

Herzlich Willkommen!



Jens Kröhnert

Principal Solution Architect
j.kroehnert@oraylis.de



Daniel Esser

Senior Consultant - Solution Architect
d.esser@oraylis.de

AZURE MEETUP #34

Struktur und Automatisierung – wie ihr ein leistungsfähiges MDWH aufbaut



1

Kurze Vorstellung

2

(Modern) Data Warehouse - Warum

3

Historie des DWH und seinen „Erfindern“

4

Das Beste aus allen Welten in einer Schichtenarchitektur



5

Aufbau eines Data-as-a-Service-Ansatzes im Consulting

6

Domain-Template-Driven Datalake Design mit Azure Synapse

7

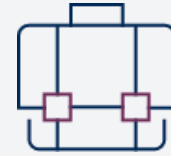
Template Driven Database Design mit Azure Synapse

ORAYLIS – Ihr Guide zur Data Driven Company



116 Mitarbeiter

davon 95 Experten, die für BI,
Data Analytics & AI brennen



22 Jahre

prall gefüllter Erfahrungsschatz



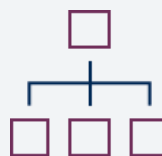
16 Jahre

Microsoft Partner mit 4
Gold-Kompetenzen



800+ Projekte

in den Bereichen BI, Data Analytics
& KI erfolgreich umgesetzt



100 Prozent

inhabergeführt und durch
flache Hierarchien geprägt



15 Prozent

Wachstum
pro Jahr



100 Prozent

Weiterempfehlung bei
kununu



3 Auszeichnungen

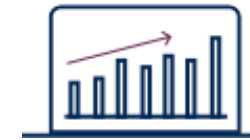
Im 3. Jahr in Folge sind wir ein
Great Place to Work

Gestalten Sie mit Ihren Daten die Zukunft

Unsere Lösungen für
Ihren Weg zur Data Driven
Company.



Datenstrategie



Business Intelligence &
Data Analytics



Künstliche Intelligence



Internet of Things



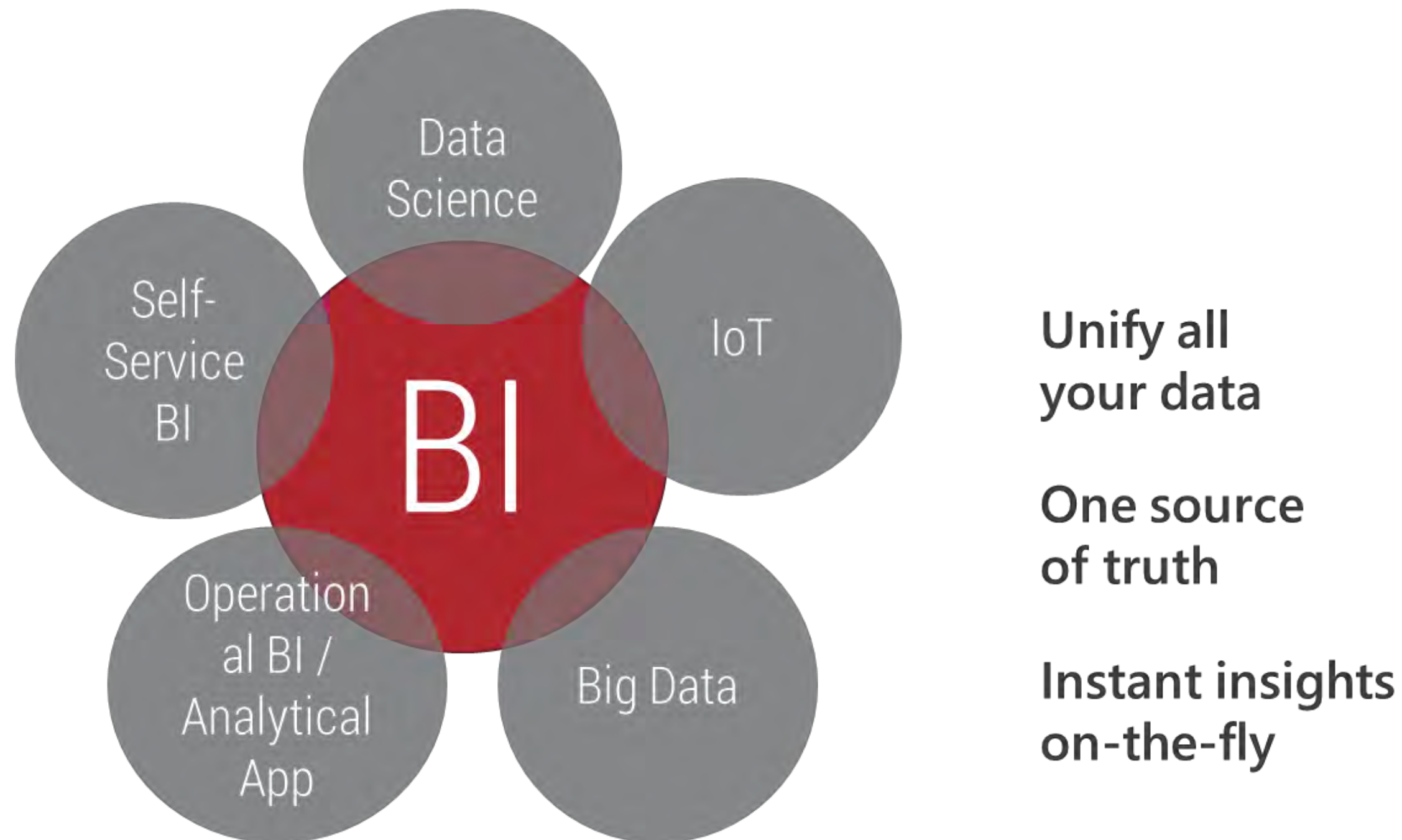
Big Data Engineering



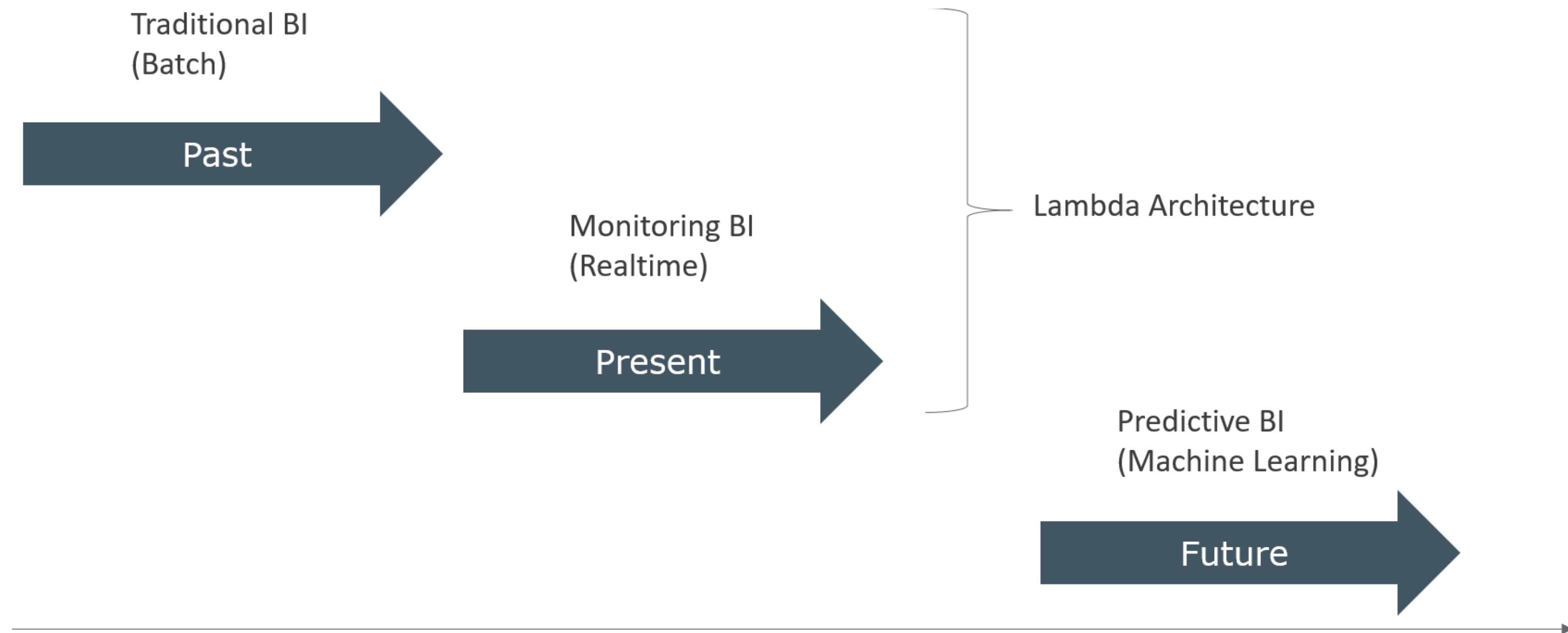
Projektmanagement

Mit Daten ist heute
machbar, was gestern nur
denkbar war.

Cloud – Basis für die Digitalisierung



From Realtime to Predictive Enterprise



Digital Feedback Loops sind die Basis der Data Driven Company



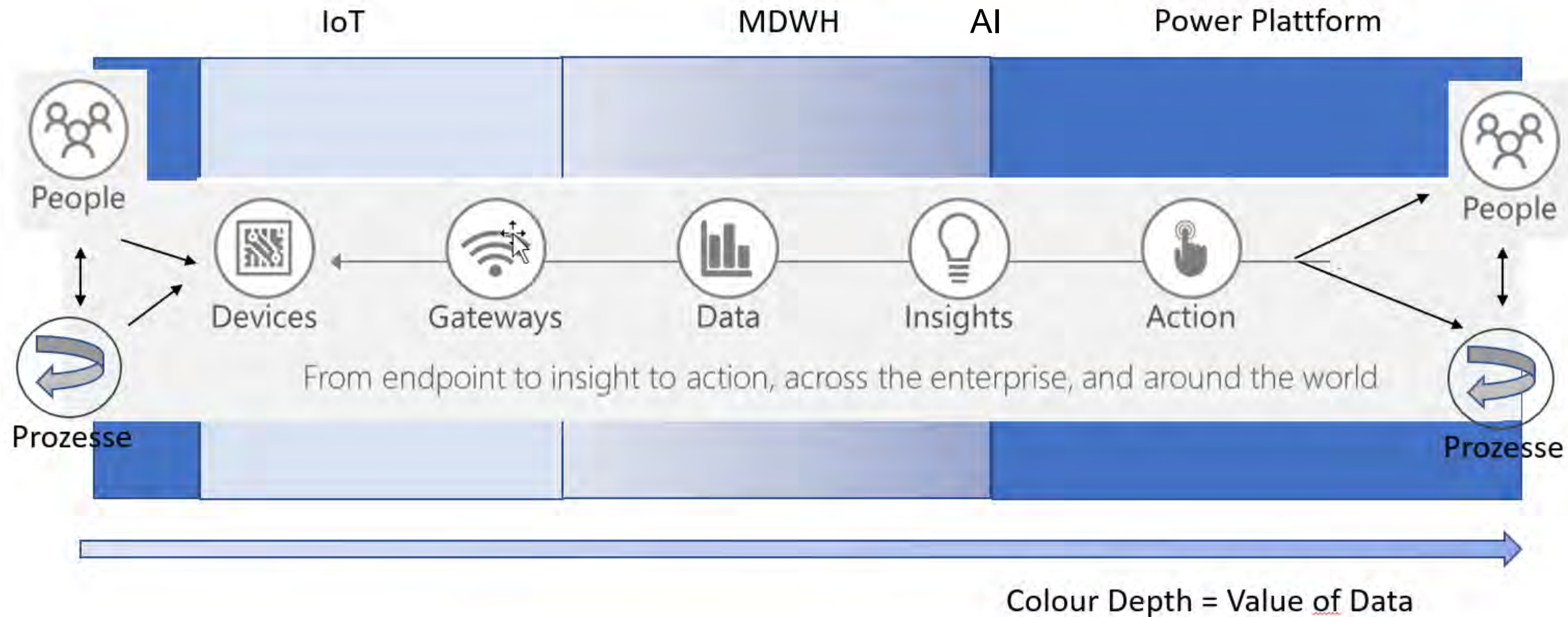


DATA DRIVEN COMPANY WHITEPAPER

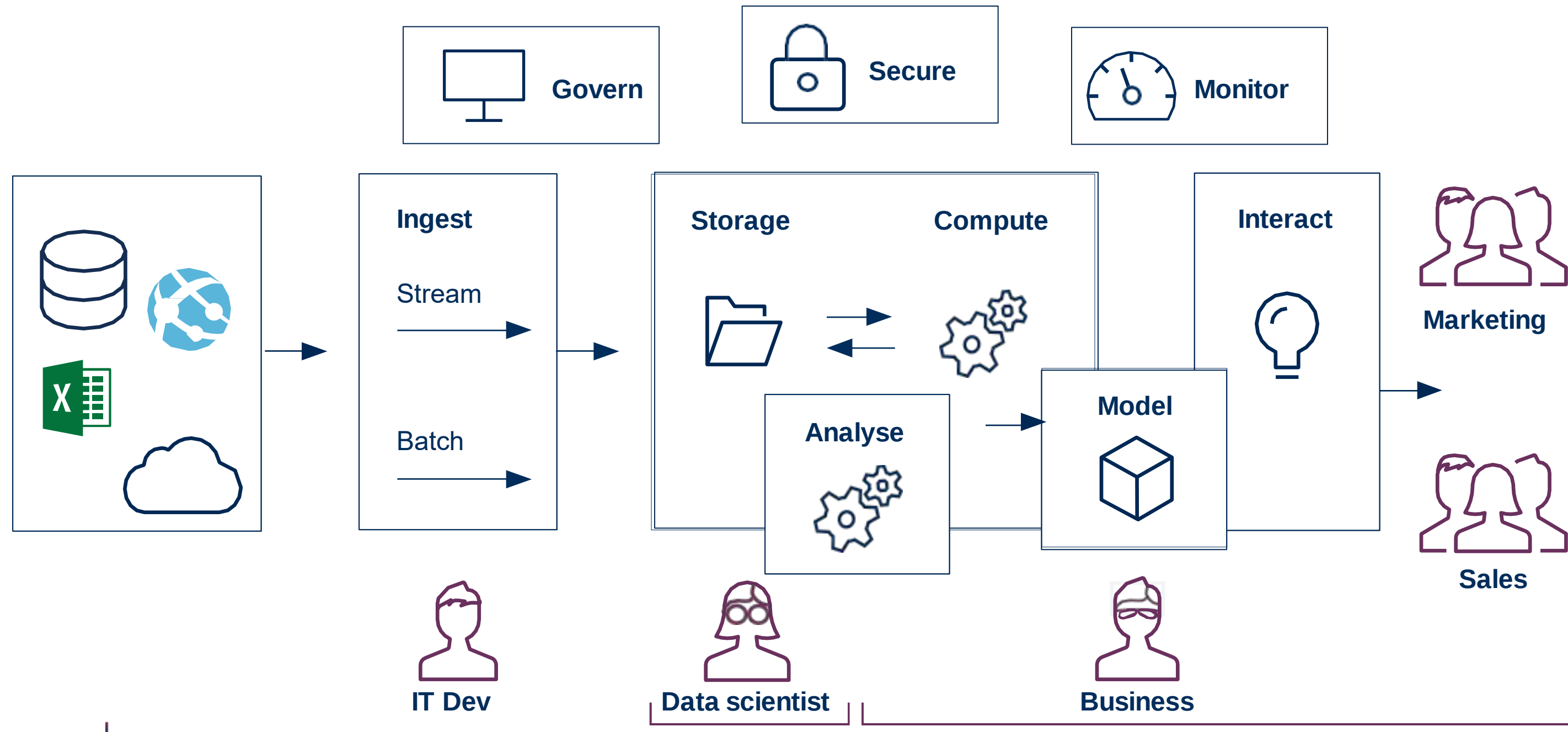
In 6 Schritten zur Data Driven Company



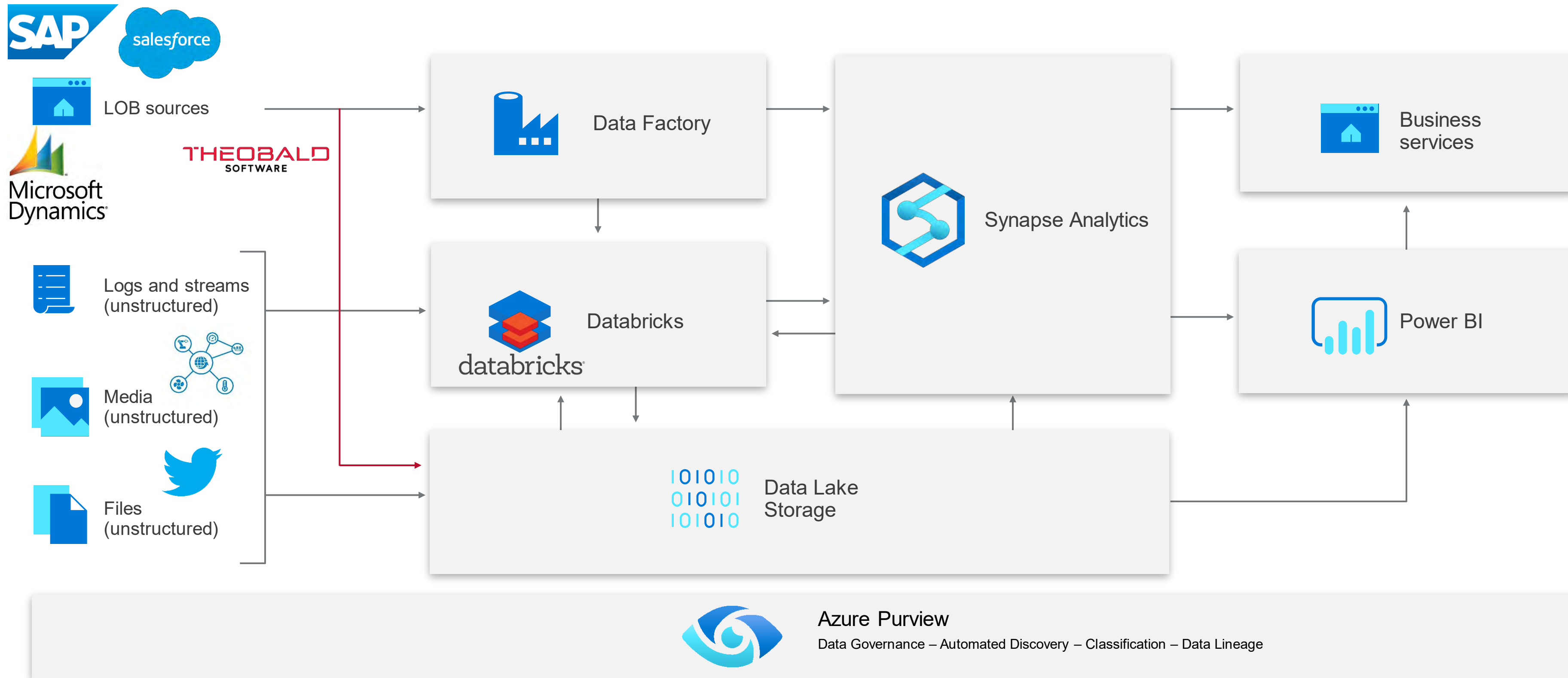
Digital Feedback Loops sind die Basis der Data Driven Company



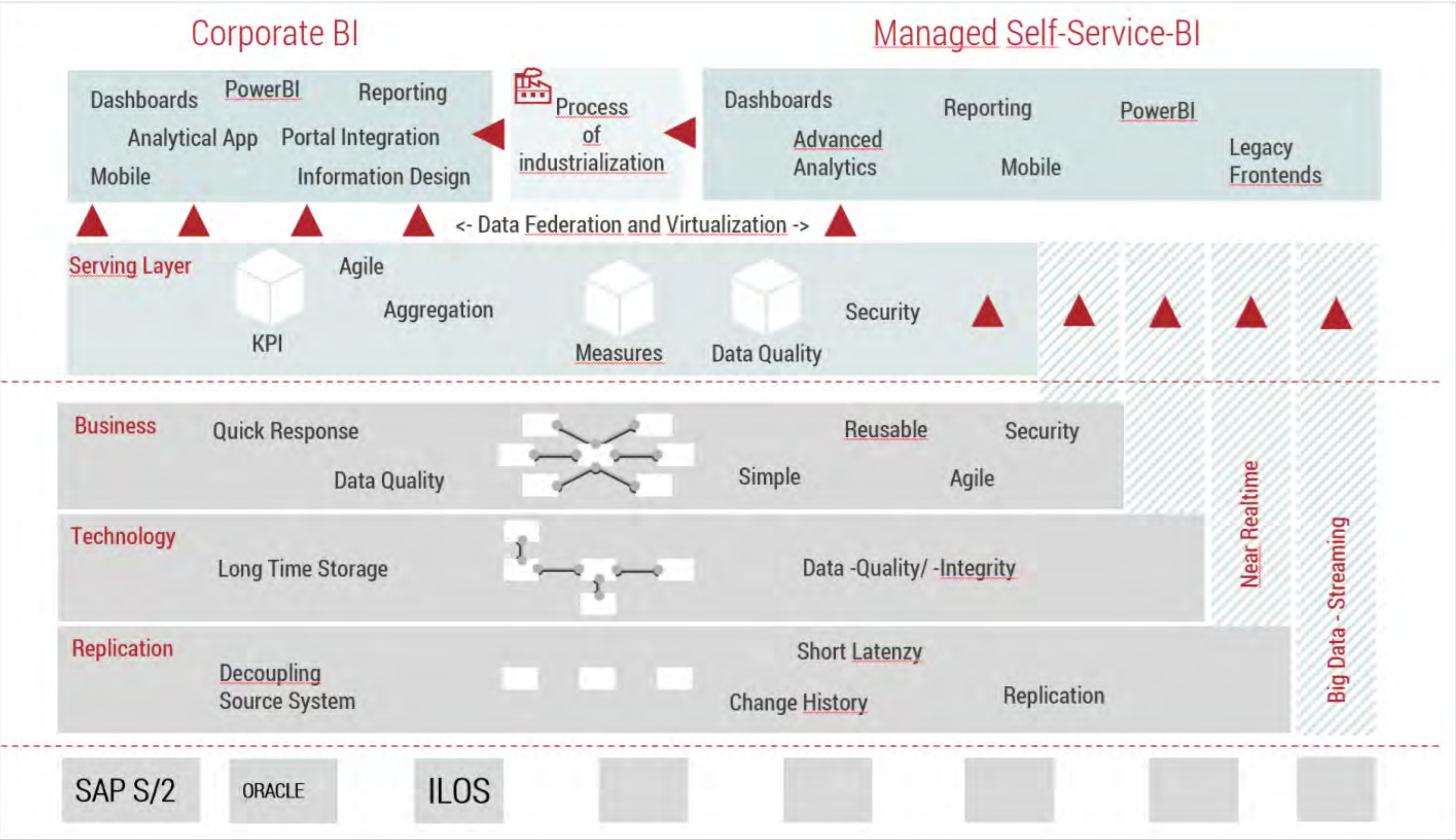
Modern Data Warehouse – Cloud und Data-Lake basiert



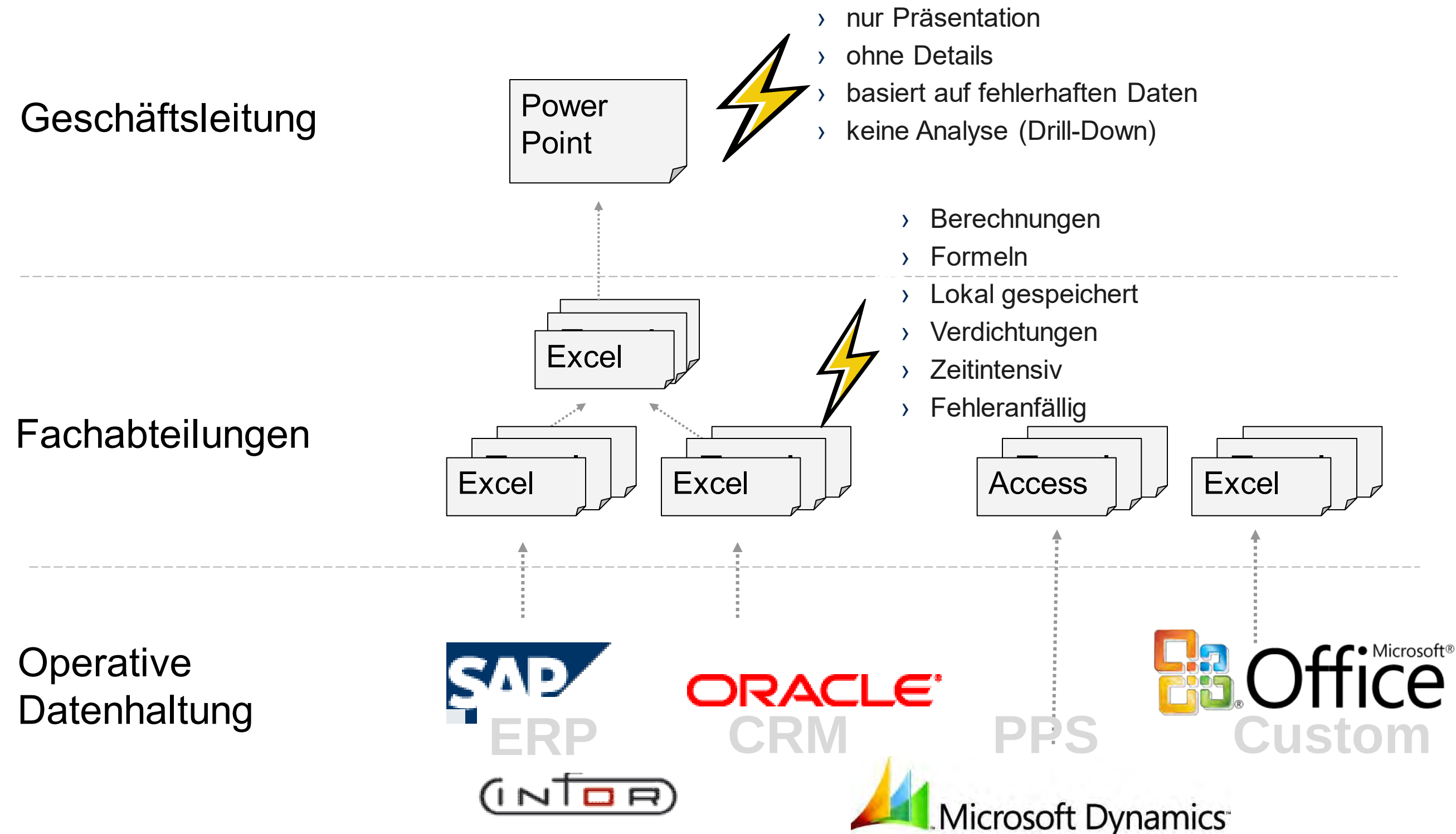
Modern Data Warehouse – Zielarchitektur



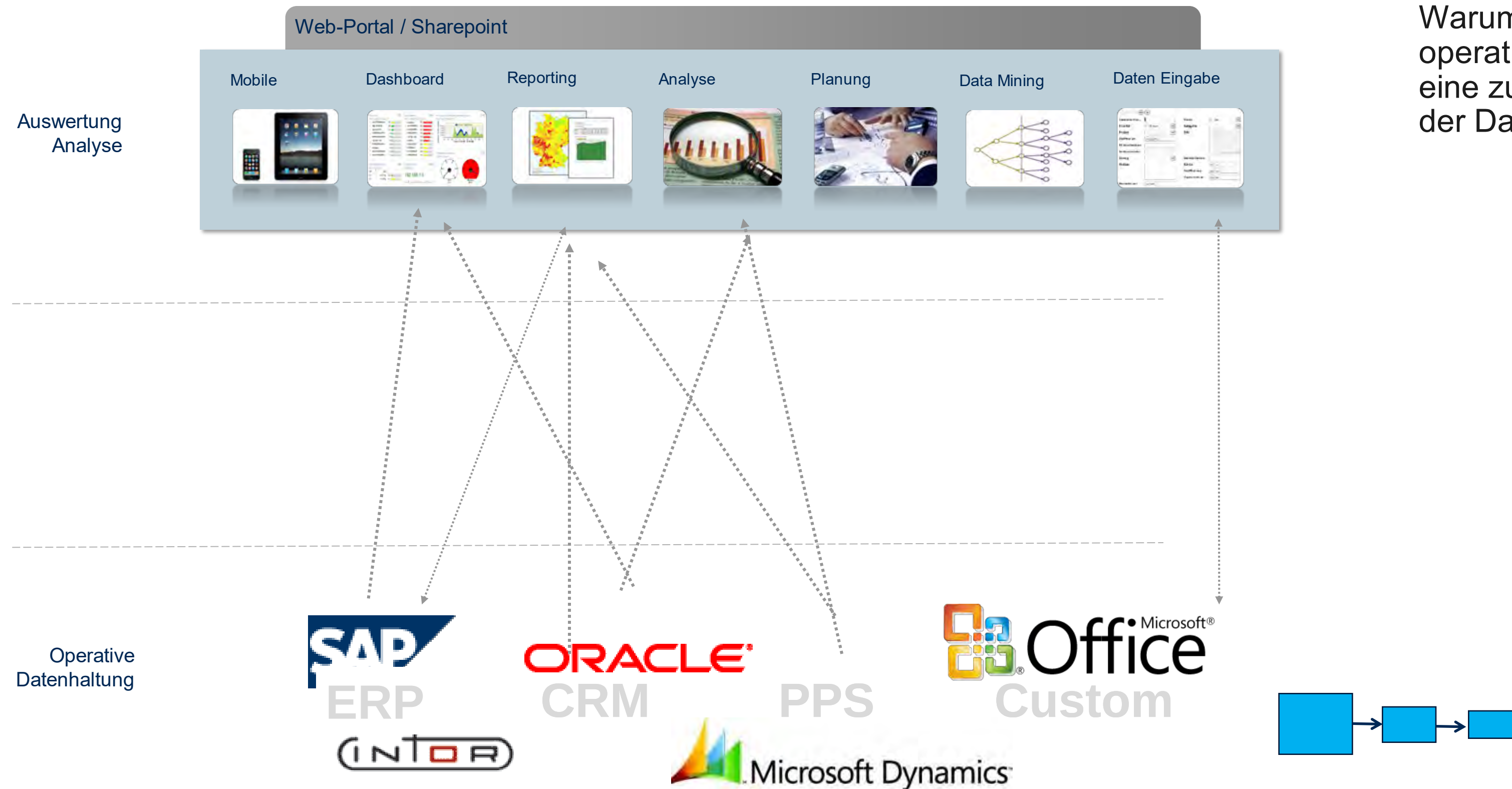
Architektur: DWH Schichten



BI manuell ohne BI Plattform

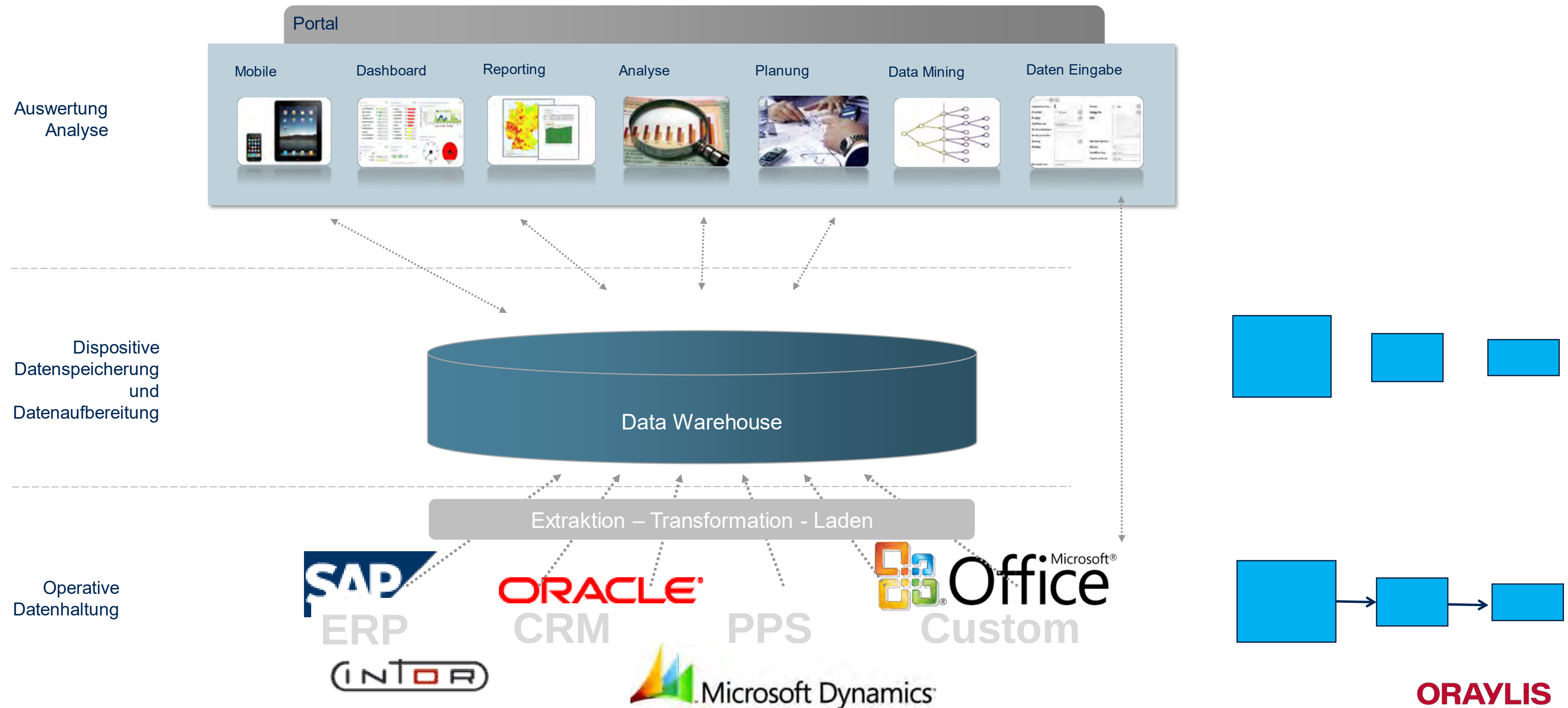


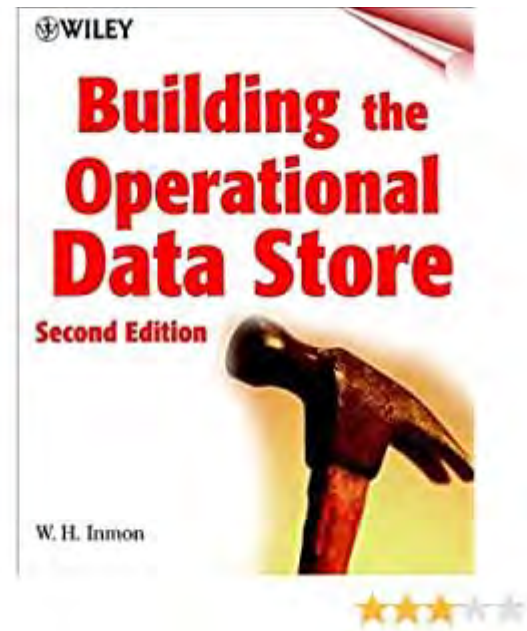
Standard Architektur BI – „Single Version of Truth“ – Situation o. BI



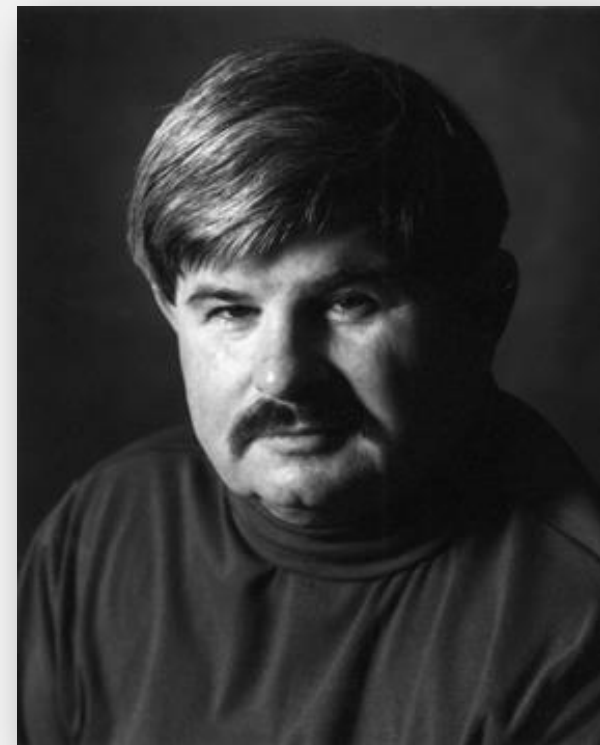
Warum macht neben den operativen Systemen eine zusätzliche Schicht der Datenhaltung Sinn?

Standard Architektur BI – „Single Version of Truth“ – Initialer Ansatz: Quellen kopieren





ODS
Core DWH
Meta-ERP



- › 1992 — Building the Data Warehouse
- › „Father of Data Warehousing“

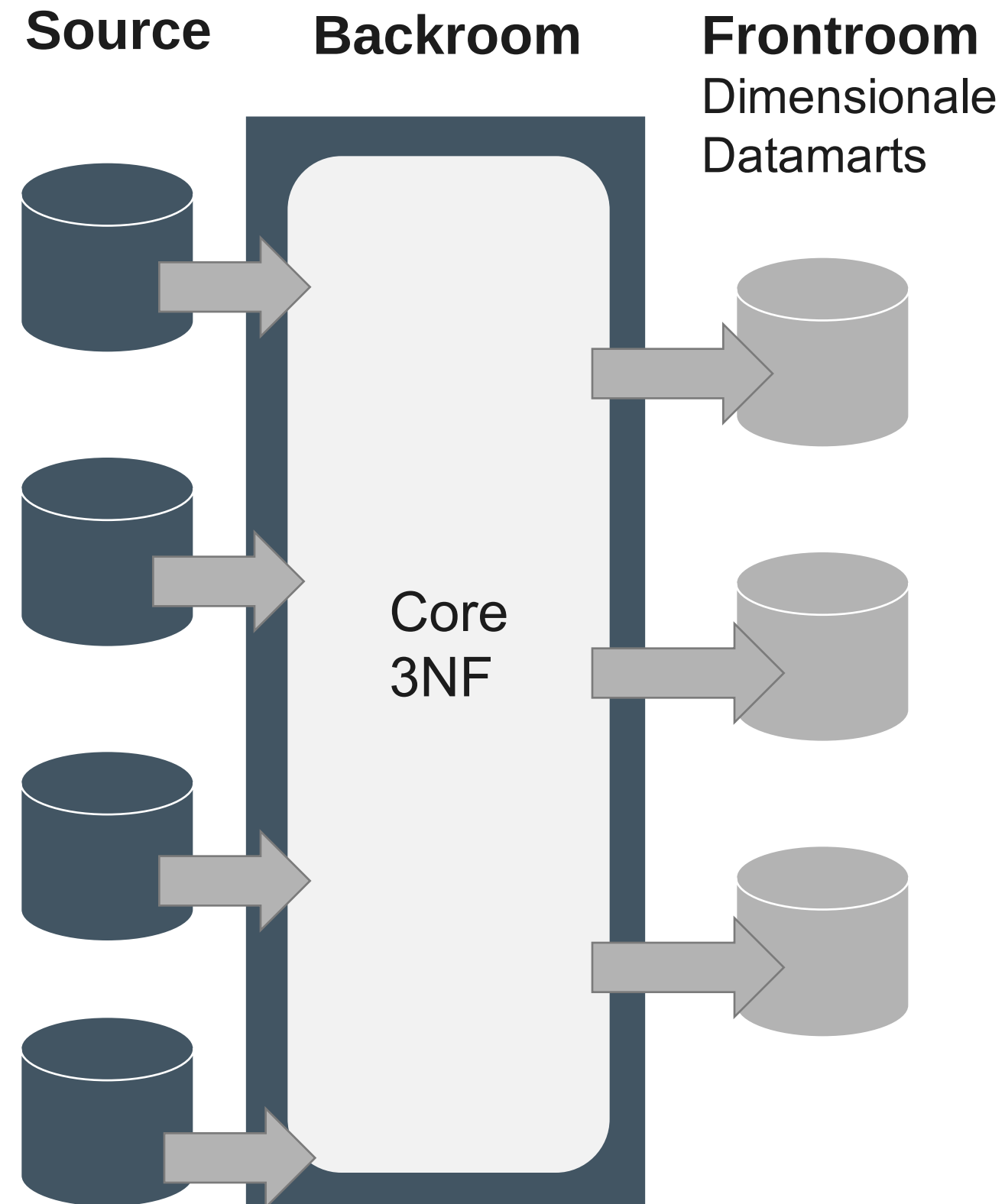
Enterprise Data Warehouse nach Bill Inmon

› Philosophie

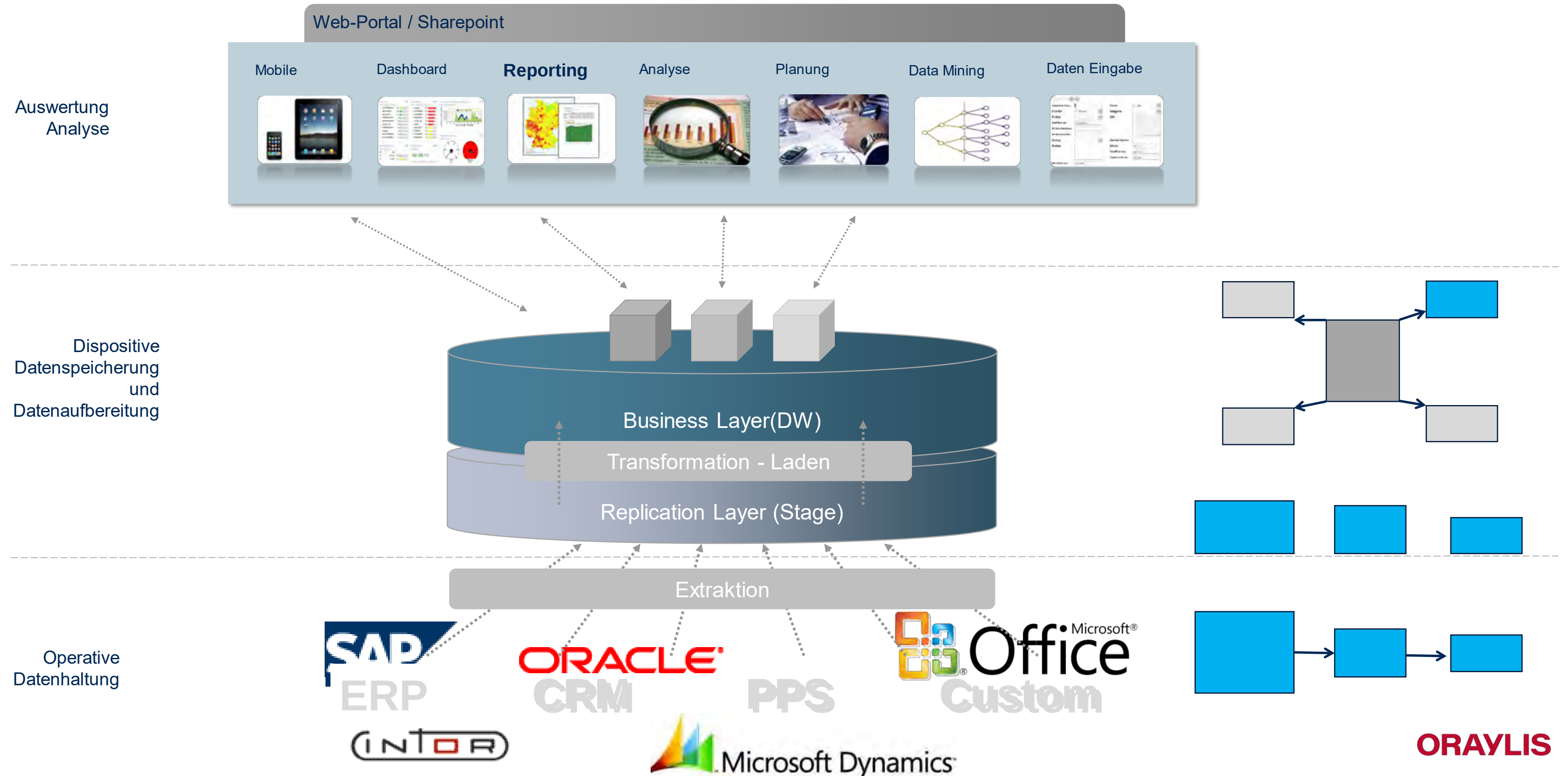
- › Ganzheitlich - Unternehmensweit
- › Modellierung des Single Point of Truth
- › Bottom-Up-Vorgehen

› Eigenschaften

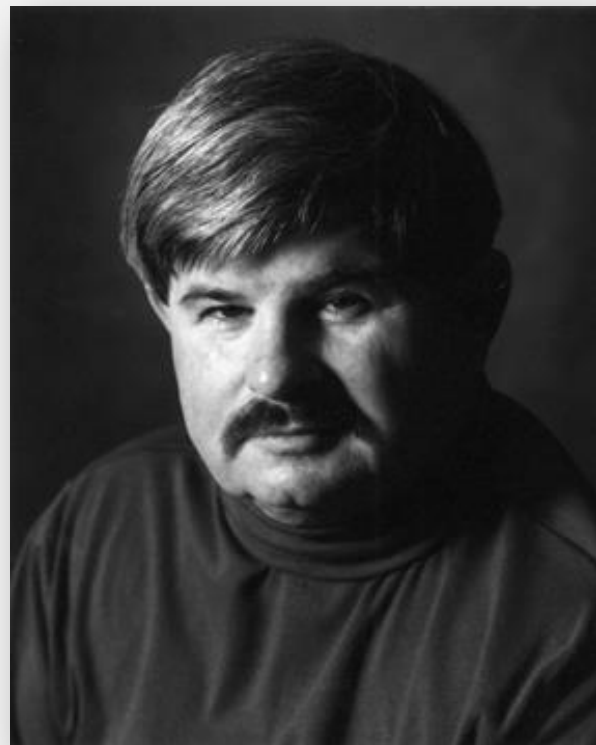
- › Quellgetrieben
- › Historische Daten
- › Integriert durch Enterprise Data Modell
- › Vollständig
- › Starke Abhängigkeiten im Datenmodell
- › Viele Data Marts



Standard Architektur BI – „Single Version of Truth“



Bill Inmon



- › 1992 — Building the Data Warehouse
- › „Father of Data Warehousing“

Ralph Kimball



- › „Father of Dimensional Modelling“
- › 1996 — The Data Warehouse Toolkit

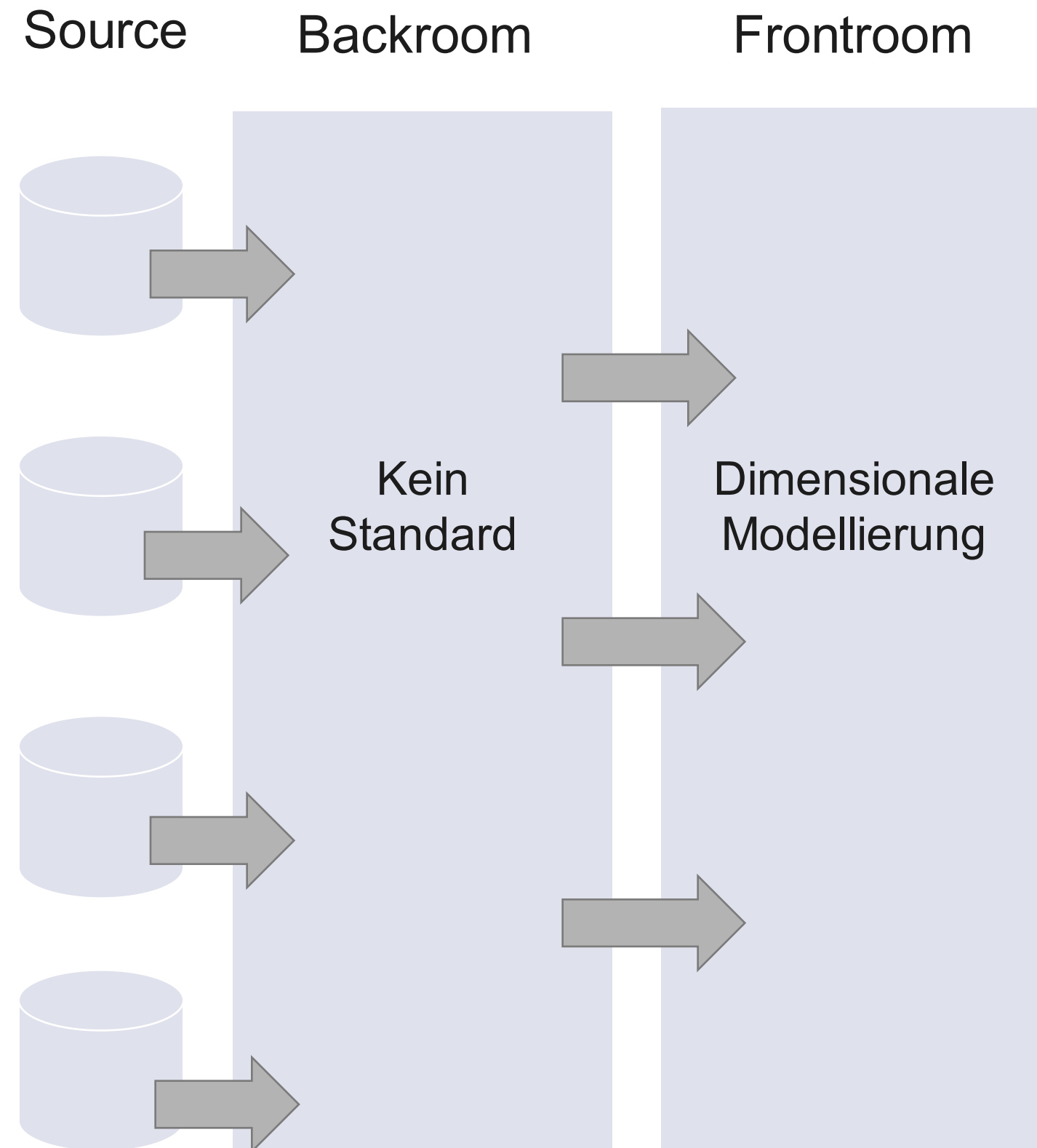
Dimensionales Data Warehouse nach Ralph Kimball

› Philosophie

- › Iterativer Ansatz
- › Schnelle Lieferung
- › Der Nutzer steht im Vordergrund
- › Top-Down-Vorgehen

› Eigenschaften

- › Geschäftsprozess orientiert
- › Anforderungsgetrieben
- › Analyseoptimiert
- › Integration durch konforme Dimensionen
- › Ein oder sehr wenige Data Marts



Inmon und Kimball im Vergleich

Bill Inmon

› Vorteile

- › Vollständig integriert
- › Der kleinste Grain der Daten liegt vor
- › Historische Daten

› Nachteile

- › Hoher Analyseaufwand zu Beginn
- › Hohe Kosten bei Änderungen
- › Agiles Vorgehen ist kaum möglich

Ralph Kimball

› Vorteile

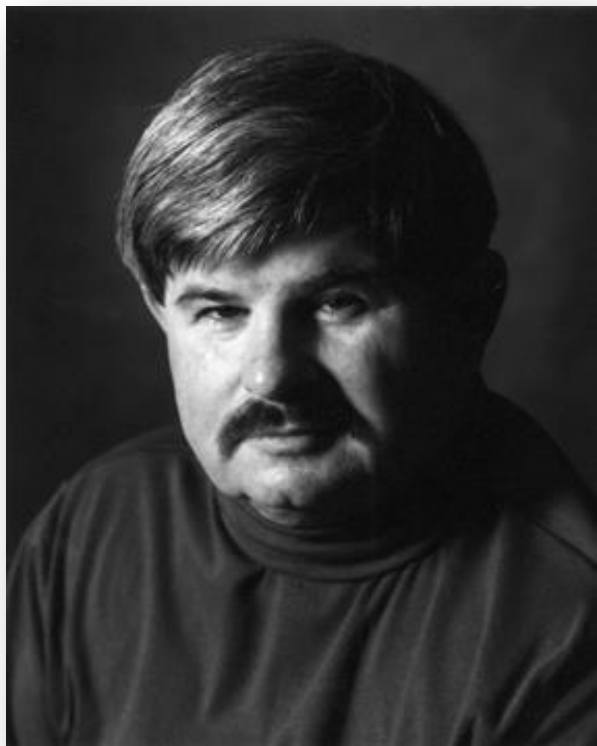
- › Abfrage- und Analyseoptimiert
- › Agiles Vorgehen im Frontroom
- › Einfaches Modell - gut zu verstehen
- › Hohe Nutzerakzeptanz

› Nachteil

- › Kein Standard im Backroom
- › Historische Daten in Dimensionen erhöhen die Komplexität

Inmon und Kimball im vergleich

Bill Inmon



Beide Ansätze sind nicht darauf ausgelegt ein Wachstum dauerhaft zu ermöglichen.

Ralph Kimball



Herausforderung	Score
Skalierbarkeit	
Agilität	
Compliance	
Komplexität	

Herausforderung	Score
Skalierbarkeit	
Agilität	
Compliance	
Komplexität	

Data Vault – Skalierbare Modellierung und Vorgehensweise

- › Was sind die Ziele bei der Data Vault Vorgehensweise
 - › Skalierbarkeit des Data Warehouse
 - › Simplifizierung der Quellenbindung
 - › Standardisierung der Datenintegration
 - › Agile Vorgehensweisen
 - › Revisionssicherheit
 - › Reduzierung von Aufwänden durch Refactoring
- › Was ist Data Vault nicht?
 - › Eine Big Data Technologie
 - › Optimiert für Abfragen
 - › Ist nicht von einer Plattform abhängig

Data Vault steht nicht in Konkurrenz zu dimensionalen Modellierung!

Dan Linstedt



- › „Inventor of the Data Vault“
- › 2010 - Supercharge your Data Warehouse

Data Vault – Reduzierung von Abhängigkeiten

› Trennung der Attribute



Identifizierende Attribute (Business Key)

› identifizieren eine Entität eindeutig

Relationale Attribute

› Beziehungen zu anderen Entitäten

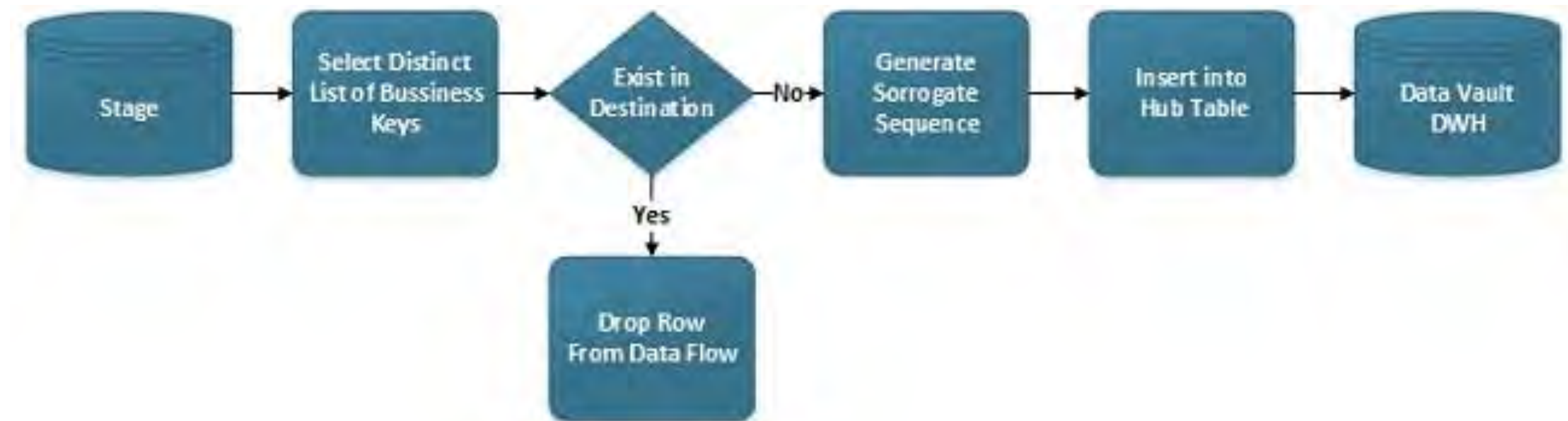
Deskriptive Attribute

› Beschreibende Kontextinformationen



Data Vault – standardisierte Ladeprozesse

Hub Load Pattern

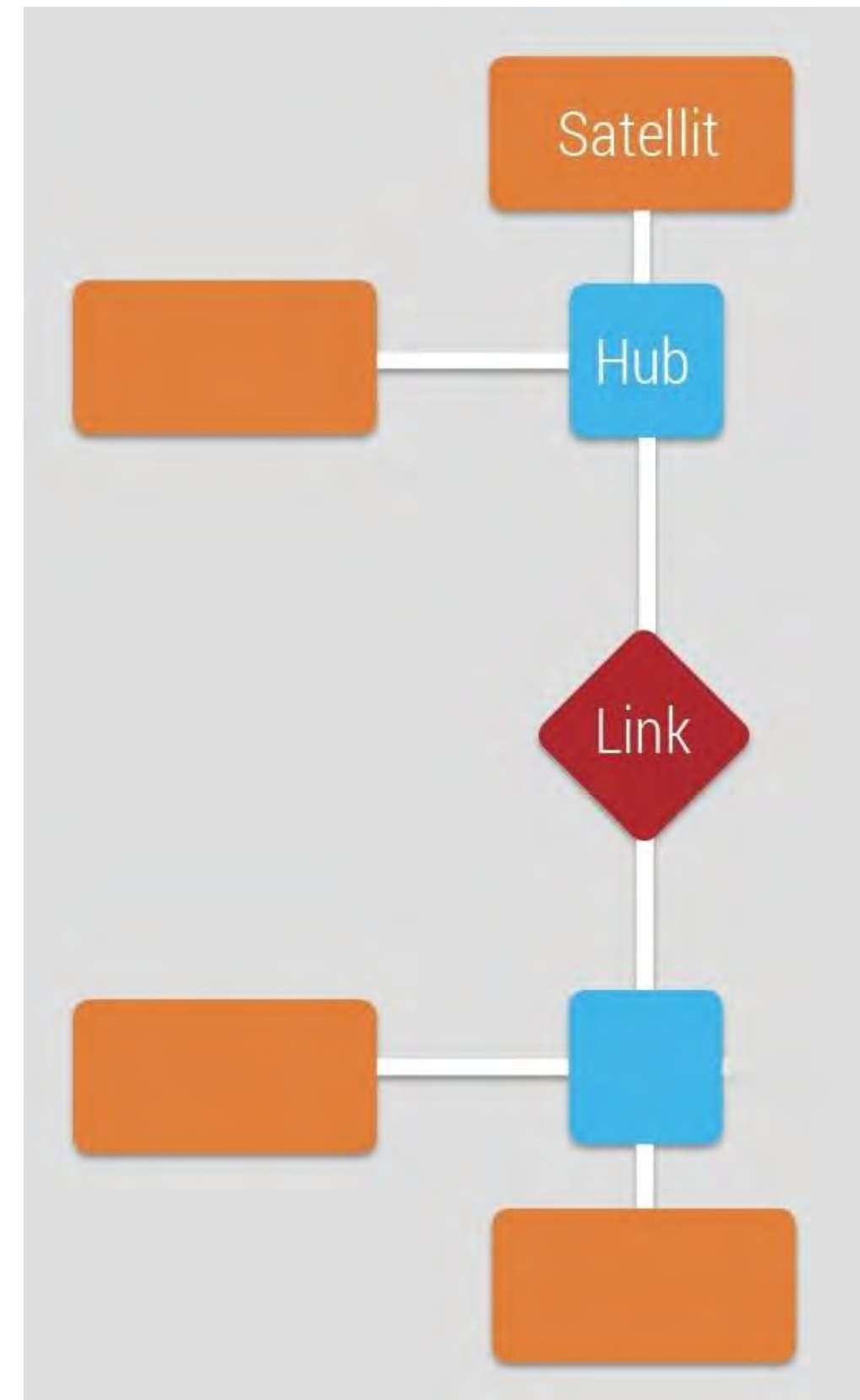


› Vorteile

- › Probleme werden nur einmal gelöst
- › Einfache Generierung von ETL-Code
- › Skalierbare Personalplanung
- › Erleichterte Off-Shore-Entwicklung
- › Unterstützte Standards
- › CMMI
- › Six Sigma
- › TQM

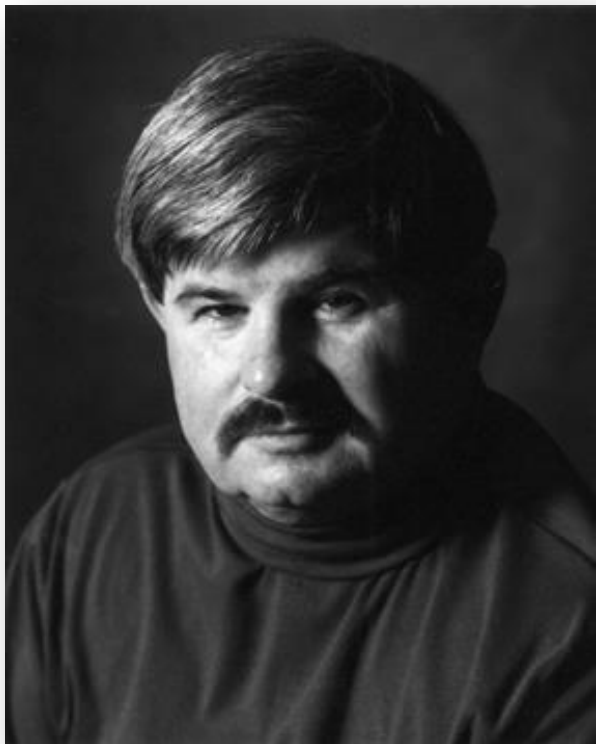
Data Vault – Modellierung

- › Hubs
 - › Distinkte Liste von Business Keys
 - › Ein Hub wird von min einem Satellit referenziert
- › Links
 - › Bilden Beziehungen zwischen Entitäten ab
 - › Links referenzieren mindestens 2 Hubs
 - › Auch andere Link-Tabellen können referenziert werden
- › Satelliten
 - › Satelliten referenzieren Hub- oder Link-Tabellen
 - › Historisierung der Entität oder Beziehung



Data Vault - das beste aus beiden Welten

Bill Inmon



Dan Linstedt



Ralph Kimball



Herausforderung	Score
Skalierbarkeit	
Agilität	
Compliance	
Komplexität	

Herausforderung	Score
Skalierbarkeit	
Agilität	
Compliance	
Komplexität	

Herausforderung	Score
Skalierbarkeit	
Agilität	
Compliance	
Komplexität	

Layer Architecture

SL – Serving Layer

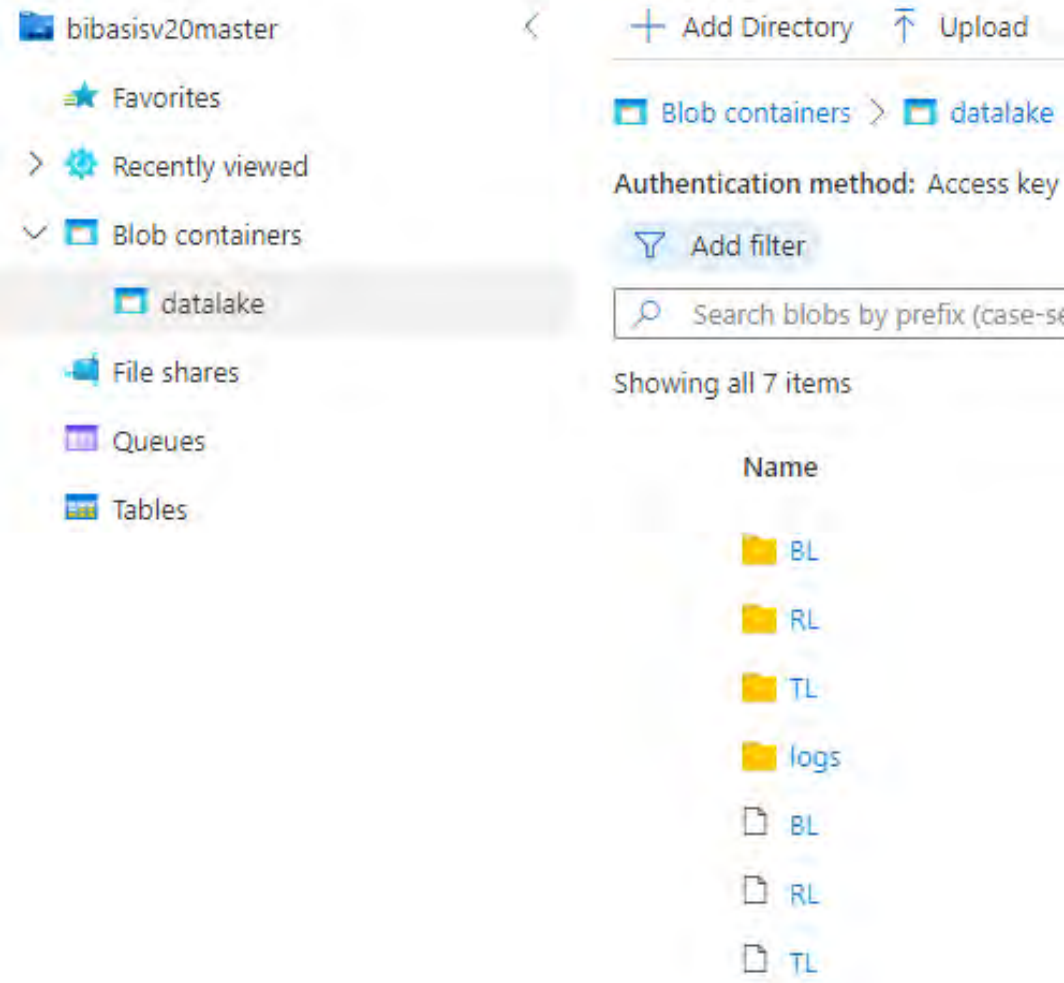
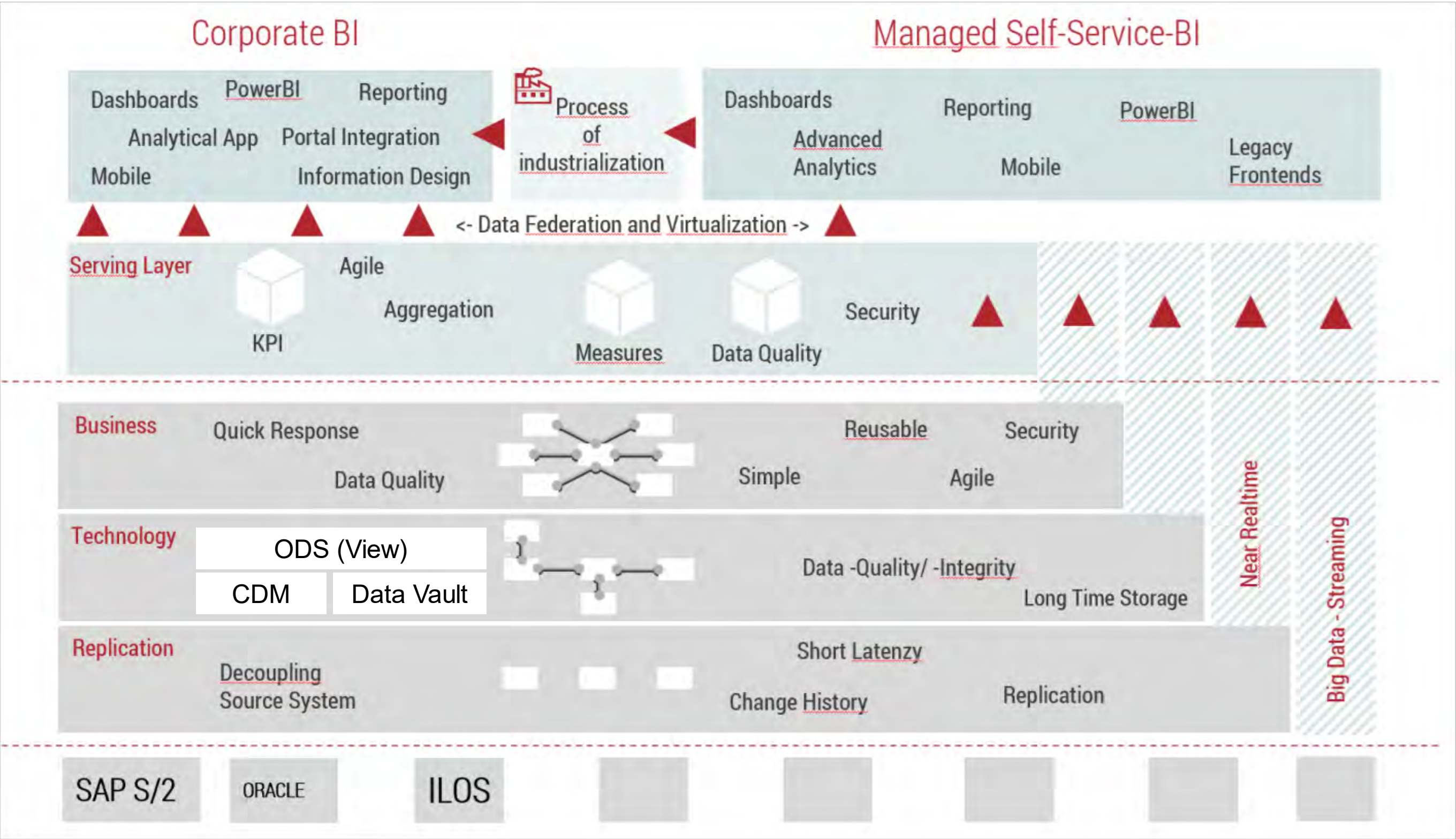
BL – Business Layer

TL – Technology Layer

RL – Replication Layer



Trennung der Schichten durch Datenbanken oder Ordner im Lake





5

Aufbau eines Data-as-a-Service-Ansatzes im Consulting

6

Domain-Template-Driven Datalake Design mit Azure Synapse

7

Automatisierung im Data Warehousing

Was ist überhaupt Data-as-a-Service?



”

Suppose you could find all the socks you ever lost.

Now suppose getting those socks back enabled you to earn a better living, or work faster and smarter.

Wouldn't you be willing to pay someone to locate those socks on a worldwide sock exchange?

”



Welchen Einfluss hat das auf die Beratung?

“Automation, digital platforms, and other innovations are changing the fundamental nature of work. Understanding these shifts can help policy makers, business leaders, and workers move forward.”

Stimmungsbild aus der Wirtschaftswissenschaft

“Consulting firms aren't just recommending software anymore, they're building it.”

“Management consulting firms like McKinsey and Booz Allen are upending the traditional consulting model and building their own software.”



Soren Kaplan

Affiliate Research Scientist

"As you look at how to scale a business, and you look at how to scale offerings, if it's people-based, it's not as scalable as if it's software-based," Kaplan said.

[protocol](#), “[Consulting Companies Building Software](#)” by Aisha Counts, November 4, 2021

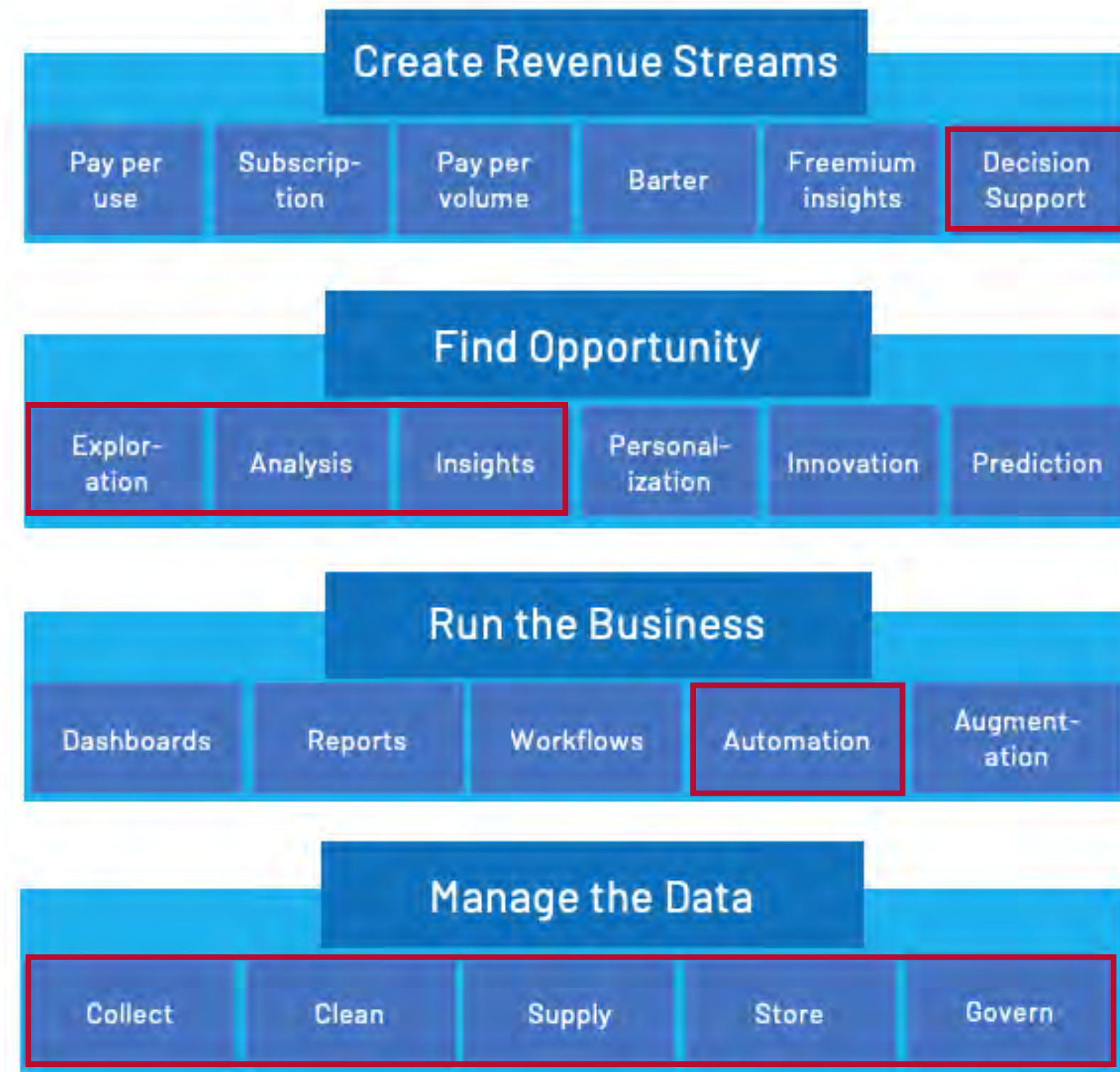
Aber wir bauen doch alle schon längst
Software!

As-a-Service? Als Produkt?

Skalierbar? Pay-per-use?

Als Teil unseres Businessmodells?

Data Value Chain



- Es wird günstiger für den Kunden
- Wir bringen einen Value-Added-Service mit
- Wir haben mehr Zeit über Innovation und Daten-Strategie zu sprechen



5

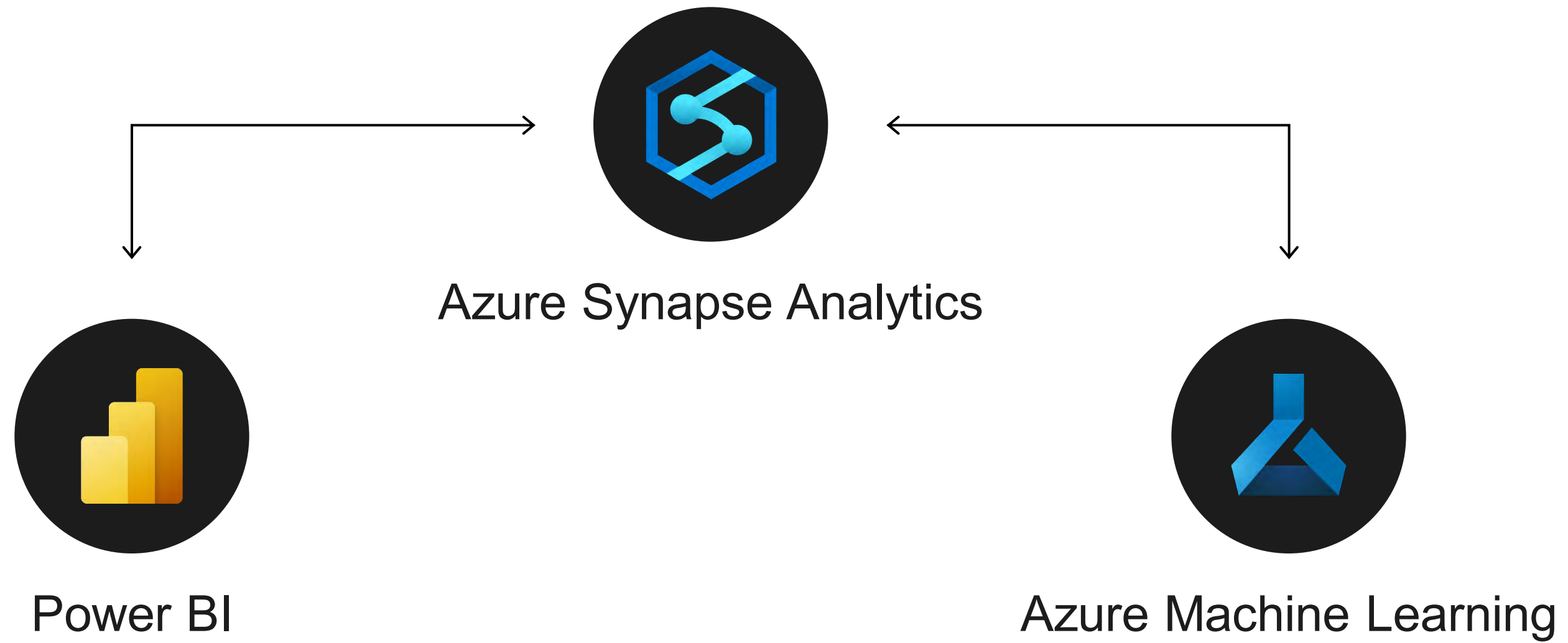
Aufbau eines Data-as-a-Service-Ansatzes im Consulting

6

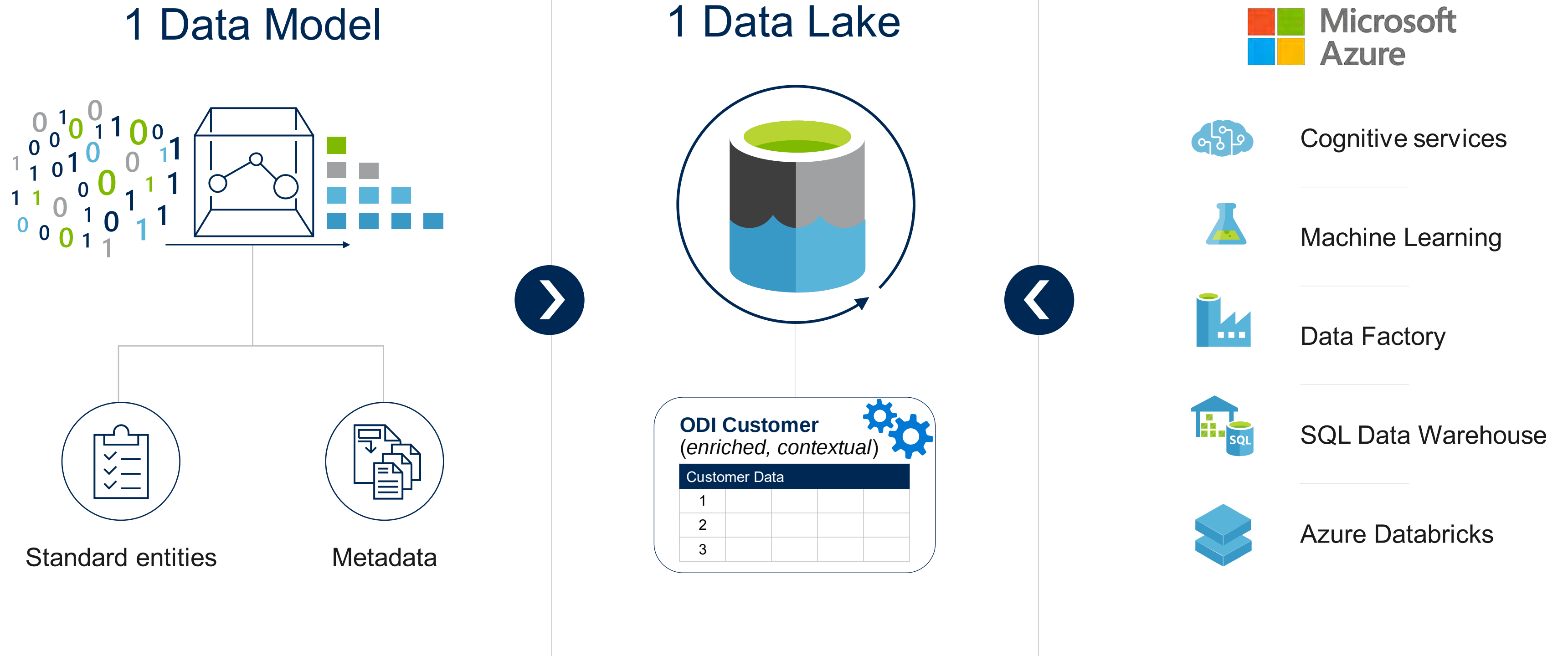
Domain-Template-Driven Datalake Design mit Azure Synapse

7

Automatisierung im Data Warehousing

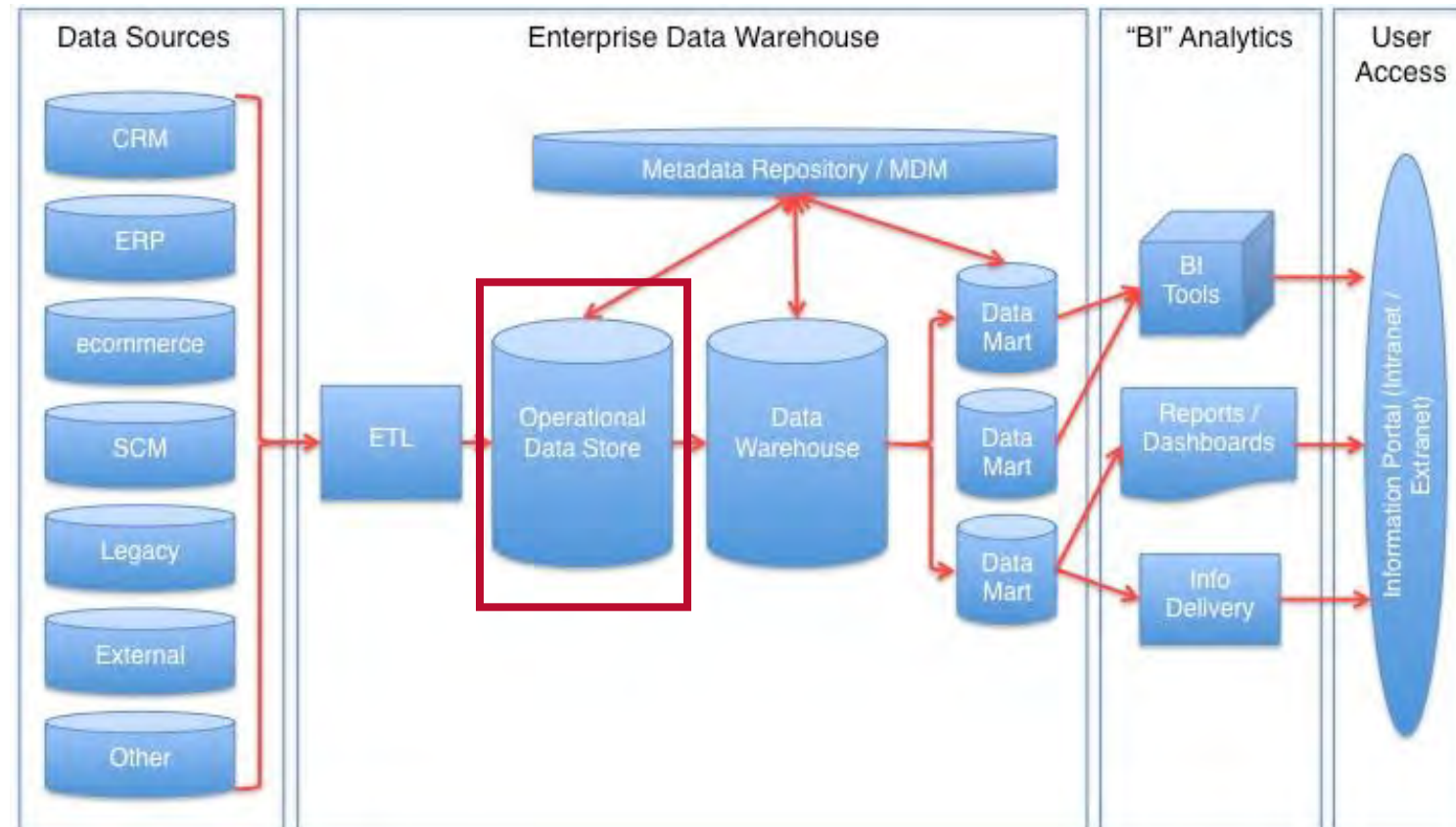


What are the components of ODI?



Template-Driven-Lake-Design hält Einzug

- › Microsoft etabliert einen Template-Driven-Lake-Design-Ansatz
 - › Die Templates decken verschiedene Geschäftsbereiche und Industrien ab
 - › Stand heute sind die Templates für ein “Rapid Prototyping” beim Aufbau eines “Operational Data Store” geeignet



ODS vs. EDW

Characteristic	Operational Data Store	Data Warehouse
Primary Purpose	Run the business on a current basis	Support managerial decision making
Design Goal	Performance throughput, availability	Easy reporting and analytics
Primary Users	Clerks, salespersons, administrators	Managers, business analysis, customers
Subject-Oriented	Yes	Yes
Integrated	Yes	Yes
Detailed Data	Yes	Yes
Summary Data	No	Yes
Time of Data	Current data	Historical snapshots
Updates	Frequent small updates	Periodic batch updates
Queries	Simple queries on a few rows	Complex queries on several rows

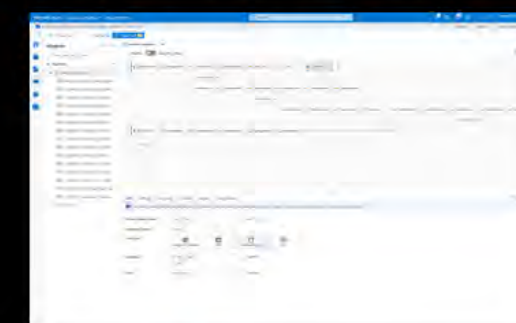


Azure Synapse compute

SQL Pools

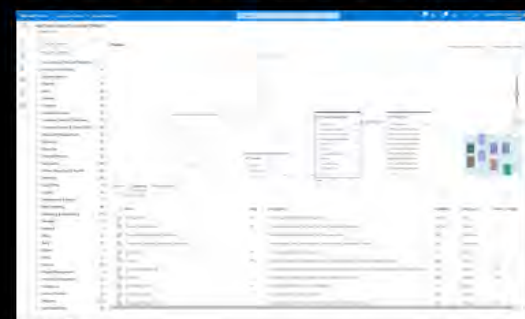
APACHE
spark

Visually create
pipeline / data flows

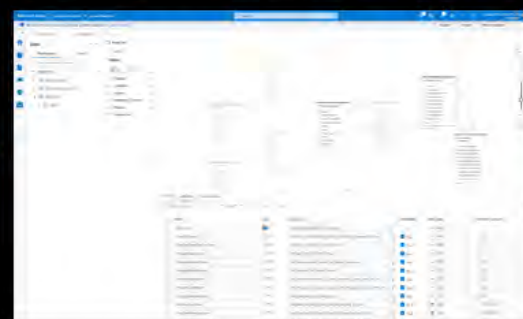


Synapse =>
Low-code/No-Code Analytics Platform (LCNC)

Browse database templates



Visually create lake database



Azure Synapse
lake database

Data sources

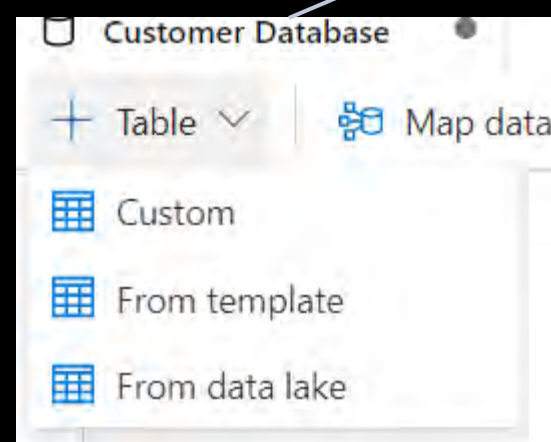
Other sources



Storage



Parquet
or CSV



Template Domains



Agriculture

For companies engaged in growing crops, raising livestock and dairy production.



Banking

For companies who are analyzing banking data.



Consumer Goods

For manufacturers or producers of goods bought and used by consumers.



Energy & Commodity Trading

For traders of energy, commodities, or carbon credits.



Freight & Logistics

For companies providing freight and logistics services.



Fund Management

For companies managing investment funds on behalf of investors.



Life Insurance & Annuities

For companies who provide life insurance, sell annuities, or both.



Oil & Gas

For companies involved in various phases of the Oil & Gas value chain.



Property & Casualty Insurance

For companies who provide insurance against risks to property and various forms of liability coverage.



Retail

For sellers of consumer goods or services to customers through multiple channels.

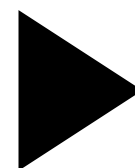


Utilities

For gas, electric and water utilities and power generators and water desalinators.

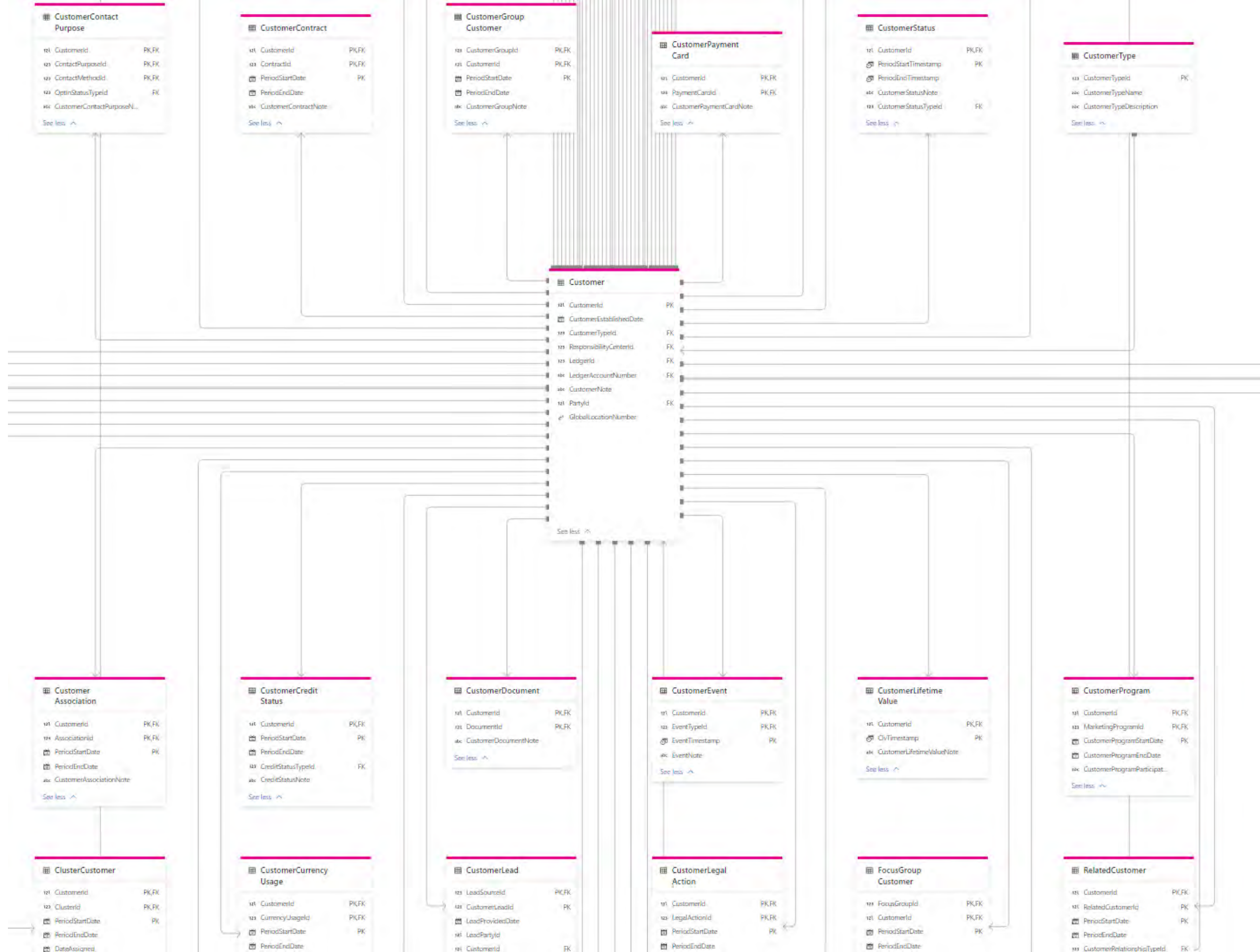
Retail-Domain

▷ Accounting & Financial Reporting	77	▷ Order	33
▷ Budget & Forecasting	42	▷ Party	217
▷ Business Metrics	4	▷ Patient	11
▷ Channel	131	▷ Policy	21
▷ Claim	55	▷ Product	535
▷ Contract	82	▷ Project Management	4
▷ Customer	103	▷ Property & Equipment	158
▷ Customer Account	91	▷ Purchasing	44
▷ Customer Credit & Collections	374	▷ Service Provider	31
▷ Customer Service & Contact Man...	158	▷ Shipping	238
▷ Document Management	57	▷ Syndicated Data	29
▷ Emissions	145	▷ Training & Education	68
▷ Encounter	133	▷ Transaction	140
▷ Financial Product	38	▷ Vendor	77
▷ Geography	220	▷ Work Order Management	5
▷ Human Resources & Payroll	291		
▷ Inventory	200		
▷ Legal Entity	11		
▷ Loyalty	54		
▷ Maintenance & Repair	6		
▷ Manufacturing	98		
▷ Marketing & Advertising	270		
▷ Member	9		
▷ Network	7		



Customer	103
☐ BenefitCriteriaType	
☒ ClusterCustomer	
☒ Customer	...
☐ CustomerActivity	
☐ CustomerAncestry	
☒ CustomerAssociation	
☒ CustomerAuthenticationMethod	
☐ CustomerAuthorizedParty	
☐ CustomerBehavioralCriteria	
☐ CustomerBiometric	

 Add related tables



Lake Database Designer

Microsoft Azure | Synapse Analytics | oraylisdemoworkspace

Synapse live | Validate all | Publish all

Customer Database

Table | Map data (Preview) | Publish

Customer

121	CustomerId	PK	
	CustomerEstablishedDate		
123	CustomerTypeId	FK	
123	ResponsibilityCenterId	FK	
123	LedgerId	FK	
abc	LedgerAccountNumber	FK	
abc	CustomerNote		
121	PartyId	FK	
abc	GlobalLocationNumber		

General | Columns | Relationships

Filter by keyword | + Column | Clone | Convert type | Delete

Name	Keys	Description		Data type	Format / Length
Standard column (9)					
CustomerId	PK	The unique identifier of	Null	121 long	
CustomerEstablishedD...	PK	The date that the	Null	date	YYYY-MM-DD
CustomerTypeId	PK FK	The unique identifier of	Null	123 integer	
ResponsibilityCenterId	PK FK	The unique identifier of	Null	123 integer	
LedgerId	PK FK	The unique identifier of	Null	123 integer	
LedgerAccountNumber	PK FK	The unique identifier of	Null	abc string	16

Properties

General | Related (0)

Choose a name for your Database. This name can be updated at any time until it is published.

Name *
Customer Database

Description
ODS for Customer

Lake database template
Retail 0.1.0

Storage settings for database

Linked service *
oraylisdemoworkspace-Workspace...

Input folder *
sapextracts/Customer Database

Data format *
Delimited text

Storage

CSV
Parquet

Map Data tpc-ds auf das Template (1/2)

New data mapping

The data mapping assistant allows you to easily map data to any of your Lake Databases. [Learn more](#)

Source type *

Azure Data Lake Storage Gen2

Source linked service *

oraylisdemoworkspace-WorkspaceDefaultStorage

Dataset type *

Parquet

Folder path * ⓘ

tpc/tpc-ds-1000-parquet/customer

Compression ⓘ

Snappy

Sources *

Filter by name

Showing 1-321 of 321 items (1 selected)

☐

Name

Preview

☐

_SUCCESS

☒

part-00000-7a679793-dbfd-4ab7-97b4-c0addc5bdab0-c0...

☐

part-00001-7a679793-dbfd-4ab7-97b4-c0addc5bdab0-c0...

☐

part-00002-7a679793-dbfd-4ab7-97b4-c0addc5bdab0-c0...

☐

part-00003-7a679793-dbfd-4ab7-97b4-c0addc5bdab0-c0...

☐

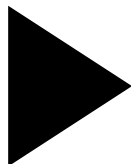
part-00004-7a679793-dbfd-4ab7-97b4-c0addc5bdab0-c0...

☐

Select multiple files for each source

Continue

Cancel



New data mapping

Data Mapping name *

Customer Mapping

Target database *

Customer Database

Map Data tpc-ds auf das Template (2/2)

Database 1

CustomerMapping

Target tables

Filter items...

Database 1

Customer

Create pipeline

Data flow debug

Customer

Primary source table *

SyntheaPatients.csv

New mapping

Additional sources

Mapping method

Source column

Target column

Direct

abc SyntheaPatients.csv@Member id

LedgerAccountNumber

Direct

abc SyntheaPatients.csv@County

CustomerNote

Data preview

Number of rows

INSERT 0

UPDATE 0

DELETE 0

UPSERT 0

LOOKUP 0

TOTAL 237

Refresh

LedgerAccountNumber

CustomerNote

b1e5c90d-1cb9-4202-bb03-26631a9bc6c8

Montgomery County

63e08f91-8a9e-40fe-af02-dcee8c53d48e

Baltimore city

4904d43b-d4c6-4395-85e9-fc021f577674

Anne Arundel County

de518aa5-6f34-4912-99e0-e640cd50765c

Baltimore city



5

Aufbau eines Data-as-a-Service-Ansatzes im Consulting

6

Domain-Template-Driven Datalake Design mit Azure Synapse

7

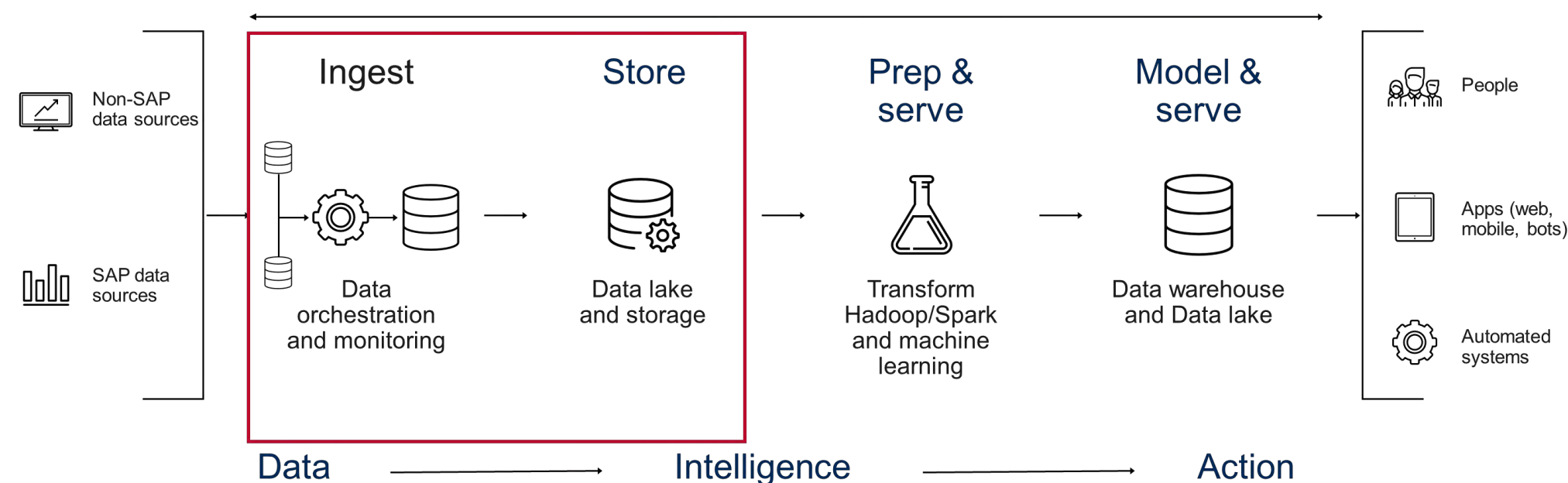
Automatisierung im Data Warehousing

Was steckt dahinter?

- › Arbeit an MDWH-Architekturen hat folgende Erkenntnis generiert:
- › Services in der Cloud sind wesentlich **vielfältiger als traditionelle Architekturen** – sie nutzen diverse Programmiersprachen und Technologien
- › Wir benötigen neben dem umfangreichen methodischen Wissen der Datenintegration, -modellierung sowie -bereitstellung auch zusätzlich Kompetenzen in der Softwareentwicklung!
- › Diese Herausforderung können wir durch wiederverwendbare sowie zugängliche Referenzarchitekturen lösen
-> **MDWH Automation Template**

Was verstehen wir unter DWH-Automatisierung?

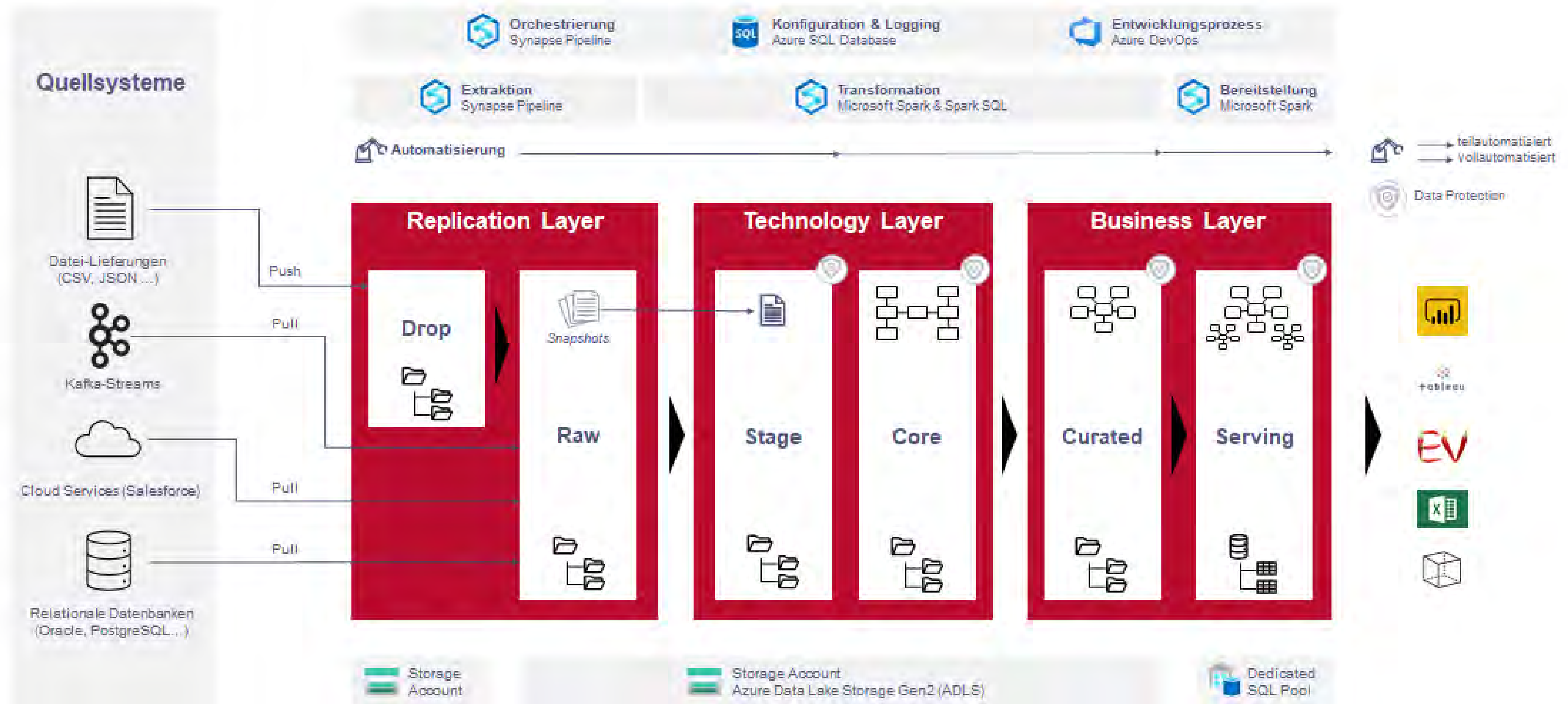
- (M)DWH-Automation ist eine Methode, um vorhersehbare und/oder wiederkehrende Lösungsbestandteile durch **Metadaten zu beschreiben**, um wiederkehrende Abläufe abzubilden und zu vereinfachen
- Lösungsbestandteile: Datenmodelle, ETL/ELT Prozesse, Schnittstellen, ...
- Automation kann dabei durch **Instanziierung** oder **Codegenerierung** stattfinden
- Bereitstellung eines zentralen Templates zur **Realisierung unserer Referenzarchitektur**



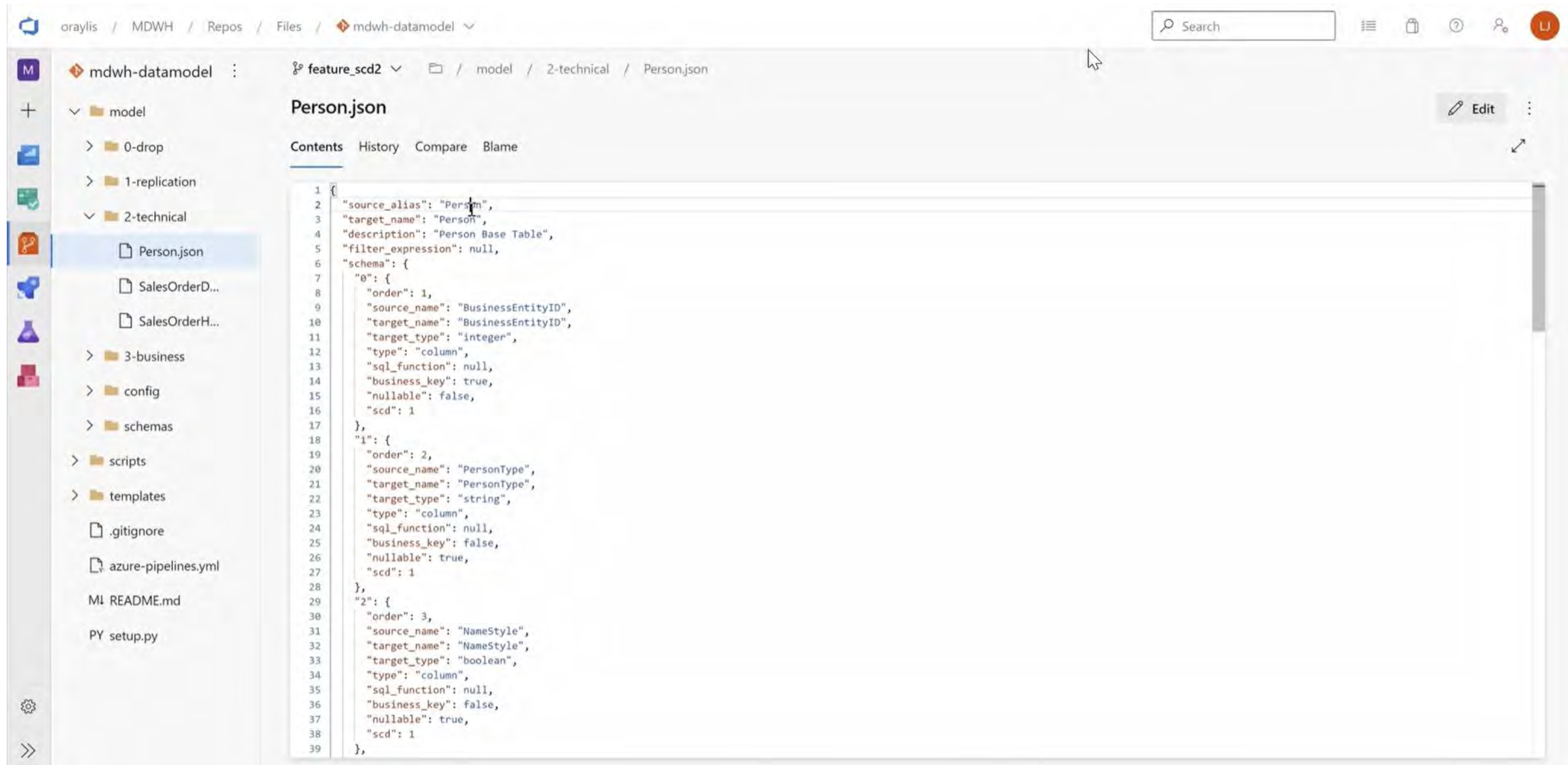
Warum sollte Automation eingesetzt werden?

1. **Einfache und schnelle Entwicklung** – Metadaten sind einfacher zu erstellen als Code (fachbereichstauglich)
2. **Gute Standardisierung** – Das Template erzeugt eine Lösung nach unserer Referenzarchitektur
3. **Hohe Qualität** – Der Prozess läuft für alle Objekte gleich, keine Fehlermöglichkeit
4. **Bessere Wartbarkeit** – Änderungen können zentral für alle Objekte durchgeführt werden
5. **Automatische Dokumentation** – Dokumentation und Lineage kann automatisch erstellt werden
6. ...

Areas of interest for automation



Definition von Metadaten



The screenshot shows a code editor interface with a sidebar on the left displaying a file tree. The main editor area shows the content of a file named `Person.json`. The file is located at `mdwh-datamodel / model / 2-technical / Person.json`. The content of the file is a JSON object defining metadata for a table named "Person".

```
1 {
2   "source_alias": "Person",
3   "target_name": "Person",
4   "description": "Person Base Table",
5   "filter_expression": null,
6   "schema": {
7     "0": {
8       "order": 1,
9       "source_name": "BusinessEntityID",
10      "target_name": "BusinessEntityID",
11      "target_type": "integer",
12      "type": "column",
13      "sql_function": null,
14      "business_key": true,
15      "nullable": false,
16      "scd": 1
17    },
18    "1": {
19      "order": 2,
20      "source_name": "PersonType",
21      "target_name": "PersonType",
22      "target_type": "string",
23      "type": "column",
24      "sql_function": null,
25      "business_key": false,
26      "nullable": true,
27      "scd": 1
28    },
29    "2": {
30      "order": 3,
31      "source_name": "NameStyle",
32      "target_name": "NameStyle",
33      "target_type": "boolean",
34      "type": "column",
35      "sql_function": null,
36      "business_key": false,
37      "nullable": true,
38      "scd": 1
39    }
40  }
41 }
```

Generator: Python Notebook

The screenshot displays the Microsoft Azure Synapse Analytics interface. The top navigation bar shows 'Microsoft Azure | Synapse Analytics | mdwh0syn0dev'. The left sidebar contains a 'Develop' section with a search bar and a list of resources: 'SQL scripts' (17) and 'Notebooks' (10). Under 'Notebooks', there are folders for '2-technical', '3-business', 'x-base', and 'y-help'. The main workspace shows a notebook titled 'NB_Create_Technical...'. The notebook's status is 'Ready'. The title bar indicates 'Attach to: mdwh0spa' and 'Language: PySpark (Python)'. The notebook content includes a title 'Create Technical Layer' and a description 'Notebook that creates and migrates the technical layer (if possible)'. The code cells are as follows:

```
1 %run "x-base/NB_Meta"
```

[1] ✓ 3 min 0 sec - Command executed in 0 sec by ljenzen on 2:31:14 PM, 8/17/21

```
1 %run "x-base/NB_Multiprocessing"
```

[2] ✓

```
1 %run "x-base/NB_Processing_Database"
```

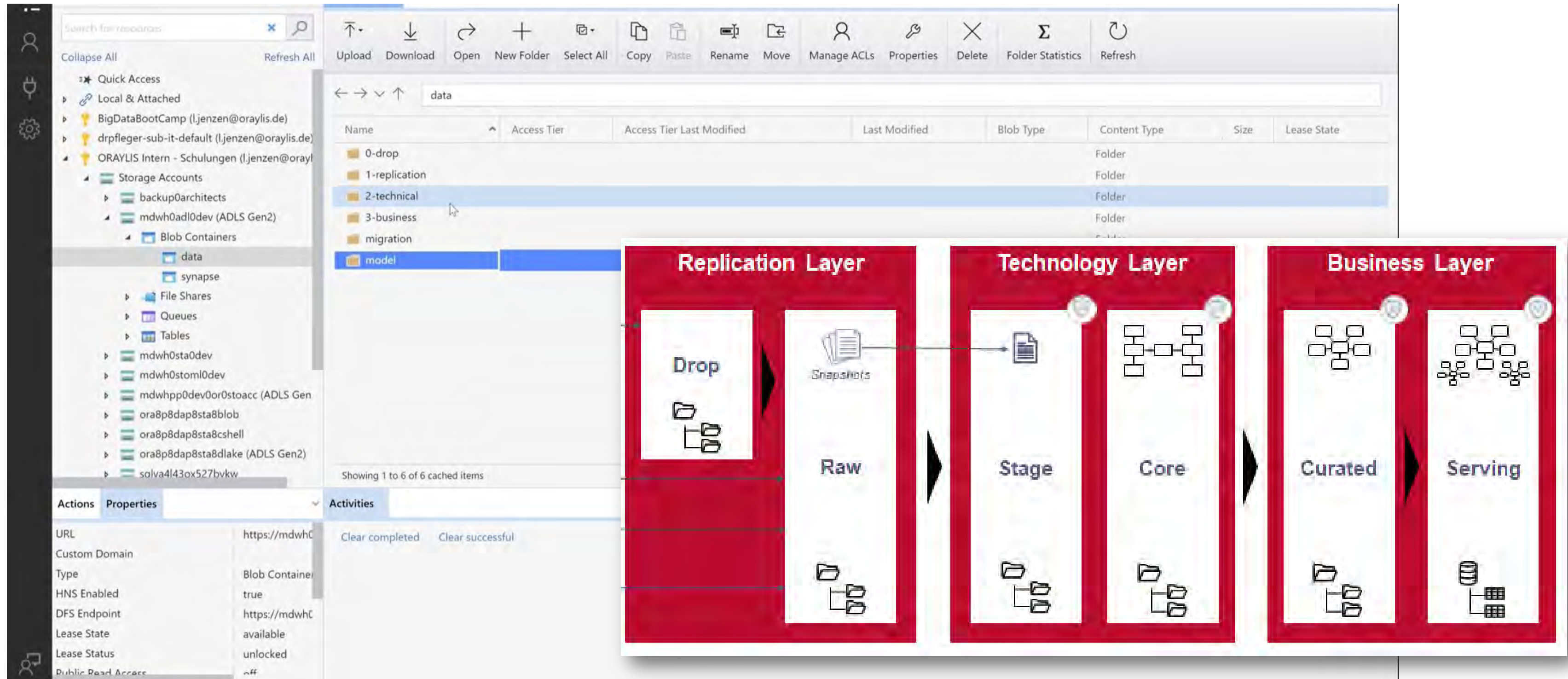
[3] ✓

```
1 storageaccount = "mdwh0ad10dev"
2 entity = ""
3 concurrency = 4
4 server = "mdwh0sql0dev.database.windows.net"
5 database = "mdwh0sql0dev0meta"
6 username = "sqladmin"
7 kvt_name = "mdwh0kvt0dev"
8 ls_processing = "LS_KVT_mdwh0kvt0dev"
```

[4] ✓ Parameters

```
1 # Get meta data
2 conf = Meta(f"abfss://{data@{storageaccount}.dfs.core.windows.net/model/")
3 t1 = conf.t1
```

Ordnerstruktur nach Schichtenkonzept



Generator: Python Notebook

The screenshot displays the Microsoft Azure Synapse Analytics interface. The top navigation bar shows 'Microsoft Azure | Synapse Analytics | mdwh0syn0dev'. The left sidebar contains a 'Develop' section with a search bar and a list of resources: 'SQL scripts' (17) and 'Notebooks' (10). Under 'Notebooks', there are folders for '2-technical', '3-business', 'x-base', and 'y-help'. The main workspace shows a notebook titled 'NB_Create_Technical...'. The notebook's toolbar includes 'Run cell', 'Commit', 'Outline', 'Attach to' (set to 'mdwh0spa'), 'Language' (set to 'PySpark (Python)'), 'Variables', and 'Preview Features'. The notebook content is titled 'Create Technical Layer' with the description 'Notebook that creates and migrates the technical layer (if possible)'. It contains four code cells, each with a 'Run cell' button and a status bar indicating successful execution. The first cell runs '%run "x-base/NB_Meta"' and shows a status of '3 min 0 sec - Command executed in 0 sec by ljenzen on 2:31:14 PM, 8/17/21'. The second cell runs '%run "x-base/NB_Multiprocessing"' and shows a status of '✓'. The third cell runs '%run "x-base/NB_Processing_Database"' and shows a status of '✓'. The fourth cell contains a Python script for configuring a database connection, with a status of '✓' and a 'Parameters' button. The script is as follows:

```
1 storageaccount = "mdwh0adl0dev"
2 entity = ""
3 concurrency = 4
4 server = "mdwh0sql0dev.database.windows.net"
5 database = "mdwh0sql0dev0meta"
6 username = "sqladmin"
7 kvt_name = "mdwh0kvt0dev"
8 ls_processing = "LS_KVT_mdwh0kvt0dev"

1 # Get meta data
2 conf = Meta(f"abfss://{data@{storageaccount}.dfs.core.windows.net/model/")
3 tl = conf.tl
```

Generator: Python Notebook

Microsoft Azure | Synapse Analytics | mdwh0syn0dev

Search

l.jenzen@oraylis.de
ORAYLIS DEV

main branch | Validate all | Commit all | Publish

Develop

Filter resources by name

SQL scripts 17

Notebooks 10

2-technical

3-business

x-base

y-help

NB_Create_Technical...

Cancel all | Commit | Outline | Attach to mdwh0spa | Language PySpark (Python) | Variables | Preview Features

Running

1 # Execute multiprocessing
2 task_exec = task_executor()
3 rc = task_exec.execute(create_technical, entities, concurrency)
4 results = task_exec.check_exception(rc)
5

[*] ✓ - Running

Job execution In progress Spark 2 executors 16 cores

ID	Description	Status	Stages	Tasks	Submission Time	Duration
Job 3	count at NativeMethodAccessorImpl.java:0	✓ Succeeded	2/2	17/17 succeeded	2:33:01 PM, 8/17/21	1 sec
Job 4	saveAsTable at NativeMethodAccessorImpl.java:0	✓ Succeeded	1/1	16/16 succeeded	2:33:10 PM, 8/17/21	1 sec
Job 10	toString at String.java:2994	✓ Succeeded	3/3	52/52 succeeded	2:33:16 PM, 8/17/21	5 sec

View in monitoring | Open Spark UI

+ Code | + Markdown

1 # Report result
2 import json
3 mssparkutils.notebook.exit(json.dumps(results, indent=4, sort_keys=True))
4

[*] ✖ - Waiting

+ Code | + Markdown

Person Entity in Data Lake

Search for resources

Collapse All

Refresh All

Quick Access

Local & Attached

BigDataBootCamp (l.jenzen@oraylis.de)

drpfleger-sub-it-default (l.jenzen@oraylis.de)

ORAYLIS Intern - Schulungen (l.jenzen@oraylis.de)

Storage Accounts

backup0architects

mdwh0adl0dev (ADLS Gen2)

Blob Containers

data

synapse

File Shares

Queues

Tables

mdwh0sta0dev

mdwh0stoml0dev

mdwhpp0dev0or0stoacc (ADLS Gen2)

ora8p8dap8sta8blob

ora8p8dap8sta8cshell

ora8p8dap8sta8dlake (ADLS Gen2)

solva4l43ox527bvk

Upload

Download

Open

New Folder

Select All

Copy

Paste

Rename

Move

Manage ACLs

Properties

Delete

Folder Statistics

Refresh

data > 2-technical

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Lease State
Person					Folder		
SalesOrderDetail					Folder		
SalesOrderHeader					Folder		

Showing 1 to 3 of 3 cached items

Actions

Properties

Activities

URL

Custom Domain

Type

HNS Enabled

DFS Endpoint

Lease State

Lease Status

Public Read Access

https://mdwhC

Blob Container

true

https://mdwhC

available

unlocked

off

Clear completed

Clear successful

Fazit

