Requirements for Data Warehouse + Data Lake as a hybrid Infrastruktur

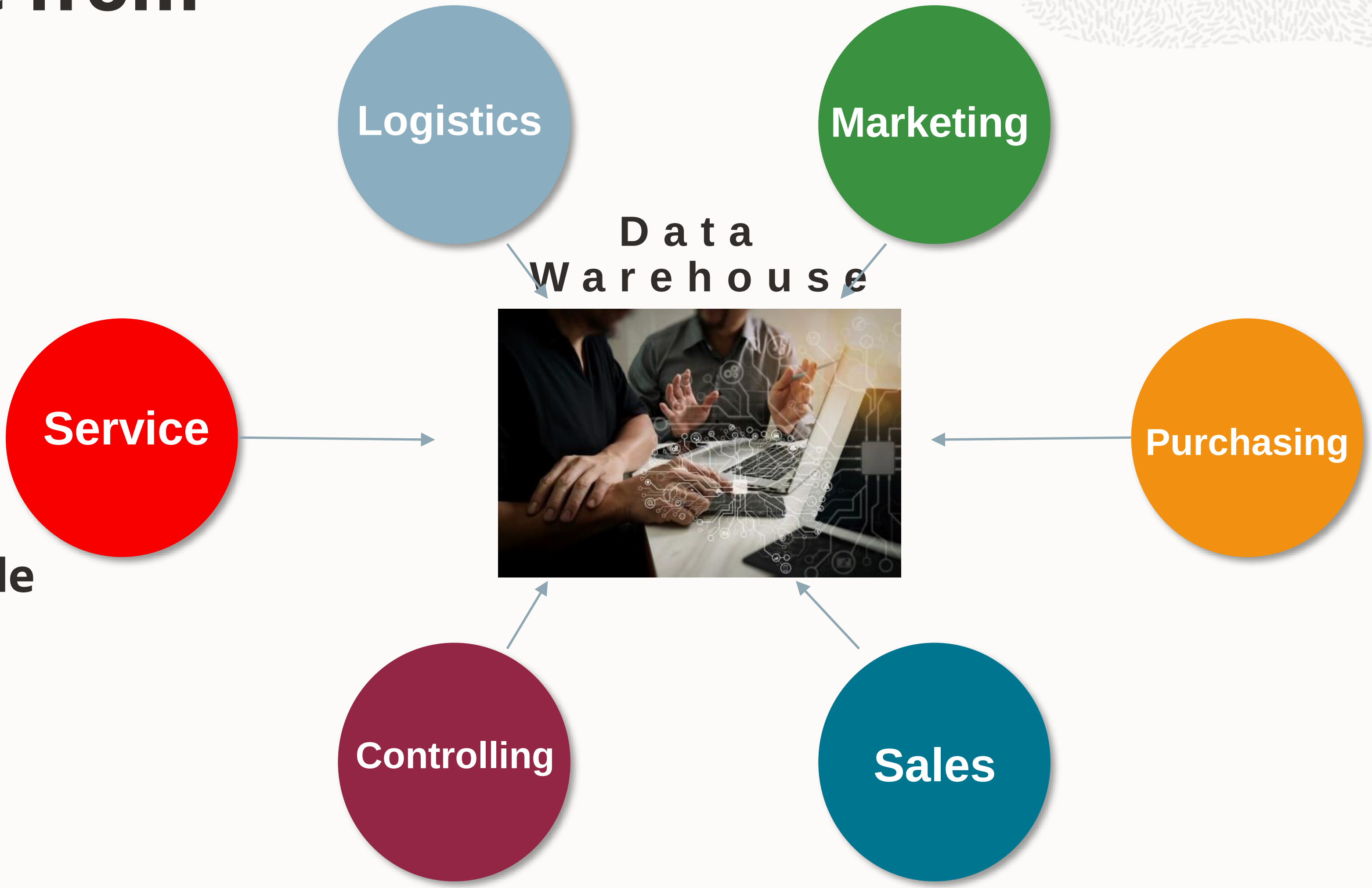
Alfred Schlaucher, Oracle

alfred.schlaucher@oracle.com

Requirements Data Lakes and Analytical Platform

Lets start with classical conceptual requirements for Data Warehouse from the 1990s to today

- Central Accesspoint
- Company-wide
- Uniform and consistent
- Understandable for all people enriched
- Historical



Classical conceptual requirements
Data Warehouse from the 1990s to today

- Central Accesspoint
- Company-wide
- Uniform and consistent
- Understandable for all people enriched
- Historical

Additional technical and organizational requirements

- Easy access from all tools
- SQL (Structured Query Language)
- Interoperability / *connection to all Apps*
- Automation
- Controllability *„Which information is stored“ (The Metadata-stuff)*

Question:

What about this requirements related to Data Lakes?

Lets have a look to what is the same and what will be changed

What should be analyzed and how

User	Management	Domain stuff/ Controller	Planning/ Controlling	Data Scientist (Cross-Lob)
Type of Usage	Standard Reporting	Interactive analysis Dashboards	Predictive Analytics	Exploratory analysis
What is used	Consolidated Data	Consolidated Data Operational Data	Consolidated Data Operational Data External Data	Operational Data External Data

Who is responsible for

- CIO (Visionary)
- Data Warehouse development (Central-IT)
- Business Analyst (Department)
- Architects (Staff Office/ Cross-Lob-Function)

What should be analyzed and how

User

- Management → **Standard Reporting**
- Domain stuff/ Controller → **Interactive analysis Dashboards**
- Planning/ Controlling → **Predictive Analytics**
- Data Scientist (Cross-Lob) → **Exploratory analysis**

Type of Usage

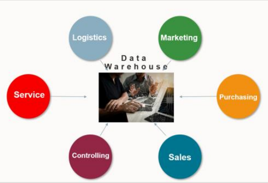
What is used

Consolidated Data

Consolidated Data

Data Warehouse

Consolidated Data



Operational Data

Operational Data

Operational Data

Data Lake

External Data

Data

Who is responsible for

- CIO (Visionary)
- Data Warehouse development (Central-IT)
- Business Analyst (Department)
- Architects (Staff Office/ Cross-Lob-Function)

What should be analyzed and how

User

- Management → **Standard Reporting**
- Domain stuff/ Controller → **Interactive analysis Dashboards**
- Planning/ Controlling → **Predictive Analytics**
- Data Scientist (Cross-Lob) → **Exploratory analysis**

Type of Usage

Standard Reporting | Interactive analysis Dashboards | Predictive Analytics | Exploratory analysis

What is used

Data Warehouse

- Data prepared for specialist issues
- Data compiled and consolidated from several specialist topics
- Historizations for the detection of trends
- Enriched by reference data for comparisons and abstractions
- Separation of data for planning, simulation and business

Data Lake

- Central and shared repository for any operational data
- Adhoc - repository for storage of fast operational data (streaming, logs)
- Bulk storage for "unclean" non-processed data
- Archives and low-cost backup storage
- Project data (e.g. for data science projects)

Who is responsible for

- CIO (Visionary)
- Data Warehouse development (Central-IT)
- Business Analyst (Department)
- Architects (Staff Office/ Cross-Lob-Function)

What should be analyzed and how

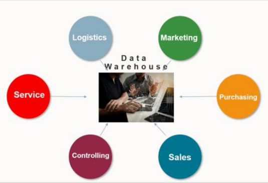
User	Management	Domain stuff/ Controller	Planning/ Controlling	Data Scientist (Cross-Lob)
Type of Usage	Standard Reporting	Interactive analysis Dashboards	Predictive Analytics	Exploratory analysis

What is used

Data Warehouse

store only the data you want to analyze

prepare it in the best possible way for this purpose



Data Lake

store all data you can get, quickly and in original form

do not worry about later use

Who is responsible for

- CIO (Visionary)
- Data Warehouse development (Central-IT)
- Business Analyst (Department)
- Architects (Staff Office/ Cross-Lob-Function)

Short Answer: Requirements are the same!

Requirements do not change due to the type and procedure of storage.

Conceptual Requirements

- Central Accesspoint
- Company-wide
- (Uniform and consistent)*
- (Understandable for all people enriched)*
- (Historical)*

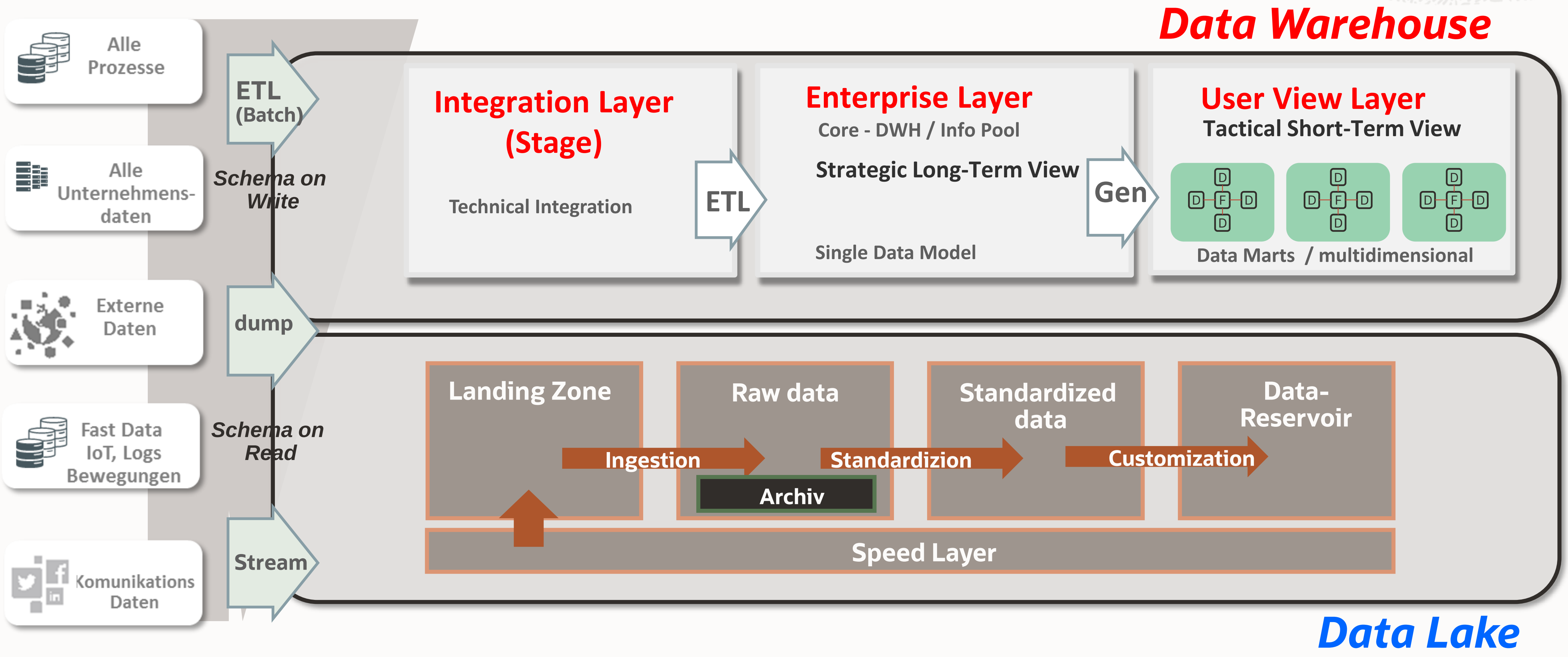
* only for Data Warehouse

Technical and organizational Requirements

- Easy access from all tools
- SQL
- Interoperability / *connection to all Apps*
- Automation
- Controllability *„Which information is stored“ (The Metadata-stuff)*

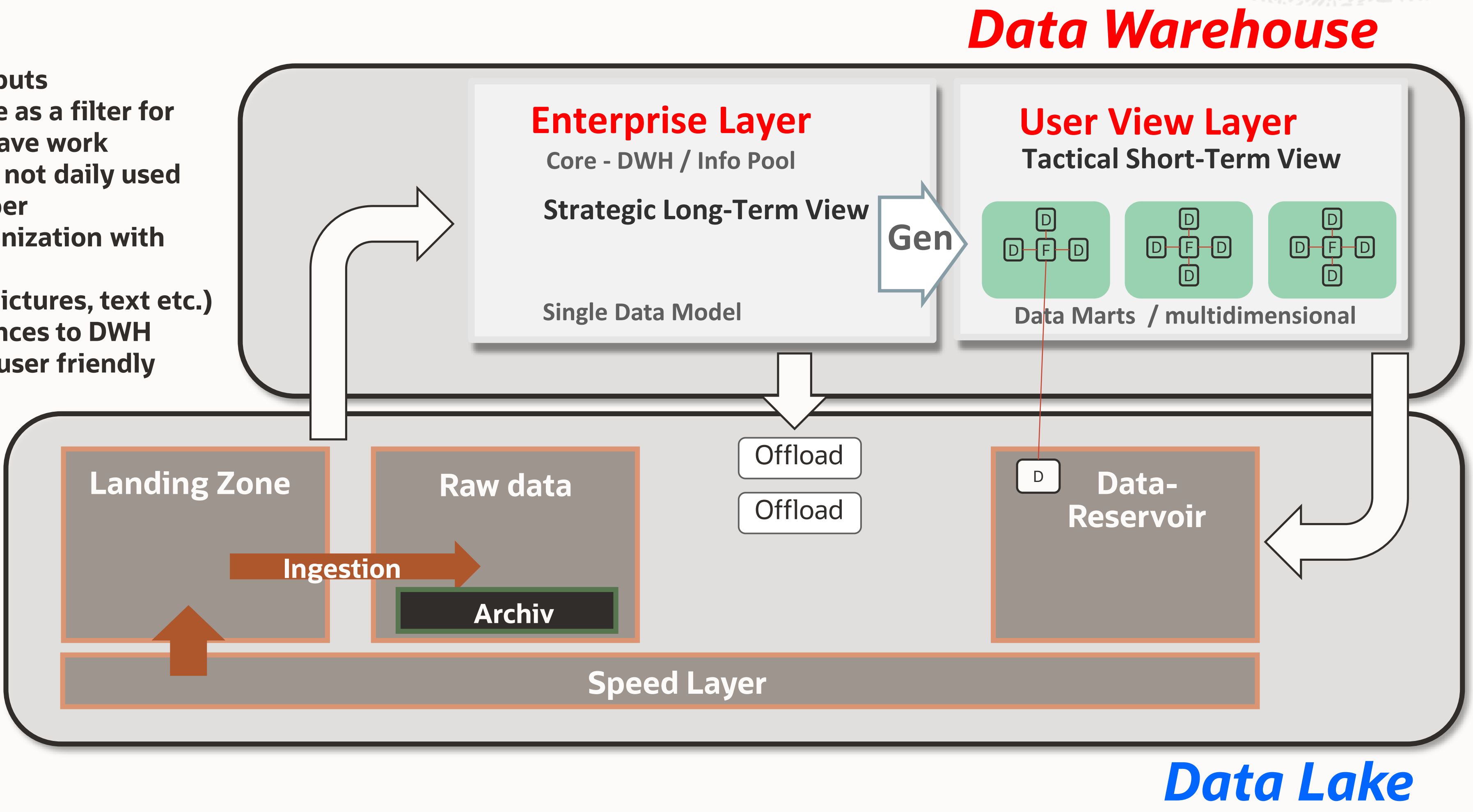
Can we combine both solutions to get a much *better approach*?

Typical architectures for Data Warehouse and Data Lake



The possibilities / advantages of an integrated (hybrid) architecture of Data Warehouse and Data Lake

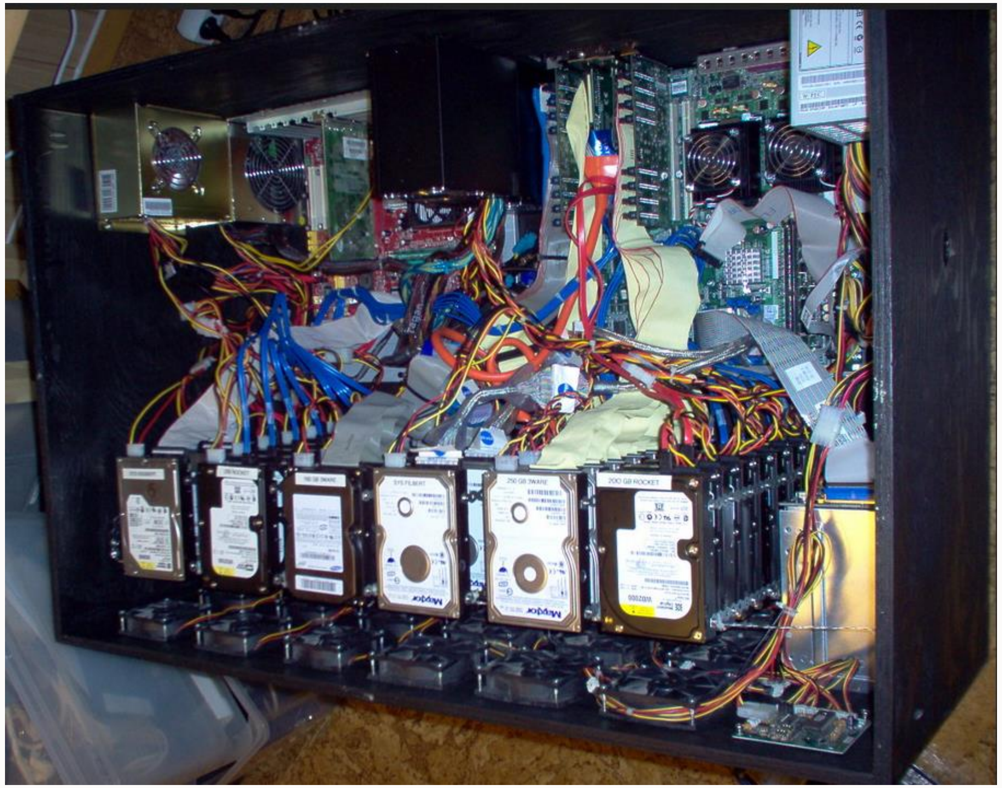
- One Landing Zone for all inputs
- Use Data Lake Landing Zone as a filter for the whole environment -> save work
- Move archive data and data not daily used in Data Lake -> this is sheaper
- Do standardization / harmonization with SQL in DWH -> it is easier
- Store not structured data (pictures, text etc.) in Data Lake -> build references to DWH
- Use DWH (Data Marts) as a user friendly Entry Gate for end user (SQL) - queries (e. g. Business Intelligence Tools)



The possibilities / advantages of an integrated architecture of data warehouse and data lake

- **Shifting staging tasks to the data lake area**
  - Not all data has to go into the DWH.
  - The data lake takes over a filter function for DWH.
  - Load data can be archived more cheaply in the data lake
  - Advantages: Cost savings and acceleration of loading runs
- **Shifting of standardization / harmonization tasks to the DWH area**
  - This task is particularly challenging in data lakes because it requires a lot of programming, whereas such tasks are easier to perform with SQL. – Define Transformation rules with declarative nearly *natural linguistic sounding* language
- **Moving older data, which is hardly ever read, from the data warehouse to the data lake**
  - The cost of storing such data in a database-driven data warehouse is usually higher than storing a file in the data lake.
- **Cost-effective provision of additional data such as images, movies, texts for data warehouse analyses**
  - Classic warehouse reports can be enriched with such additional types of data. For example, reports with article images or brochures and catalogs for individual customer approaches can be generated.
- **Data Marts as a multidimensional, SQL – based Entry Gate for non-technical users**
  - SQL – based is best for Business Intelligence Tools
  - Direct access to Data Lake through SQL

Build it yourself or flexibly from the cloud *On Demand*



or

Regarding build and running:
Analytics environments are among the most challenging

But Cloud environments offers perfect solutions here

- Ressourcen-On-Demand
- Always the right technology for project purposes
- Automatic Scaling without administration in your project
- Much cheaper: Pay As You Go
- Always the newest technology

Therefore:
Cloud environments fits best for Analytics

Today:
Most new Analytics initiatives start in Cloud

Example from the Deep Learning space

Jupyter Anaconda Notebook Session – switch in seconds

jupyterlab
Version 2.2.6

CPU calculated

```
model = Sequential()
model.add(Dense(200,input_dim=784,activation='relu'))
model.add(BatchNormalization())
model.add(Dense(64,activation='relu'))
model.add(BatchNormalization())
model.add(Dense(64,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(10,activation='softmax'))
```

```
start = timer()
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
#model.fit(XTrain, YTrain, batch_size=256, epochs=10, verbose=True)
model.fit(XTrain, YTrain, batch_size=512, epochs=10, verbose=True)
vector_add_cpu_time = timer() - start
print("Zeitverbrauch: ", vector_add_cpu_time)
```

```
Epoch 1/10
118/118 [==============================] - 9s 53ms/step - loss: 0.4547 - accuracy: 0.8678
Epoch 2/10
118/118 [==============================] - 6s 48ms/step - loss: 0.1449 - accuracy: 0.9587
Epoch 3/10
118/118 [==============================] - 6s 51ms/step - loss: 0.0923 - accuracy: 0.9736
Epoch 4/10
118/118 [==============================] - 6s 55ms/step - loss: 0.0676 - accuracy: 0.9804
Epoch 5/10
118/118 [==============================] - 6s 53ms/step - loss: 0.0506 - accuracy: 0.9848
Epoch 6/10
118/118 [==============================] - 7s 56ms/step - loss: 0.0381 - accuracy: 0.9887
Epoch 7/10
118/118 [==============================] - 6s 51ms/step - loss: 0.0326 - accuracy: 0.9902
Epoch 8/10
118/118 [==============================] - 6s 52ms/step - loss: 0.0247 - accuracy: 0.9926
Epoch 9/10
118/118 [==============================] - 6s 52ms/step - loss: 0.0197 - accuracy: 0.9946
Epoch 10/10
118/118 [==============================] - 6s 52ms/step - loss: 0.0145 - accuracy: 0.9962
Zeitverbrauch:  86.72603639098816
```

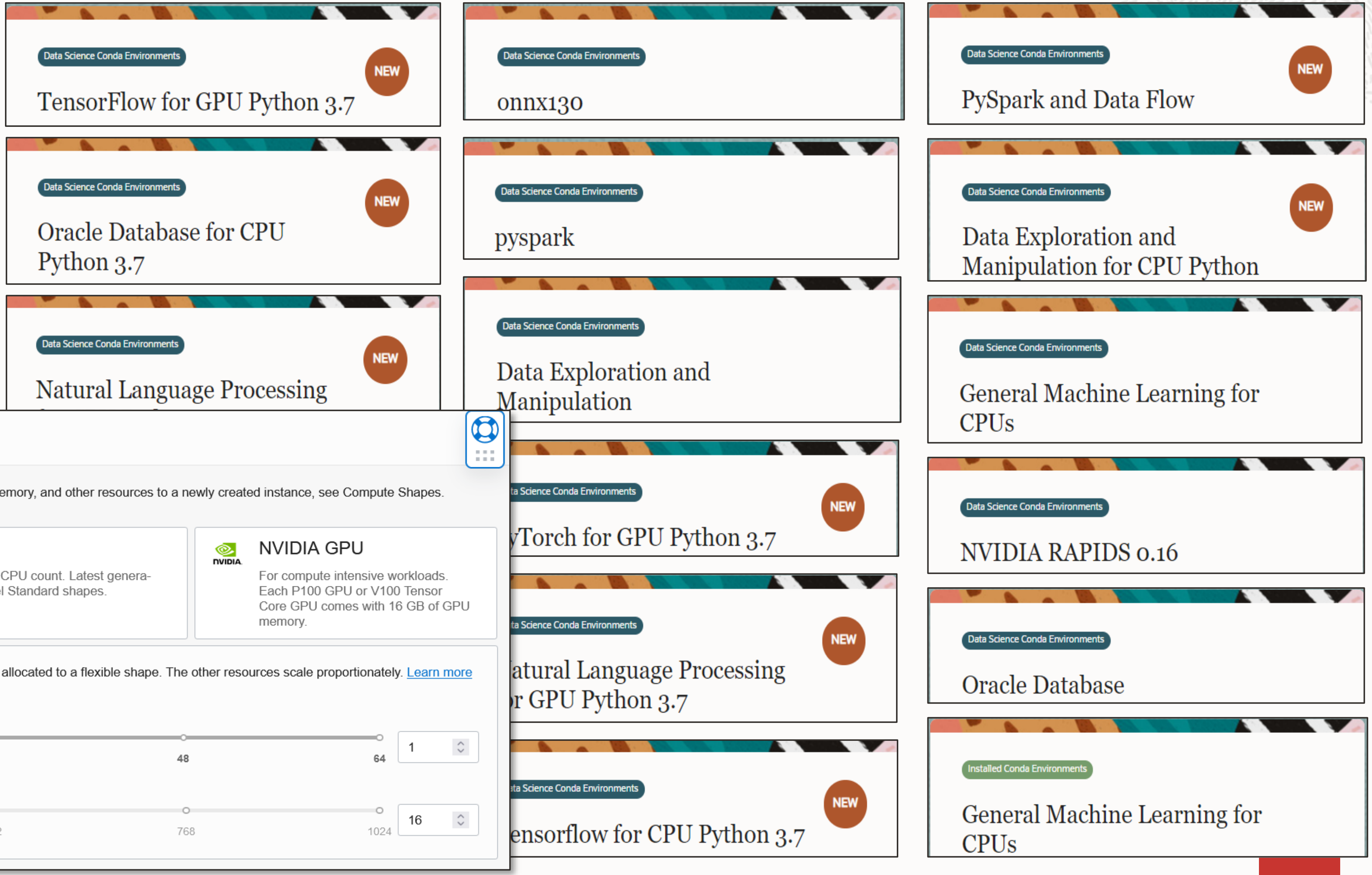
GPU calculated

```
model = Sequential()
model.add(Dense(200,input_dim=784,activation='relu'))
model.add(BatchNormalization())
model.add(Dense(64,activation='relu'))
model.add(BatchNormalization())
model.add(Dense(64,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(10,activation='softmax'))
```

```
start = timer()
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
#model.fit(XTrain, YTrain, batch_size=256, epochs=10, verbose=True)
model.fit(XTrain, YTrain, batch_size=512, epochs=10, verbose=True)
vector_add_cpu_time = timer() - start
print("Zeitverbrauch: ", vector_add_cpu_time)
```

```
Epoch 1/10
118/118 [==============================] - 1s 4ms/step - loss: 0.4684 - accuracy: 0.8668
Epoch 2/10
118/118 [==============================] - 0s 4ms/step - loss: 0.1466 - accuracy: 0.9586
Epoch 3/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0960 - accuracy: 0.9729
Epoch 4/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0689 - accuracy: 0.9805
Epoch 5/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0514 - accuracy: 0.9849
Epoch 6/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0389 - accuracy: 0.9888
Epoch 7/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0343 - accuracy: 0.9895
Epoch 8/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0250 - accuracy: 0.9930
Epoch 9/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0210 - accuracy: 0.9937
Epoch 10/10
118/118 [==============================] - 0s 4ms/step - loss: 0.0166 - accuracy: 0.9953
Zeitverbrauch:  6.225744802999543
```

Data Science Environments on demand in Seconds



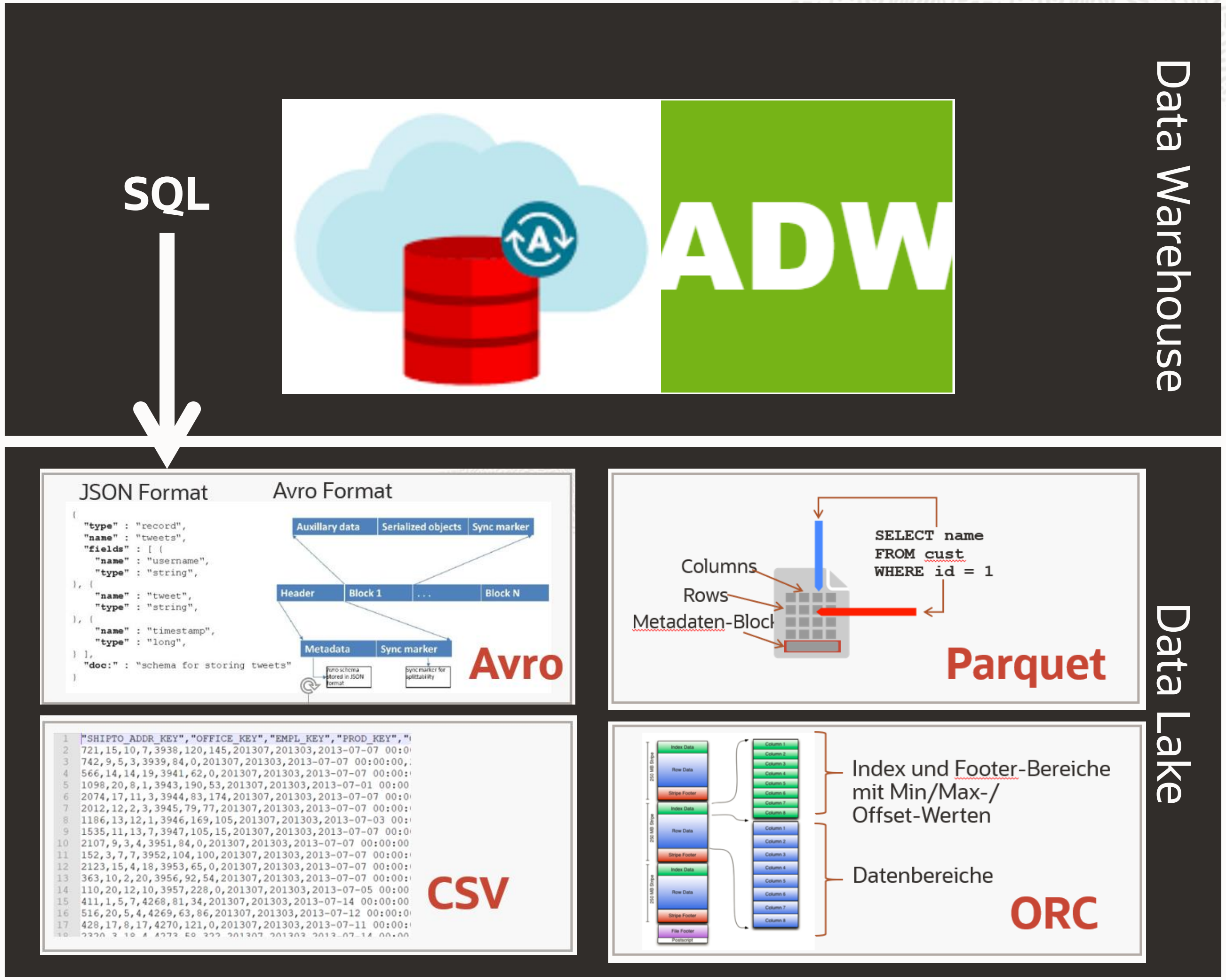
Store data where you want – you have the freedom

Object Storage replaces HDFS

- Relational model for consolidated data
- Autonomous Data Warehouse database
- Self-managing database
- Auto Scaling
- Running nearly without administration

- Data Lakes are mostly a collection of files
- Object Storage (S3) replaces HDFS (Hadoop) (in cloud we are not using HDFS anymore)

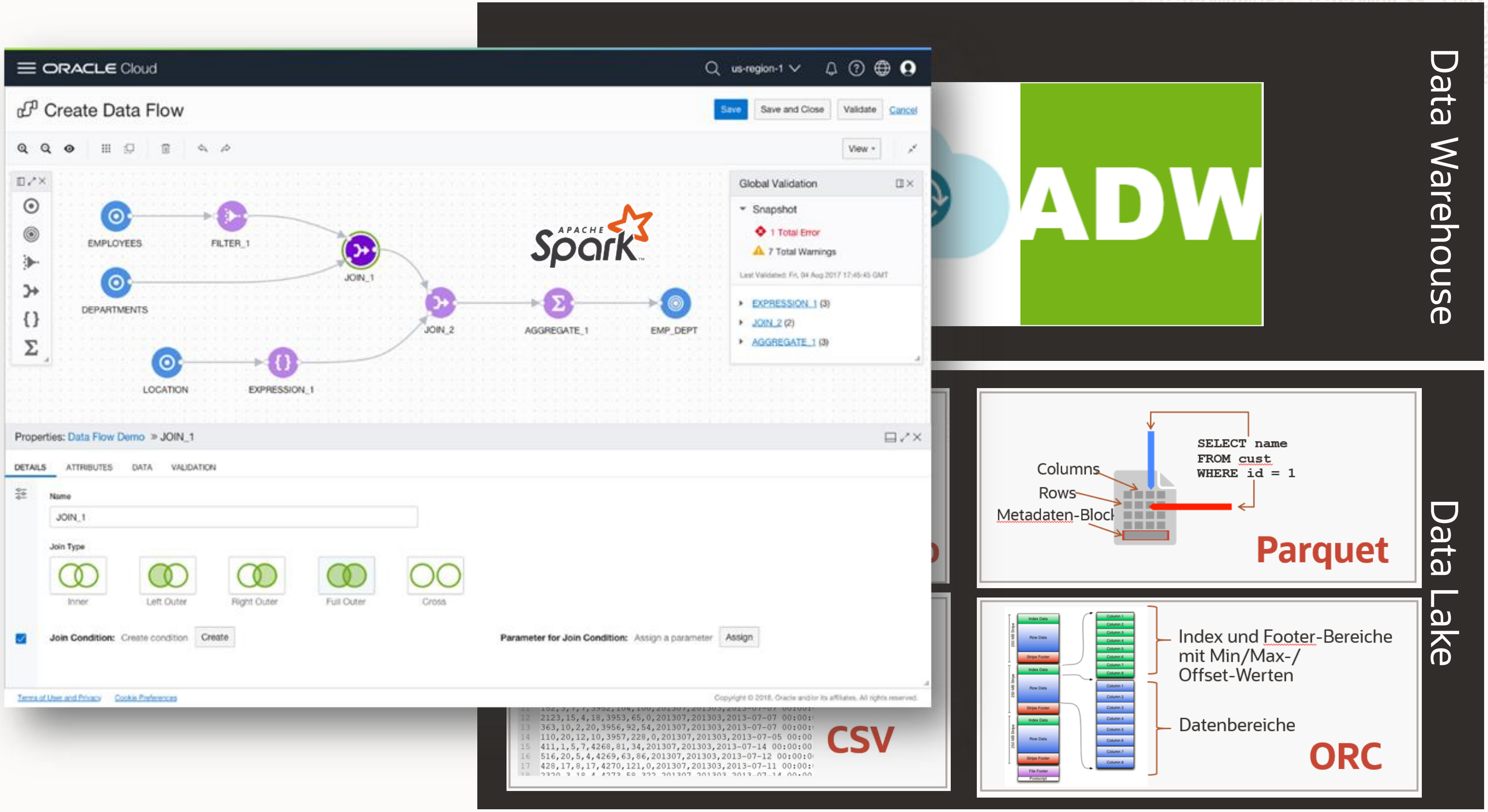
- Use Parquet instead of CSV (much smaller and faster)
- Access files with SQL



Move data from any place to any other place or transform it to a suitable structure

- prevent complexity by declare rules instead of programming language
- Use wizzards

- Use Spark environments for large datasets instead of old Hadoop Map Reduce



One of the most difficult challenges must be solved:
How to find the right information at the right time?

Consider this scenario

Employees are the most important resource in the company!
How to prevent key employees from quitting the job?
Where is the data on the quitting job behavior of employees?

Controlling dep.

employees cost?

Hr. Meier

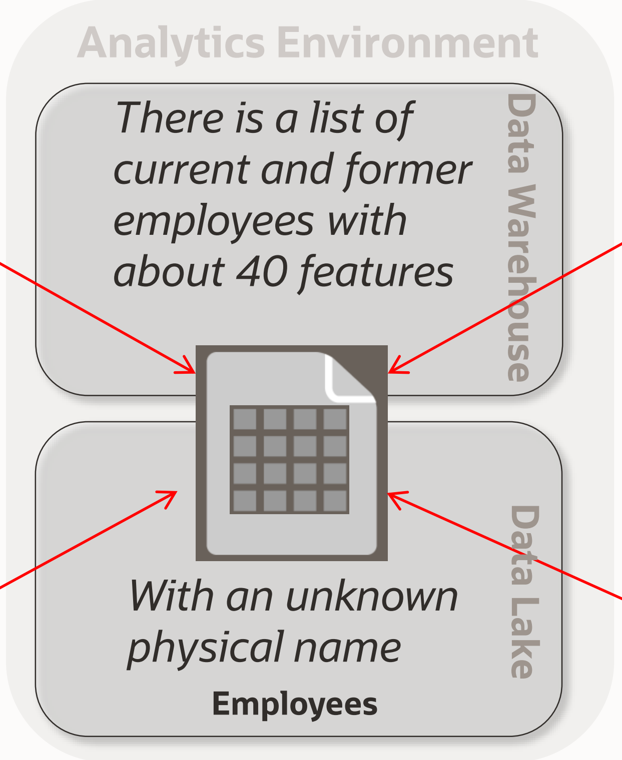
Hiring cost
payroll

Customer Care / Sales dep.

skills required for customer projects

Fr. Müller

Resource
Knowledge
Skills



Human resources dep.

Employees' affinity for quitting

Fr. Schmidt

Quitting job
Cancel job

Management

Employee structure, male, female, number

Hr. Bauer

Master data
Employee

You need business context around your objects in Data Lake and Data Warehouse

The important role of a Glossar

Collect all relevant terms in a Business Glossar

Controlling

employees cost?

Hr. Meier

Hiring cost
payroll

Human resources

Employees' affinity for quitting

Fr. Schmidt

Quitting job
Cancel job

Business Glossar

Hiring cost
payroll

Quitting job
Cancel job

Resource
Knowledge
Skills

Master data
Employee

Customer Care / Sales

skills required for customer projects

Fr. Müller

Resource
Knowledge
Skills

Management

Employee structure,
male female Number

Hr. Bauer

Master data
Employee

Define several routes between the physical objects and the Business Glossar

Business Glossar

Hiring cost
payroll

Quitting job
Cancel job

Resource
Knowledge
Skills

Master data
Employee

Different Tags

Linking objects

Machine Learning based Recommendations

Businessname

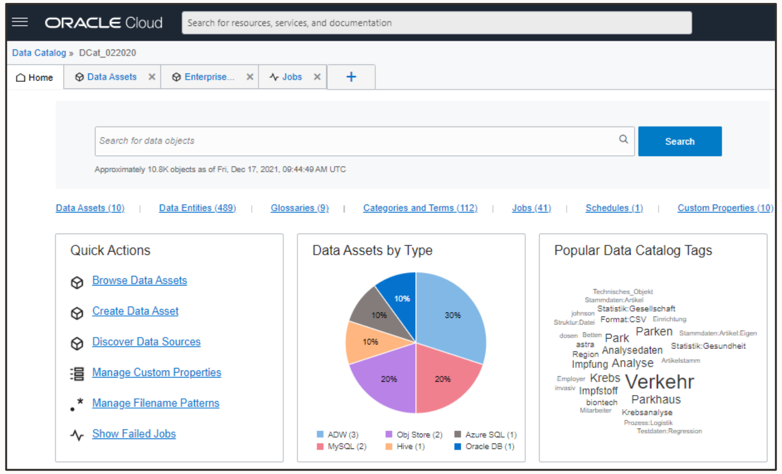
Alias

Synonyms

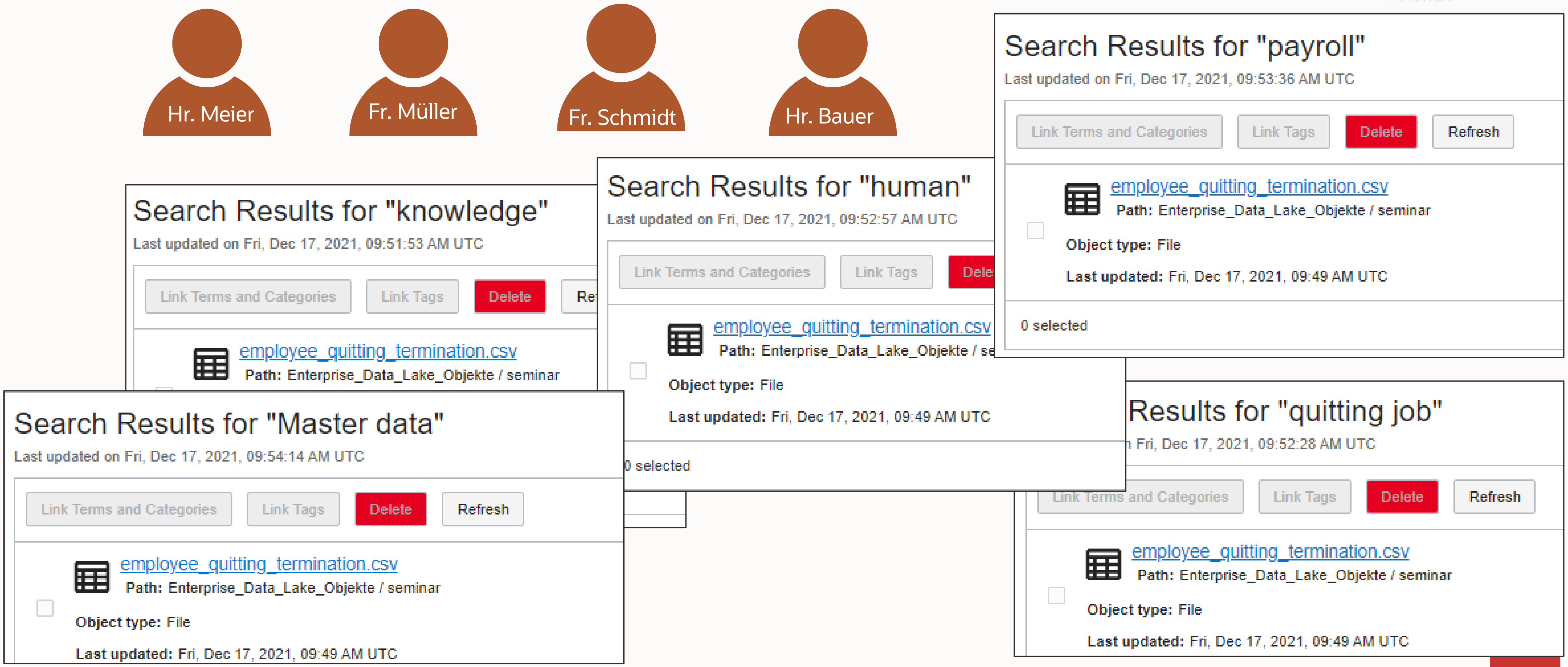
User defined Attributes

Employees
+
40 Columns

Data Catalog

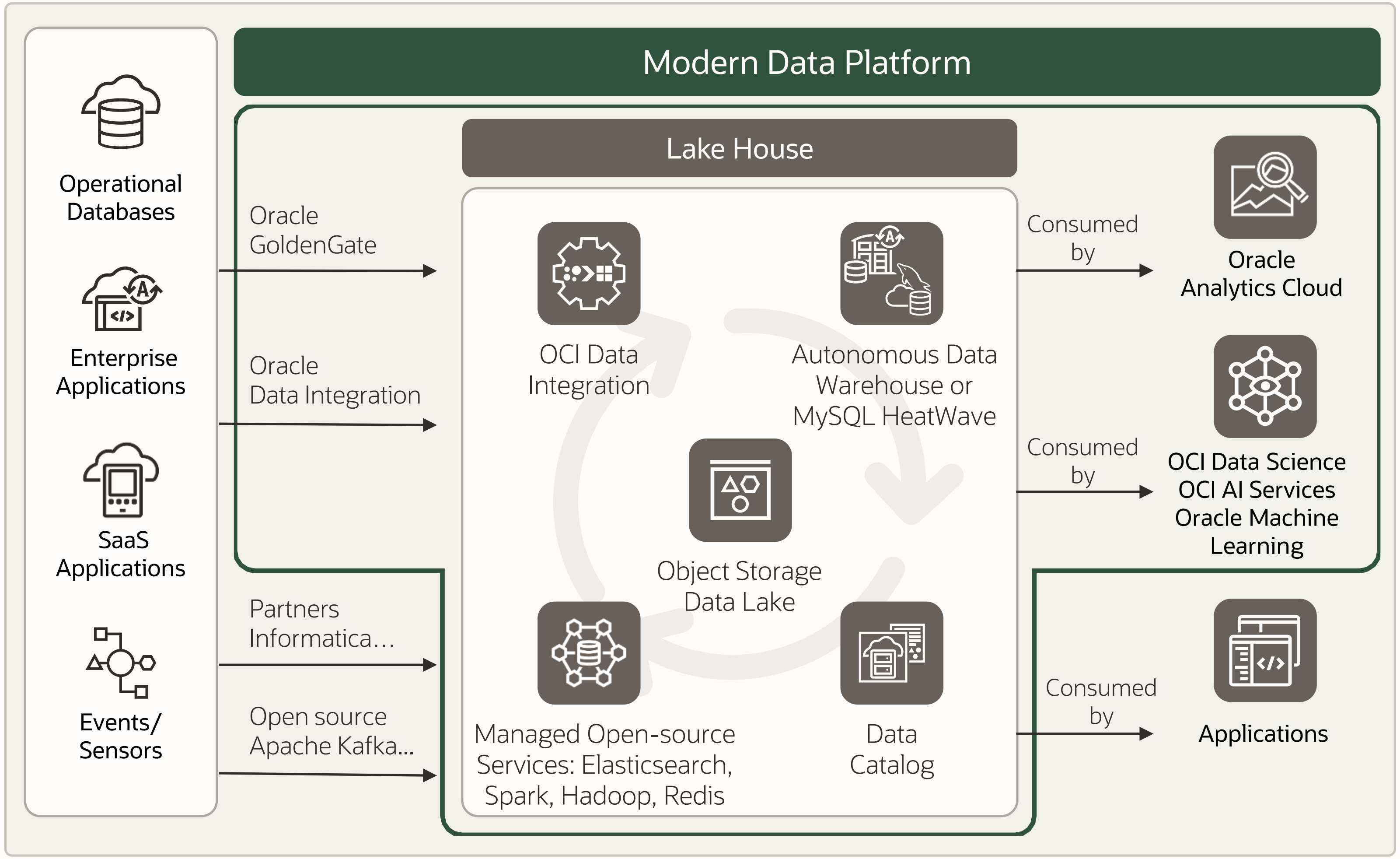


And you have it:
All people will find the same information but with different search - strategies



The “Oracle Lake House”

Oracle Cloud Infrastructur (OCI) to build a modern Data-Platform



Autonomous Data Warehouse: Autonomous high-performance Database for Analytics

MySQL Heatwave: high-performance Analytics for MySQL – Apps

Object Storage Data Lake: Low-Cost Storage

Managed Open-source Services: Classical Hadoop Tools (Spark, Hadoop, Elasticsearch, Redis)

OCI Data Integration: Extract, Transform, Load, Declarive rules. Move data between Data Lake and DWH

OCI Data Catalog: Business driven Data Discovery for Enduser and Data Scientists in the whole environment

Thank you

Alfred.Schlaucher@oracle.com