Das Apache Hadoop Framework im industriellen Einsatz

Daniel Müller, Marvin Follmann | Business Analytics Day 2019

Über uns

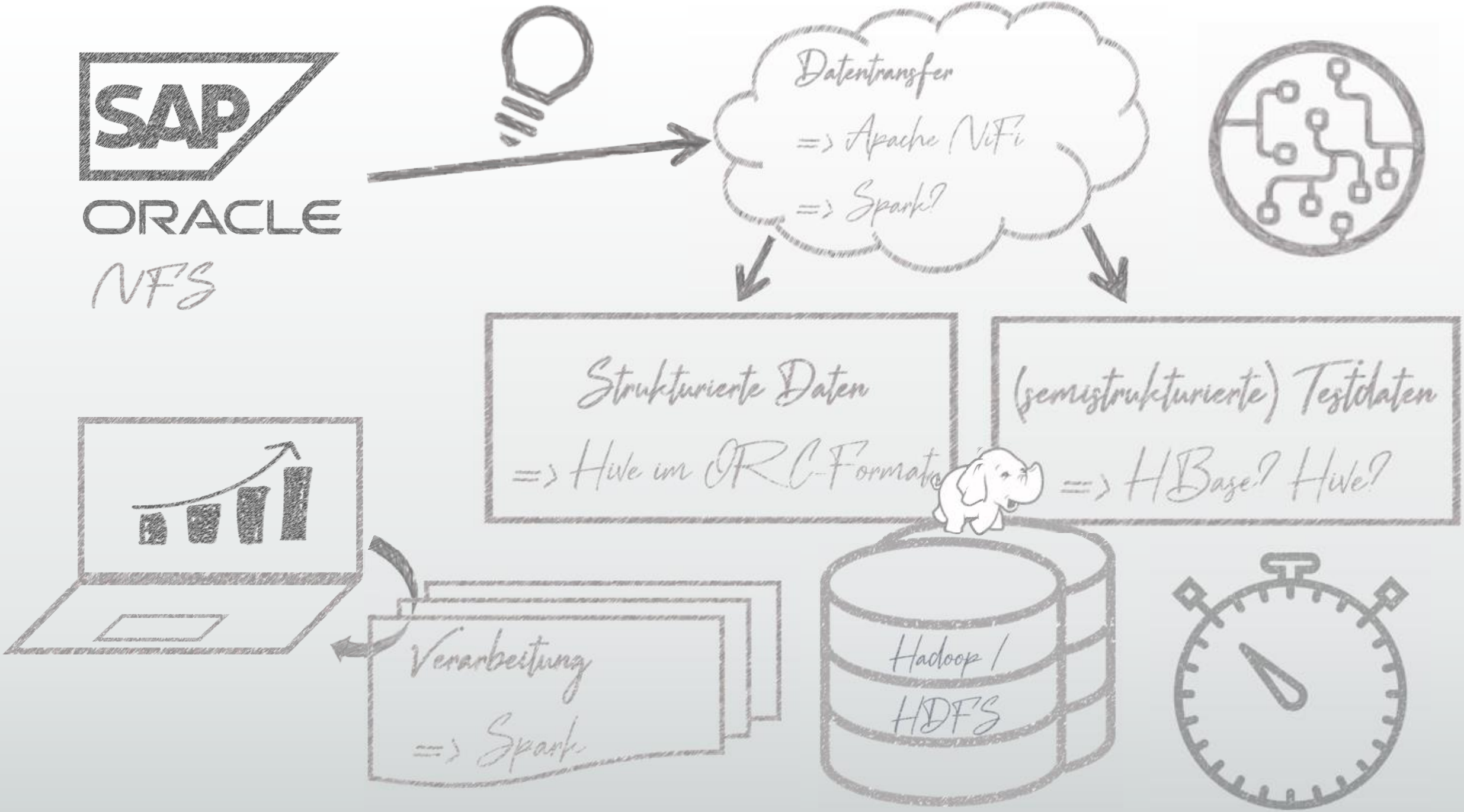
Seamless Analytics GmbH

- Gegründet 2018
  - Daniel Müller, M. Sc.
  - Marvin Follmann, M. Sc.

- Beratung und Dienstleistungen zu Datenspeicherung und -analyse
  - Projektplanung, Umsetzung, Schulungen
  - Apache Hadoop (HDP, Spark, Hive, HBase)
  - Maschinelles Lernen (TF, CNTK, Keras)

- Idee entstand durch Forschungsprojekt und wissenschaftliche Arbeiten
  - Hochschule Offenburg

https://seamless-analytics.de


SAP
ORACLE
NFS
Datentransfer
=> Apache NiFi
=> Spark?
Strukturierte Daten
=> Hive im ORC-Format
(semistrukturierte) Testdaten
=> HBase? Hive?
Verarbeitung
=> Spark
Hadoop /
HDFS

Über das Projekt

- **Hightech Fertigungsbetrieb aus Süddeutschland**
  - Halbleiter Hersteller
  - Sensorüberwachte Maschinen produzieren große Datenmengen
    - Daten: Maschinenmesswerte, Produktionsereignisse, Qualitätstests

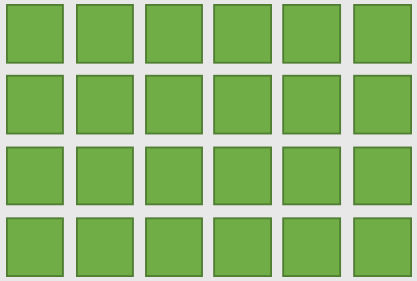
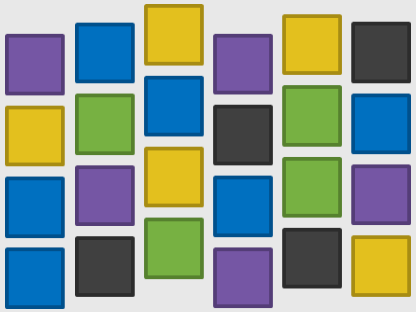
- **Problemstellung**
  - Daten wurden teilweise automatisiert, teilweise manuell für Analysen zusammengetragen
    - Hohe zeitliche Belastung der Mitarbeiter
    - Zeitnahe Auswertungen schwer möglich
  - System kam an Kapazitätsgrenzen

Über das Projekt

- **Projektziel**
  - Basiswissen zu Hadoop innerhalb der Firma aufbauen
  - Evaluation der Machbarkeit
    - Aufwand, Kostenpunkte (Hardware, Speicher, Lizenzen), Zukunftsfähigkeit
  - Aufbau einer Hadoop-basierten Plattform zur Erweiterung der bisherigen Analyselandschaft
    - Datenauslagerung aus SAP-System und Oracle-Datenbank
  - Schaffen neuer Analysemöglichkeiten
    - Reduzierung von Arbeitsaufwänden und Reaktionszeiten durch automatisierte Analysen

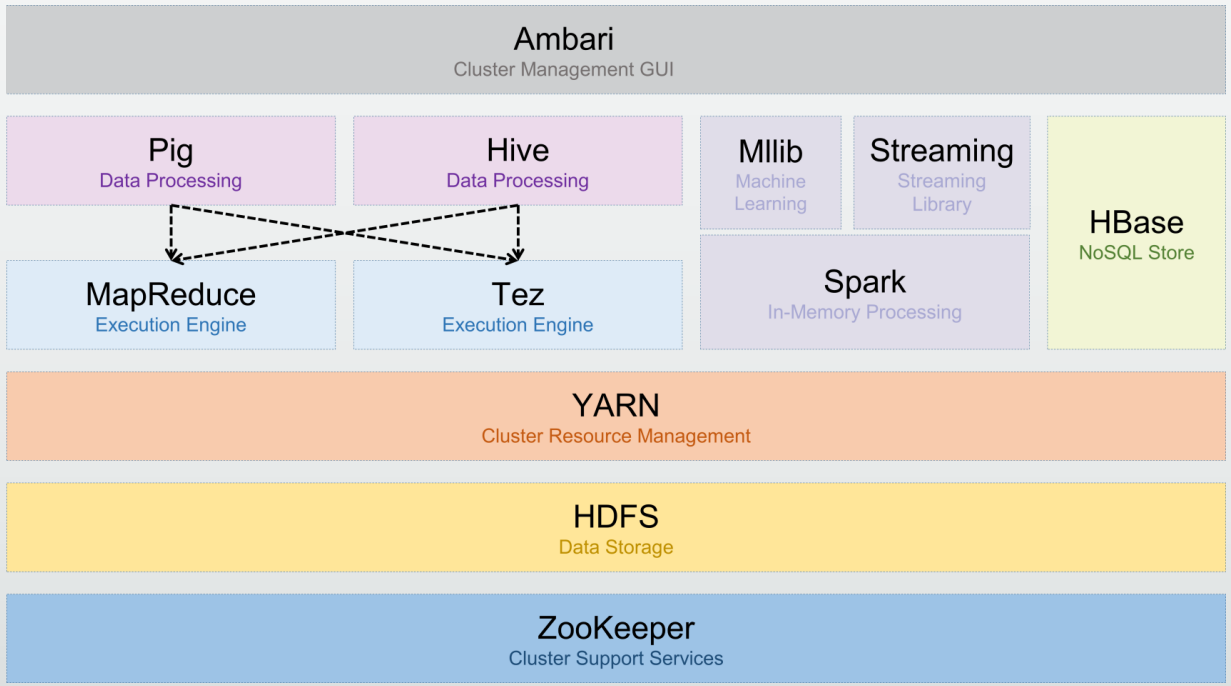
Über das Projekt

Datenbasis

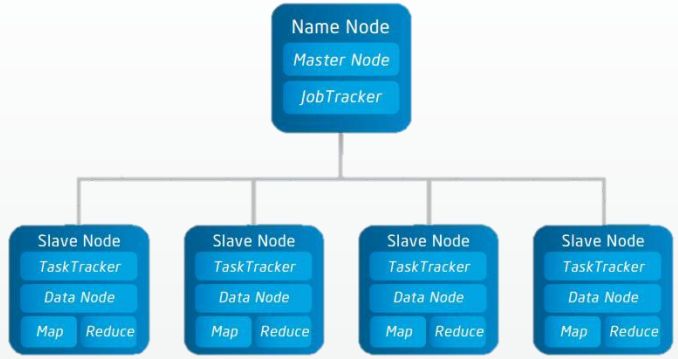
Strukturiert	Semistrukturiert
Loshistorien • Welche Schritte ist welches Los auf welcher Maschine wann gelaufen? **Maschinendaten** • Welche Ereignisse sind aufgetreten? 	**Testdaten** • Ergebnisse der Produkttests zu einem Los • Eine CSV pro Los und Testdurchlauf • Aufbau jeder Testdatei kann theoretisch unterschiedlich sein 

Allgemeines zum Apache Hadoop Framework

- Open-Source Framework zur verteilten Datenspeicherung und -analyse
  - Sammlung an Tools zum Speichern und Verarbeiten von Daten
    - **HDFS**: Verteilter Datenspeicher im Cluster
    - **YARN**: Ressourcen-Manager
    - **Hive**: SQL-Schnittstelle für strukturierte Daten
    - **HBase**: verteilte NoSQL Datenbank
    - **Spark**: Verarbeitungs-Engine
- Verteiltes Rechnen in Clustern
  - Master-Slave Architektur
  - Datenreplikation
  - Horizontale Skalierung
- Hadoop Distributionen
  - Hortonworks Data Platform / Data Flow + Ambari
  - Cloudera, MapR



Das Hadoop Cluster



- **Hortonworks Data Platform 2.6.4**
  - Hadoop 2.7.3

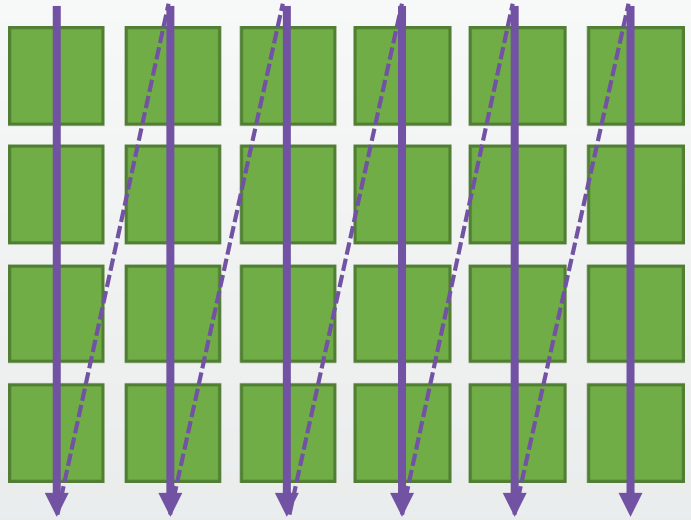
- **1 x NameNode**
  - CPU: 2 x Intel Xeon Silver 4114 @ 2.2 GHz (je 10 Kerne, 20 Threads)
  - RAM: 192 GB
  - System: 2 x 600 GB HDD (RAID-1)
  - /hadoop: 4 x 2 TB HDD (RAID-10)
  - OS: Red Hat Enterprise Linux Server 7.5

- **4 x DataNode**
  - CPU: 2 x Intel Xeon Silver 4114 @ 2.2 GHz (je 10 Kerne, 20 Threads)
  - RAM: 192 GB
  - System: 2 x 600 GB HDD (RAID-1)
  - HDFS: 10 x 2 TB
  - OS: Red Hat Enterprise Linux Server 7.5

	Current
	HDP-2.6.4.0
	Show Details
HDFS	2.7.3
YARN	2.7.3
MapReduce2	2.7.3
Tez	0.7.0
Hive	1.2.1000
HBase	1.1.2
Pig	0.16.0
Sqoop	1.4.6
ZooKeeper	3.4.6
Ambari Infra	0.1.0
Ambari Metrics	0.1.0
Log Search	0.5.0
Spark	1.6.x
Spark2	2.x
Zeppelin Notebook	0.7.0
NiFi	1.5.0
NiFi Registry	0.1.0
Slider	0.92.0

Speicherung der strukturierten Daten

- Anlegen von internen Hive Tabellen
  - **/apps/hive/warehouse**/<dbname>.db/<tablename>/<partCol>=<val>



- Speicherung der Daten im ORC Format
  - Spaltenbasiertes Speicherformat
  - Einteilung in sog. Stripes
  - Kompression (ZLIB, Snappy)
  - Predicate Pushdown

- Partitionierung der Tabelle
  - PartitionKey = concat(StartStep, substring(LosNr, 0, 3))
  - Anfragen beinhalten immer den StartStep und die Losnummer → PartitionKey berechenbar

Speicherung der semistrukturierten Messdaten

- Dateigröße 0,5 kB – 200 MB
  - → 0 – 30.000 Zeilen (PIDs)
  - → Teilweise >1.000 Spalten
  - → Datenmenge: ~300 GB / Monat (unkomprimiert)

```
Date;      2016_11_18 17:35:26
Lot;       281655.000
Sublot;    04
WaferID;   04
UserText;  P2N
Revision;  202
...
```

Parameter;	HBIN;	DIE_X;	DIE_Y;	TIME;	KO1;	KS1;	B1O	...
TestNR;	;	;	;	;	3;	4;	5;	
TestArt;	;	;	;	;	P;	P;	P;	
Unit;	;	;	;	msec;	V;	V;	V;	
HighL;	;	;	;	;	;	-0.2;	-0.540706;	
LowL;	;	;	;	;	-0.9;	;	-0.571229;	
HighS;	;	;	;	;	;	;	;	
LowS;	;	;	;	;	;	;	;	
PID-1;	1;	42;	12;	0;	-0.734;	-0.734;	-0.5535;	
PID-2;	1;	43;	12;	0;	-0.7334;	-0.7334;	-0.5529;	
PID-3;	1;	43;	13;	0;	-0.7334;	-0.7334;	-0.5527;	
...								

- **Problem: kein „gewöhnlicher" CSV-Aufbau**
  - Headerinformationen → Datei-/Testidentifikation
  - Einheiten, Grenzwerte etc. (zwischen Header und Tests) → Filtern
  - Einteilung der Spalten
    - Basisspalten: In jeder Datei vorhanden (+ für Filterung verwendet)
    - Testspalten: **Können pro Datei unterschiedlich sein**

→ Standard Bulk-Load Möglichkeiten nicht verwendbar →

Speicherung der semistrukturierten Messdaten – Variante 1

Prototyp erfolgreich umgesetzt mit ~100 GB Daten.

Zusätzlicher Aufwand durch HBase
→ „Ein Tool mehr zu verwalten“

- **Aufgaben**
  - Einlesen neuer Daten aus dem HDFS
  - Einbinden der Header-Attribute in Zeilen
    - Zusammensetzen des Row-Keys
  - Löschen der Attribut-Metainformationen
  - Einteilen der Spalten in HBase ColumnFamilies
  - Schreiben nach HBase

```
Date;       2016_11_18 17:35:26
Lot;        281655.000
Sublot;     04
WaferID;    04
UserText;   P2N
Revision;   202
...
```

Parameter;	HBIN;	DIE_X;	DIE_Y;	TIME;	KO1;	KS1;	B1O	...
TestNR;	;	;	;	;	3;	4;	5;	
TestArt;	;	;	;	;	P;	P;	P;	
Unit;	;	;	;	msec;	V;	V;	V;	
HighL;	;	;	;	;	;	-0.2;	-0.540706;	
LowL;	;	;	;	;	-0.9;	;	-0.571229;	
HighS;	;	;	;	;	;	;	;	
LowS;	;	;	;	;	;	;	;	
PID-1;	1;	42;	12;	0;	-0.734;	-0.734;	-0.5535;	
PID-2;	1;	43;	12;	0;	-0.7334;	-0.7334;	-0.5529;	
PID-3;	1;	43;	13;	0;	-0.7334;	-0.7334;	-0.5527;	
...								

Parameter;	HBIN;	DIE_X;	DIE_Y;	TIME;	LOT_ID;	WAFER_ID;	Date;	Sublot;	Usertext;	KO1;	KS1;	B1O	...
PID-1;	1;	42;	12;	0;	281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;	-0.734;	-0.734;	-0.5535;	
PID-2;	1;	43;	12;	0;	281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;	-0.7334;	-0.7334;	-0.5529;	
PID-3;	1;	43;	13;	0;	281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;	-0.7334;	-0.7334;	-0.5527;	
...													

Speicherung der semistrukturierten Messdaten

- Dateigröße 0,5 kB – 200 MB
  - → 0 – 30.000 Zeilen (PIDs)
  - → Teilweise >1.000 Spalten
  - → Datenmenge: ~300 GB / Monat

```
Date;      2016_11_18 17:35:26
Lot;       281655.000
Sublot;    04
WaferID;   04
UserText;  P2N
Revision;  202
...
```

Parameter;	HBIN;	DIE_X;	DIE_Y;	TIME;	KO1;	KS1;	B1O	...
TestNR;	;	;	;	;	3;	4;	5;	
TestArt;	;	;	;	;	P;	P;	P;	
Unit;	;	;	;	msec;	V;	V;	V;	
HighL;	;	;	;	;	;	-0.2;	-0.540706;	
LowL;	;	;	;	;	-0.9;	;	-0.571229;	
HighS;	;	;	;	;	;	;	;	
LowS;	;	;	;	;	;	;	;	
PID-1;	1;	42;	12;	0;	-0.734;	-0.734;	-0.5535;	
PID-2;	1;	43;	12;	0;	-0.7334;	-0.7334;	-0.5529;	
PID-3;	1;	43;	13;	0;	-0.7334;	-0.7334;	-0.5527;	
...								

- **Problem: kein „gewöhnlicher" CSV-Aufbau**
  - Headerinformationen → Datei-/Testidentifikation
  - Einheiten, Grenzwerte etc. (zwischen Header und Tests) → Filt
  - Einteilung der Spalten
    - Basisspalten: In jeder Datei vorhanden
    - Testspalten: **Können pro Datei unterschiedlich sein**

Sind aber für **eine** Datei (= 1 Test) gleich!

Es wird immer **eine ganze** Testspalte ausgelesen!

→ Standard Bulk-Load Möglichkeiten nicht verwendbar →

Speicherung der semistrukturierten Messdaten – Variante 2

- **Aufgaben**
  - Einlesen neuer Daten aus dem HDFS
  - Einbinden der Header-Attribute in Zeilen
    - Datei- bzw. Testidentifikation
  - Löschen der Attribut-Metainformationen
  - Transformieren der Test-Spalten in Werte-Arrays
    - **Key** Spalte = Testname
    - **ValueList** Spalte = Ergebnis-Array

```
Date;       2016_11_18 17:35:26
Lot;        281655.000
Sublot;     04
WaferID;    04
UserText;   P2N
Revision;   202
...
```

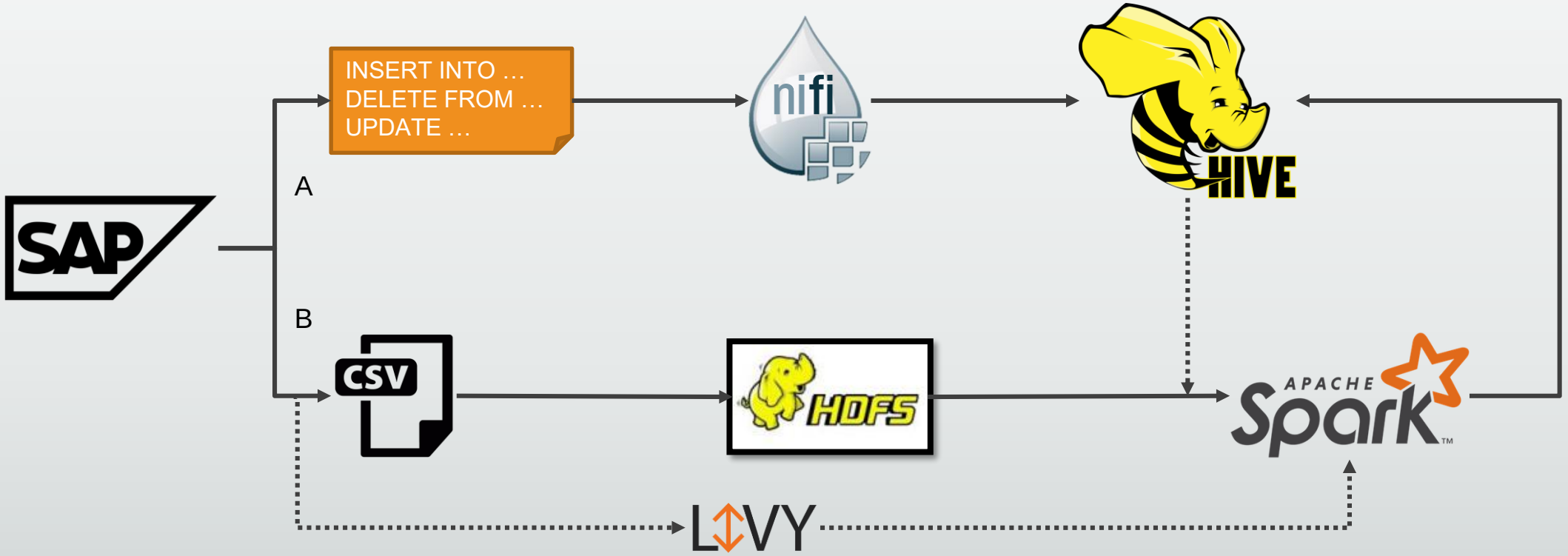
Parameter;	HBIN;	DIE_X;	DIE_Y;	TIME;	KO1;	KS1;	B1O	...
TestNR;	;	;	;	;	3;	4;	5;	
TestArt;	;	;	;	;	P;	P;	P;	
Unit;	;	;	;	msec;	V;	V;	V;	
HighL;	;	;	;	;	;	-0.2;	-0.540706;	
LowL;	;	;	;	;	-0.9;	;	-0.571229;	
HighS;	;	;	;	;	;	;	;	
LowS;	;	;	;	;	;	;	;	
PID-1;	1;	42;	12;	0;	-0.734;	-0.734;	-0.5535;	
PID-2;	1;	43;	12;	0;	-0.7334;	-0.7334;	-0.5529;	
PID-3;	1;	43;	13;	0;	-0.7334;	-0.7334;	-0.5527;	
...								

LOT_ID;	WAFER_ID;	Date;	Sublot;	Usertext;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;
281655.000;	04;	2016_11_18 17:35:26;	04;	P2N;

Key;	**ValueList**
Parameter;	[PID-1, PID-2, PID-3, …]
HBIN;	[1, 1, 1, …]
DIE_X;	[42, 43, 43, …]
DIE_Y;	[12, 12, 13, …]
TIME;	[0, 0, 0, …]
KO1;	[-0.734, -0.7334, -0.7334, …]
KS1;	[-0.734, -0.7334, -0.7334, …]
B1O;	[-0.5535, -0.5529, -0.5527, …]

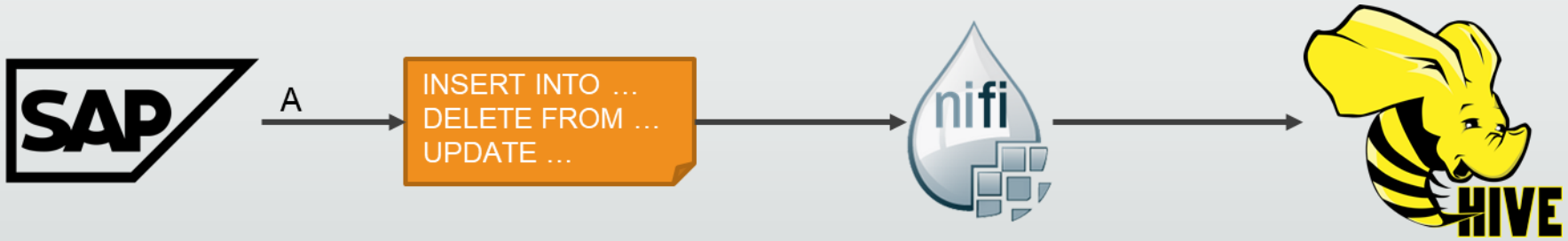
Datenimport

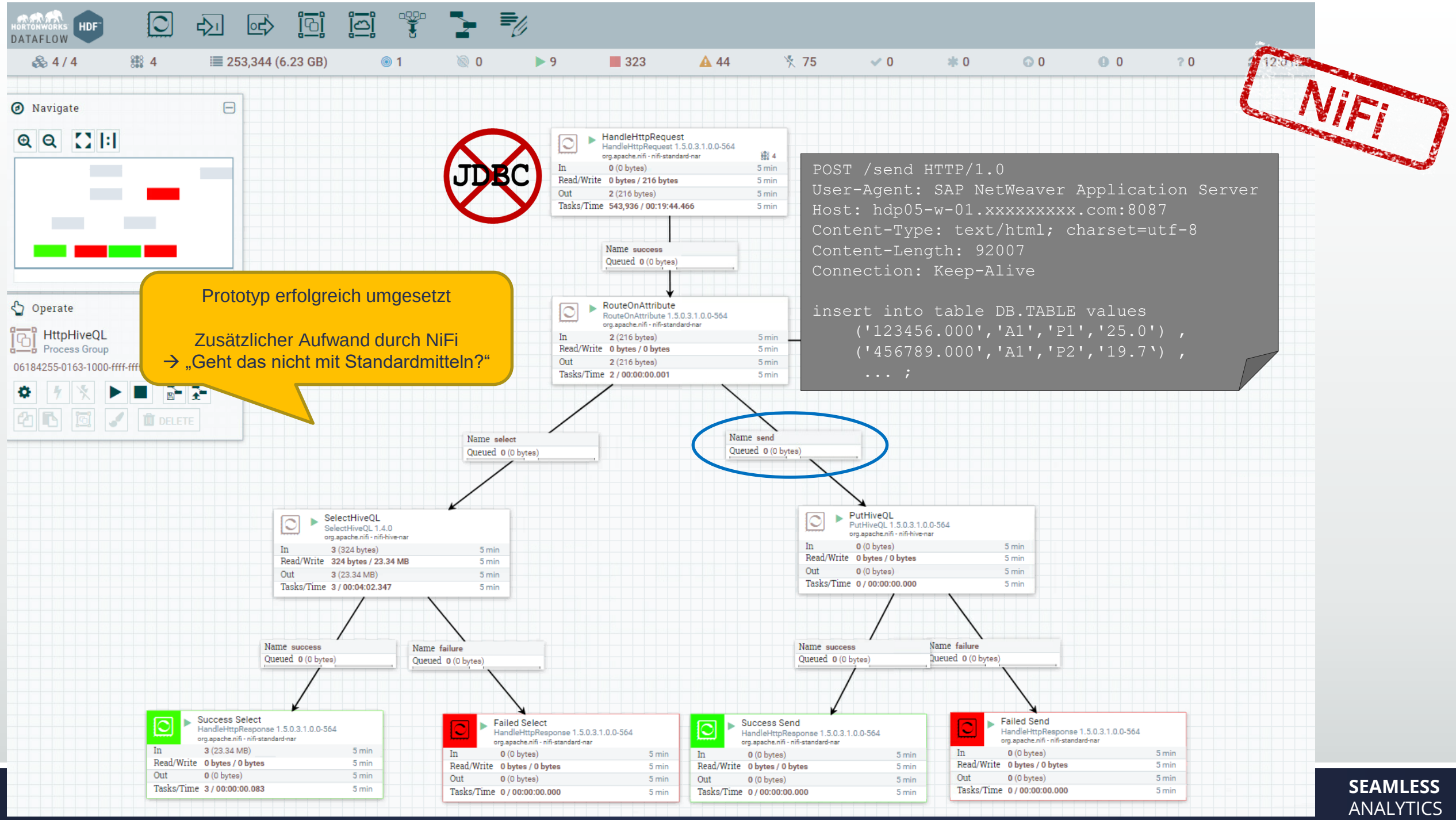
- Wie kommen neue Daten ins Hadoop?
  - SAP System als Datenlieferant | Apache Hive als Ziel-Speicher



Datenimport – Variante 1

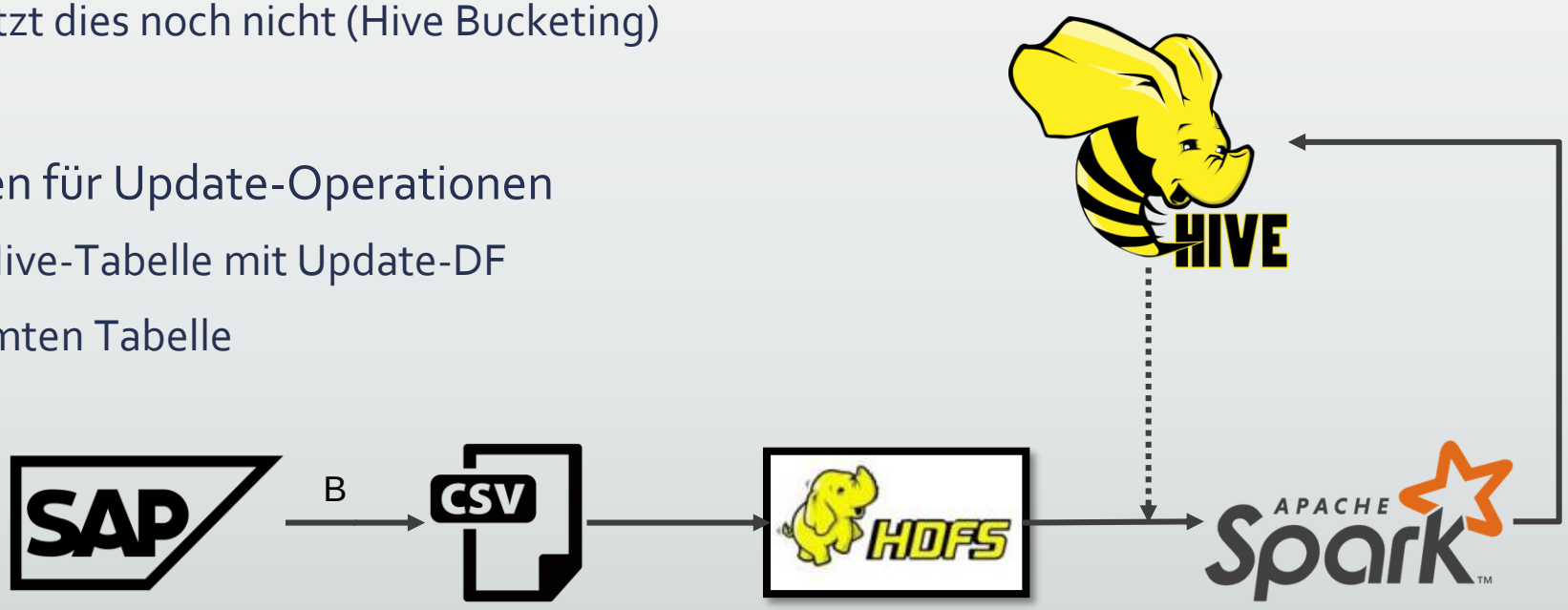
- Verwendung des Workflow Tools **Apache NiFi**
  - Enthalten im Hortonworks Data Flow (HDF) Paket
  - Workflow-Erstellung per Drag&Drop sog. Prozessoren
    - Config, Input/Output
    - Verbindung zu anderen Prozessoren = Weiterreichen der Daten



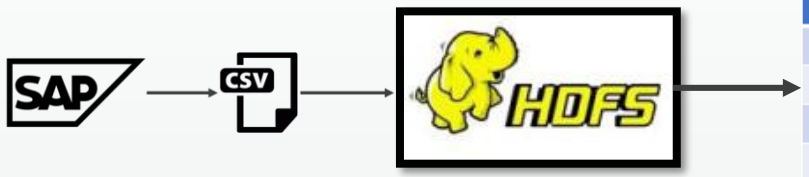

HORTONWORKS DATAFLOW HDF
4 / 4
4
253,344 (6.23 GB)
1
0
9
323
44
75
0
0
0
0
0
Navigate
Operate
HttpHiveQL
Process Group
DELETE
JDBC
NiFi
Prototyp erfolgreich umgesetzt
Zusätzlicher Aufwand durch NiFi
→ „Geht das nicht mit Standardmitteln?“
HandleHttpRequest
HandleHttpRequest 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 216 bytes 5 min
Out 2 (216 bytes) 5 min
Tasks/Time 543,936 / 00:19:44.466 5 min
Name success
Queued 0 (0 bytes)
RouteOnAttribute
RouteOnAttribute 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-standard-nar
In 2 (216 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 2 (216 bytes) 5 min
Tasks/Time 2 / 00:00:00.001 5 min
POST /send HTTP/1.0
User-Agent: SAP NetWeaver Application Server
Host: hdp05-w-01.xxxxxxxxx.com:8087
Content-Type: text/html; charset=utf-8
Content-Length: 92007
Connection: Keep-Alive
insert into table DB.TABLE values
('123456.000','A1','P1','25.0') ,
('456789.000','A1','P2','19.7') ,
... ;
Name select
Queued 0 (0 bytes)
Name send
Queued 0 (0 bytes)
SelectHiveQL
SelectHiveQL 1.4.0
org.apache.nifi - nifi-hive-nar
In 3 (324 bytes) 5 min
Read/Write 324 bytes / 23.34 MB 5 min
Out 3 (23.34 MB) 5 min
Tasks/Time 3 / 00:04:02.347 5 min
PutHiveQL
PutHiveQL 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-hive-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name success
Queued 0 (0 bytes)
Name failure
Queued 0 (0 bytes)
Name success
Queued 0 (0 bytes)
Name failure
Queued 0 (0 bytes)
Success Select
HandleHttpResponse 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-standard-nar
In 3 (23.34 MB) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 3 / 00:00:00.083 5 min
Failed Select
HandleHttpResponse 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Success Send
HandleHttpResponse 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Failed Send
HandleHttpResponse 1.5.0.3.1.0.0-564
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
SEAMLESS
ANALYTICS

Datenimport – Variante 2

- **Idee**: Verwendung von Spark SQL zum Datenabgleich
  - Datenabgleich über Spark Job mit Hive Zugriff
- **Problem**: Für Updates werden transaktionale Tabellen benötigt
  - Spark SQL 2.2 unterstützt dies noch nicht (Hive Bucketing)
- **Workaround**: CSV-Dateien für Update-Operationen
  - Zusammenführen der Hive-Tabelle mit Update-DF
  - Neuschreiben der gesamten Tabelle



Datenimport – Variante 2



Operation	Lot-ID	Step	Date	PID	Key	Value
INS	123456.002	A1	2019-03-07	1	Temp	23.0
DEL	123456.001	A1	2019-03-06	2	Temp	24.0
INS	123456.001	A2	2019-03-07	2	Temp	24.0
DEL	123456.000	A1	2019-03-06	0	null	null

Operation	Lot-ID	Step	Date	PID	Key	Value
DEL	123456.001	A1	2019-03-06	2	Temp	24.0
DEL	123456.000	A1	2019-03-06	0	null	null

Operation	Lot-ID	Step	Date	PID	Key	Value
INS	123456.002	A1	2019-03-07	1	Temp	23.0
INS	123456.001	A2	2019-03-07	2	Temp	24.0

Lot-ID	Step	Date	PID	Key	Value
111222.000	A5	2019-03-06	1	P	12.3
123456.000	A1	2019-03-06	0	P	11.9
123456.000	A1	2019-03-06	0	U	5.1
123456.001	A1	2019-03-06	2	Temp	24.0
111555.000	A4	2019-03-06	3	L	2.2

Datenimport – Variante 2

Operation	Lot-ID	Step	Date	PID	Key	Value
INS	123456.002	A1	2019-03-07	1	Temp	23.0
DEL	123456.001	A1	2019-03-06	2	Temp	24.0
INS	123456.001	A2	2019-03-07	2	Temp	24.0
DEL	123456.000	A1	2019-03-06	0	null	null

Operation	Lot-ID	Step	Date	PID	Key	Value
DEL	123456.001	A1	2019-03-06	2	Temp	24.0
DEL	123456.000	A1	2019-03-06	0	null	null

Operation	Lot-ID	Step	Date	PID	Key	Value
INS	123456.002	A1	2019-03-07	1	Temp	23.0
INS	123456.001	A2	2019-03-07	2	Temp	24.0

Lot-ID	Step	Date	PID	Key	Value
111222.000	A5	2019-03-06	1	P	12.3
~~123456.000~~	~~A1~~	~~2019-03-06~~	~~0~~	~~P~~	~~11.9~~
~~123456.000~~	~~A1~~	~~2019-03-06~~	~~0~~	~~U~~	~~5.1~~
~~123456.001~~	~~A1~~	~~2019-03-06~~	~~2~~	~~Temp~~	~~24.0~~
111555.000	A4	2019-03-06	3	L	2.2

Datenimport – Variante 2

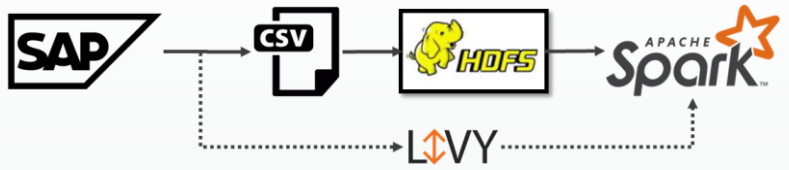
Operation	Lot-ID	Step	Date	PID	Key	Value
INS	123456.002	A1	2019-03-07	1	Temp	23.0
DEL	123456.001	A1	2019-03-06	2	Temp	24.0
INS	123456.001	A2	2019-03-07	2	Temp	24.0
DEL	123456.000	A1	2019-03-06	0	null	null

Operation	Lot-ID	Step	Date	PID	Key	Value
DEL	123456.001	A1	2019-03-06	2	Temp	24.0
DEL	123456.000	A1	2019-03-06	0	null	null

Operation	Lot-ID	Step	Date	PID	Key	Value
INS	123456.002	A1	2019-03-07	1	Temp	23.0
INS	123456.001	A2	2019-03-07	2	Temp	24.0

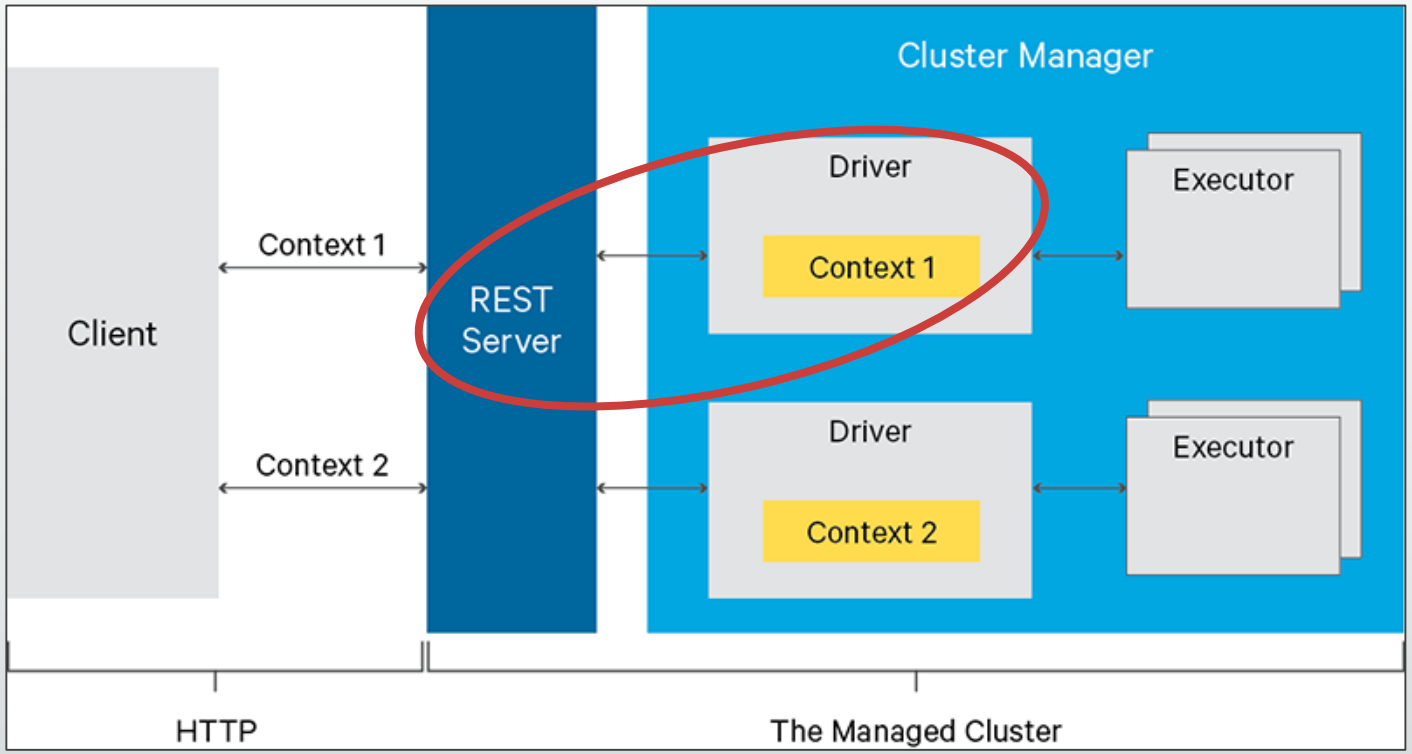
Lot-ID	Step	Date	PID	Key	Value
111222.000	A5	2019-03-06	1	P	12.3
111555.000	A4	2019-03-06	3	L	2.2
123456.001	A2	2019-03-07	2	Temp	24.0
123456.002	A1	2019-03-07	1	Temp	23.0

Datenimport – Variante 2



SPARK + LIVY

- Livy zum Triggern des Spark Jobs aus externem System heraus
  - Livy = REST Schnittstelle zur Verwaltung von Spark Sessions




```
POST /batches
{
  "className"       : "examples.SparkPi",
  "driverMemory"    : "4g",
  "executorMemory"  : "4g",
  "numExecutors"    : "10",
  "file"    : "/path/to/examples.jar",
  "args"    : ["a", "b", "100"]
}
```

```
$ spark-submit
  --class examples.SparkPi
  --driver-memory 4g
  --executor-memory 4g
  --num-executors 10
  --deploy-mode yarn-cluster
  /path/to/examples.jar
  a b 100
```

Benchmarks

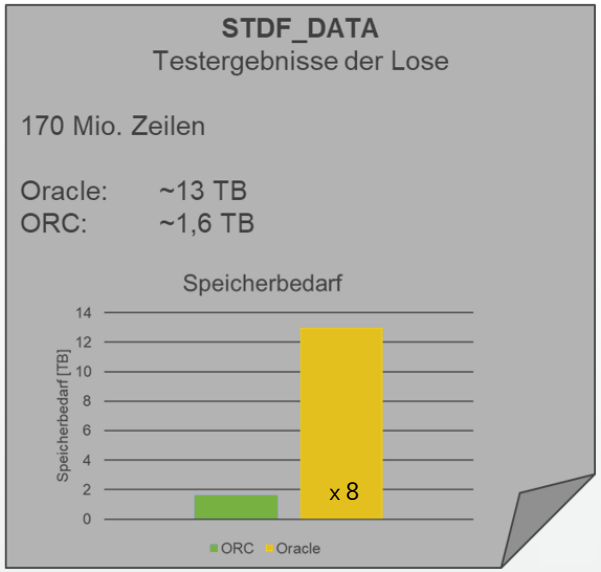
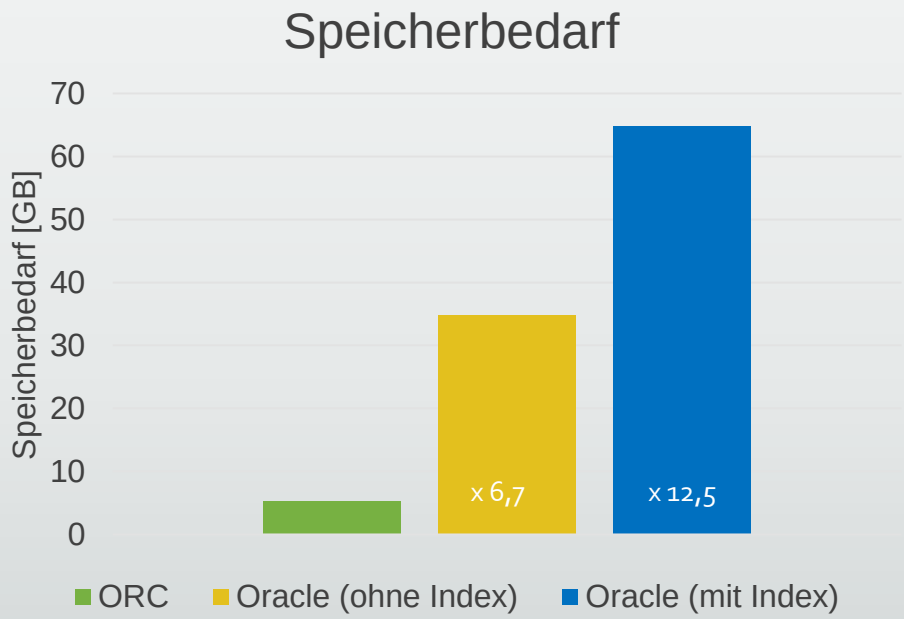
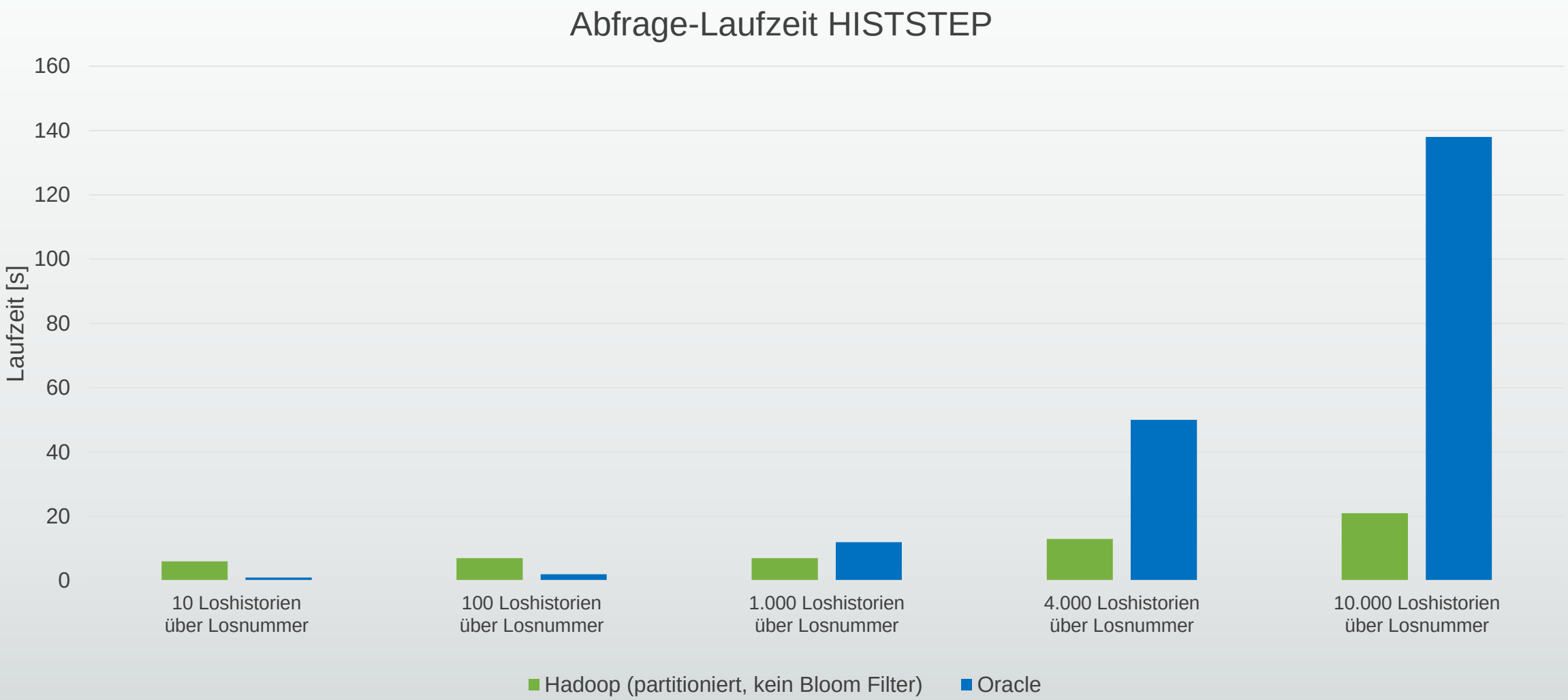


Tabelle HISTSTEP

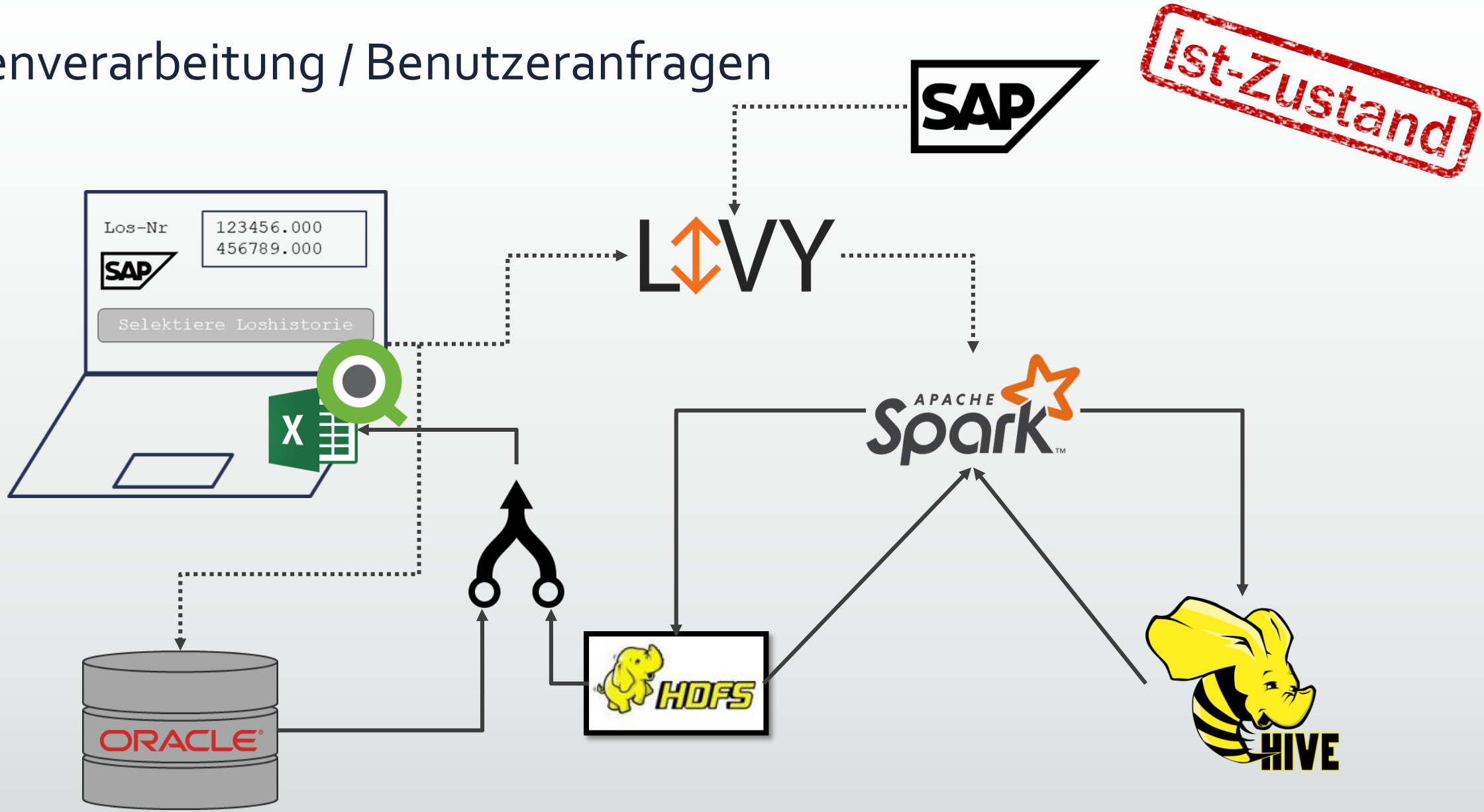
- Enthält die durchlaufenen Arbeitsschritte von Losen
- ~275 Mio. Zeilen
- 17 Spalten
  - STRING, VARCHAR(1-20)
- Speicherbedarf
  - Oracle: 34,9 GB (+29,9 GB Index)
  - ORC: 5,2 GB (partitioniert, ohne Bloom Filter Columns)



Benchmarks

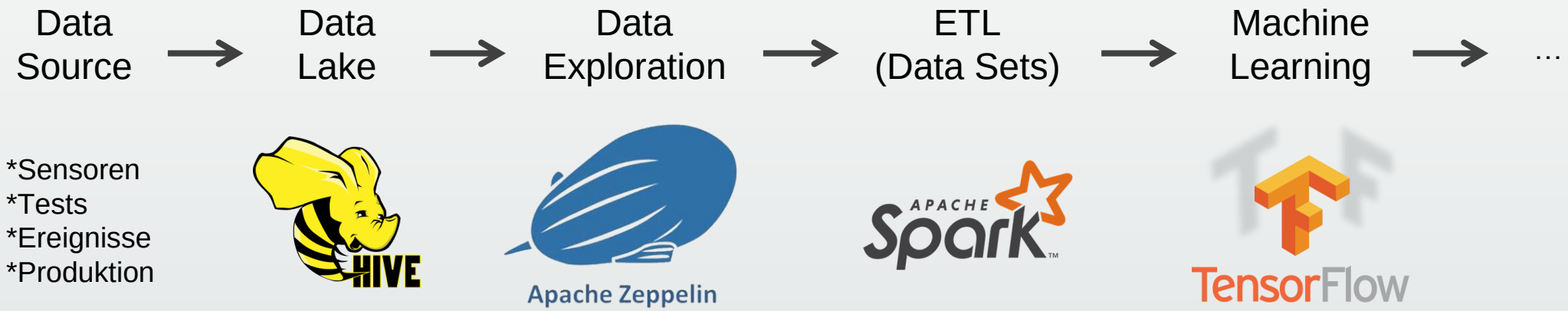

Abfrage-Laufzeit HISTSTEP
Laufzeit [s]
160
140
120
100
80
60
40
20
0
10 Loshistorien über Losnummer
100 Loshistorien über Losnummer
1.000 Loshistorien über Losnummer
4.000 Loshistorien über Losnummer
10.000 Loshistorien über Losnummer
Hadoop (partitioniert, kein Bloom Filter)
Oracle

Datenverarbeitung / Benutzeranfragen



Maschinelles Lernen mit Hadoop

Typischer Data Flow



Maschinelles Lernen mit Hadoop

- Export für ML
  - Exportieren der Daten im passenden Format z.B. mittels Spark

+ Keine neue Komponente im Hadoop

+ Gleichermaßen möglich mit Hadoop 2 und 3

+ Der ML-Entwickler/Data-Scientist kann die Daten in seiner gewohnten Umgebung mit beliebigen Tools analysieren und nutzen.

- Zwei getrennt Architekturen

Maschinelles Lernen mit Hadoop

- Innerhalb des bestehenden Clusters
  - Verwendung von Submarine um ML direkt im Hadoop Cluster zu betreiben

+ Durch Hadoop 3 GPU Unterstützung
+ Submarine: Zeppelin Notebook Integration
+ Der ganze ML Prozess findet „Standardisiert" im Cluster statt.
- Noch nicht ganz ausgereift

Vielen Dank!

Wir suchen nach neuen, spannenden Projekten!

Sprechen Sie uns an... ☺

Daniel Müller
daniel.mueller@seamless-analytics.de

Marvin Follmann
marvin.follmann@seamless-analytics.de