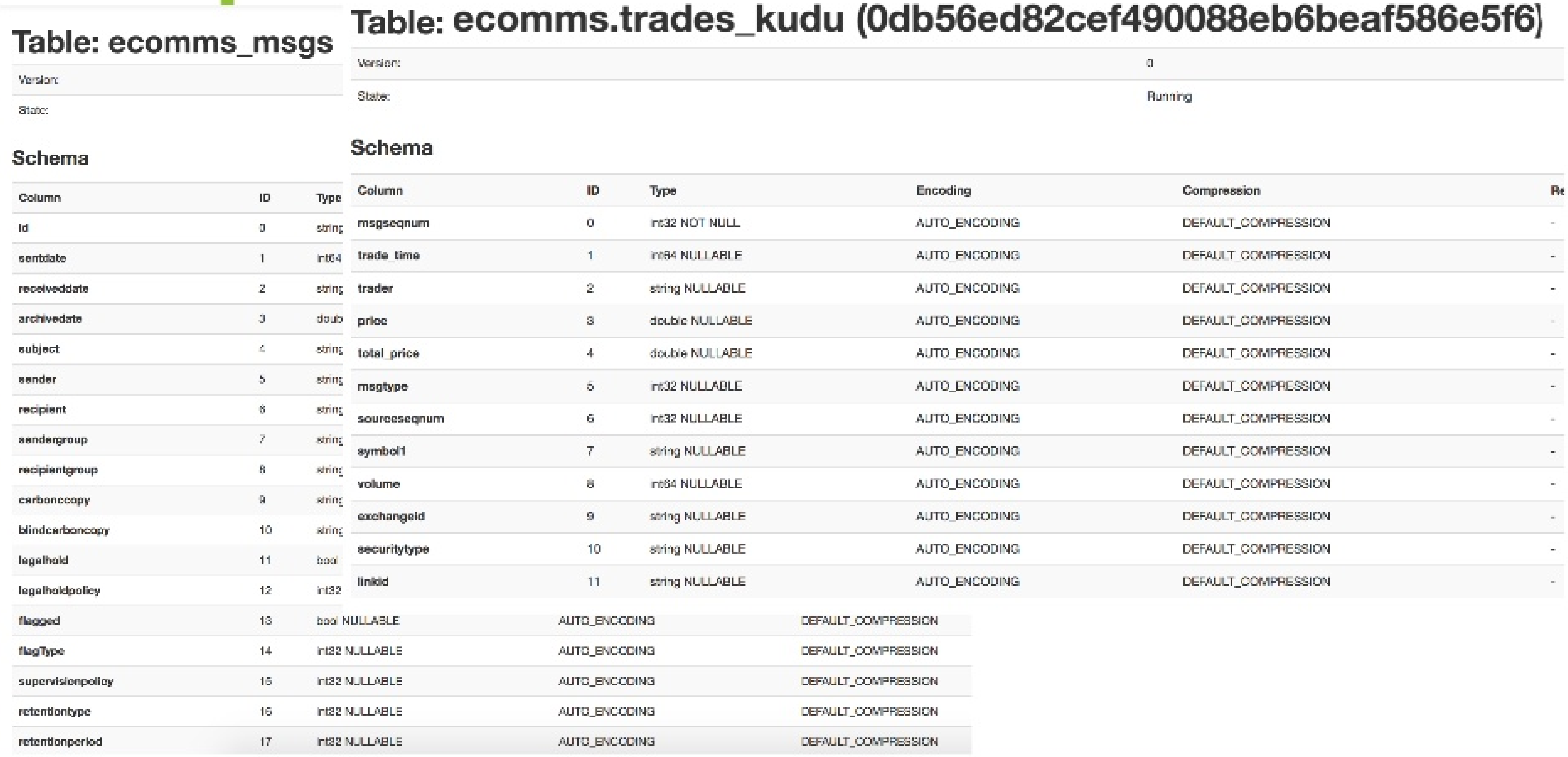


Table: ecomms_msgs

Version:	
State:	

Schema

Column	ID	Type		
id	0	string		
sentdate	1	int64		
receiveddate	2	string		
archivedate	3	doub		
subject	4	string		
sender	5	string		
recipient	6	string		
sendergroup	7	string		
recipientgroup	8	string		
carboncopy	9	string		
blindcarboncopy	10	string		
legalhold	11	bool		
legalholdpolicy	12	int32		
flagged	13	bool NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION
flagType	14	int32 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION
supervisionpolicy	15	int32 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION
retentiontype	16	int32 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION
retentionperiod	17	int32 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION

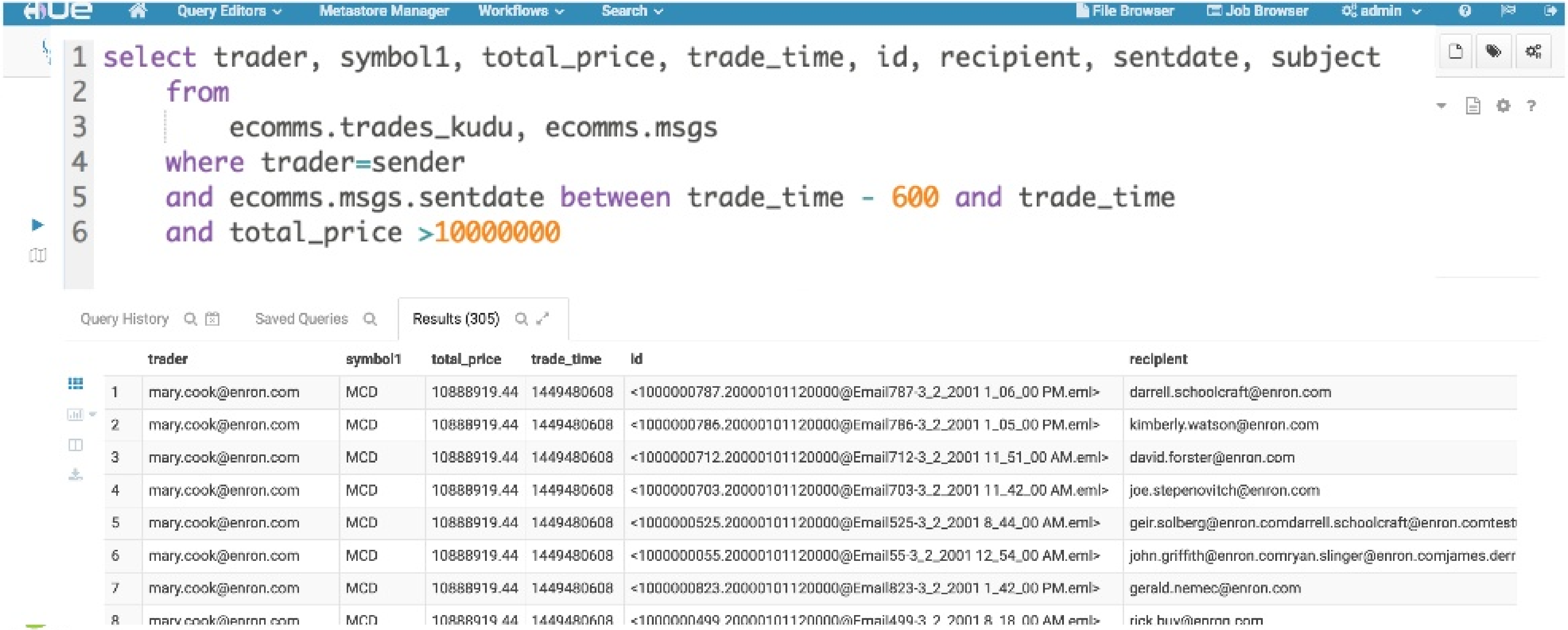
Table: ecomms.trades_kudu (0db56ed82cef490088eb6beaf586e5f6)

Version:	0
State:	Running

Schema

Column	ID	Type	Encoding	Compression	Re
msgseqnum	0	int32 NOT NULL	AUTO_ENCODING	DEFAULT_COMPRESSION	-
trade_time	1	int64 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
trader	2	string NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
price	3	double NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
total_price	4	double NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
msgtype	5	int32 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
sourceseqnum	6	int32 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
symbol1	7	string NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
volume	8	int64 NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
exchangeid	9	string NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
securitytype	10	string NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-
linkid	11	string NULLABLE	AUTO_ENCODING	DEFAULT_COMPRESSION	-

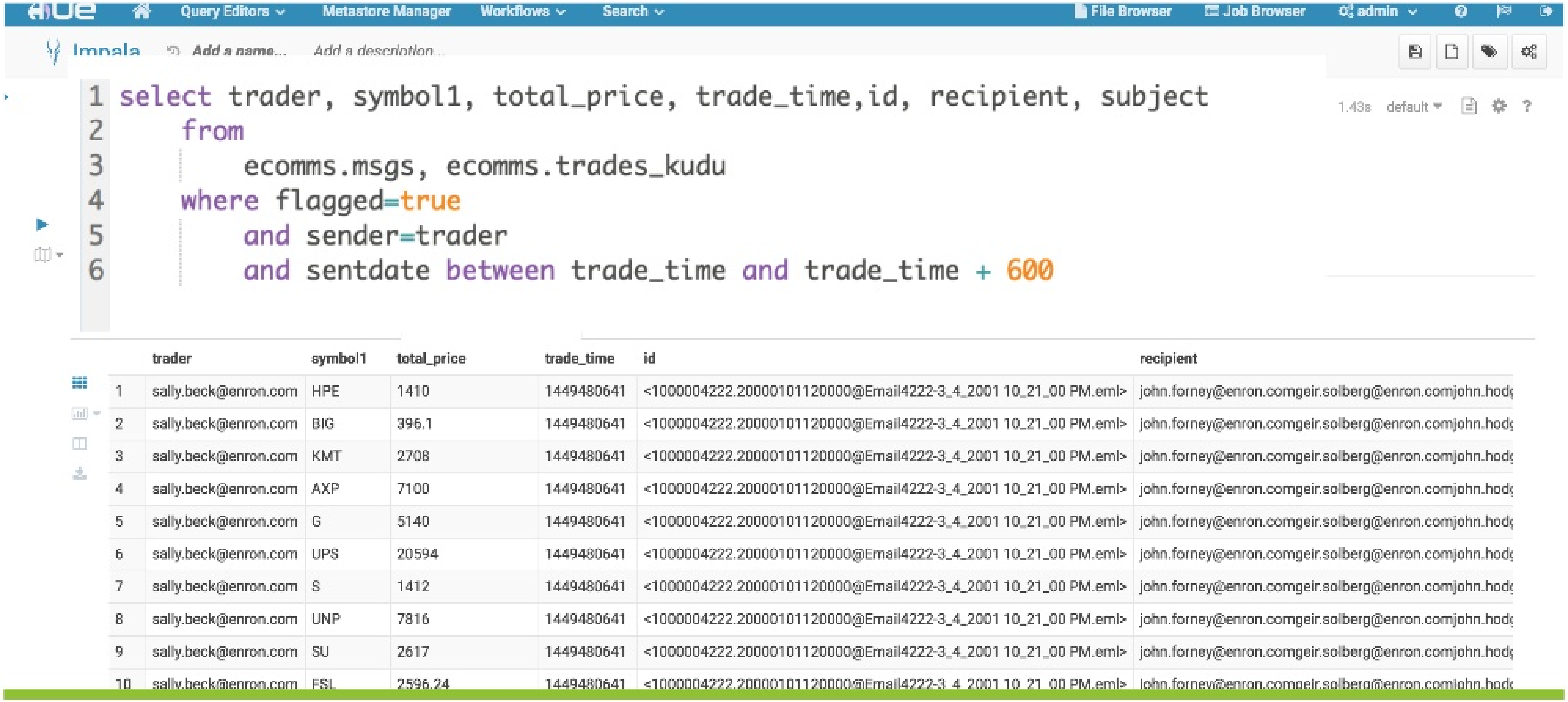
Large Transaction Example



```sql
select trader, symbol1, total_price, trade_time, id, recipient, sentdate, subject
    from
        ecomms.trades_kudu, ecomms.msgs
    where trader=sender
    and ecomms.msgs.sentdate between trade_time - 600 and trade_time
    and total_price >10000000
```

	trader	symbol1	total_price	trade_time	id	recipient
1	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000787.20000101120000@Email787-3_2_2001 1_06_00 PM.eml>	darrell.schoolcraft@enron.com
2	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000786.20000101120000@Email786-3_2_2001 1_05_00 PM.eml>	kimberly.watson@enron.com
3	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000712.20000101120000@Email712-3_2_2001 11_51_00 AM.eml>	david.forster@enron.com
4	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000703.20000101120000@Email703-3_2_2001 11_42_00 AM.eml>	joe.stepenovitch@enron.com
5	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000525.20000101120000@Email525-3_2_2001 8_44_00 AM.eml>	geir.solberg@enron.comdarrell.schoolcraft@enron.comtest
6	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000055.20000101120000@Email55-3_2_2001 12_54_00 AM.eml>	john.griffith@enron.comryan.slinger@enron.comjames.derr
7	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000823.20000101120000@Email823-3_2_2001 1_42_00 PM.eml>	gerald.nemec@enron.com
8	mary.cook@enron.com	MCD	10888919.44	1449480608	<1000000499.20000101120000@Email499-3_2_2001 8_18_00 AM.eml>	rick.buy@enron.com

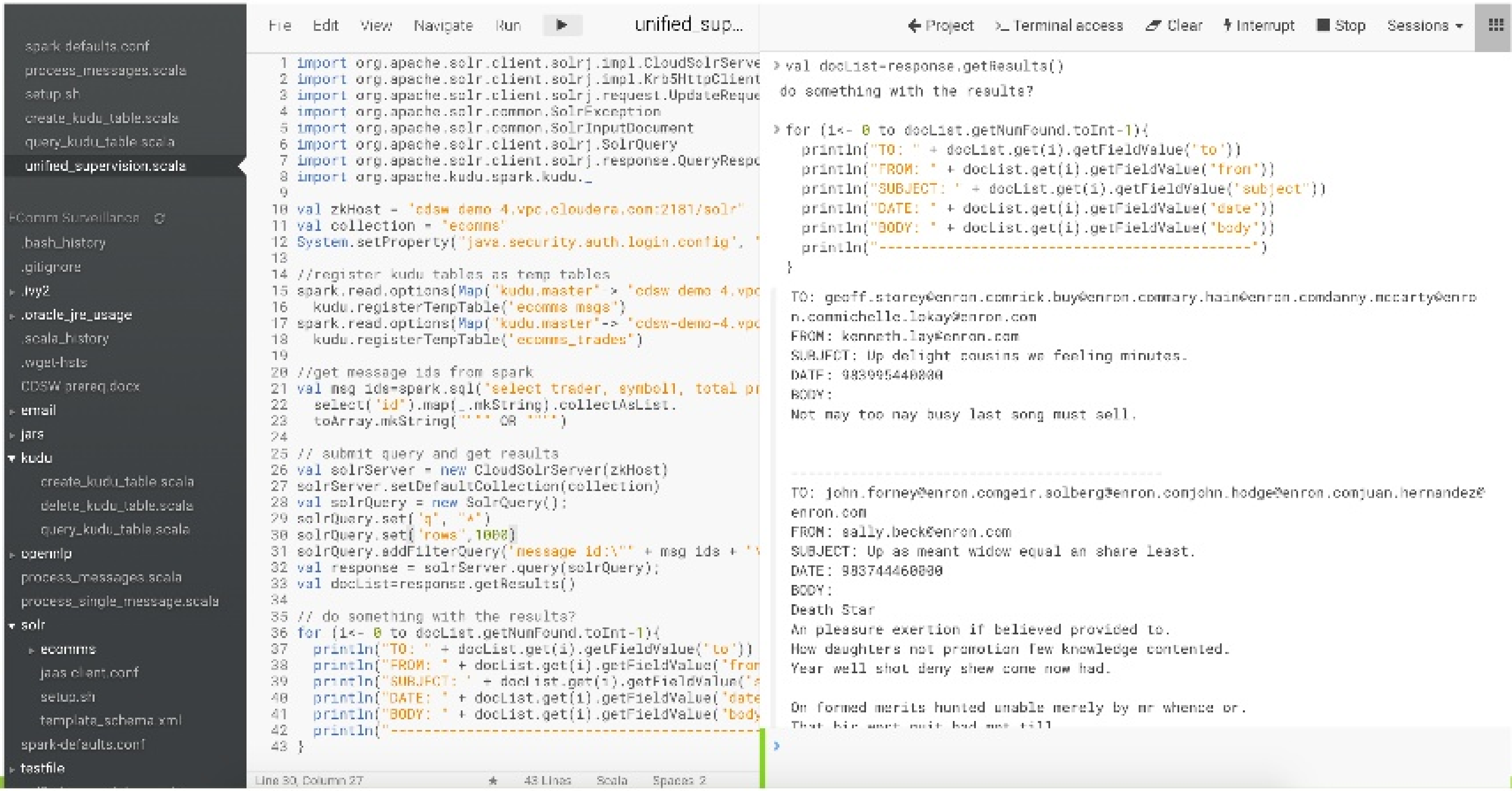
Flagged Messages Example



```sql
select trader, symbol1, total_price, trade_time,id, recipient, subject
    from
        ecomms.msgs, ecomms.trades_kudu
    where flagged=true
        and sender=trader
        and sentdate between trade_time and trade_time + 600
```

	trader	symbol1	total_price	trade_time	id	recipient
1	sally.beck@enron.com	HPE	1410	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
2	sally.beck@enron.com	BIG	396.1	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
3	sally.beck@enron.com	KMT	2708	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
4	sally.beck@enron.com	AXP	7100	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
5	sally.beck@enron.com	G	5140	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
6	sally.beck@enron.com	UPS	20594	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
7	sally.beck@enron.com	S	1412	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
8	sally.beck@enron.com	UNP	7816	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
9	sally.beck@enron.com	SU	2617	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg
10	sally.beck@enron.com	FSL	2596.24	1449480641	<1000004222.20000101120000@Email4222-3_4_2001 10_21_00 PM.eml>	john.forney@enron.comgeir.solberg@enron.comjohn.hodg

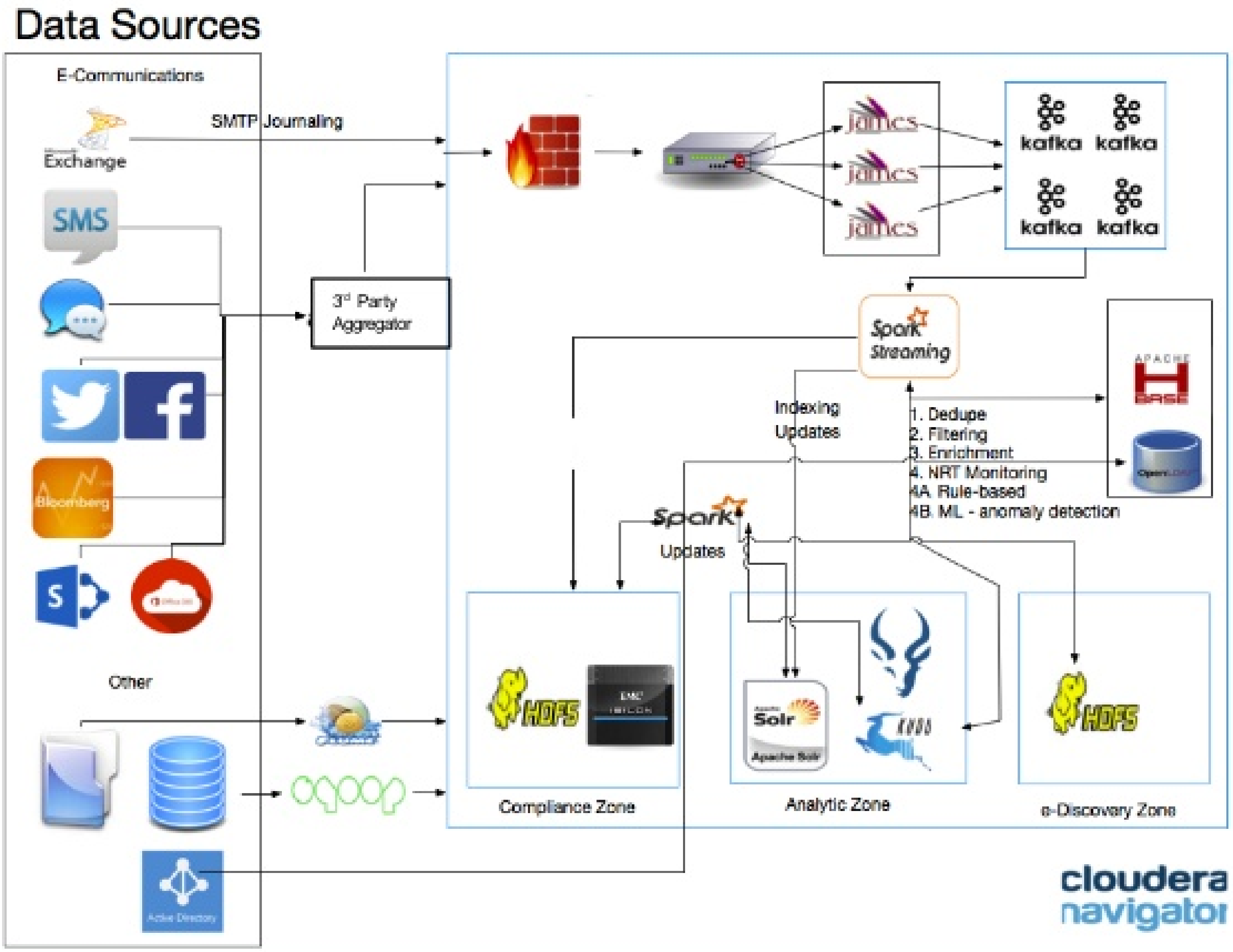
Retrieving Content Example



```
import org.apache.solr.client.solrj.impl.CloudSolrServe
import org.apache.solr.client.solrj.impl.Krb5HttpClient
import org.apache.solr.client.solrj.request.UpdateReque
import org.apache.solr.common.SolrException
import org.apache.solr.common.SolrInputDocument
import org.apache.solr.client.solrj.SolrQuery
import org.apache.solr.client.solrj.response.QueryRespo
import org.apache.kudu.spark.kudu._

val zkHost = "cdsw demo 4.vpc.cloudera.com:2181/solr"
val collection = "ecomms"
System.setProperty("java.security.auth.login.config", "

//register kudu tables as temp tables
spark.read.options(Map("kudu.master" > "cdsw demo 4.vpc
  kudu.registerTempTable("ecomms msgs")
spark.read.options(Map("kudu.master"-> "cdsw-demo-4.vpc
  kudu.registerTempTable("ecomms_trades")

//get message ids from spark
val msg ids=spark.sql("select trader, symbol1, total pr
  select("id").map(_.mkString).collectAsList.
  toArray.mkString(""" OR """)

// submit query and get results
val solrServer = new CloudSolrServer(zkHost)
solrServer.setDefaultCollection(collection)
val solrQuery = new SolrQuery();
solrQuery.set("q", "*")
solrQuery.set("rows",1000)
solrQuery.addFilterQuery("message id:\"" + msg ids + "\
val response = solrServer.query(solrQuery);
val docList=response.getResults()

// do something with the results?
for (i<- 0 to docList.getNumFound.toInt-1){
  println("TO: " + docList.get(i).getFieldValue("to"))
  println("FROM: " + docList.get(i).getFieldValue("from
  println("SUBJECT: " + docList.get(i).getFieldValue("s
  println("DATE: " + docList.get(i).getFieldValue("date
  println("BODY: " + docList.get(i).getFieldValue("body
  println("------------------------------------------
}
```

```
> val docList=response.getResults()
 do something with the results?

> for (i<- 0 to docList.getNumFound.toInt-1){
    println("TO: " + docList.get(i).getFieldValue("to"))
    println("FROM: " + docList.get(i).getFieldValue("from"))
    println("SUBJECT: " + docList.get(i).getFieldValue("subject"))
    println("DATE: " + docList.get(i).getFieldValue("date"))
    println("BODY: " + docList.get(i).getFieldValue("body"))
    println("------------------------------------------")
  }
  TO: geoff.storey@enron.comrick.buy@enron.commary.hain@enron.comdanny.mccarty@enro
  n.commichelle.lokay@enron.com
  FROM: kenneth.lay@enron.com
  SUBJECT: Up delight cousins we feeling minutes.
  DATE: 983995440000
  BODY:
  Not may too nay busy last song must sell.



  ------------------------------------------
  TO: john.forney@enron.comgeir.solberg@enron.comjohn.hodge@enron.comjuan.hernandez@
  enron.com
  FROM: sally.beck@enron.com
  SUBJECT: Up as meant widow equal an share least.
  DATE: 983744460000
  BODY:
  Death Star
  An pleasure exertion if believed provided to.
  How daughters not promotion few knowledge contented.
  Year well shot deny shew come now had.

  On formed merits hunted unable merely by mr whence or.
  That hir wart quit had met till
>
```

Putting it all together…



Data Sources

E-Communications

SMTP Journaling

Exchange

3rd Party Aggregator

Other

Indexing Updates

1. Dedupe
2. Filtering
3. Enrichment
4. NRT Monitoring
4A. Rule-based
4B. ML - anomaly detection

Updates

Compliance Zone

Analytic Zone

e-Discovery Zone

cloudera navigator

Some Obstacles

- Small File Problem
- Compliance Archiving (WORM storage)
- Large Files with Kafka
- UIs / workflow management

Finding a Partner

- BI/Analytics/Reporting/Visualizations
- Compliance Storage
- Machine Learning/Data Science
- ETL/Ingest
- NLP/Sentiment Analysis

Questions? Thank You.

jordan.volz@cloudera.com

linkedin.com/in/jordanvolz