Big Data and Analytics on AWS

KD Singh
Solutions Architect
Amazon Web Services

What is big data?

When your data sets become **so large that you have to start innovating** around how to collect, store, organize, analyze, and share it

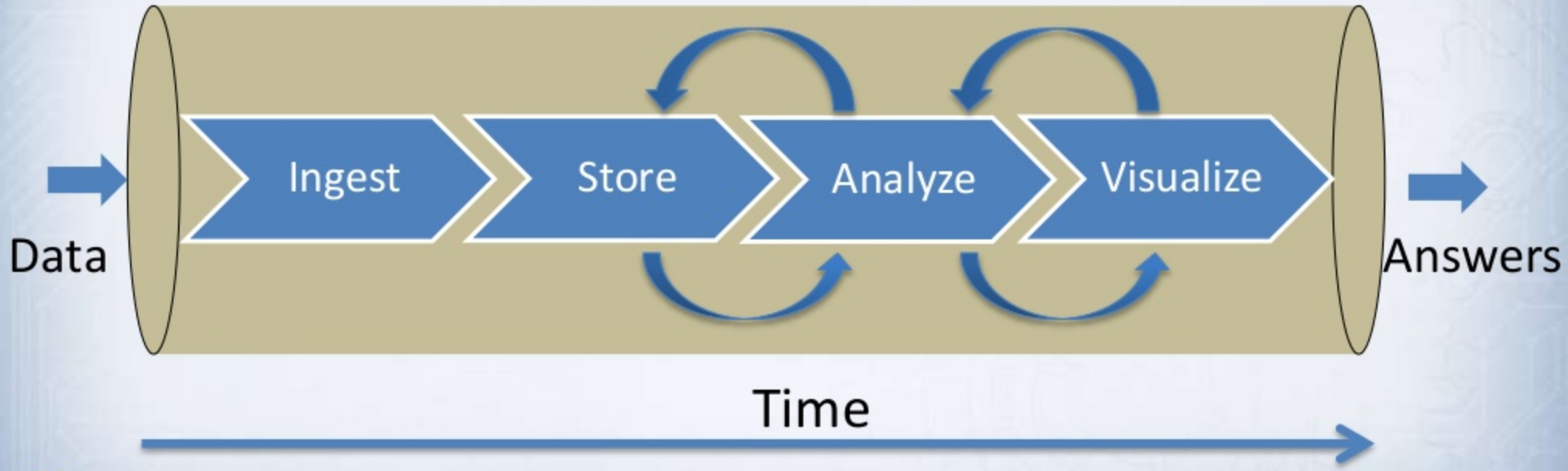
- Velocity
  - Rate of data flow in
- Latency
  - High or Low
- Volume
  - High or Low
- Variety
  - Diversity of source data
- Item Size
  - KB or MB
- Request Rate
  - Access patterns
- Change Rate
  - How much is the data changing?
- Processing Requirements
  - How much computation?
- Durability
  - Preservation of source data?
- Availability
  - Tolerance for downtime?
- Growth Rate
  - Rate of data growth?
- Views
  - The diversity of consumers?

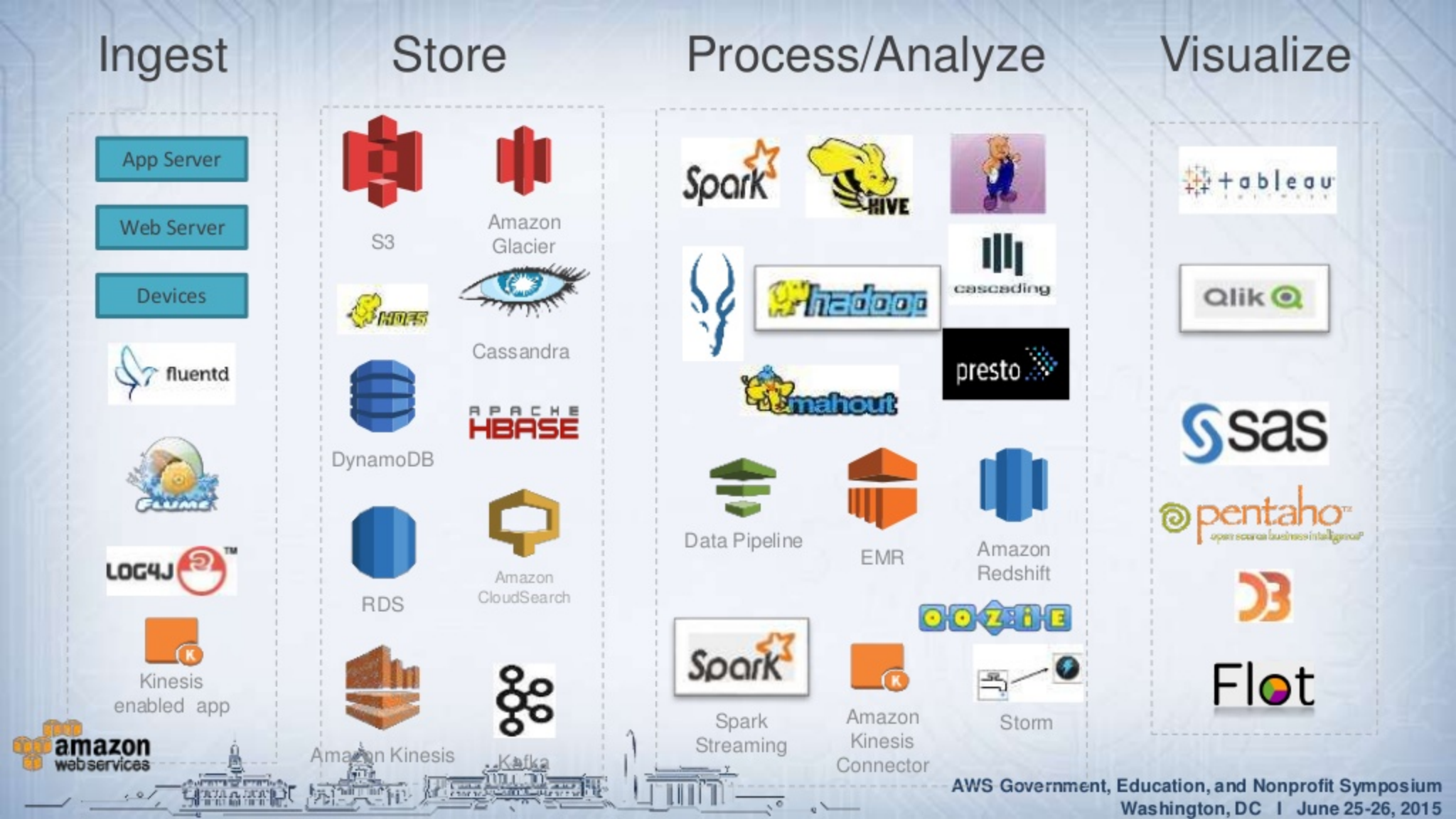
Plethora of tools

Cassandra

Simplify data analytics flow

Multiple stages
Storage decoupled from processing




Ingest
Store
Process/Analyze
Visualize
App Server
Web Server
Devices
fluentd
Flume
LOG4J
Kinesis enabled app
S3
Amazon Glacier
HDFS
Cassandra
DynamoDB
APACHE HBASE
RDS
Amazon CloudSearch
Amazon Kinesis
Kafka
Spark
HIVE
hadoop
cascading
presto
mahout
Data Pipeline
EMR
Amazon Redshift
OOZIE
Spark Streaming
Amazon Kinesis Connector
Storm
tableau
Qlik
sas
pentaho
Flot
amazon web services
AWS Government, Education, and Nonprofit Symposium
Washington, DC | June 25-26, 2015

AWS big data portfolio

Collect / Ingest

Amazon Kinesis

AWS Import/Export

AWS Direct Connect

Amazon SQS

Store

Amazon S3

Amazon DynamoDB

Amazon Glacier

Amazon RDS

Process / Analyze

Amazon EMR

Amazon EC2

Amazon Redshift

AWS Data Pipeline

Visualize / Report

tableau

sas

INFORMATICA

MicroStrategy

SAP Business Objects

JASPERSOFT

Industries using AWS for data analysis

Mobile / Cable Telecom

SAMSUNG
NOKIA
COMCAST
orange
vodafone

Oil and Gas Industrial Manufacturing

Schneider Electric
SIEMENS
LAMBORGHINI

Retail/Consumer Entertainment Hospitality

amazon
Unilever
ticketmaster
RED LION HOTELS
airbnb

Life Sciences Scientific Exploration

Pfizer
NOVARTIS
Bristol-Myers Squibb
illumina
NASA

Financial Services

FINRA
NASDAQ
bankinter
SUNCORP
Washington Trust Bank

Publishing Media Advertising

4
News International
Discovery COMMUNICATIONS
THOMSON REUTERS
The Washington Post
gettyimages
CONDÉ NAST

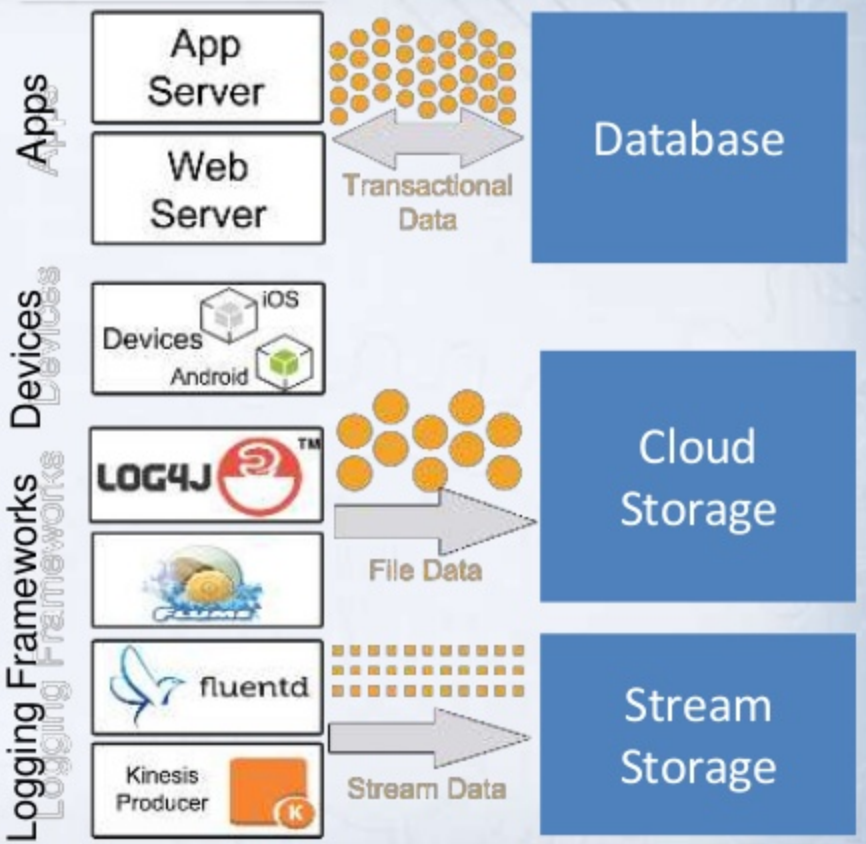
Online Media Social Network Gaming

NETFLIX
Flipboard
Spotify
foursquare
ROVIO

Ingest: The act of collecting and storing data

Types of data ingest

- Transactional
  - Database reads/writes
- File
  - Media files; log files
- Stream
  - Click-stream logs (sets of events)



Amazon Kinesis

Real-time processing of streaming data

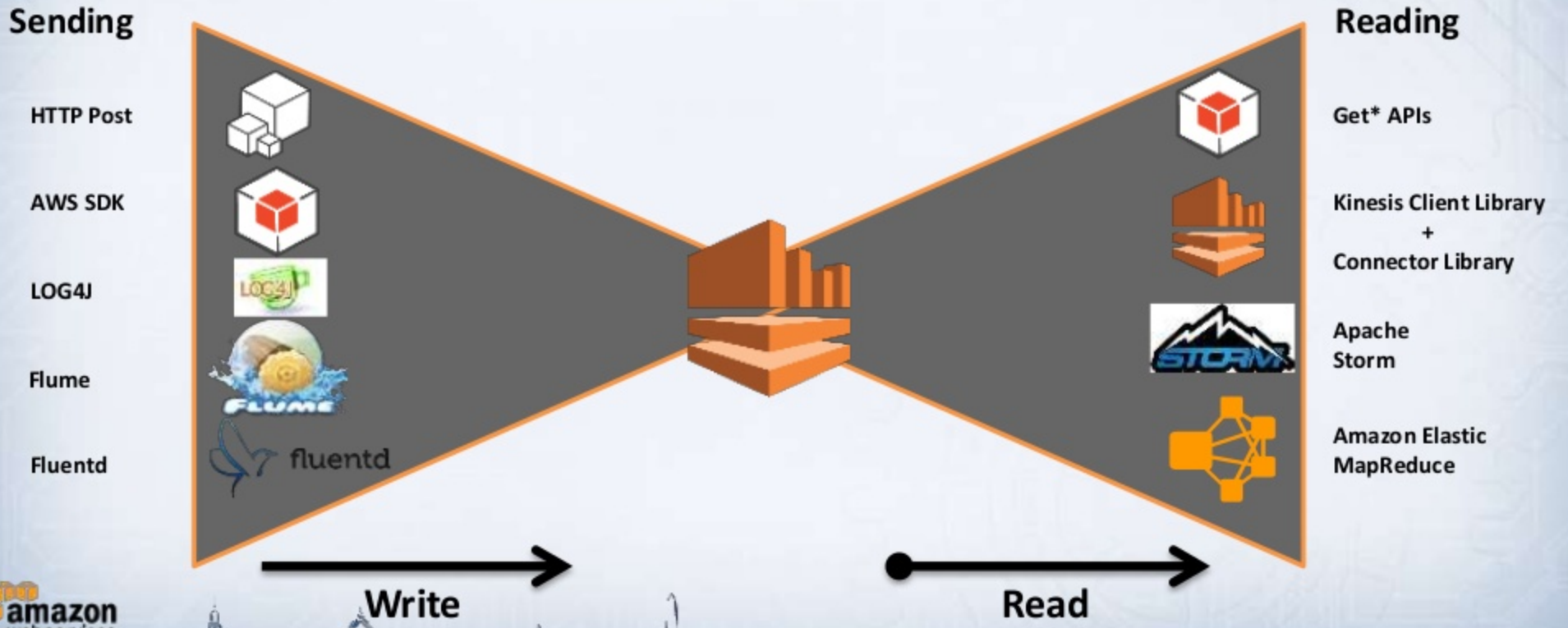
High throughput

Elastic

Easy to use

Connectors for EMR, S3, Amazon Redshift, DynamoDB

Sending and reading data from Amazon Kinesis streams



AWS Partners for data ingest, load, and transformation

ATTUNITY

pentaho

birst

sas

snapLogic

talend
integration at any scale

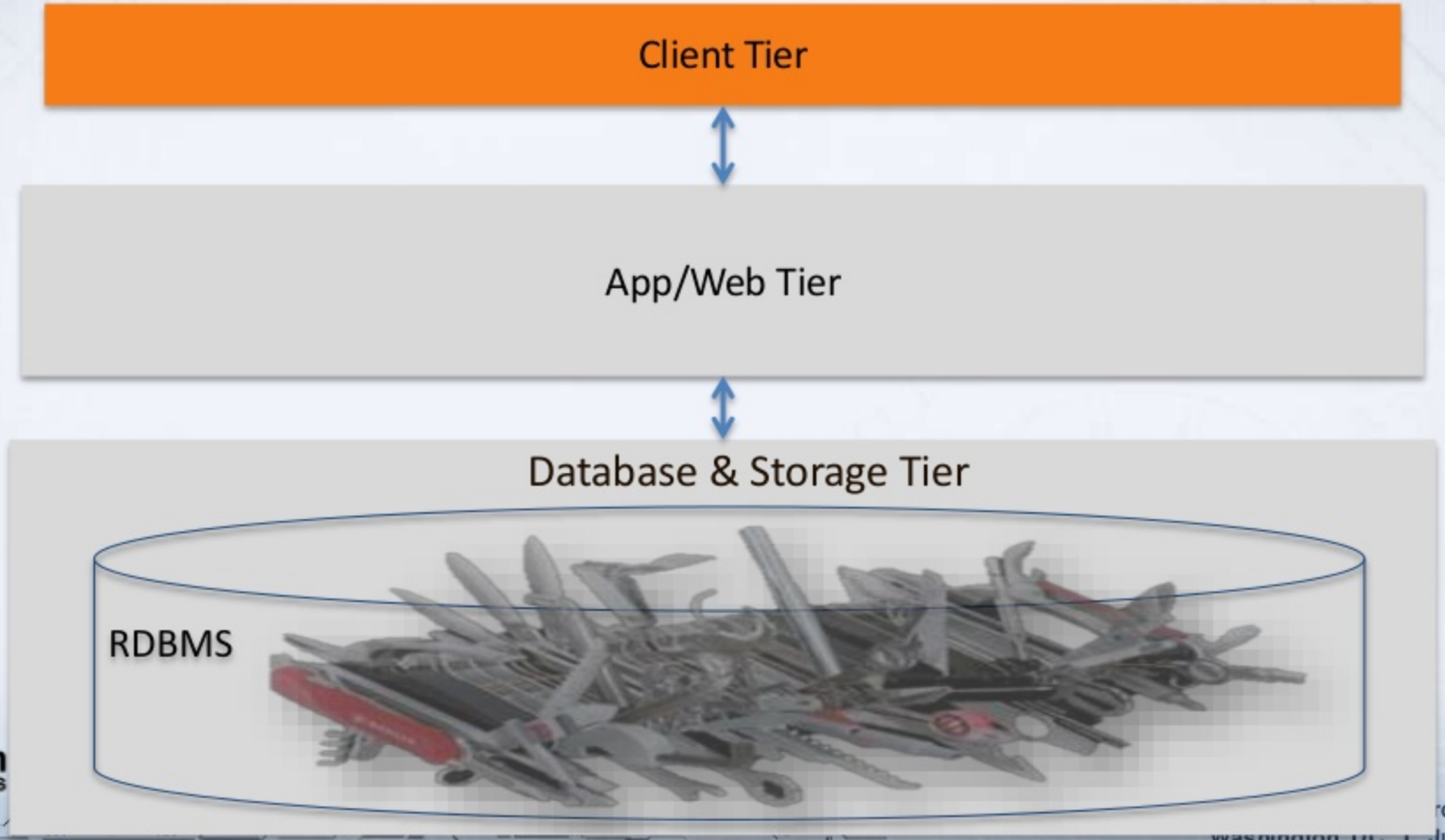
cloudera
Flume, Sqoop

INTEGRATING BIG DATA... SMARTER
syncsort

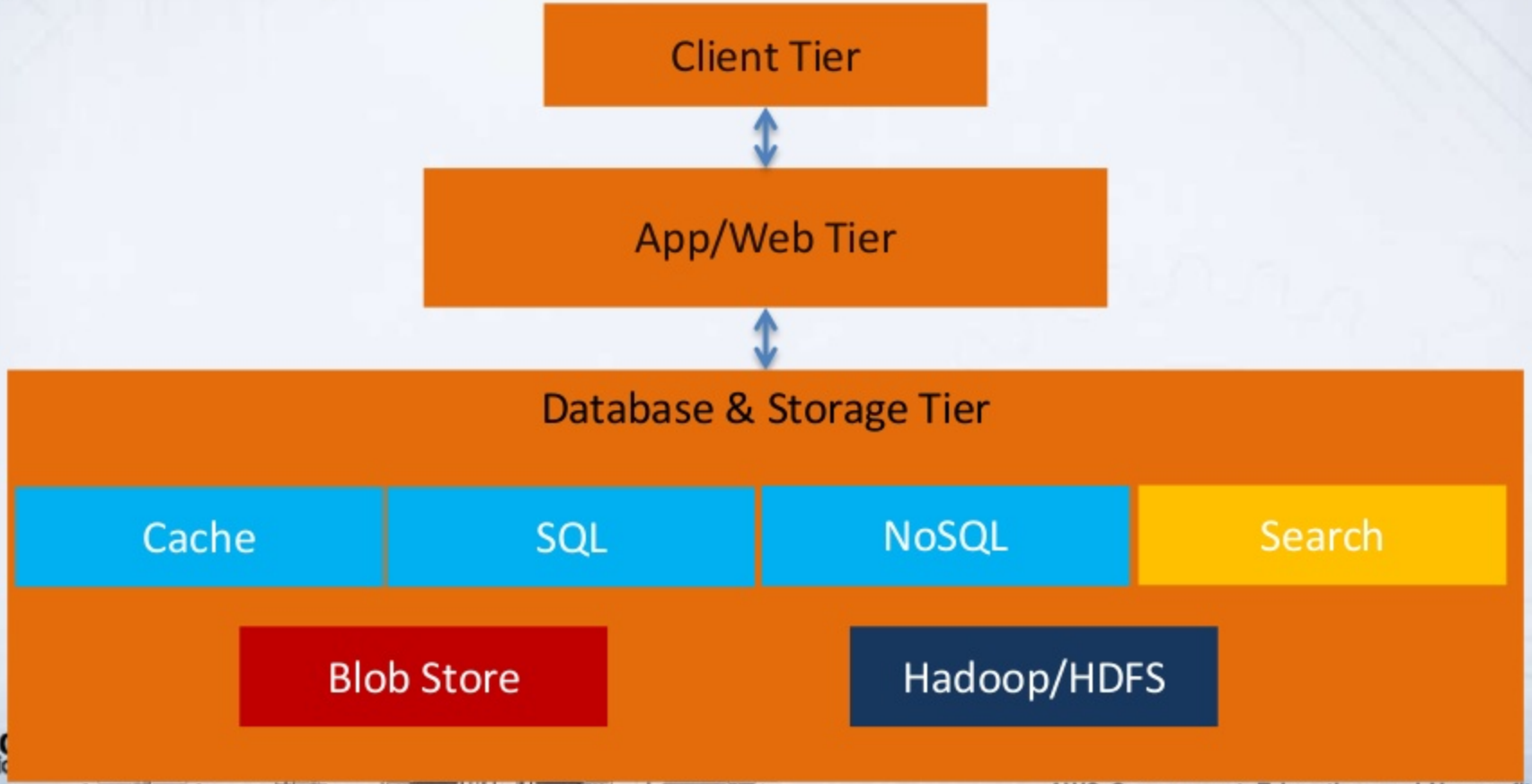
INFORMATICA
Hparser, Big Data Edition

Storage

Cloud database and storage tier anti-pattern



Cloud database and storage tier — use the right tool for the job!



Cloud database and storage tier — use the right tool for the job!

Amazon
S3

Store anything

Object storage

Scalable

Designed for 99.999999999% durability

Aggregate all data in S3 surrounded by a collection of the right tools

- No limit on the number of objects
- Object size up to 5 TB
- Central data storage for all systems
- High bandwidth
- 99.999999999% durability
- Versioning; lifecycle policies
- Amazon Glacier integration

Amazon
DynamoDB

Fully managed NoSQL database service

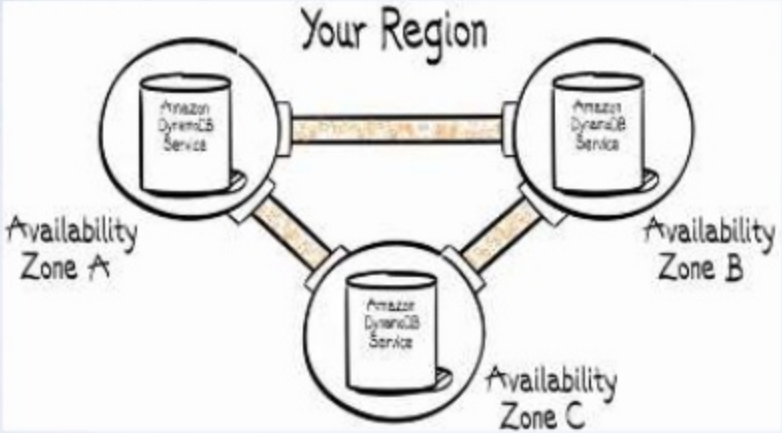
Built on solid-state drives (SSDs)

Consistent low-latency performance

Any throughput rate

No storage limits

DynamoDB: managed high availability and durability



- Regional service
- Synchronous replication to three Availability Zones
- Writes acknowledged only when they are on disk in at least two AZs

- Scaling without downtime
- Automatic sharding
- Security inspections, patches, upgrades
- Automatic hardware failover
- Multi-AZ replication
- Hardware configuration designed specifically for DynamoDB
- Performance tuning

Amazon
RDS

Relational databases

Fully managed; zero admin

MySQL, PostgreSQL, Oracle, SQL Server

Aurora

Process and analyze

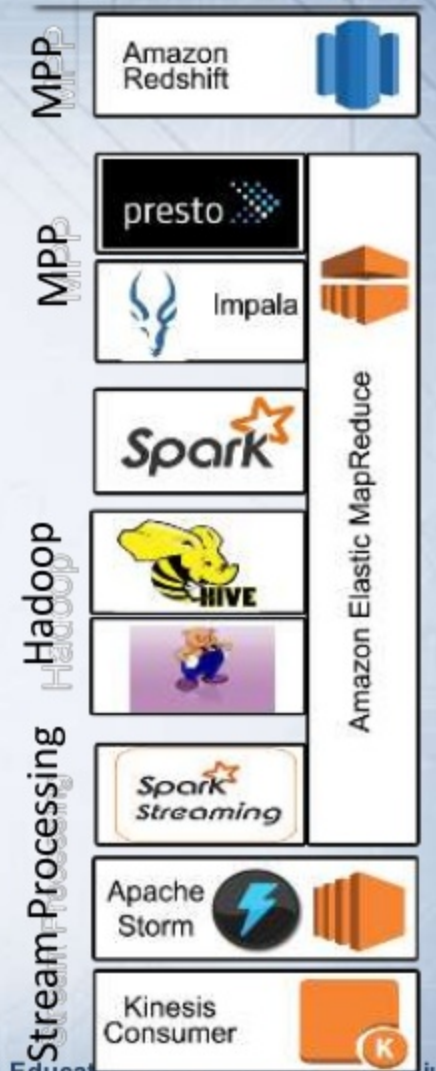
Processing frameworks

- Batch processing
  - Take large amount (>100 TB) of cold data and ask questions
  - Takes minutes or hours to get answers back
  - Example: Generating hourly, daily, weekly reports

- Stream processing (real-time)
  - Take small amount of hot data and ask questions
  - Takes short amount of time to get your answer back
  - Example: 1 min metrics

Processing frameworks

- Batch processing/analytic
  - Amazon Redshift
  - Amazon EMR (Hadoop)
  - Spark, Hive/Tez, Pig, Impala, Presto, ….
- Stream processing
  - Amazon Kinesis client and connector library
  - Spark Streaming
  - Storm (+Trident)

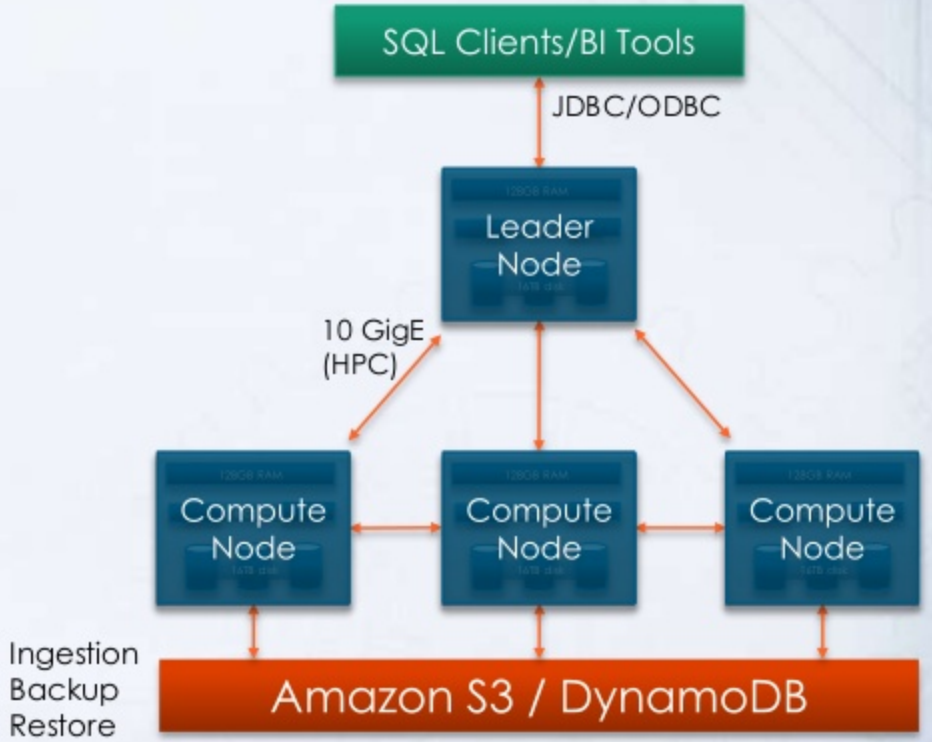


Amazon
Redshift

- Columnar data warehouse
- ANSI SQL compatible
- Massively parallel
- Petabyte scale
- Fully managed
- *Very* cost-effective

Amazon Redshift architecture

- Leader Node
  - SQL endpoint
  - Stores metadata
  - Coordinates query execution
- Compute Nodes
  - Local, columnar storage
  - Execute queries in parallel
  - Load, backup, restore via Amazon S3
  - Parallel load from Amazon DynamoDB
- Hardware optimized for data processing
- Two hardware platforms
  - DS2 (dense storage): HDD; scale to 1.6PB
  - DC1 (dense compute): SSD; scale to 256TB



Amazon
Elastic
MapReduce

Hadoop/HDFS clusters

Hive, Pig, Impala, HBase

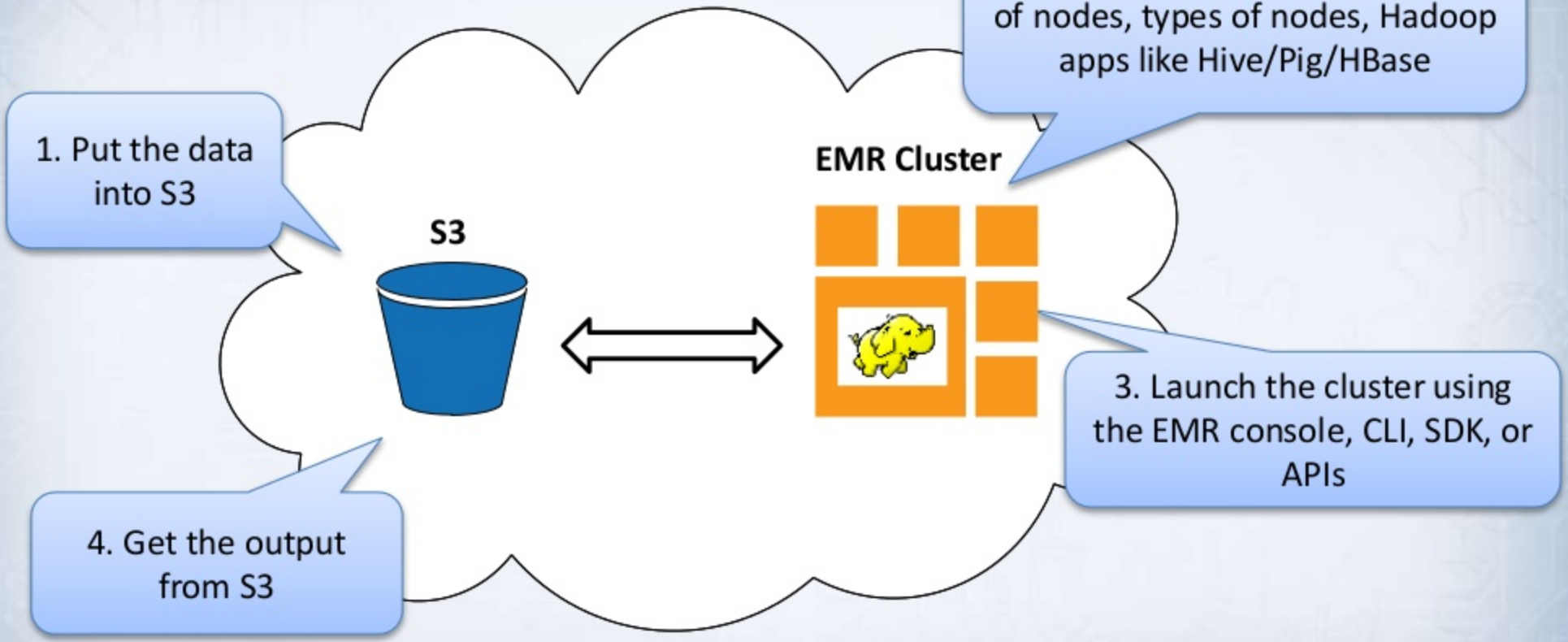
Easy to use; fully managed

On-demand and spot pricing

Tight integration with S3,

DynamoDB, and Amazon Kinesis

How does EMR work?



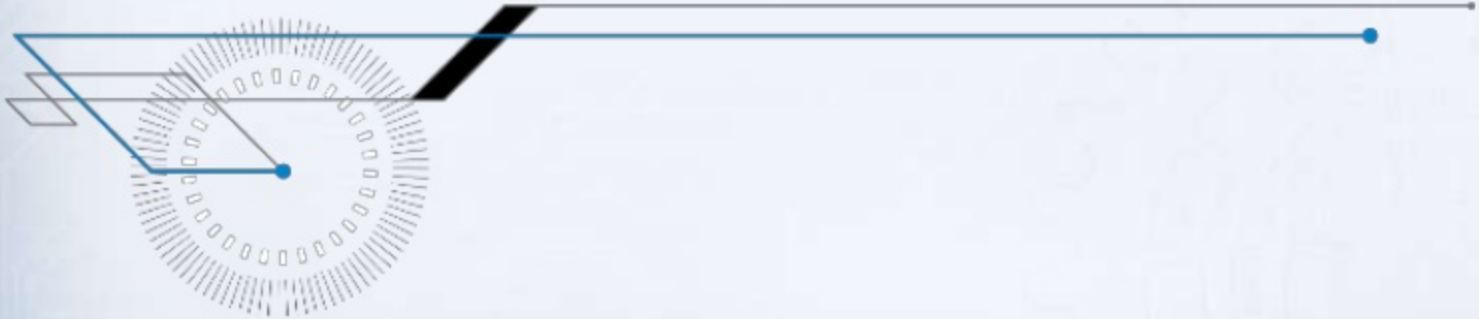
The Hadoop ecosystem works with EMR

Partners – advanced analytics

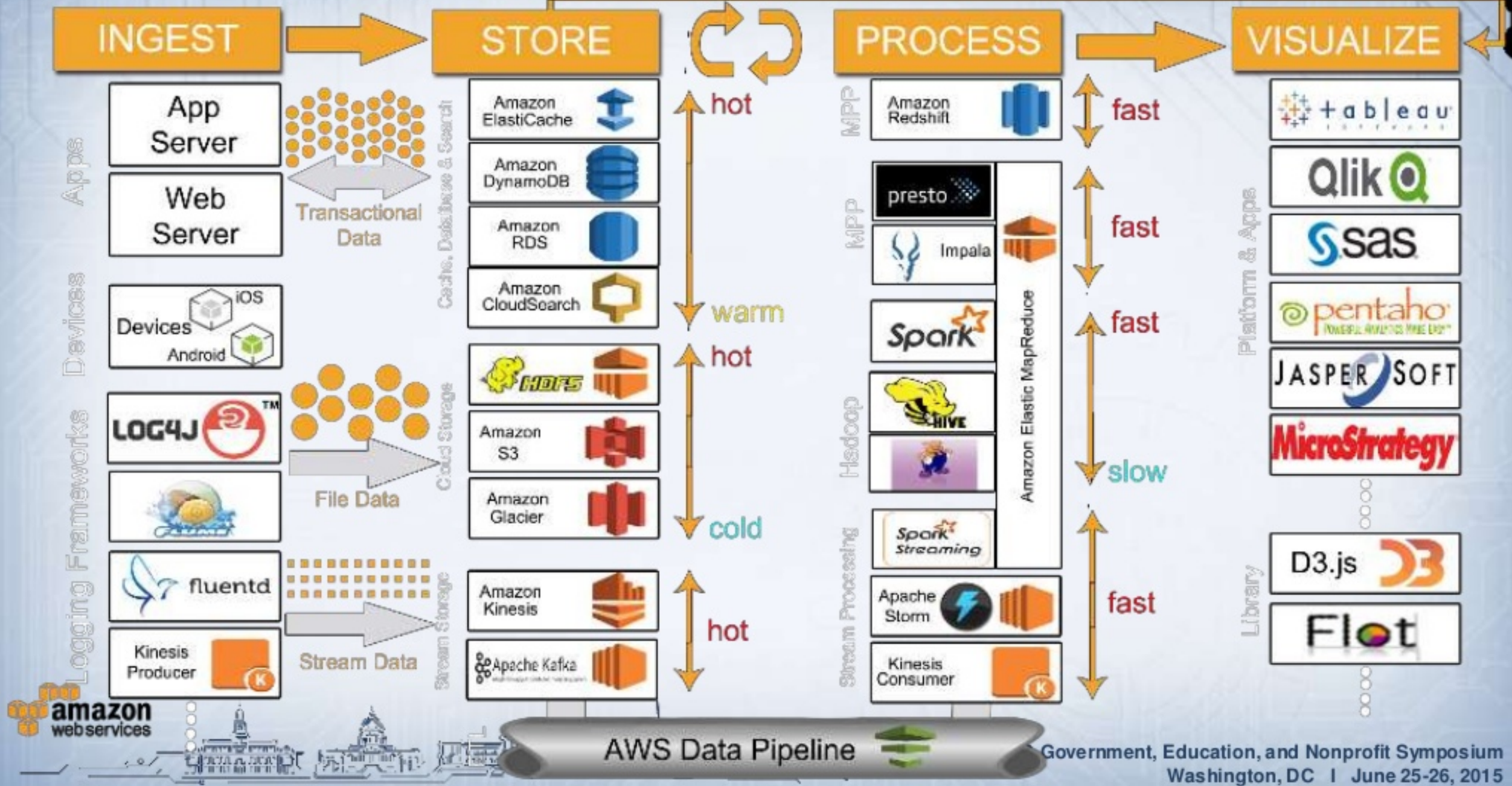
Visualize

AWS Partners for BI & data visualization

Putting it all together



Big Data Reference Architecture



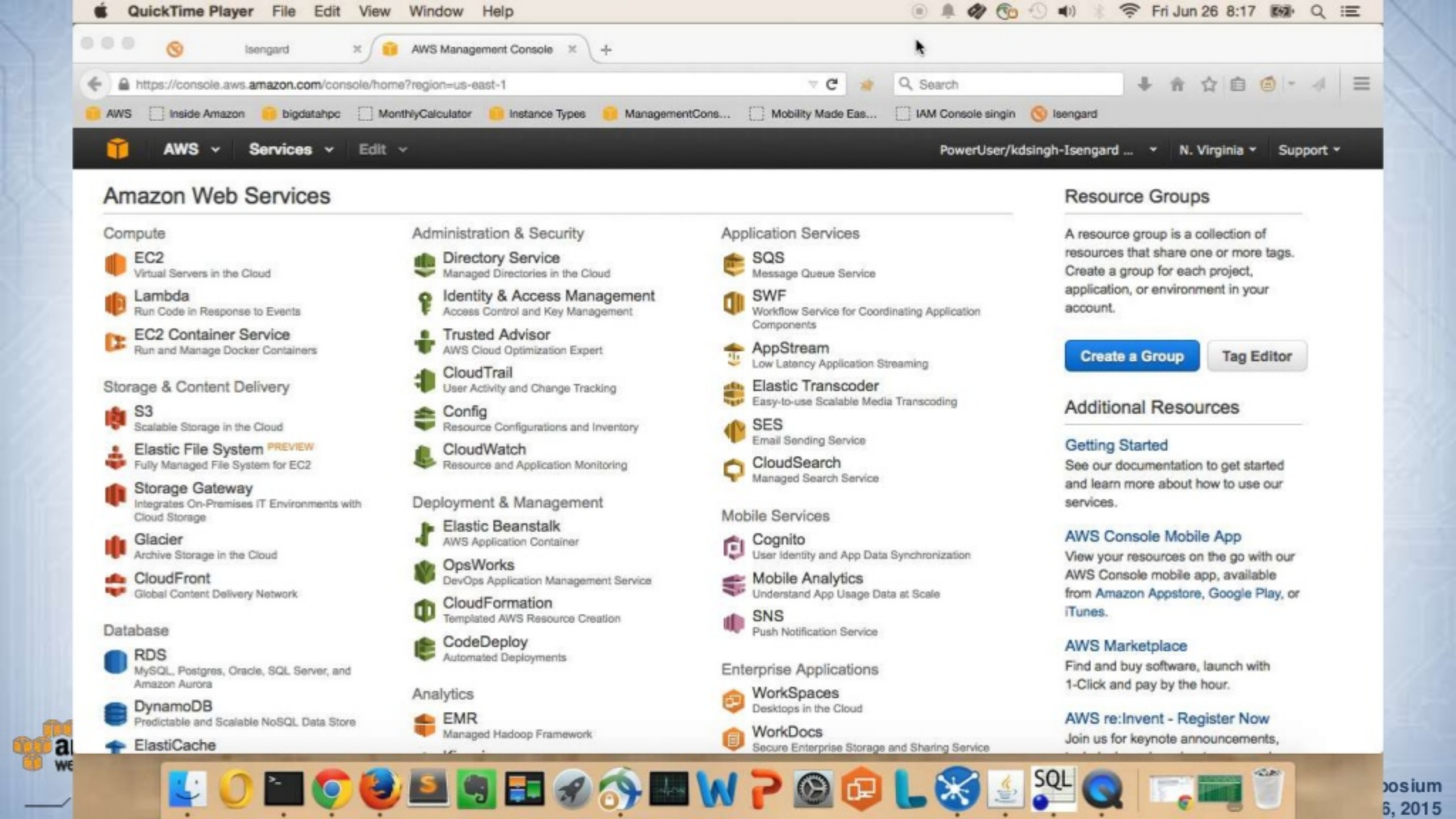
Demo

Traffic Statistics

Amazon EMR as ETL Grid and Analysis

Amazon Redshift – Production DWH

Visualization


QuickTime Player File Edit View Window Help
Fri Jun 26 8:17
Isengard
AWS Management Console
https://console.aws.amazon.com/console/home?region=us-east-1
Search
AWS
Inside Amazon
bigdatahpc
MonthlyCalculator
Instance Types
ManagementCons...
Mobility Made Eas...
IAM Console singin
Isengard
AWS
Services
Edit
PowerUser/kdsingh-Isengard ...
N. Virginia
Support
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
Lambda
Run Code in Response to Events
EC2 Container Service
Run and Manage Docker Containers
Storage & Content Delivery
S3
Scalable Storage in the Cloud
Elastic File System PREVIEW
Fully Managed File System for EC2
Storage Gateway
Integrates On-Premises IT Environments with Cloud Storage
Glacier
Archive Storage in the Cloud
CloudFront
Global Content Delivery Network
Database
RDS
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
DynamoDB
Predictable and Scalable NoSQL Data Store
ElastiCache
Administration & Security
Directory Service
Managed Directories in the Cloud
Identity & Access Management
Access Control and Key Management
Trusted Advisor
AWS Cloud Optimization Expert
CloudTrail
User Activity and Change Tracking
Config
Resource Configurations and Inventory
CloudWatch
Resource and Application Monitoring
Deployment & Management
Elastic Beanstalk
AWS Application Container
OpsWorks
DevOps Application Management Service
CloudFormation
Templated AWS Resource Creation
CodeDeploy
Automated Deployments
Analytics
EMR
Managed Hadoop Framework
Application Services
SQS
Message Queue Service
SWF
Workflow Service for Coordinating Application Components
AppStream
Low Latency Application Streaming
Elastic Transcoder
Easy-to-use Scalable Media Transcoding
SES
Email Sending Service
CloudSearch
Managed Search Service
Mobile Services
Cognito
User Identity and App Data Synchronization
Mobile Analytics
Understand App Usage Data at Scale
SNS
Push Notification Service
Enterprise Applications
WorkSpaces
Desktops in the Cloud
WorkDocs
Secure Enterprise Storage and Sharing Service
Resource Groups
A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.
Create a Group
Tag Editor
Additional Resources
Getting Started
See our documentation to get started and learn more about how to use our services.
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.
AWS re:Invent - Register Now
Join us for keynote announcements,

ICAO and Hadoop

Marco Merens
Chief (Acting) Integrated Analysis
International Civil Aviation Organization

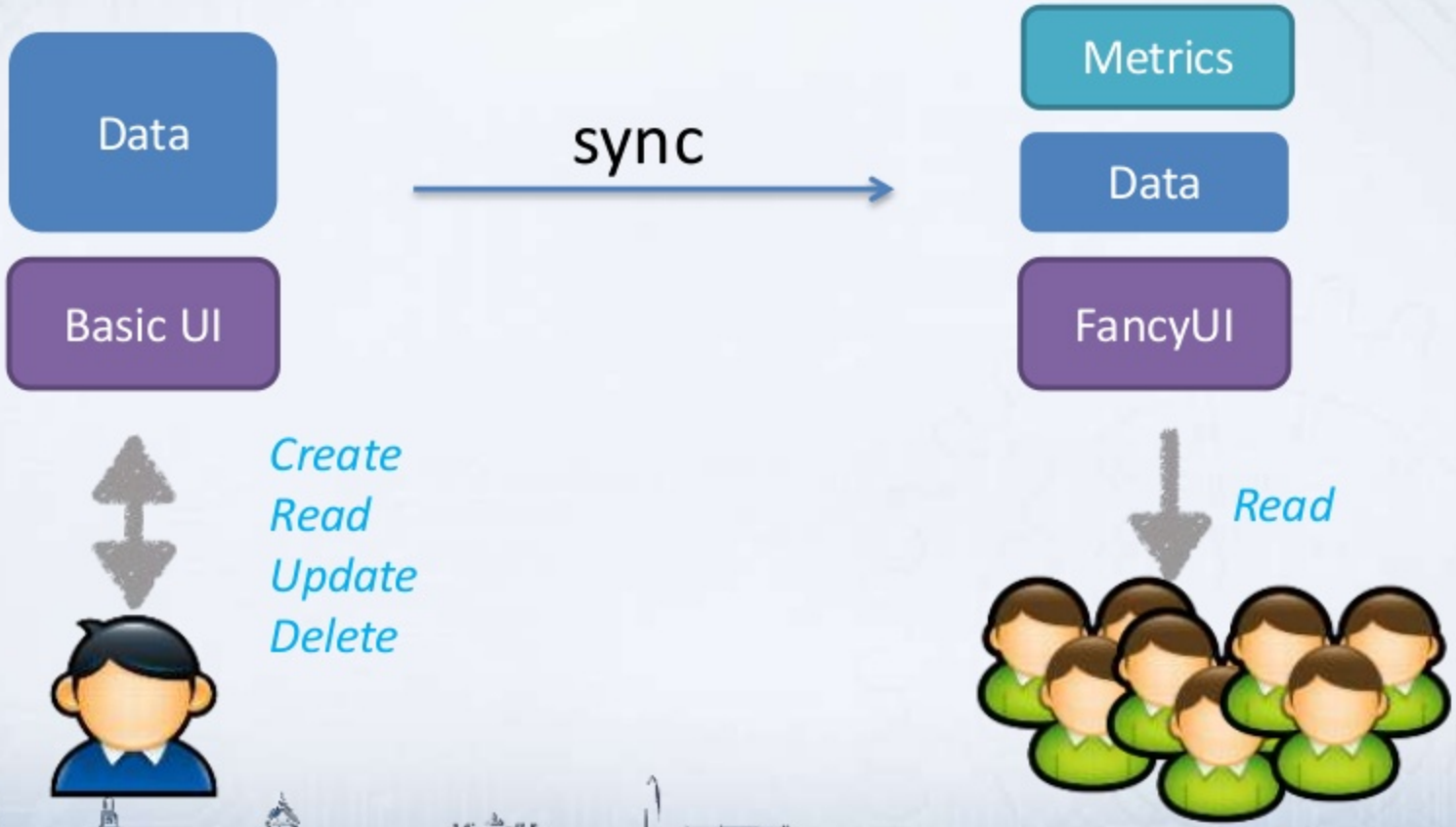
ICAO in the cloud

It's faster, scales better, and is more flexible than in-house infrastructure

Cloudability principles at ICAO

1. What comes from the cloud, can stay in the cloud
2. What comes from in-house
   A. should stay in-house if private, or
   B. can be **synced** with the cloud if public

Data sync


Data
Basic UI
sync
Metrics
Data
FancyUI
Create
Read
Update
Delete
Read

EMR example: blended accident list

⊖	2014-09-06	Mogadishu Somalia	FOKKERF50 5Y-BYE Jubba Airways Somalia	Landing Accident	RS

Narratives
ADREP: A Fokker 50 aircraft operated by Jubba Airways inbound from Galkayo to Mogadishu sustained

ASCEND: The aircraft apparently veered to the right and ran off the right side of the runway on landing at Mogadishu International Airport. After leaving the runway the aircraft ran down a small embankment and its nose/forward fuselage impacted a concrete perimeter wall. The left main undercarriage collapsed and the left wing/engine struck the ground. The 24 passengers and crew suffered no injuries in the accident which happened in daylight (1425L). The aircraft was operating a domestic flight from Galkayo.

ASN: A Fokker 50 passenger plane, operating on a flight for Jubba Airways, sustained substantial damage in a runway excursion accident at Mogadishu International Airport (MGQ), Somalia.The airplane performed flight 3J715 from Galkayo Airport (GLK) to Mogadishu. Upon landing the airplane suffered a runway excursion. It went down a small embankment and collided with a concrete perimeter fence. The right hand main landing gear collapsed, the nose was crushed and the forward fuselage sustained structural damage.

AVHERALD: A Jubba Airways Fokker 50, registration 5Y-BYE performing flight 6J-715 from Galkayo (Puntland/Somalia) to Mogadishu (Somalia) with 24 people on board, landed on Mogadishu's runway 05 at about 10:30L (07:30Z), however, the right main gear collapsed causing the aircraft to veer right off the runway. The aircraft came to a stop in a ditch receiving additional damage to nose section and right wing and right engine/propeller. There were no injuries, the aircraft sustained substantial damage.Airport officials reported the aircraft had safely landing when during rollout the landing gear collapsed and the aircraft arriving from Galkayo went off the runway. All passengers disembarked in good condition.No weather data are available for Mogadishu (neither Metars nor local weather station data), photographic evidence suggests clear weather however (winds unknown).5Y-BYE shortly after the runway excursion (Photo: bizjets101):5Y-BYE being recovered suggesting right wing fractured (Photo: Neil Wigan):

Input format

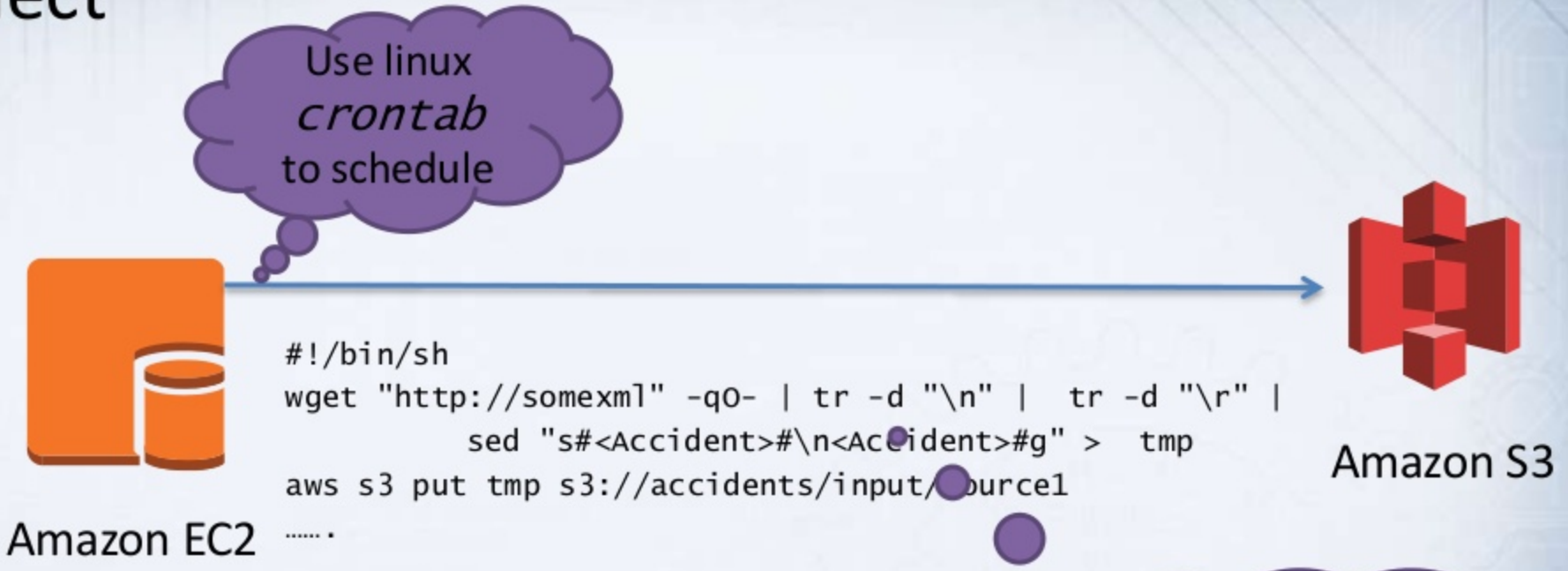
XML

```
<?xml version="1.0" encoding="utf-8"?>
<root>
<ADREP><FilingInformation
State="XX"><ReportingOrganization>Ascend</ReportingOrganization>
<StateFileNumber>S1982045</StateFileNumber>
<Headline>MU-2, Collision with high ground, (near) Kelowna</Headline>
</FilingInformation>
….
</root>
```

CSV

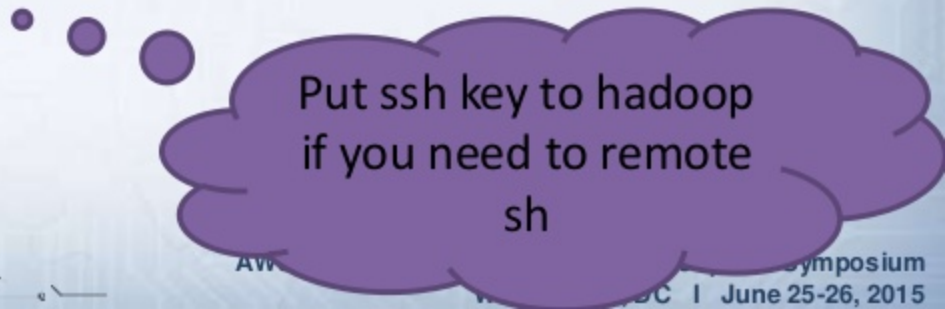
```
|26/12/2001|Germany|Germany|"ICE:Icing"|Accident|Fatal|8|Germany|Bremerhav
en|D-IAAI|"BRITTEN NORMAN"||||"2 251 to 5 700 Kg"|Scheduled|Airplane|Take-
off||
```

Collect


Use linux
crontab
to schedule
Amazon EC2
#!/bin/sh
wget "http://somexml" -qO- | tr -d "\n" | tr -d "\r" |
sed "s#<Accident>#\n<Accident>#g" > tmp
aws s3 put tmp s3://accidents/input/source1
……
Amazon S3
Make One XML
element per line for
EMR

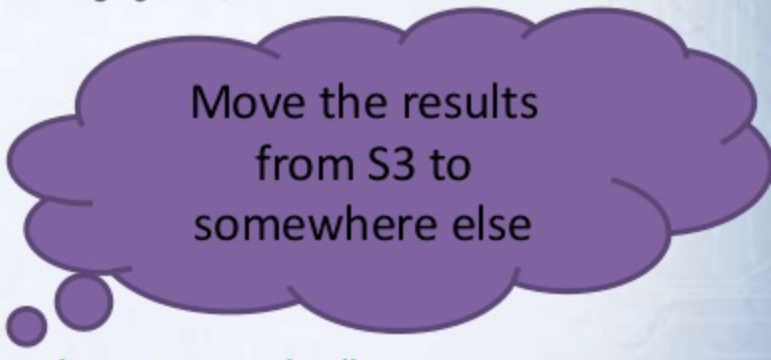
EMR command line

```
elastic-mapreduce
--create
--bootstrap-action
s3://elasticmapreduce/samples/node/install-node-bin-
x86.sh
--instance-type m1.small --instance-count 3
--json job.json
--put /home/ec2-user/key/newtest.pem
--to /home/hadoop
--enable-debugging
```



EMR json config file

```
[{
        "Name": "Make accident map",
        "ActionOnFailure": "CANCEL_AND_WAIT",
        "HadoopJarStep": {
        "Jar":"/home/hadoop/contrib/streaming/hadoop-streaming.jar",
            "Args": [
                "-input",
                "s3://accidentstats/input/*", …
               ]},{
        "Name": "Store in mongo",
        "ActionOnFailure": "CANCEL_AND_WAIT",
        "HadoopJarStep": {
        "Jar":"s3://elasticmapreduce/libs/script-runner/script-runner.jar" ,
            "Args": [
                "s3://edmscripts/uploadtomongo.sh",
                "accidentstats/output",
                "NEWACCIDENTLIST"
        ]}}
```



Map

Amazon S3

sourceX

```
#!/usr/bin/env node
function treatline(line) {
If (line.indexOf("<ADREP>"))
            {
            source1(line)
            }
……
```

Check which mapper to use

```
Function source1(line)
{
Var data=xml2json(line)
data.records.forEach(function(v){
            Var el={   Date:v.Date,

            Registration:v.Registration,
                                    Model:v.Model,
                                    Source:"Source1",
                                    Priority:1
                        }
            var key=el.Date+"#"+el.Registration
            process.stdout.write(key+"/t"+JSON.stringify(el)
)
            })
}
```

Build the data and the key and emit

Amazon Elastic MapReduce

mapped

Reduce

Amazon Elastic MapReduce

Mapped and sorted

```
#!/usr/bin/env node
Var oldkey,key,array=[]
function treatline(line) {
key=line.split("/t")[0]
data=JSON.parse(line.split("/t")[1])
If ((key==oldkey) || !oldkey)
            {
            array.push(data)}
Else {
            treat(array)
            array=[]}
oldkey=key
……}
```

Collect the same keys

Amazon S3

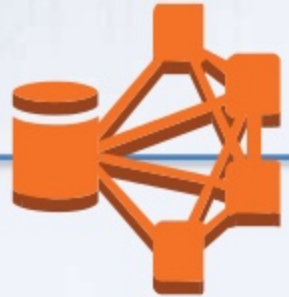
Reduced

```
Function treat(array)
{
el={}
array=array.sort(prioritysort)
array.forEach(function(v){
            el=updateresult(el,v)
})
process.stdout.write(JSON.stringify(el)+"\n")
}
```

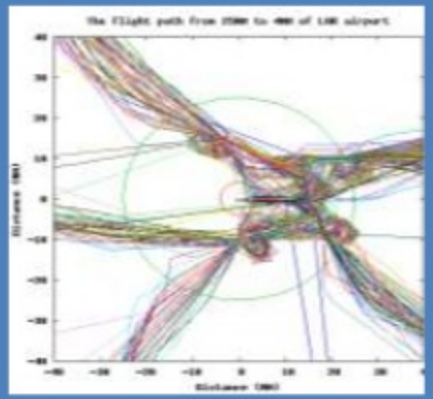
Sort the data according to priority
Build the output

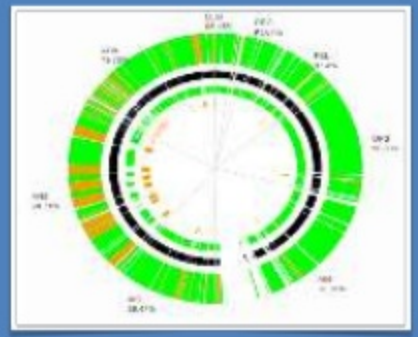
amazon web services

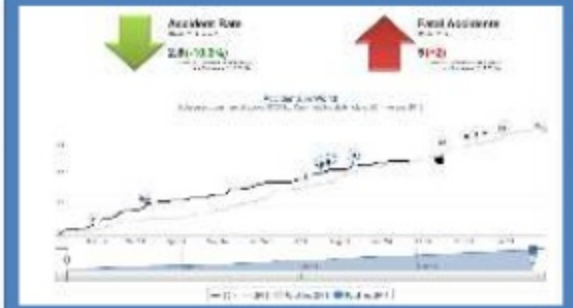
Real-time statistics



Amazon Elastic MapReduce







Thank You.

This presentation will be loaded to SlideShare the week following the Symposium.
http://www.slideshare.net/AmazonWebServices