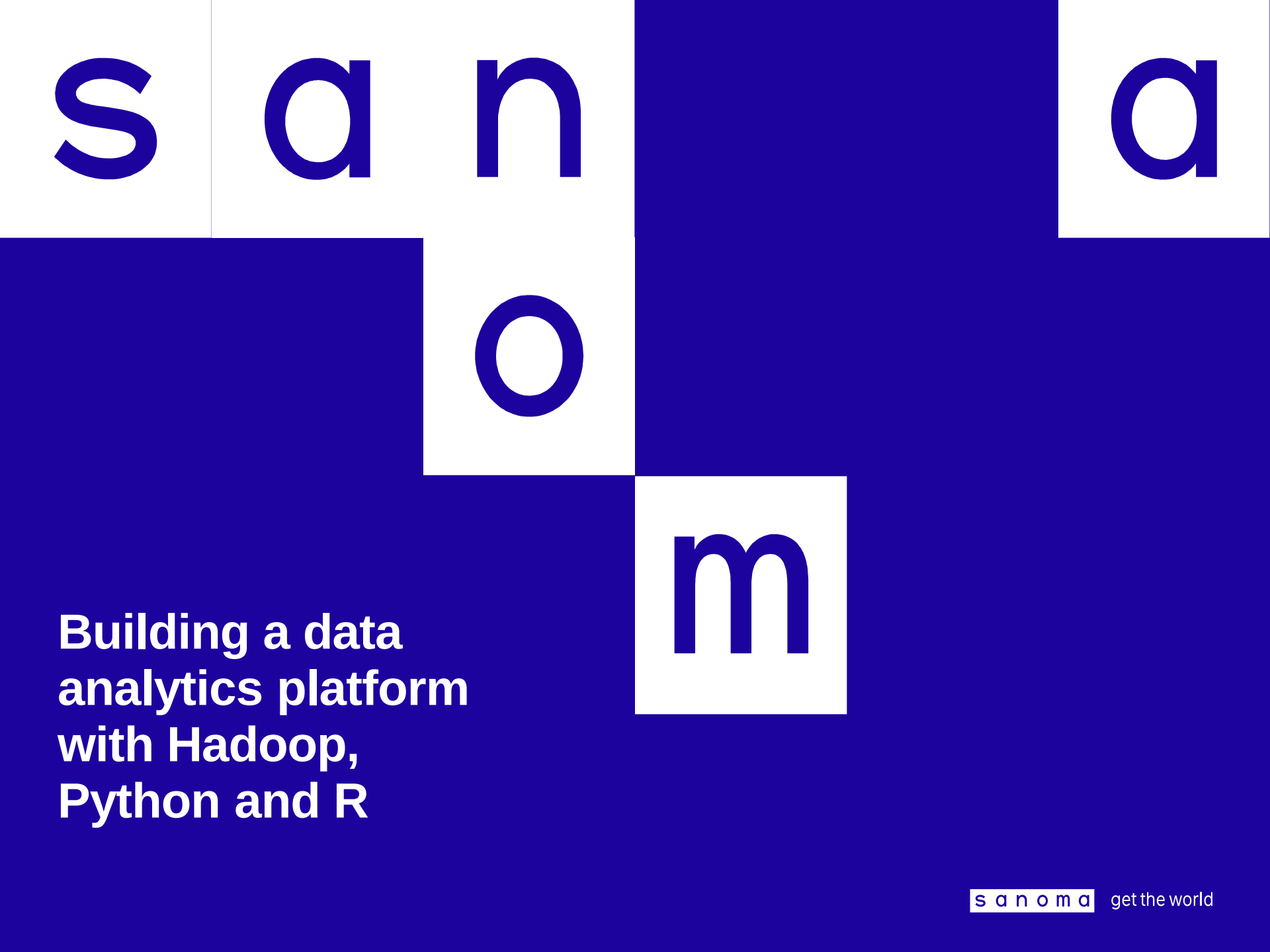


Building a data analytics platform with Hadoop, Python and R

sanoma get the world

Agenda

- Me
- Sanoma
- Past
- Present
- Future

/me
@skieft

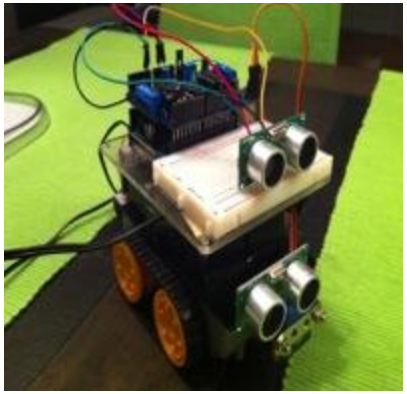
- Software architect for Sanoma
- Managing the data and search team
- Focus on the digital operation

- Work:
  - Centralized services
  - Data platform
  - Search

- Like:
  - Work
  - Water(sports)
  - Whiskey
  - Tinkering: Arduino, Raspberry PI, soldering stuff

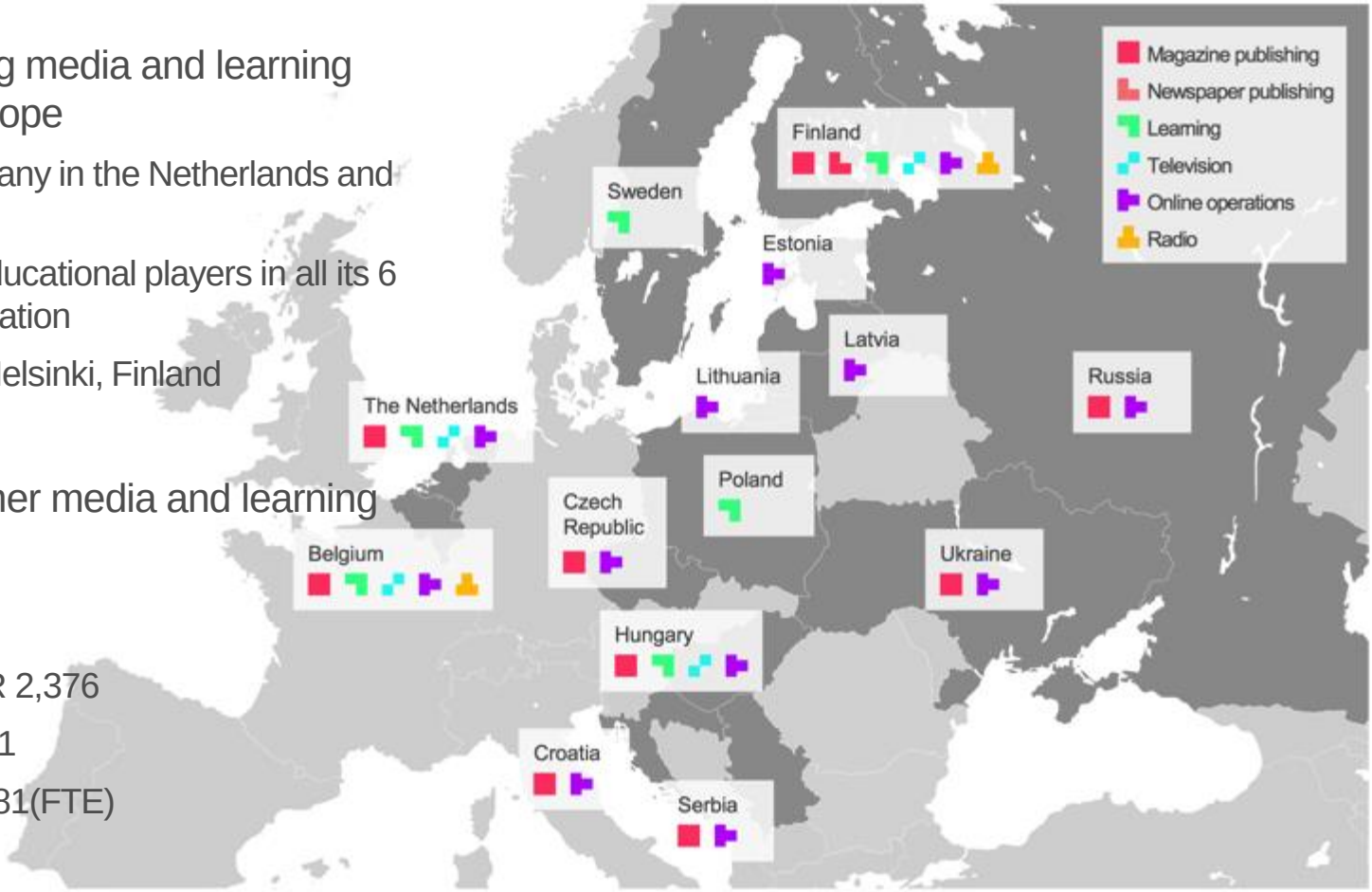







Sanoma: Market leader in chosen businesses and markets

- One of the leading media and learning companies in Europe
  - #1 media company in the Netherlands and Finland
  - Among top 2 educational players in all its 6 markets of operation
  - Head office in Helsinki, Finland
- Focus on consumer media and learning
- 2012 financials
  - Netsales mEUR 2,376
  - EBIT mEUR 231
  - Personnel 10,381(FTE)



sanoma get the world

Over 300 magazines


Goeie
Vraag!


overzicht.nl


ilse


Olcsóbbat.hu
Spórolni tudni kell!


STYLE
TODAY


startlap


Startpagina


KIESKEURIG.NL


HÍR 24


ruutu


.NL


ZIE.NL


nu
nl


SchoolBANK


Belegger.nl


GAMER.NL
ALTIJD EEN EIGEN MENING


HUUTO

Past

History

< 2008 | 2009 | 2010 | 2011 | 2012 | 2013

Self service

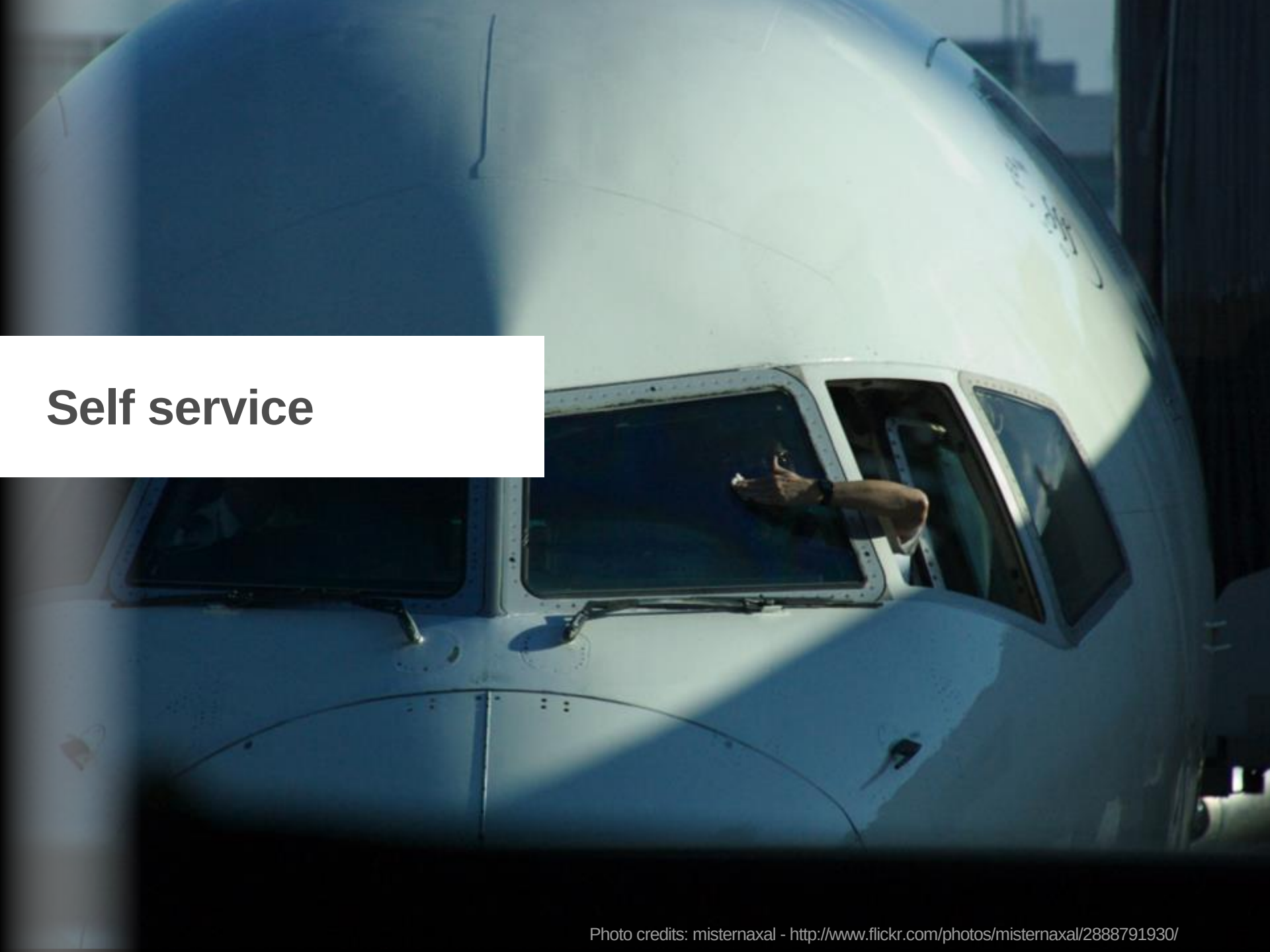


Photo credits: misternaxal - http://www.flickr.com/photos/misternaxal/2888791930/

Self service levels

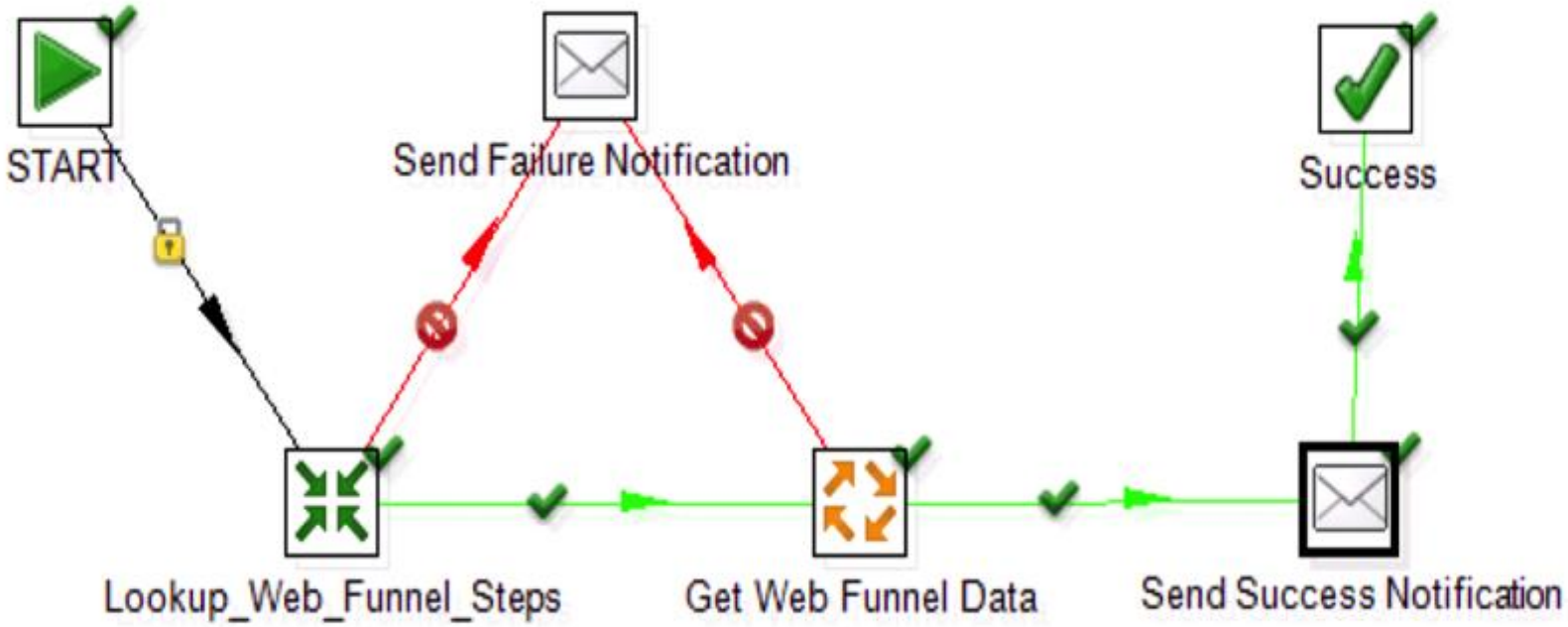
Personal	Departmental	Corporate
Full self service	**Support with publishing dashboards and data loading**	**Full service and support on dashboard**
Information is created by end users with little or no oversight. Users are empowered to integrate different data sources and make their own calculations.	Information has been created by end users and is worth sharing, but has not been validated.	Information that has gone through a rigorous validation process can be disseminated as official data.

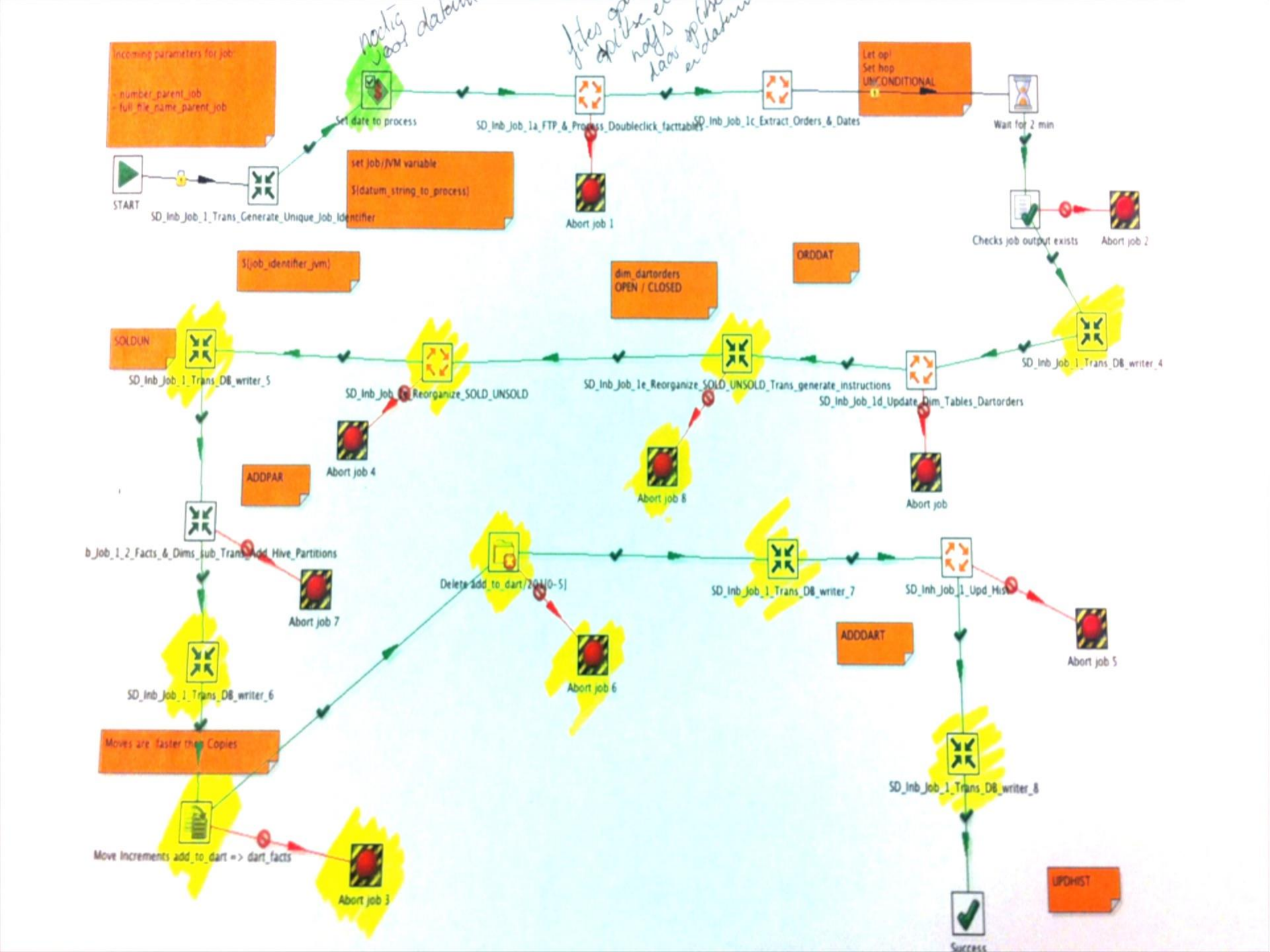
History

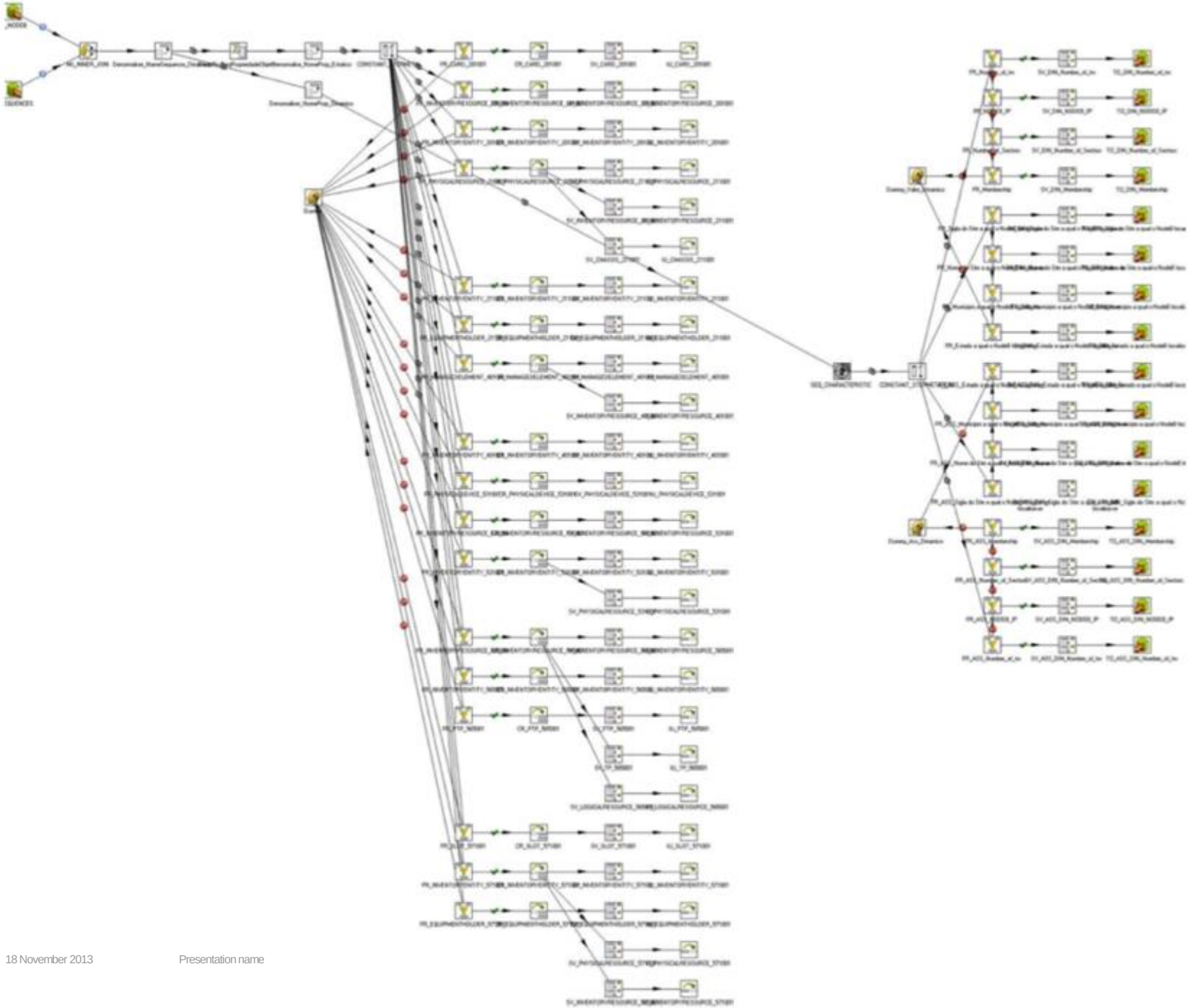
< 2008 | 2009 | 2010 | 2011 | 2012 | 2013

Glue: ETL

- EXTRACT
- TRANSFORM
- LOAD


START
Send Failure Notification
Success
Lookup_Web_Funnel_Steps
Get Web Funnel Data
Send Success Notification


Incoming parameters for job:
- number_parent_job
- full_file_name_parent_job
START
SD_Inb_Job_1_Trans_Generate_Unique_Job_Identifier
Set date to process
set job/JVM variable
${datum_string_to_process}
SD_Inb_Job_1a_FTP_&_Process_Doubleclick_facttables
Abort job 1
SD_Inb_Job_1c_Extract_Orders_&_Dates
Let op!
Set hop
UNCONDITIONAL
Wait for 2 min
Checks job output exists
Abort job 2
${job_identifier_jvm}
dim_dartorders
OPEN / CLOSED
ORDDAT
SOLDUN
SD_Inb_Job_1_Trans_DB_writer_5
SD_Inb_Job_1b_Reorganize_SOLD_UNSOLD
Abort job 4
SD_Inb_Job_1e_Reorganize_SOLD_UNSOLD_Trans_generate_instructions
Abort job 8
SD_Inb_Job_1d_Update_Dim_Tables_Dartorders
Abort job
SD_Inb_Job_1_Trans_DB_writer_4
ADDPAR
b_Job_1_2_Facts_&_Dims_sub_Trans_Add_Hive_Partitions
Abort job 7
Delete add_to_dart/20[0-5]
Abort job 6
SD_Inb_Job_1_Trans_DB_writer_7
SD_Inh_Job_1_Upd_Hist
Abort job 5
ADDDART
SD_Inb_Job_1_Trans_DB_writer_6
Moves are faster than Copies
Move Increments add_to_dart => dart_facts
Abort job 3
SD_Inb_Job_1_Trans_DB_writer_8
UPDHIST
Success



Shell


#fail

Learnings

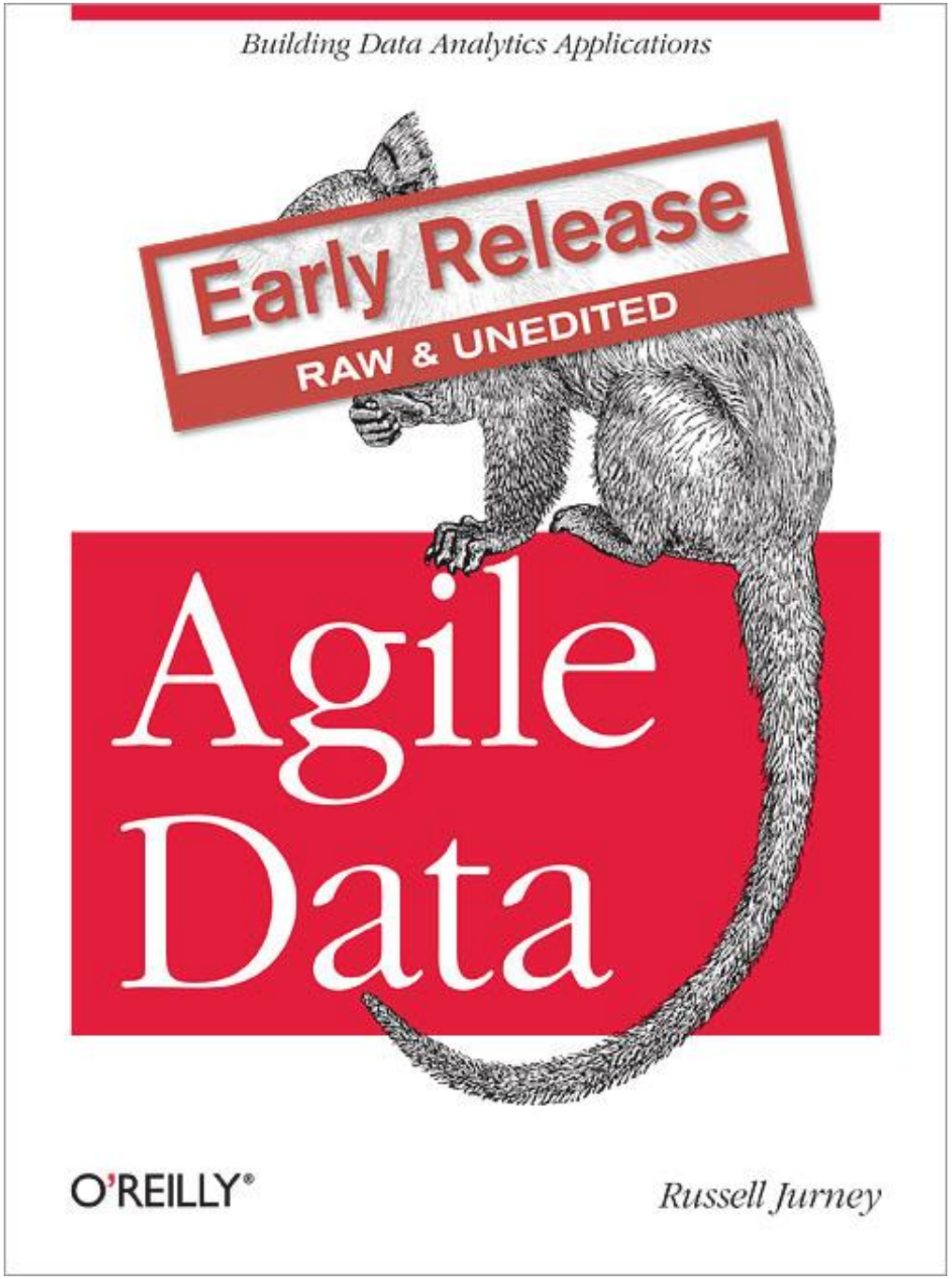
- Traditional ETL tools don’t scale and are not effective for Big Data sources
- Big Data projects are not BI projects
- Doing full end-to-end integrations and dashboard development doesn’t scale
- Qlikview was not good enough as the front-end to the cluster
- Hadoop requires developers not BI consultants

History

< 2008 | 2009 | 2010 | 2011 | 2012 | 2013


RESET

Russell Jurney
Agile Data



New glue

Photo credits: Sheng Hunglin - http://www.flickr.com/photos/shenghunglin/304959443/

ETL Tool features

- Processing
- Scheduling
- Data quality
- Data lineage
- Versioning
- Annotating

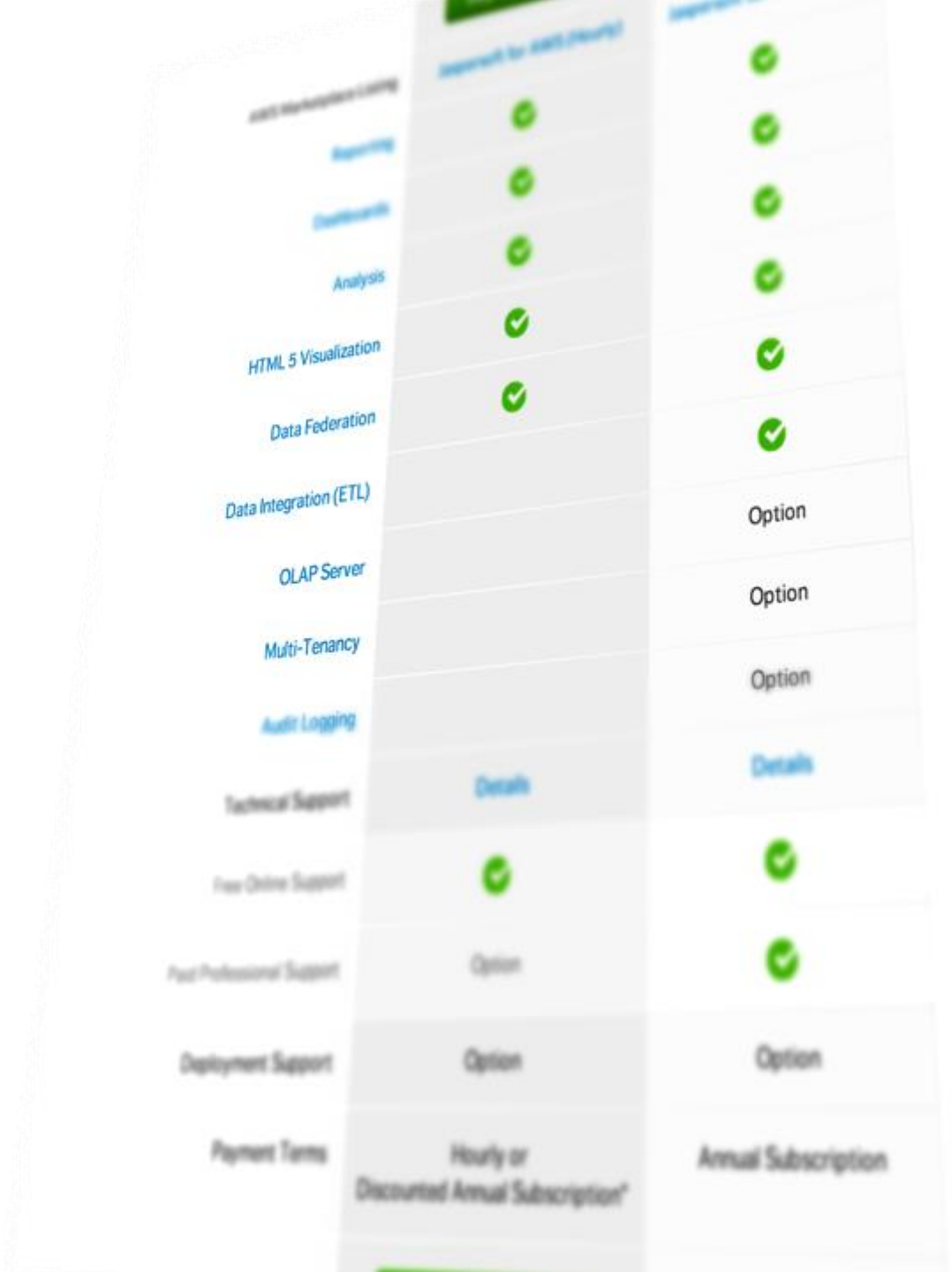


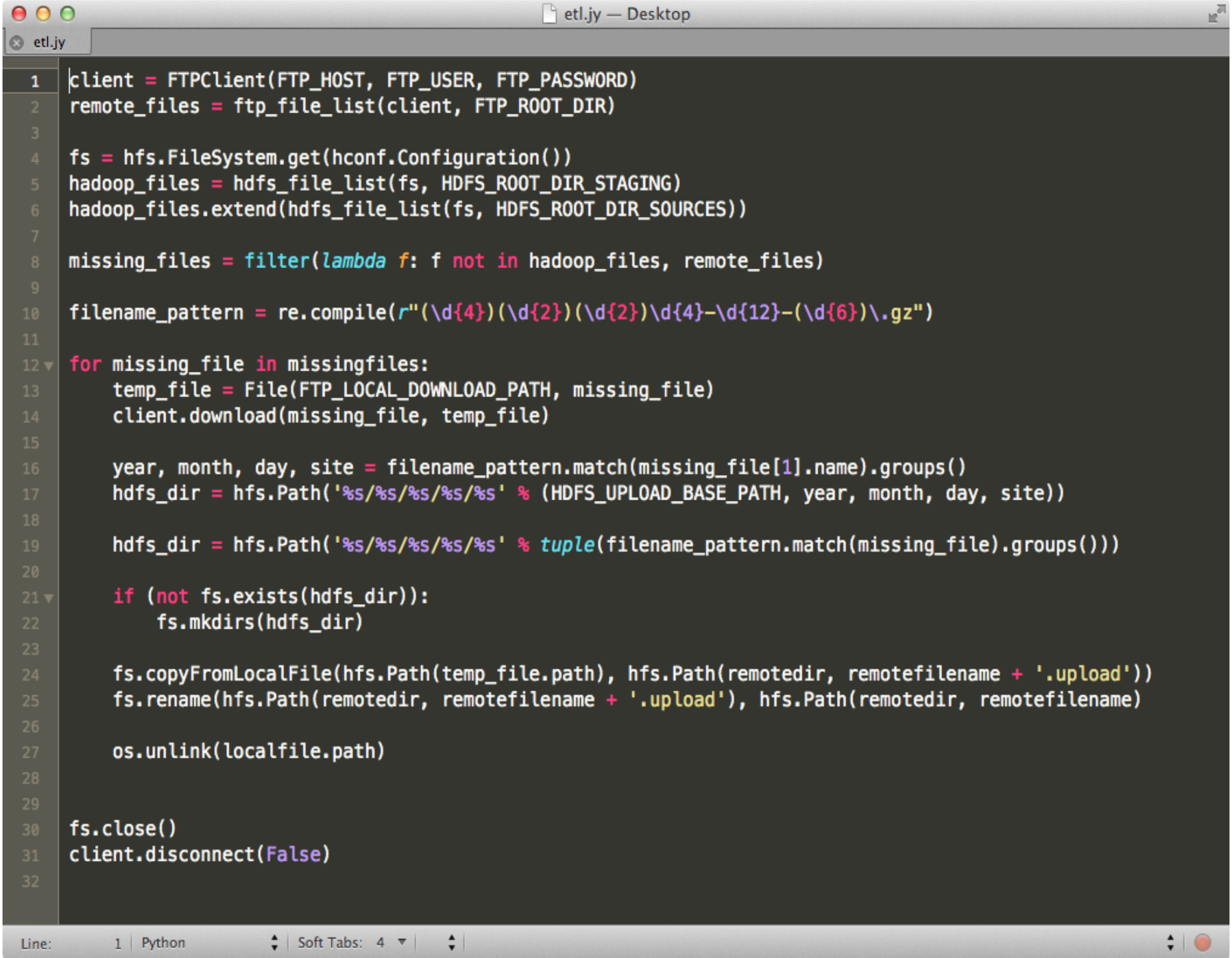
Photo credits: atomicbartbeans - http://www.flickr.com/photos/atomicbartbeans/71575328/

etl.sh — Desktop

etl.sh

```
#!/bin/bash
set -ex

echo \
'open ftp.example.com
user username password
cd my-files
find /' > /tmp/ftp-list-command.txt

lftp -f /tmp/ftp-list-command.txt | grep "\.gz" > /tmp/full-names-on-ftp.txt

hadoop fs -ls -R /staging/my-files/ > /tmp/full-names-on-hadoop.txt
hadoop fs -ls -R /sources/my-files/ >> /tmp/full-names-on-hadoop.txt

cat /tmp/full-names-on-hadoop.txt | grep "\.gz" | awk '{print $8}' | awk -F '/' '{print $NF}' > /tmp/basenames-on-hadoop.txt

cat /tmp/full-names-on-ftp.txt | fgrep -v -f /tmp/basenames-on-hadoop.txt > /tmp/missing-files-on-hadoop.txt

echo \
'open ftp.example.com
user username password
' > /tmp/ftp-get-files-command.txt
awk '{print "get -c " $0}' /tmp/missing-files-on-hadoop.txt >> /tmp/ftp-get-files-command.txt


cd /local-data/staging/my-files/
lftp -vf /tmp/ftp-get-files-command.txt
cd -

while read f
do
  filename=$(echo $f | awk -F '/' '{print $NF}')
  year=${filename:0:4}
  month=${filename:4:2}
  day=${filename:6:2}
  site=$(echo $f | awk -F '-' '{print $NF}' | sed s/\.gz//g)

  hadoop fs -mkdir "/staging/my-files/$year/$month/$day/$site"
  hadoop fs -put "/local-data/staging/my-files/$filename" "/staging/my-files/$year/$month/$day/$site/$filename.upload"
  hadoop fs -mv "/staging/my-files/$year/$month/$day/$site/$filename.upload" "/staging/my-files/$year/$month/$day/$site/$filename"
done < /tmp/missing-files-on-hadoop.txt
```

Line: 1 | Shell Script (Bash) | Tab Size: 4


etl.jy — Desktop
etl.jy
client = FTPClient(FTP_HOST, FTP_USER, FTP_PASSWORD)
remote_files = ftp_file_list(client, FTP_ROOT_DIR)
fs = hfs.FileSystem.get(hconf.Configuration())
hadoop_files = hdfs_file_list(fs, HDFS_ROOT_DIR_STAGING)
hadoop_files.extend(hdfs_file_list(fs, HDFS_ROOT_DIR_SOURCES))
missing_files = filter(lambda f: f not in hadoop_files, remote_files)
filename_pattern = re.compile(r"(\d{4})(\d{2})(\d{2})\d{4}-\d{12}-(\d{6})\.gz")
for missing_file in missingfiles:
temp_file = File(FTP_LOCAL_DOWNLOAD_PATH, missing_file)
client.download(missing_file, temp_file)
year, month, day, site = filename_pattern.match(missing_file[1].name).groups()
hdfs_dir = hfs.Path('%s/%s/%s/%s/%s' % (HDFS_UPLOAD_BASE_PATH, year, month, day, site))
hdfs_dir = hfs.Path('%s/%s/%s/%s/%s' % tuple(filename_pattern.match(missing_file).groups()))
if (not fs.exists(hdfs_dir)):
fs.mkdirs(hdfs_dir)
fs.copyFromLocalFile(hfs.Path(temp_file.path), hfs.Path(remotedir, remotefilename + '.upload'))
fs.rename(hfs.Path(remotedir, remotefilename + '.upload'), hfs.Path(remotedir, remotefilename)
os.unlink(localfile.path)
fs.close()
client.disconnect(False)
Line: 1
Python
Soft Tabs: 4

Processing - Jython

- No JVM startup overhead for Hadoop API usage
- Relatively concise syntax (Python)
- Mix Python standard library with any Java libs

Jython

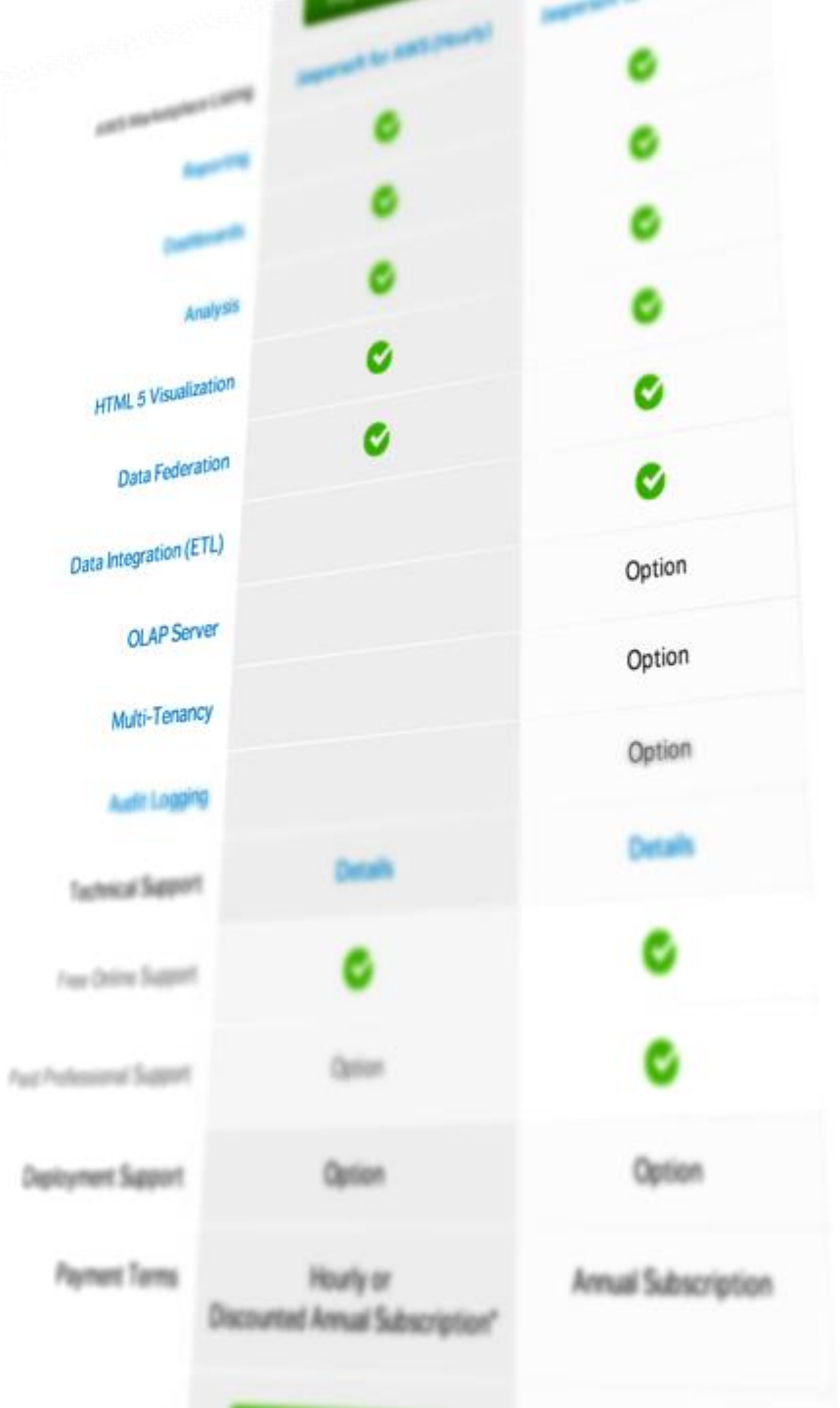
Scheduling - Jenkins

- Flexible scheduling with dependencies
- Saves output
- E-mails on errors
- Scales to multiple nodes
- REST API
- Status monitor
- Integrates with version control

Jenkins

ETL Tool features

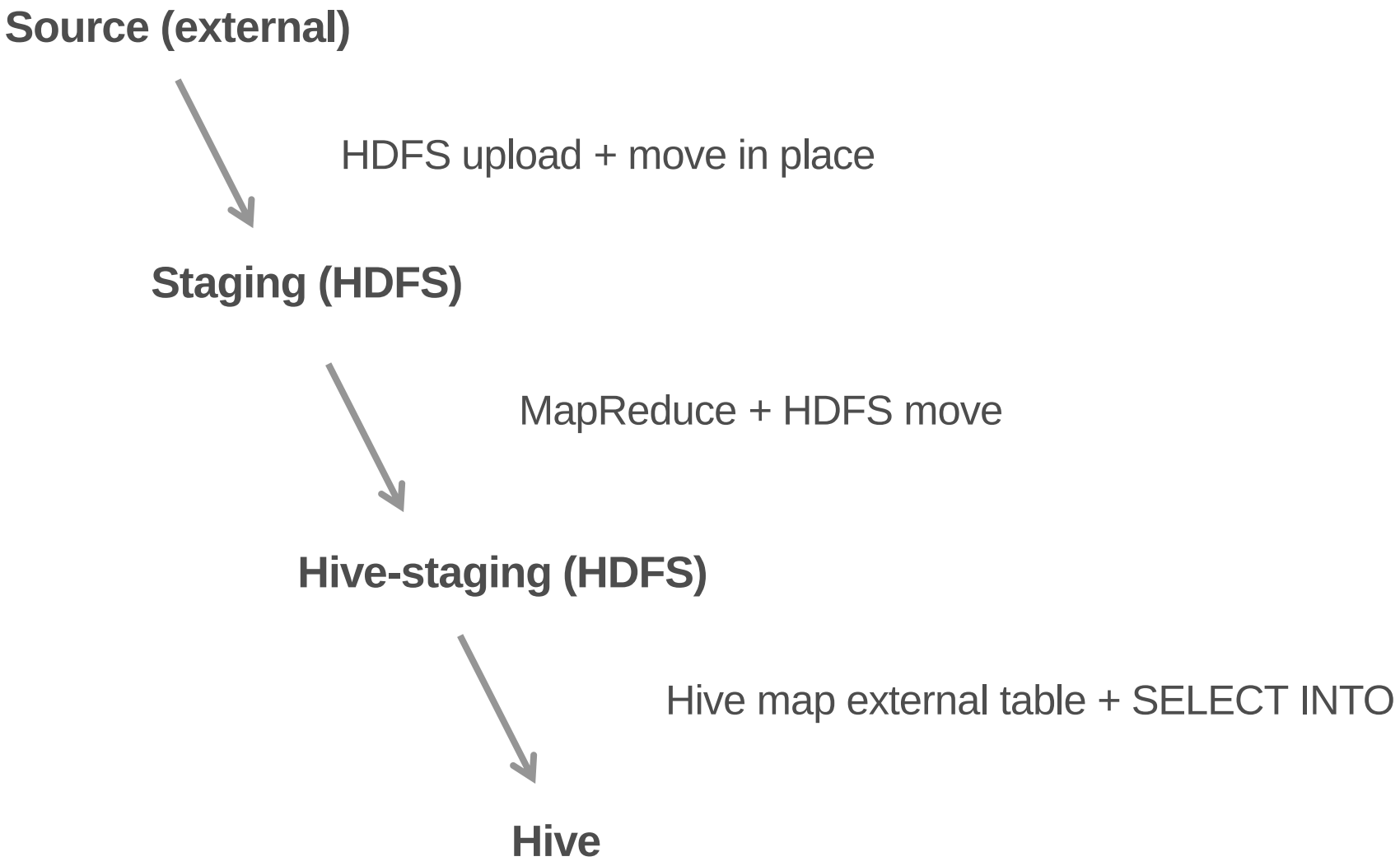
- Processing – **Bash & Jython**
- Scheduling – **Jenkins**
- Data quality
- Data lineage
- Versioning – **Mecurial (hg)**
- Annotating – **Commenting the code ☺**



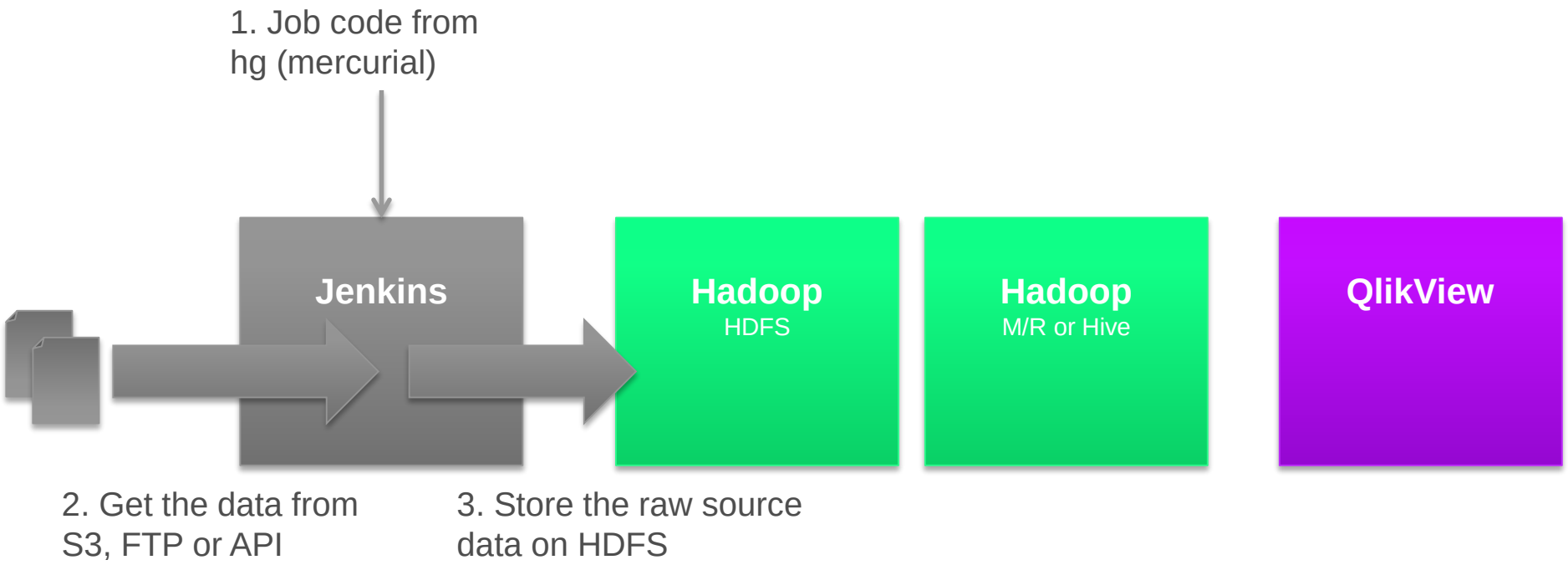
Processes

Photo credits: Paul McGreevy - http://www.flickr.com/photos/48379763@N03/6271909867/

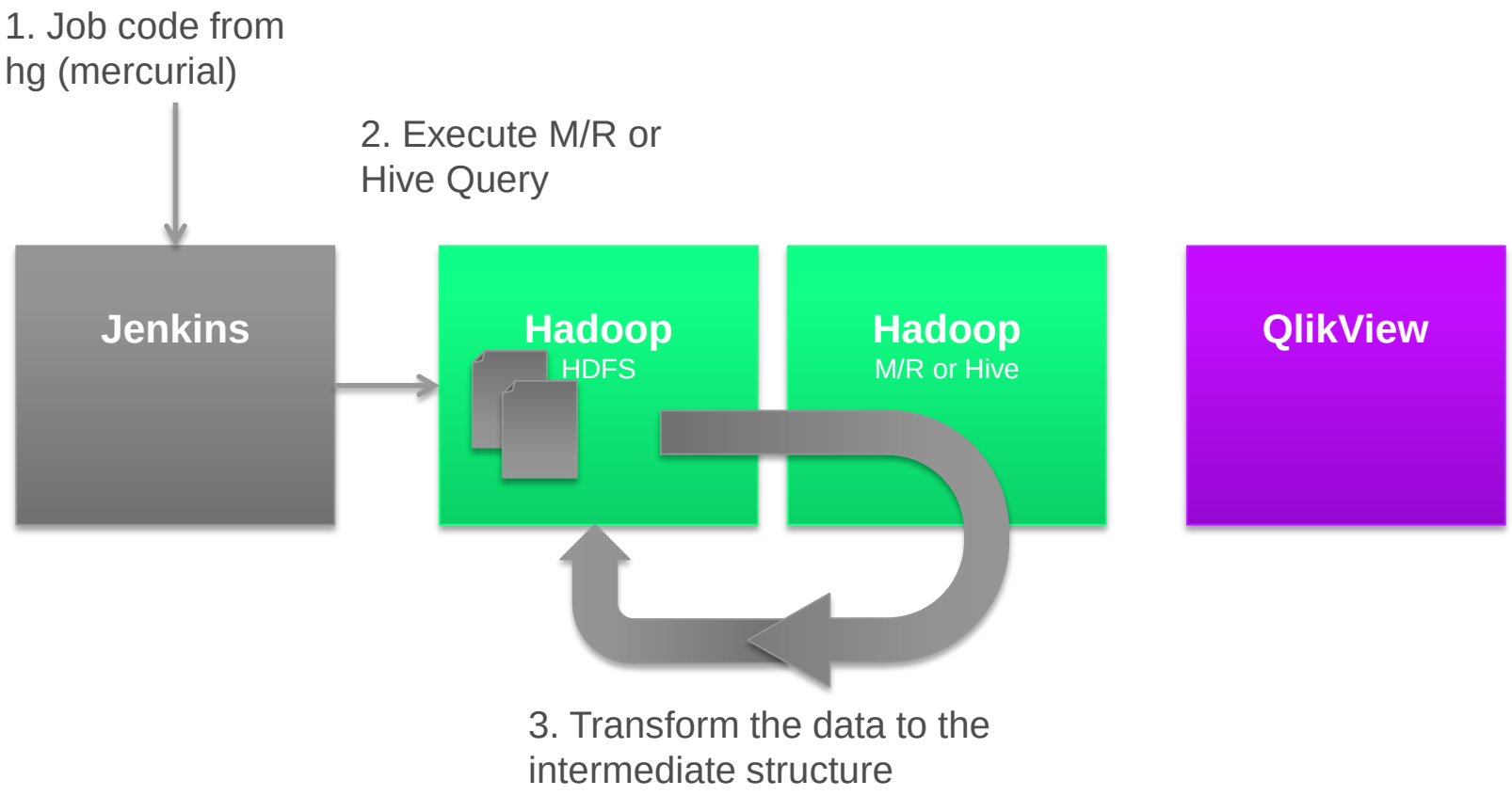
Independent jobs



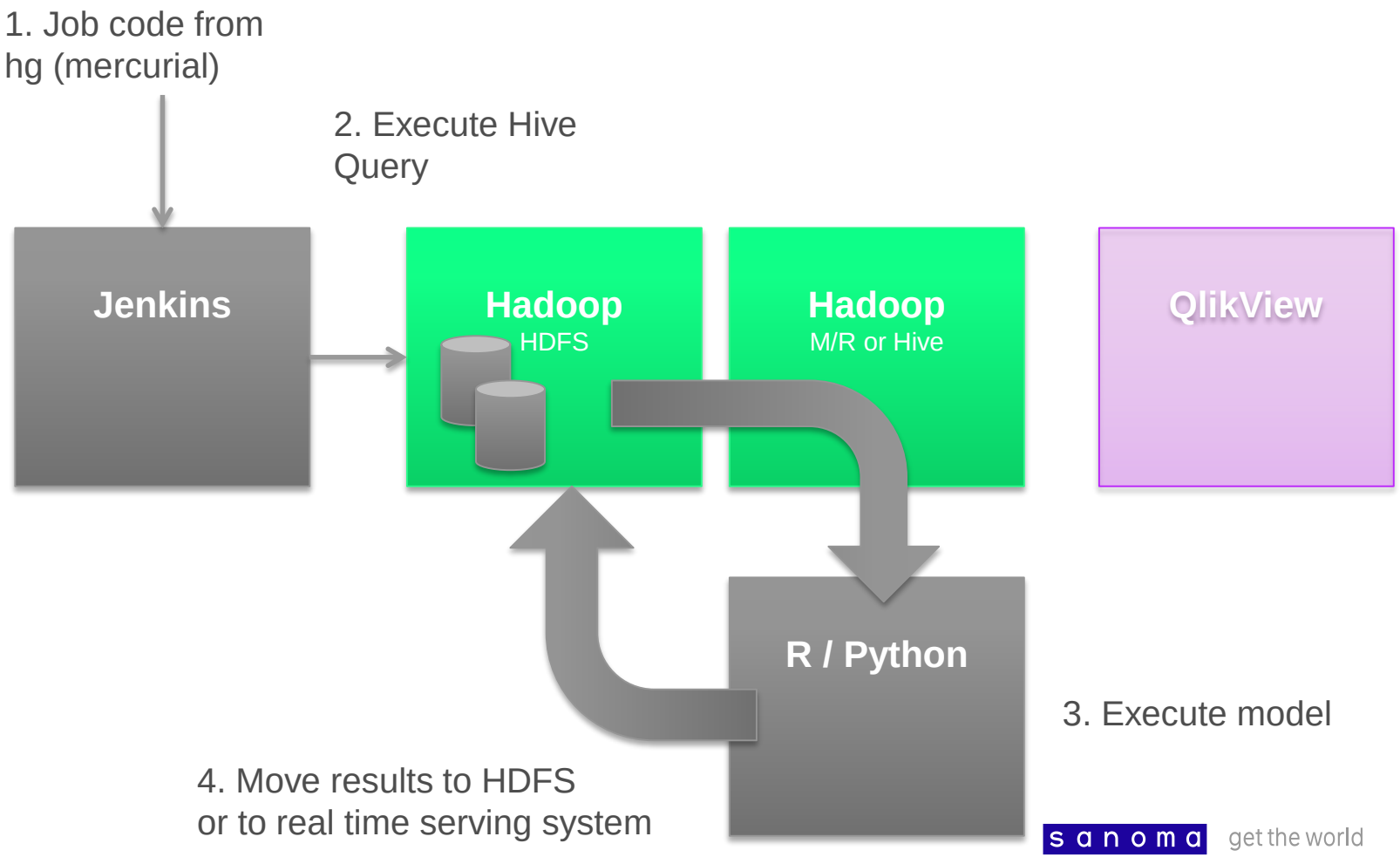
Typical data flow - Extract



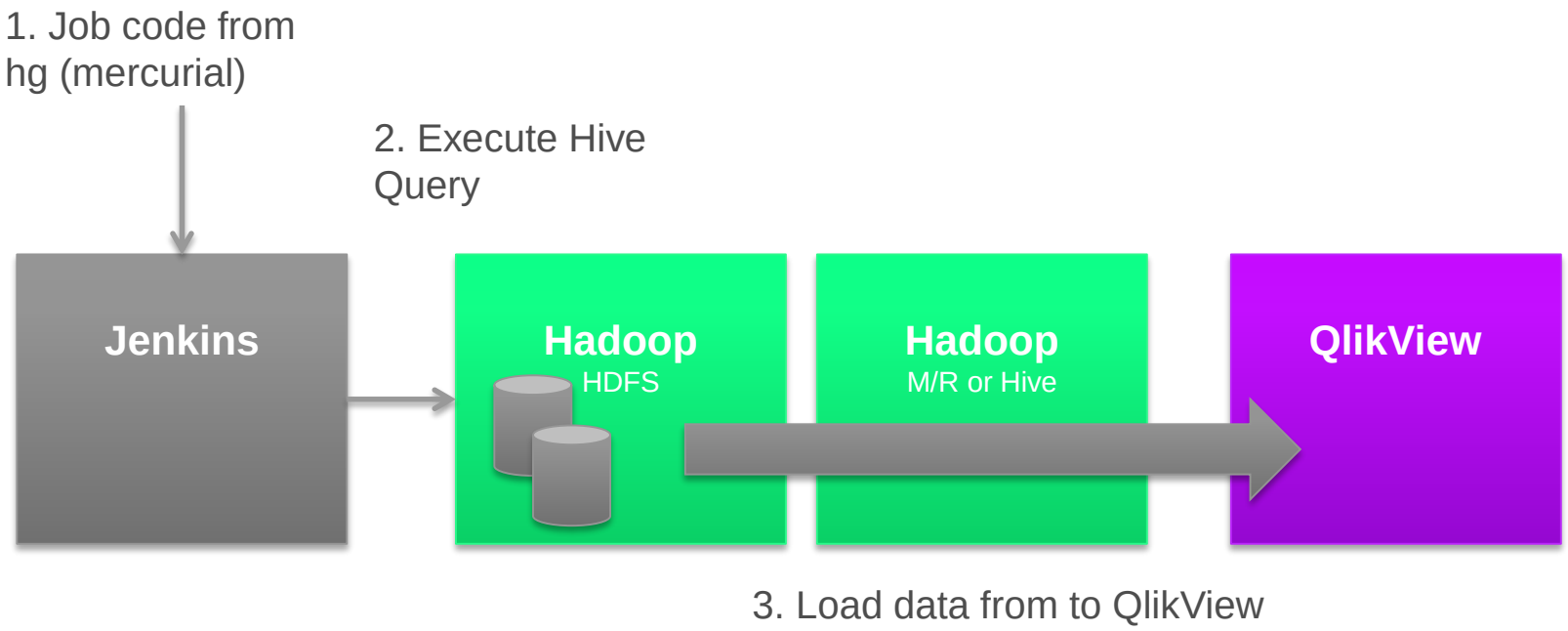
Typical data flow - Transform



Typical data flow - Load



Typical data flow - Load



Out of order jobs

- At any point, you don’t really know what ‘made it’ to Hive
- Will happen anyway, because some days the data delivery is going to be three hours late
- Or you get half in the morning and the other half later in the day
- It really depends on what you do with the data
- This is where metrics + fixable data store help...

Fixable data store

- Using Hive partitions
- Jobs that move data from staging create partitions
- When new data / insight about the data arrives, drop the partition and re-insert
- Be careful to reset any metrics in this case
- Basically: instead of trying to make everything transactional, repair afterwards
- Use metrics to determine whether data is fit for purpose

Photo credits: DL76 - http://www.flickr.com/photos/dl76/3643359247/

My Queries | Saved Queries | History | Settings

Query Editor

```
1 SELECT to_date(creation_tstamp) dt
2   ,COUNT(*) clicks
3 FROM kis_clicks
4 GROUP BY to_date(creation_tstamp)
```

HUE

IZATION
Parameterization
FICATION
e on completion

Execute | Execute Into QV | Save as... | Explain | or create a | New query

Hadoop job scheduling

- Schedulers to spread the load on your cluster
- CapacityScheduler:

  vs

- FairScheduler:
  - The default since: CDH 4.1

- Use scheduling pools to seperate workloads. ETL vs User based vs Consuming Applications

Quotas

- Since user can break stuff, easily..

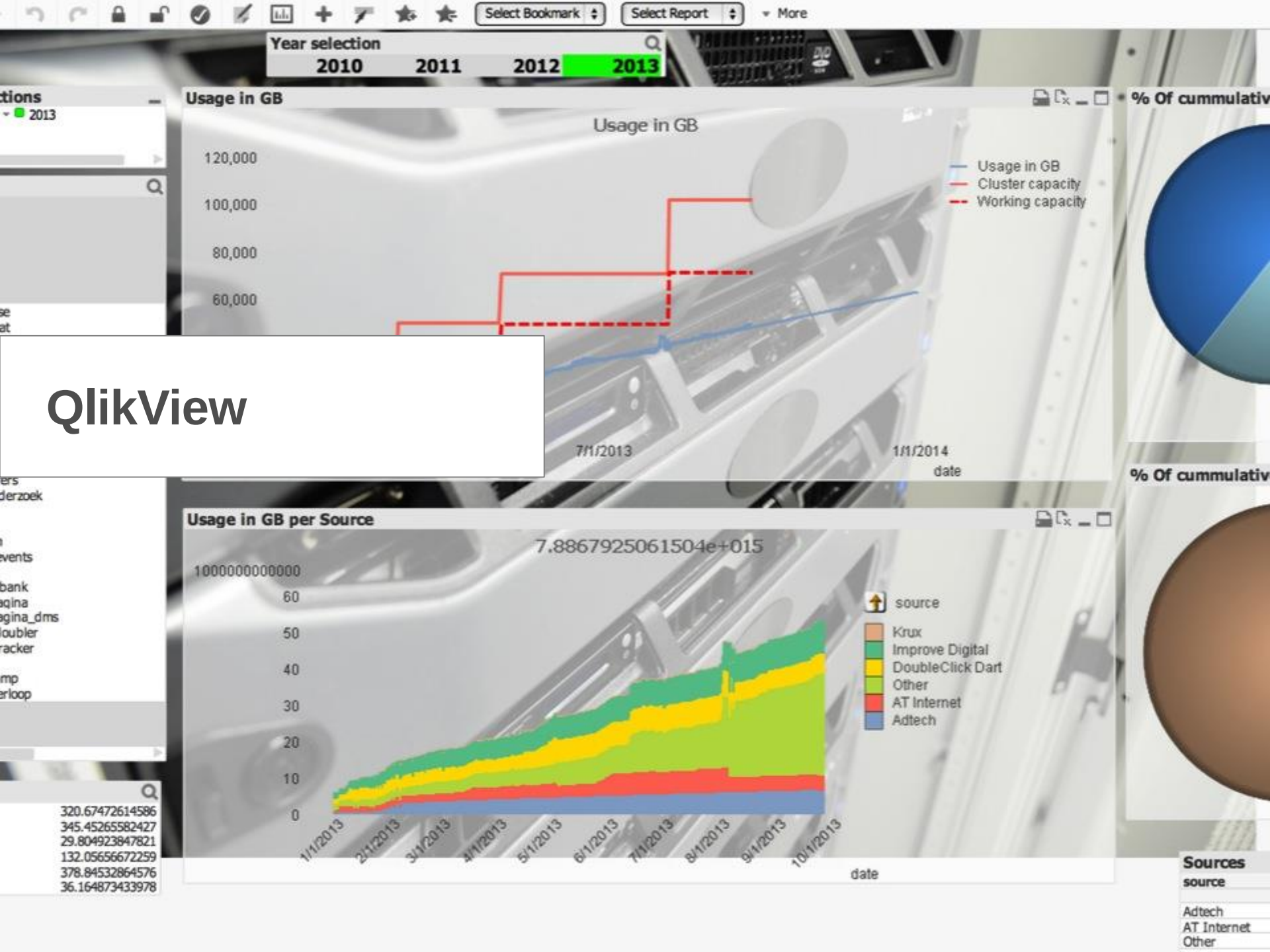
..and SQL skills get rusty

```
SELECT *
FROM views
  LEFT JOIN clicks
WHERE day = '2013-09-26’
```

Views table is 10TB and $36x10^9$ rows

Clicks table is 100GB and $36x10^6$ rows

Or use Strict mode = 1 in hive


Select Bookmark
Select Report
More
Year selection
2010
2011
2012
2013
Usage in GB
Usage in GB
120,000
100,000
80,000
60,000
Usage in GB
Cluster capacity
Working capacity
7/1/2013
1/1/2014
date
% Of cummulativ
% Of cummulativ
QlikView
Usage in GB per Source
7.8867925061504e+015
1000000000000
60
50
40
30
20
10
0
source
Krux
Improve Digital
DoubleClick Dart
Other
AT Internet
Adtech
1/1/2013
2/1/2013
3/1/2013
4/1/2013
5/1/2013
6/1/2013
7/1/2013
8/1/2013
9/1/2013
10/1/2013
date
320.67472614586
345.45265582427
29.804923847821
132.05656672259
378.84532864576
36.164873433978
Sources
source
Adtech
AT Internet
Other

R Studio Server

lit Code View Plots Session Project Build Tools Help

Go to file/function

rce on Save | Run | Source

```
JDBC")

C(driverClass = "org.apache.hive.jdbc.HiveDriver",
  classPath = paste("/opt/cloudera/parcels/CDH/lib/hive/lib/hive-
                    "/opt/cloudera/parcels/CDH/lib/hive/lib/hive-
                    "/opt/cloudera/parcels/CDH/lib/hive/lib/libth
                    "/opt/cloudera/parcels/CDH/lib/hive/lib/hive-
                    "/opt/cloudera/parcels/CDH/lib/hive/lib/libfb
                                                    ib/slf4j
                                                    ib/commo
```

Script

e/skieft/R Test/

```
(range(x))

(range(y))

1]/xdelta

2]/ydelta

scale, yscale)

in[1]/scale - xdelta)

in[2]/scale - ydelta)

), numeric(0),
nge(x)+c(-1,1)*xadd, ylim = range(y)+c(-1,1)*yadd,
", ann = FALSE)
see next plot:
```

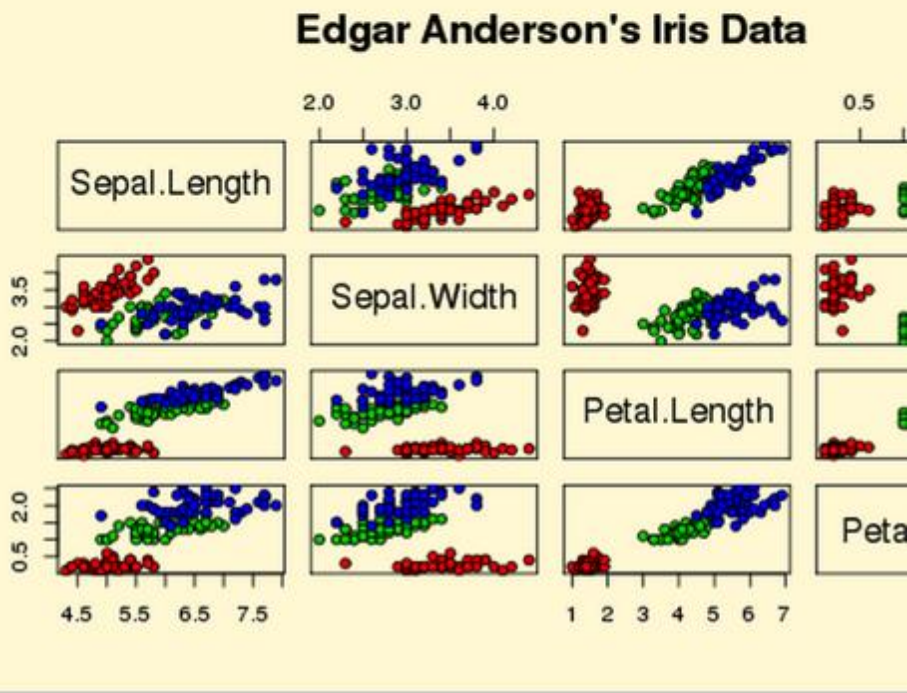
Workspace | History

Import Dataset

Values	
conn	JDBCConnection[1]
drv	JDBCDriver[1]
g	factor[1000]
lev	numeric[12]
n	100
opar	list[1]
pie.sales	numeric[6]

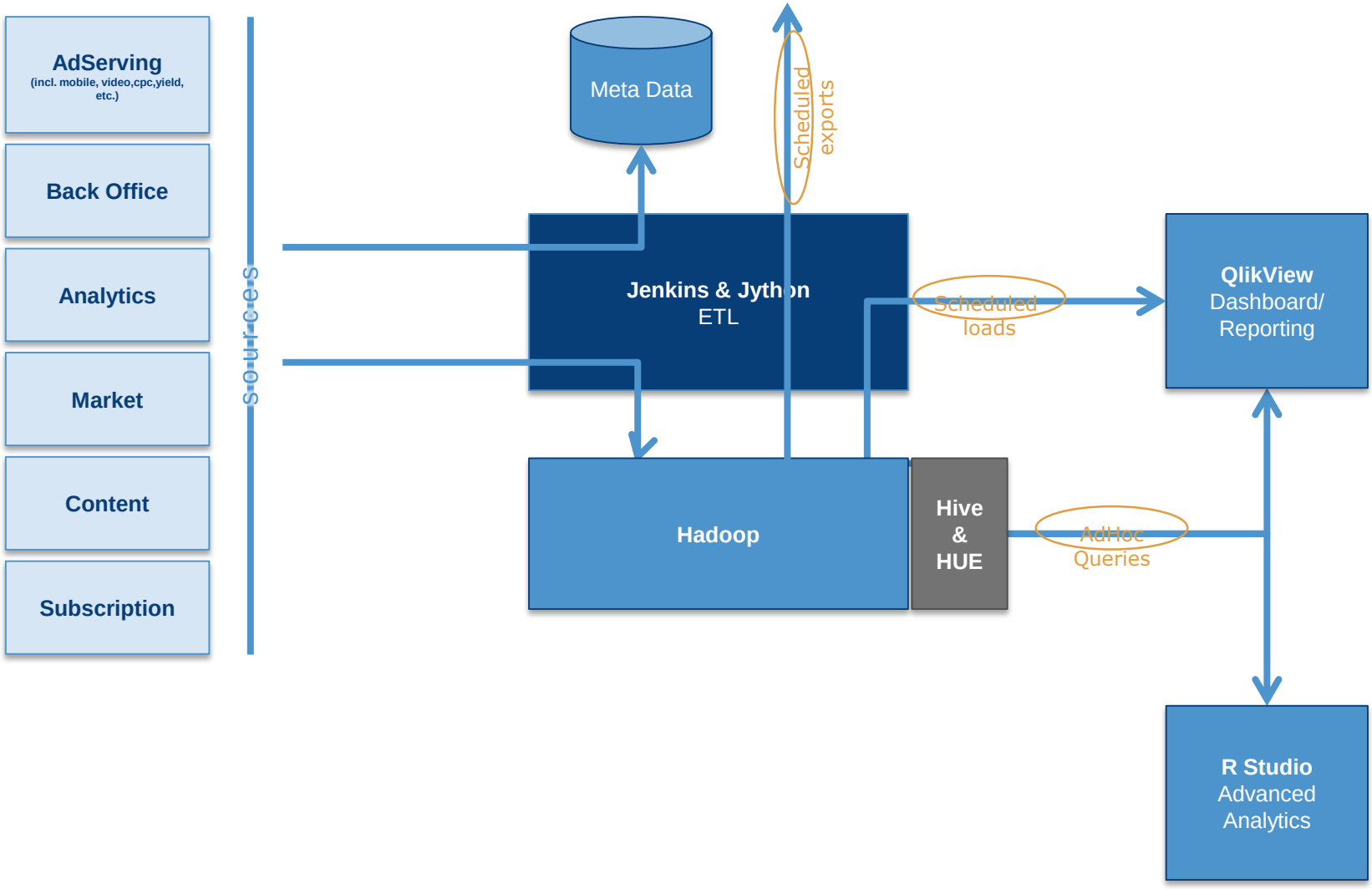
Files | Plots | Packages | Help

Zoom | Export | Clear All

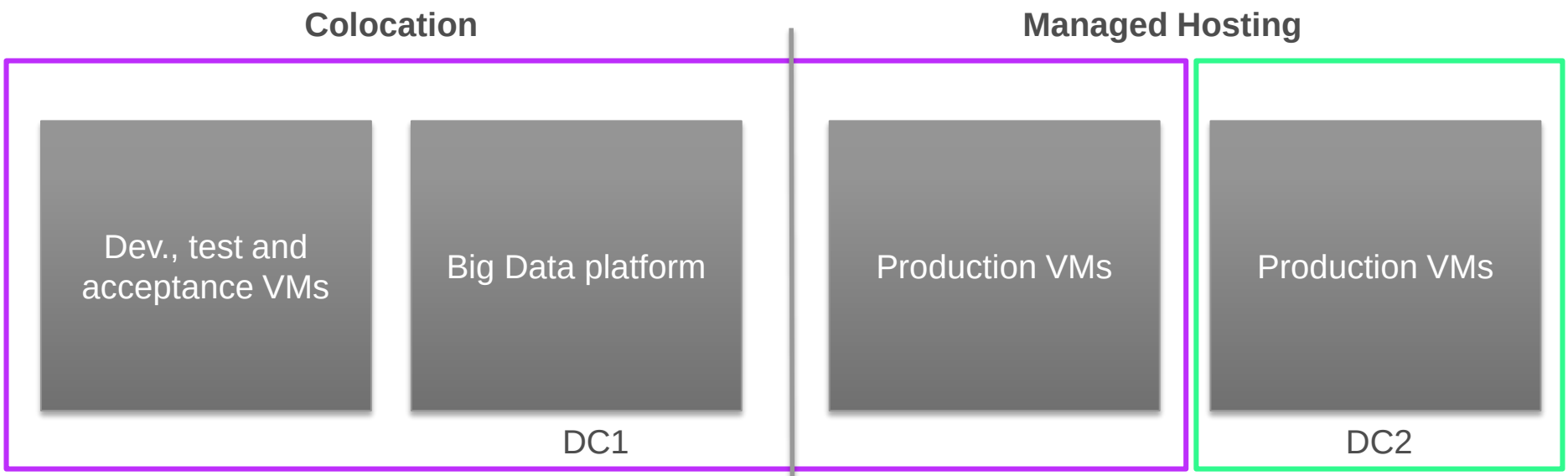


Architecture

High Level Architecture



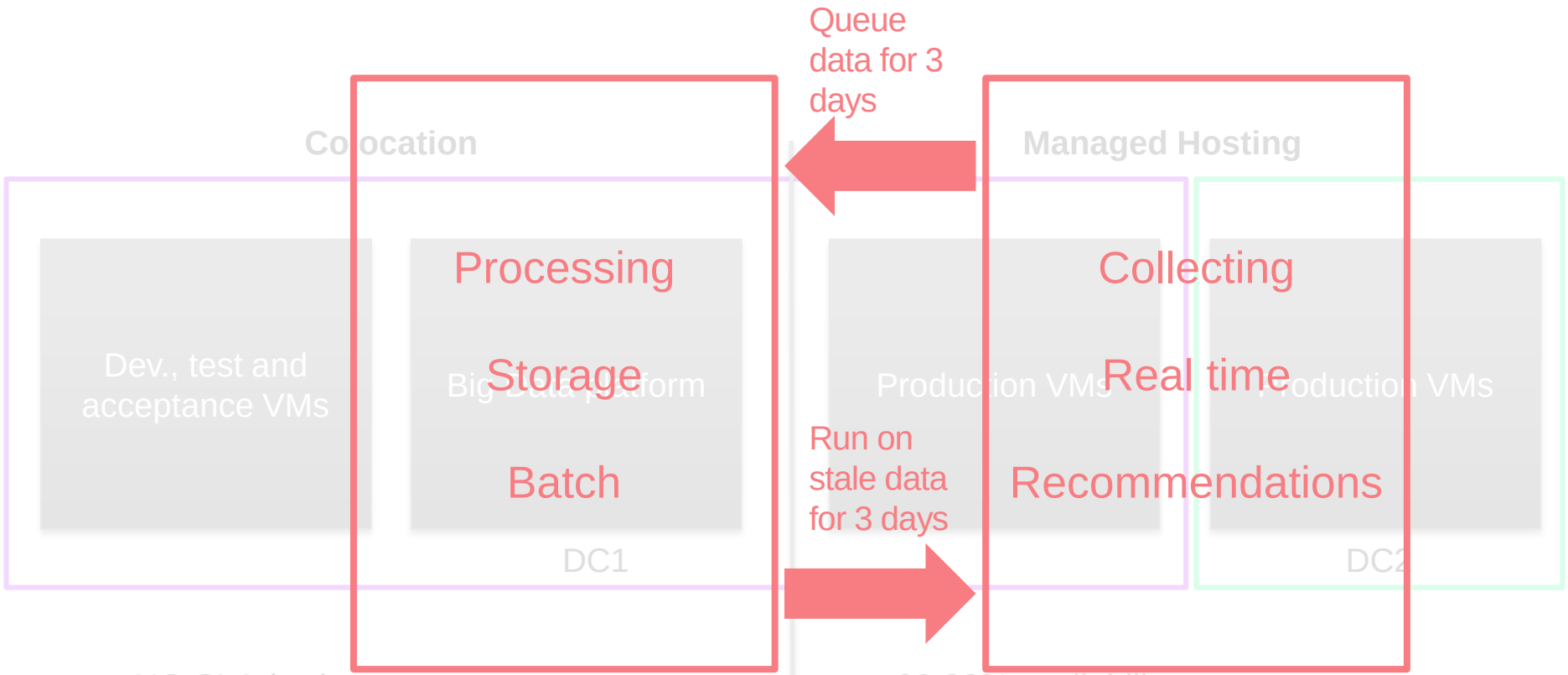
Sanoma Media The Netherlands Infra



- NO SLA (yet)
- Limited support BD9-5
- No Full system backups
- Managed by us with systems department help

- 99,98% availability
- 24/7 support
- Multi DC
- Full system backups
- High performance EMC SAN storage
- Managed by dedicated systems department

Sanoma Media The Netherlands Infra



- NO SLA (yet)
- Limited support BD9-5
- No Full system backups
- Managed by us with systems department help

- 99,98% availability
- 24/7 support
- Multi DC
- Full system backups
- High performance EMC SAN storage
- Managed by dedicated systems department

Present

Current state – Use case

- A/B testing + deeper analyses
- Ad auction price optimization
- Recommendations
- Search optimizations

Current state - Usage

- Main use case for reporting and analytics
- Sanoma standard data platform, used by other Sanoma countries too: Finland, Hungary, ..
- ~ 100 Users: analysts & developers
- 25 daily users
- 43 source systems, with 125 different sources
- 300 tables in hive

Current state – Tech and team

- Team:
  - 1 product manager
  - 2 developers
  - 2 data scientists
  - 1 Qlikview application manager
  - ½ architect

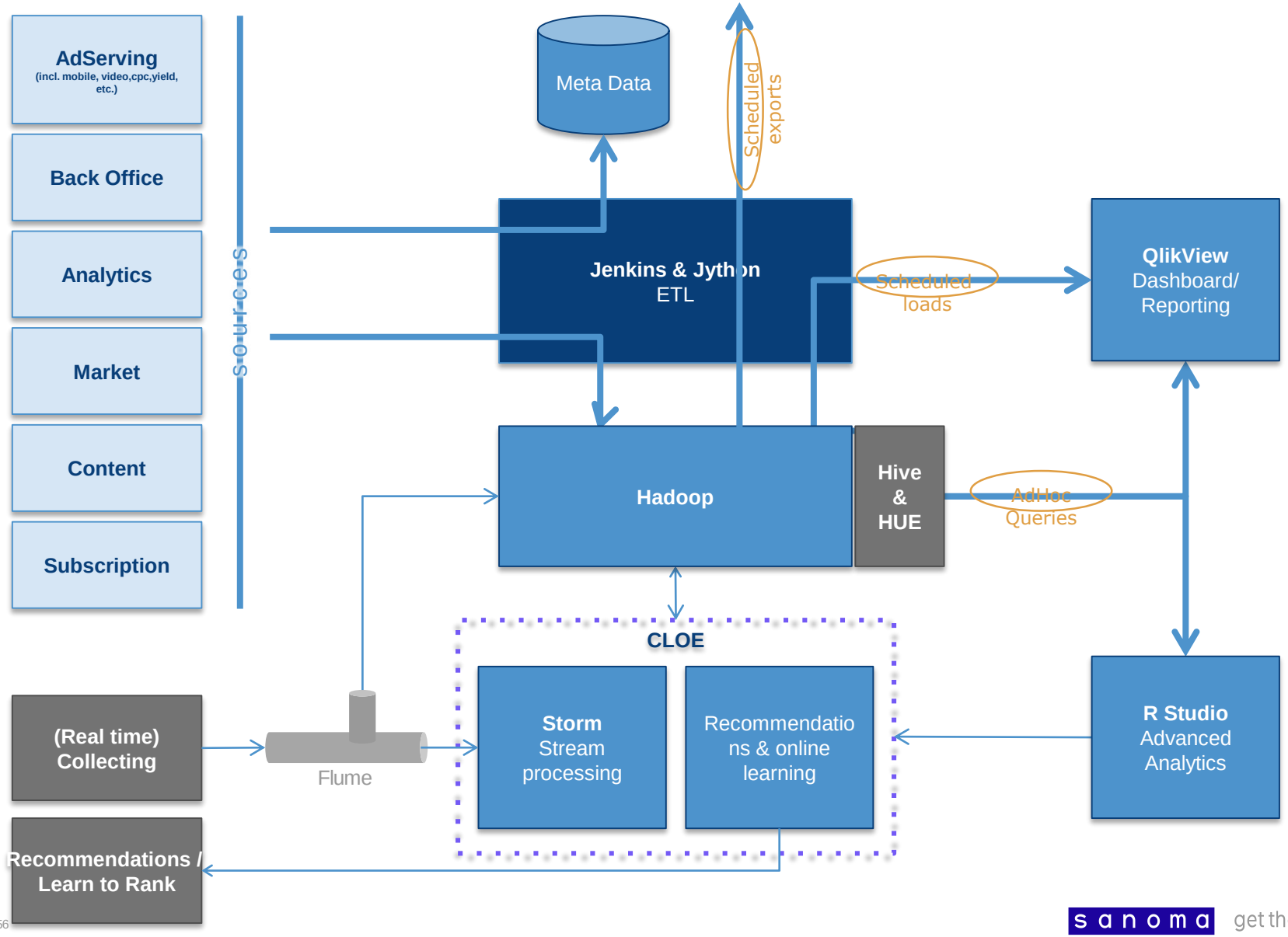
- Platform:
  - 30-50 nodes
  - > 300TB storage
  - ~2000 jobs/day

- Typical data node / task tracker:
  - 2 system disks (RAID 1)
  - 4 data disks (2TB, 3TB or 4TB)
  - 24GB RAM

Current State - Real time

- Extending own collecting infrastructure
- Using this to drive recommendations, user segementation and targetting in real time
- Moving from Flume to Kafka
- First production project with Storm

High Level Architecture



Future

What's next

- Cloudera search (Solr Cloud on Hadoop)
  - Index creation
  - External Ranker
  - Easier scaling and maintainance

- Moving some NLP (Natural Language Processing) and Image recognition workload to hadoop
- Optimizing Job scheduling (Fair Scheduling Pools)
- Automated query optimization tips for analysts
- Full roll out R integration, with rmr2

- More support for: cascading, scalding, pig, etc.


Thank you!
Questions?
sanoma
get the world

sanoma