The pulse of Munich

SWM – Municipal Utility Company

Nils Hendrik Rottgardt
15.12.2021

Nils Hendrik Rottgardt – BI Architect

Nils Hendrik Rottgardt
BI Architect

Stadtwerke München
Data Analytics & BI

‣ I have been working in the SAP BI environment for 15 years. In the beginning, my focus was on BW and integration with third-party SQL databases. The introduction of SAP HANA was an important basis for personal development and ultimately lead into product responsibility of the SAP HANA native subarea of the enterprise data warehouse (UDWH = Unternehmens Data Warehouse) at Stadtwerke München.

SW//M

Strong legacy: Major business areas

Energy*

Mobility

Telecommunications

* Electricity, natural gas, district heating, district cooling

Drinking water

Public pools

SW//M
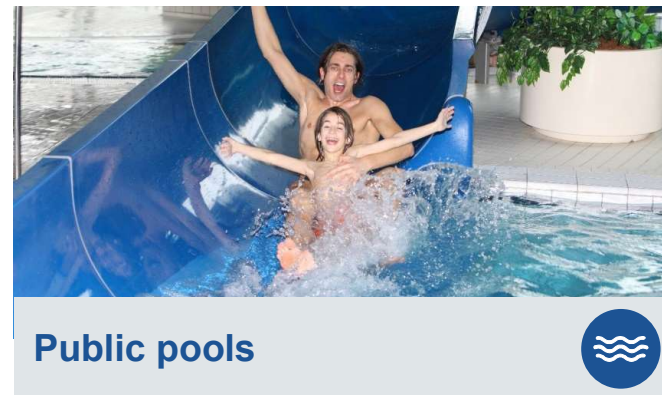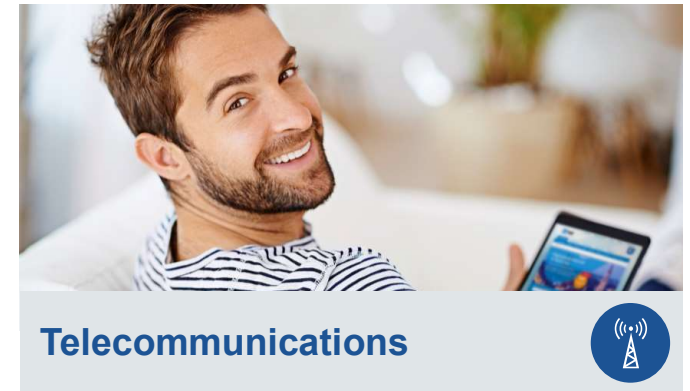
Green electricity plants operated by SWM

Munich and region

≋ 14 hydropower plants

🌀 2 wind power plants

🔥 6 geothermal power plants (one under construction)

🌱 1 biogas plant, 1 biomass cogeneration plant

☀ 33 photovoltaic plants

🌡 12 district cooling plants from ground water/underground streams

Germany

🌀 3 offshore wind parks (North Sea)

🌀 Onshore wind parks (Brandenburg, North Rhine-Westphalia, Rhineland-Palatinate and Saxony-Anhalt)

☀ 2 solar power plants (Bavaria and Saxony)

Europe

🌀 1 offshore wind park (United Kingdom)

🌀 Onshore wind parks (Belgium, Finland, France, Croatia, Norway, Poland, Sweden)

🌞 Parabolic trough power plant (Spain)

Including shares; as of 01/2020

Utility company owned by the city of Munich

👤	**Customer contracts**	1.3 million (energy and drinking water)
🛡	**Shareholder**	City of Munich (100 %)
🏢	**Offices**	Munich, Oslo, London, Copenhagen, Berlin, Brussels

⚡	**Electricity grid**	~13,000 km	🔥	**Natural gas grid**	~6,000 km
🏠	**District heating grid**	~900 km	💧	**Water grid**	~3,400 km
🌡	**District cooling grid**	~30 km	🚌	**Transport network**	~700 km

SW//M

Financials and employees

		2019	2020
Revenues*	in EUR million	**10,711.2**	**10,711.2**
EBITDA	in EUR million	**928,0**	**905,2**
EBIT	in EUR million	**458.6**	**414,4**
Net income	in EUR million	**116,1**	**-152,0**
Investments**	in EUR million	**759.8**	**1,107.2**
Equity ratio	in percent	**52**	**53**
Employees		**9,444**	**10,004**

* excl. electricity and energy tax | ** in property, plant and equipment and intangible assets

SW//M

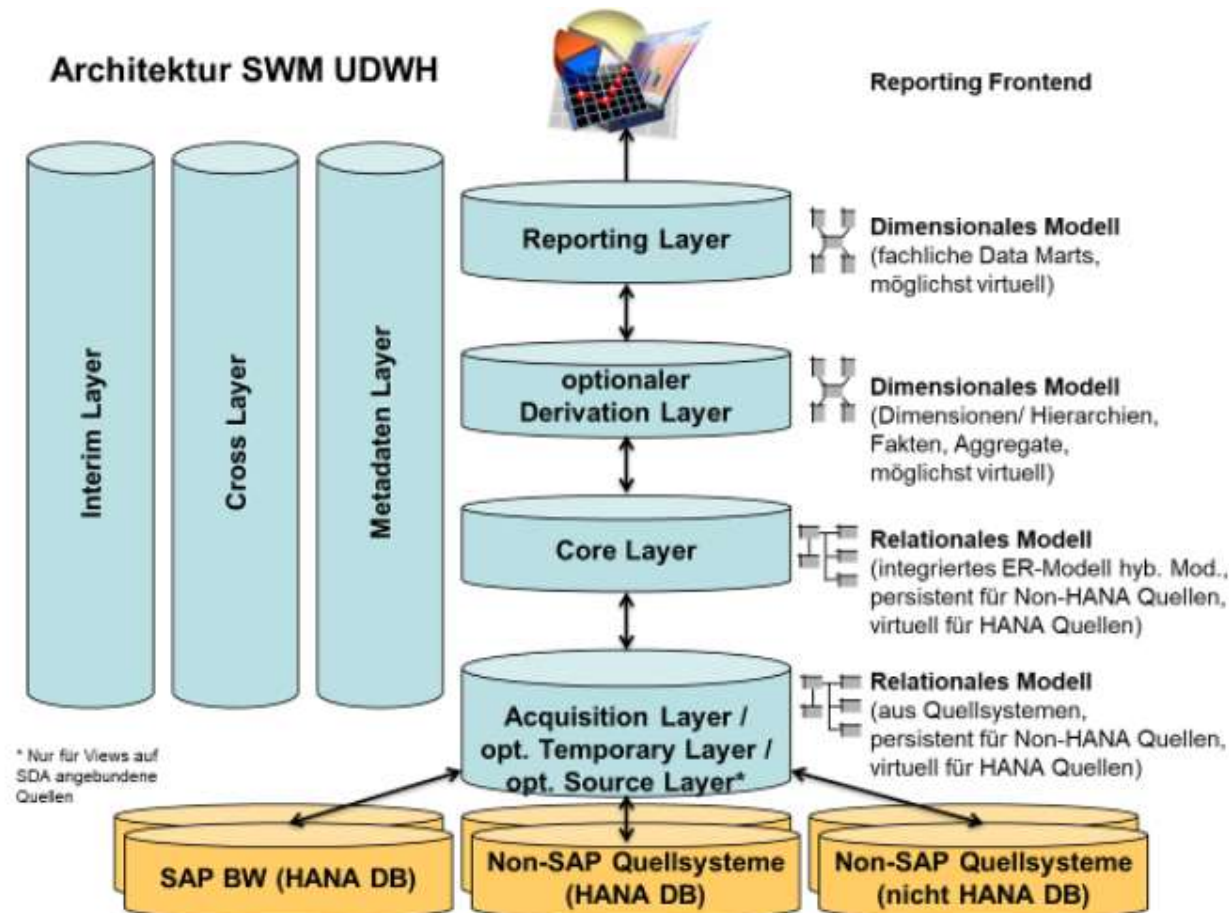# History of SAP BI @SWM
or how we get to agile datawarehousing based on XSA/HANA

‣ Heterogenous system architecture

  ‣ Oracle, SQL Server, PostGre, MariaDB, TM1… → high integration efforts

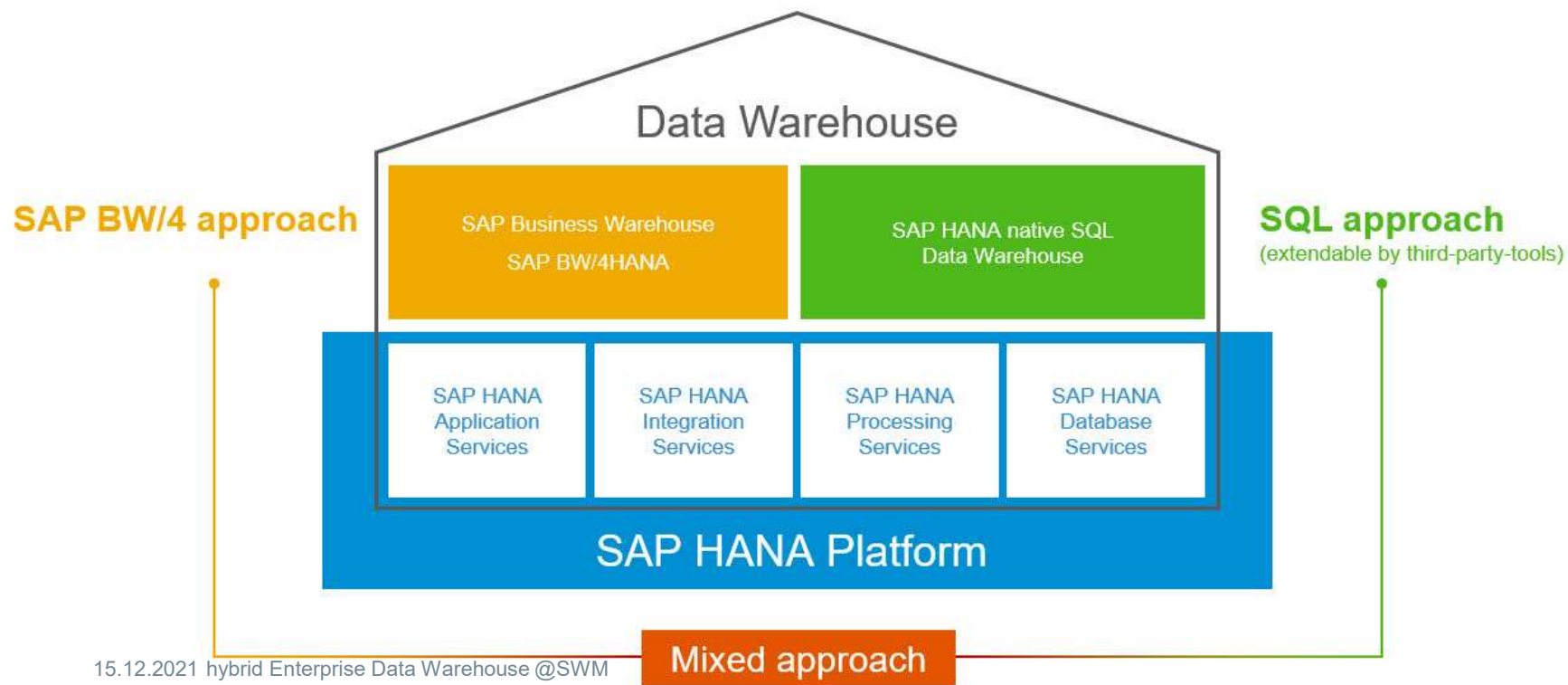‣ BW since 2003

‣ HANA native since 2017

‣ XSA since 2019
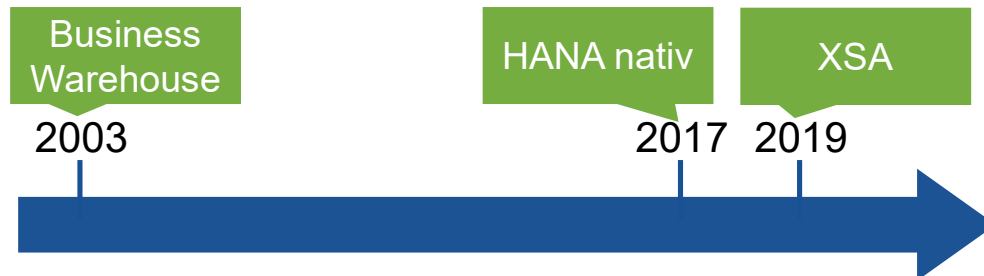
‣ Pre-XSA era

  ‣ Introduction of HANA as central warehouse

  ‣ manual deployment process, no source versioning, no parallel development in one area

  ‣ BW und SQL DWH Silos

    → UDWH introduced as single SQL DWH

‣ XSA

  ‣ Introduction of XSA to define a consistent deployment workflow and stabilize development between multiple projects

  ‣ Containerization of SQL development

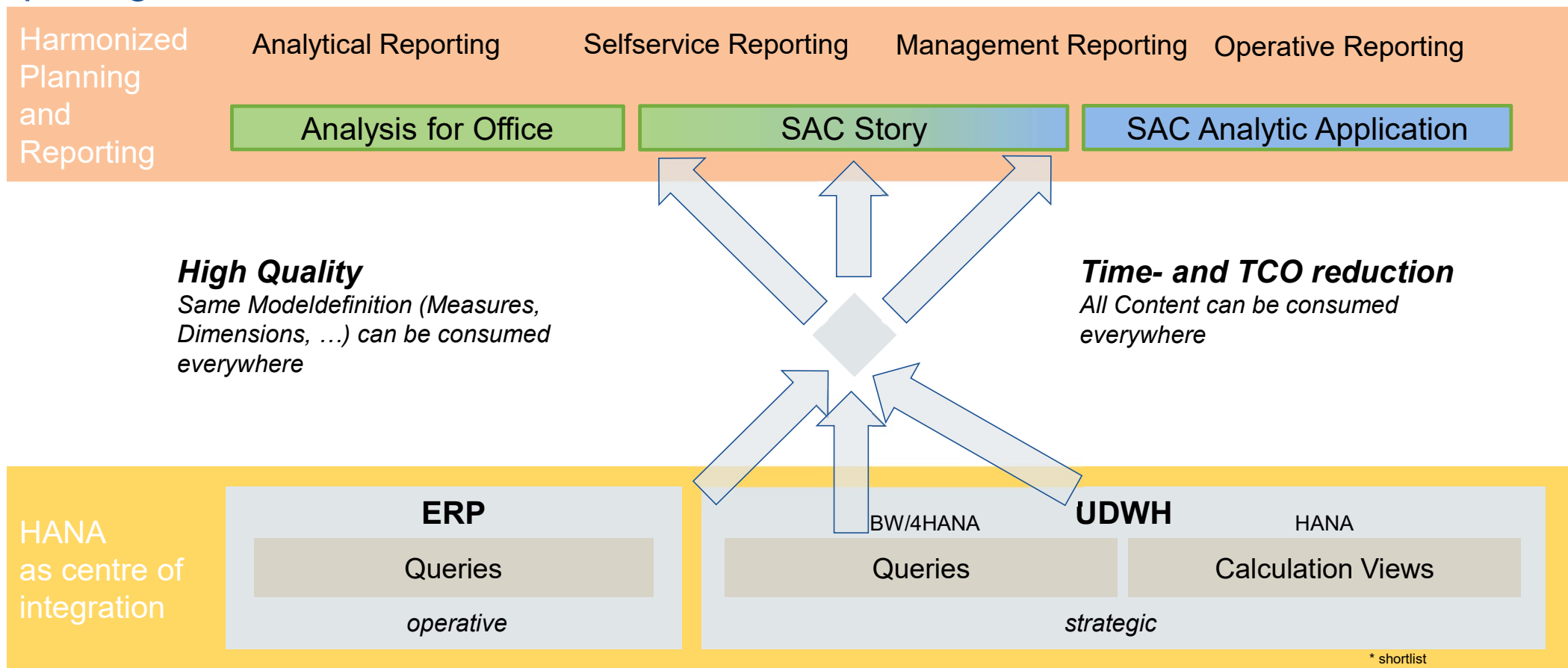 ‣ Merge of BW und SQL DWH as a single mixed scenario Data Warehouse

Business Warehouse
2003

HANA nativ
2017

XSA
2019

SW//M

Target Architecture as of 2017

Architektur SWM UDWH

Reporting Frontend

Interim Layer

Cross Layer

Metadaten Layer

Reporting Layer — **Dimensionales Modell** (fachliche Data Marts, möglichst virtuell)

optionaler Derivation Layer — **Dimensionales Modell** (Dimensionen/ Hierarchien, Fakten, Aggregate, möglichst virtuell)

Core Layer — **Relationales Modell** (integriertes ER-Modell hyb. Mod., persistent für Non-HANA Quellen, virtuell für HANA Quellen)

Acquisition Layer / opt. Temporary Layer / opt. Source Layer* — **Relationales Modell** (aus Quellsystemen, persistent für Non-HANA Quellen, virtuell für HANA Quellen)

* Nur für Views auf SDA angebundene Quellen

SAP BW (HANA DB)

Non-SAP Quellsysteme (HANA DB)

Non-SAP Quellsysteme (nicht HANA DB)

Target Architecture as of 2019

The three approach-strategy for SAP HANA Data Warehousing

SAP BW/4 approach

SQL approach
(extendable by third-party-tools)

Data Warehouse

SAP Business Warehouse
SAP BW/4HANA

SAP HANA native SQL
Data Warehouse

SAP HANA
Application
Services

SAP HANA
Integration
Services

SAP HANA
Processing
Services

SAP HANA
Database
Services

SAP HANA Platform

Mixed approach

15.12.2021 hybrid Enterprise Data Warehouse @SWM

Satisfy Poweruser to Endusers with one Frontend during the whole lifecycle of the reporting

Harmonized Planning and Reporting	Analytical Reporting	Selfservice Reporting	Management Reporting	Operative Reporting
	Analysis for Office	SAC Story		SAC Analytic Application

High Quality
Same Modeldefinition (Measures, Dimensions, …) can be consumed everywhere

Time- and TCO reduction
All Content can be consumed everywhere

HANA as centre of integration	**ERP**		**UDWH** BW/4HANA	HANA
	Queries		Queries	Calculation Views
	operative		*strategic*	

* shortlist

SW//M

What is our focus of SAC together with our UDWH (BW/4HANA, HANA XSA)
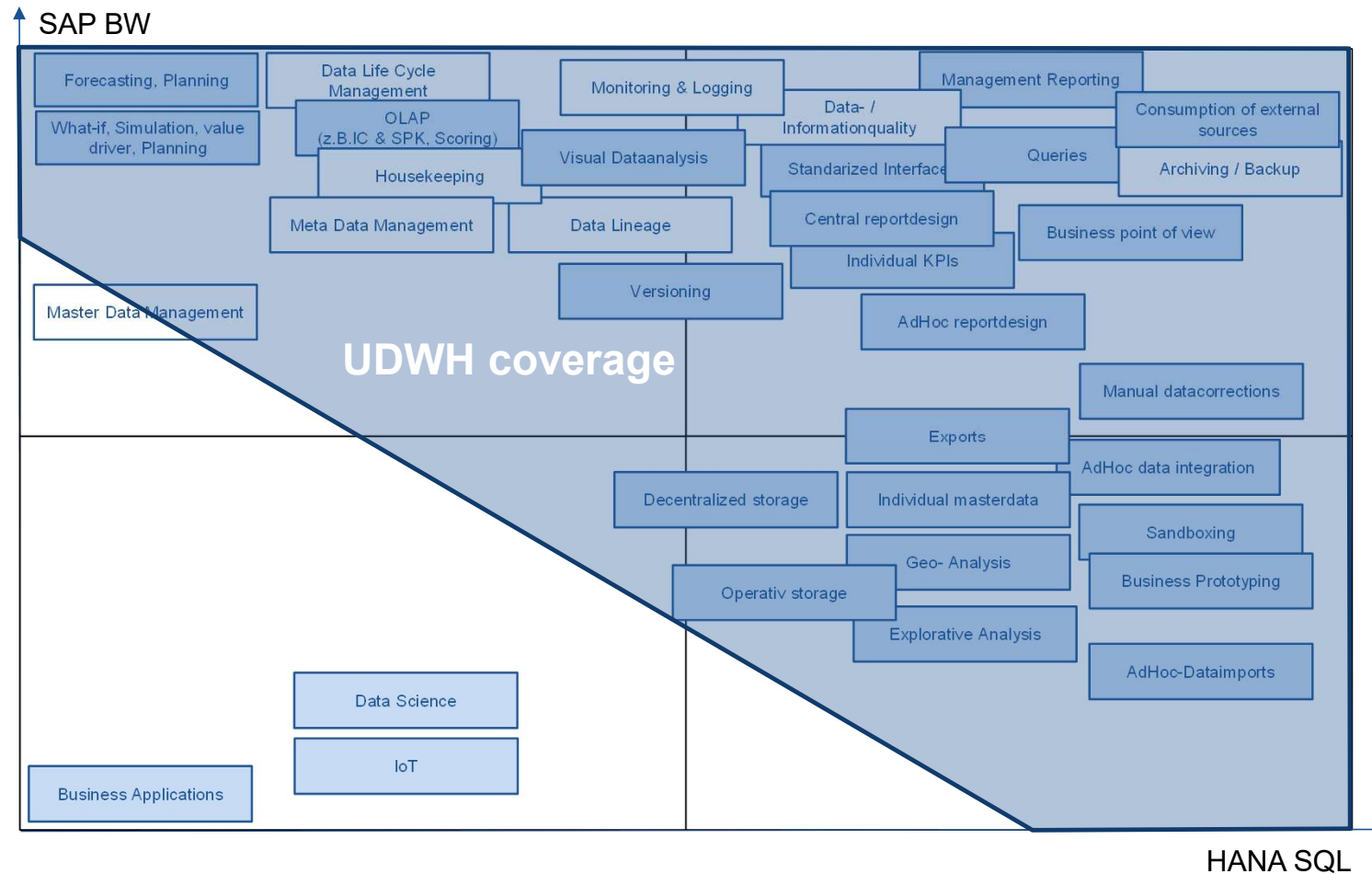
Actual (2021) architecture blueprint
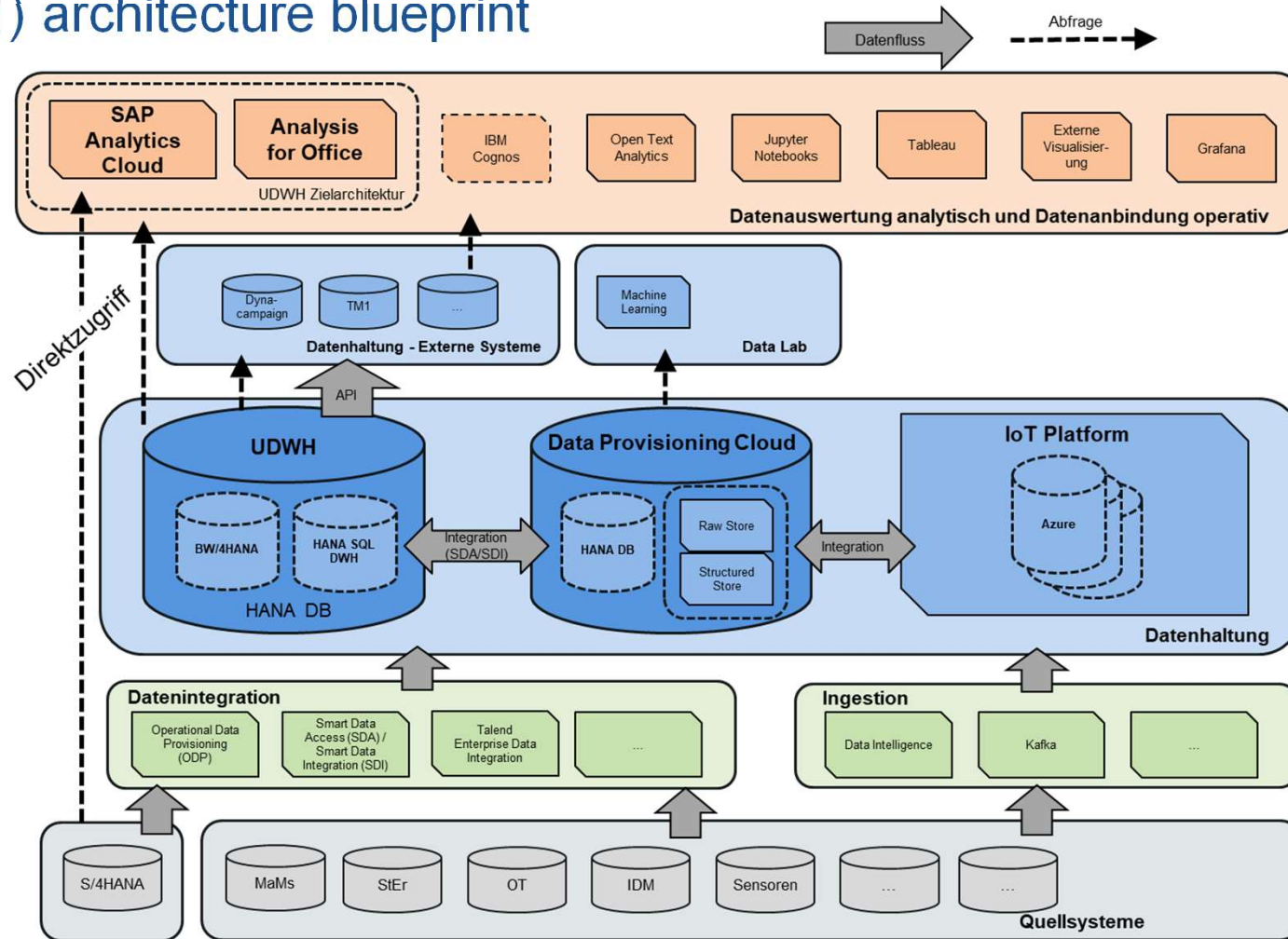
* nach *Strategie Data Management SWM*: dort heißt dieser Datentopf „Data Lake" und enthält nur anonymisierte Daten und ist grundsätzlich SWM-weit zugreifbar

Why two pillers of Data Warehousing? The best of two worlds!
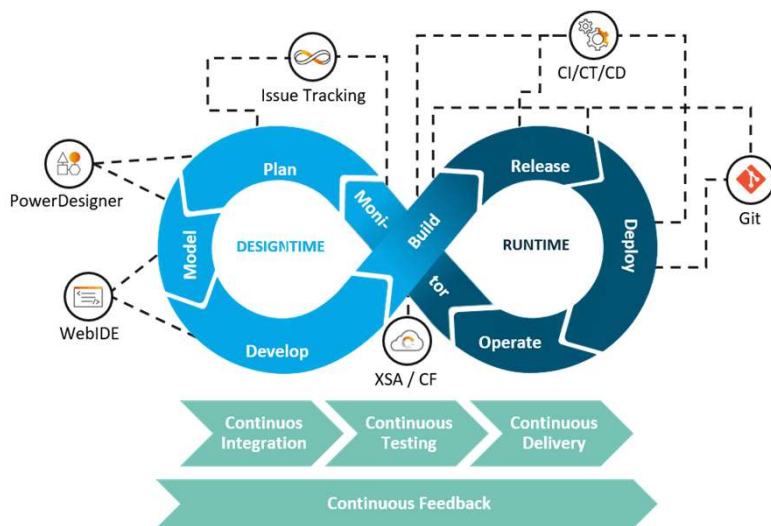
XSA

‣ Flexible
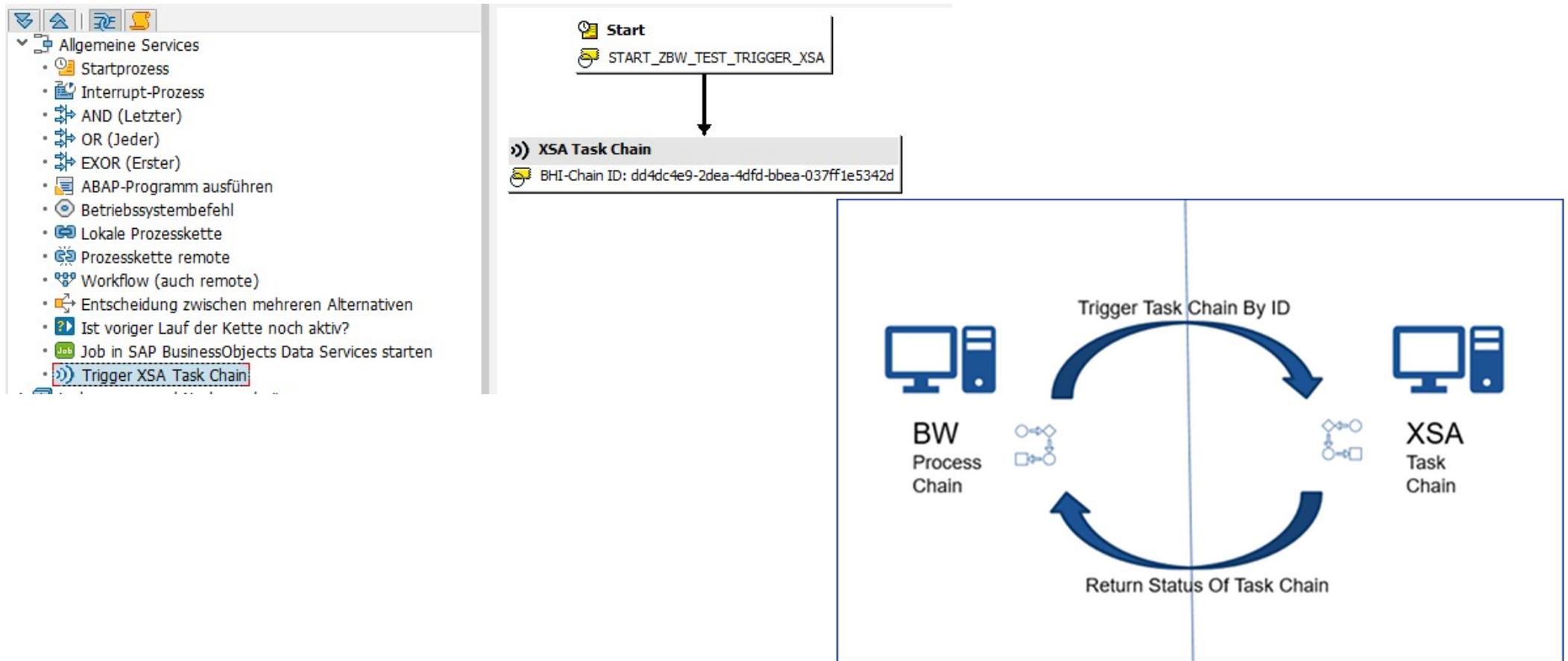
‣ Short implementation cycles (agile)

‣ Evolutional approach

BW

‣ Data Warehouse Framework

‣ High automatization

‣ Stable approach

‣ Metadata driven

Integration of DWF Task Chains in BW Process Chains for a seamless orchestration

The glue in between… Cross Database Access (CDA)

‣ Single Core Layer managed by BW <u>and</u> XSA

‣ Bidirectional access between BW and native developments

   ‣ BW OpenODS View → CDA → SQL Data Warehouse

   ‣ SQL Data Warehouse → CDA → BW DB Artifacts (ADSO View 7, ADSO View 8, InfoObject ext. HANA Views)

‣ No shared access rights → Full Core Layer Access for developers

‣ Access Management is done in Reporting Layer

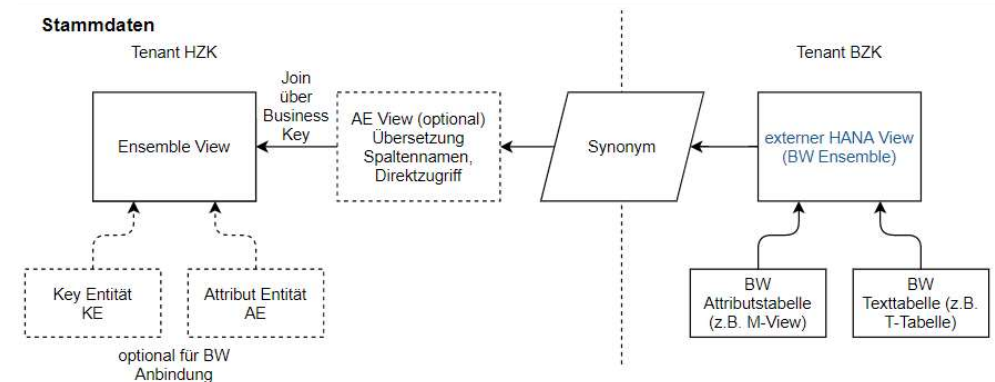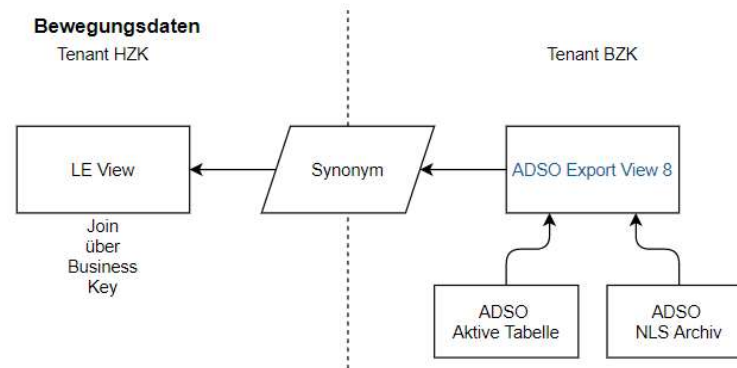   ‣ BW Analysis Authorization

   ‣ HANA Analytical Privileges

Fast access to BW Objects from XSA/HANA to reduce data redundancy

Advantages

‣ Replication of data is not necessary anymore

‣ Direct integration of data between BW and HANA

‣ No redundancies leads into a small data footprint and a cheaper architecture

Preconditions

‣ Introduction of BW/4 (or an ADSO driven BW 7.5 on HANA)

‣ Unified Core-Layer

‣ Implementation of an SID (Integer) and/or BK based HANA modelling
(Standard-Data Vault with HASH, etc. is not possible)

‣ CDA connection between both Tenants

UDWH

SAP HANA Modeling

RL Calculation View

DL   DL

CL   Joins

AL

UDWH Core Layer

SAP BW

BW Modeling

HCPR RL

ADSO DL

ADSO CL

ADSO AL

Metadatenrepository

Similar architecture according LSA++ in both worlds and strong integrated data model. Unified Core Layer, so ADSO and also HANA Objects could used in both ETL processes independent where the object is managed (HANA/DWF or BW).

Fast access to BW Objects from XSA/HANA to reduce data redundancy

Advantages

▸ Replication of data is not necessary anymore

▸ Direct integration of data between BW and HANA

▸ No redundancies leads into a small data footprint and a cheaper architecture

Preconditions

▸ Introduction of BW/4 (or an ADSO driven BW 7.5 on HANA)

▸ Unified Core-Layer

▸ Implementation of an SID (Integer) and/or BK based HANA modelling
(Standard-Data Vault with HASH, etc. is not possible)

▸ CDA connection between both Tenants

Fast access to XSA/HANA Objects from BW to reduce data redundancy

Advantages

‣ Replication of data is not necessary anymore

‣ Direct integration of data between BW and HANA

‣ No redundancies leads into a small data footprint
and a cheaper architecture

Preconditions

‣ Introduction of BW/4 (or an ADSO driven BW 7.5 on HANA)

‣ Unified Core-Layer

‣ Implementation of an SID (Integer) and/or BK
based HANA modelling
(Standard-Data Vault with HASH, etc. is not possible)

‣ CDA connection between both Tenants

Beispiel STER im Core Layer

SW//M
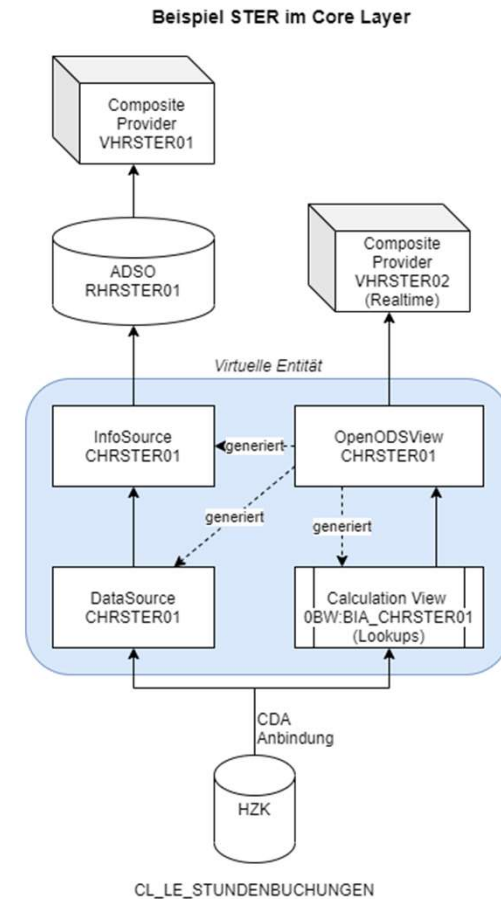
Official Smart Data Access FAQ – Joins are bad

SAP Smart Data Access
Modeling Recommendations

- Avoid JOINs between tables from multiple remote sources

  Joining multiple remote sources would create performance issue. It would require moving data from one remote source to another remote source via HANA to execute the join.

  For example, lets say the analytic use requires building HANA model to combine data from HANA, Teradata, and IQ.

  In this case joining Teradata tables with IQ tables in a model would create performance overhead because data from either Teradata or IQ will need to be brought over to HANA, then push it to IQ or Teradata where JOIN will execute.

  Instead you should create encapsulated calc view for only IQ tables, and create encapsulated calc view for only Teradata tables, and use these encapsulated models to build final analytic model.

- Avoid large data transfer between SDA remote sources

  In case of joins on high-cardinal its columns, there will be significant data transfer from SDA remote source to HANA. Consider moving these joins to the remote source, by replicating the HANA table to the remote source

9

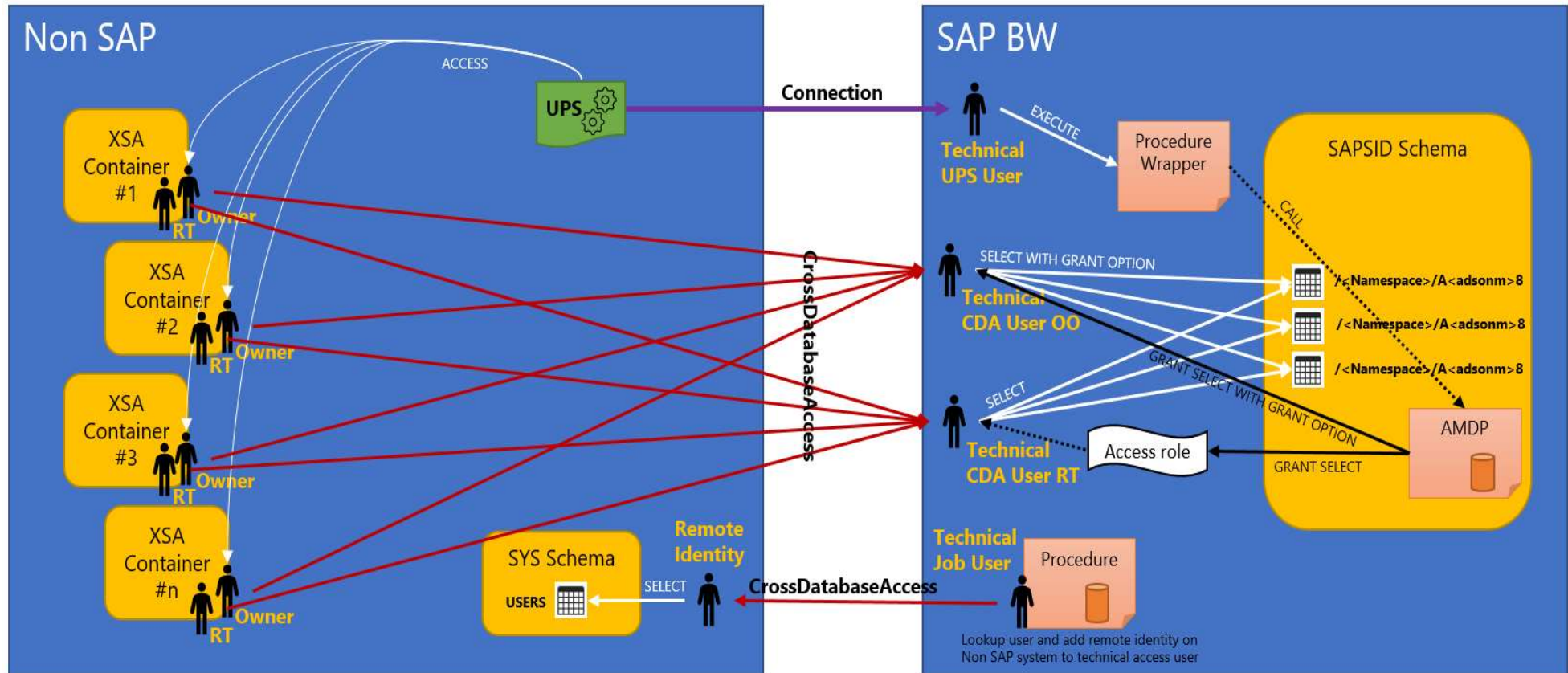# CDA bahavior is like everything is stored on the same Tenant
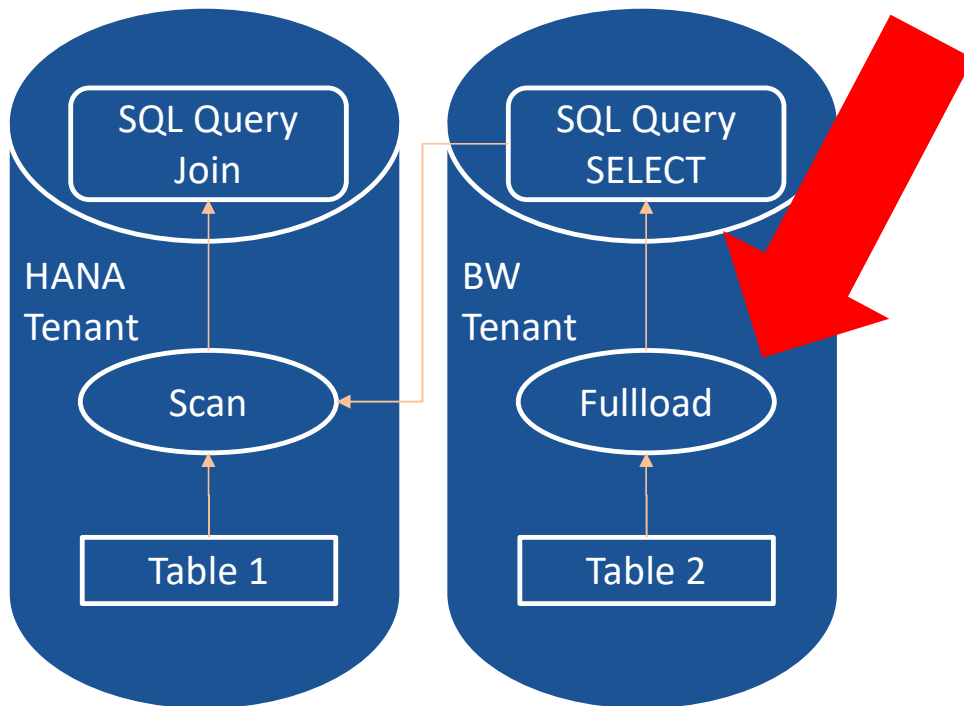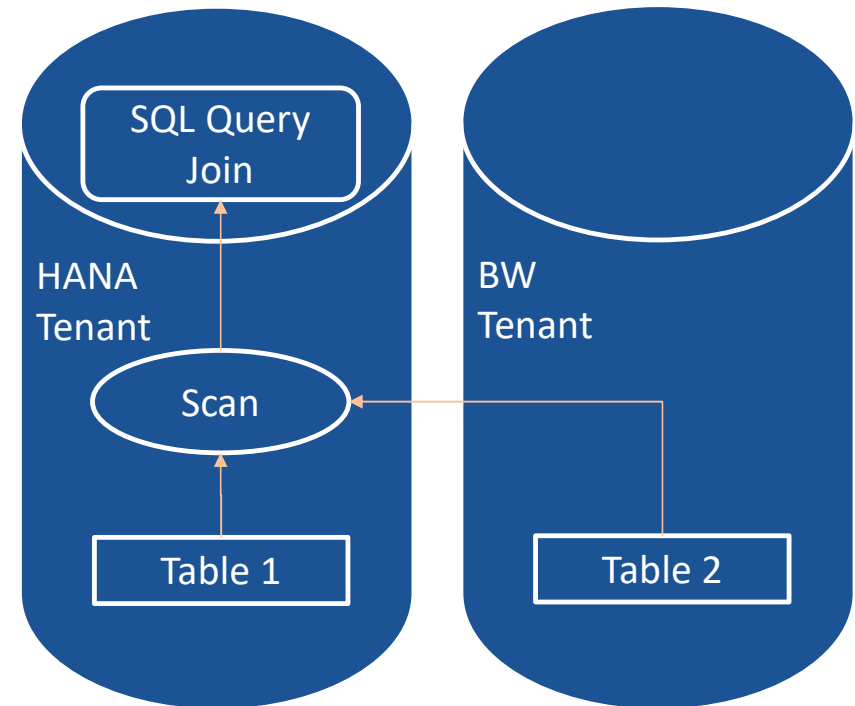
How we work in XSA (with Git)
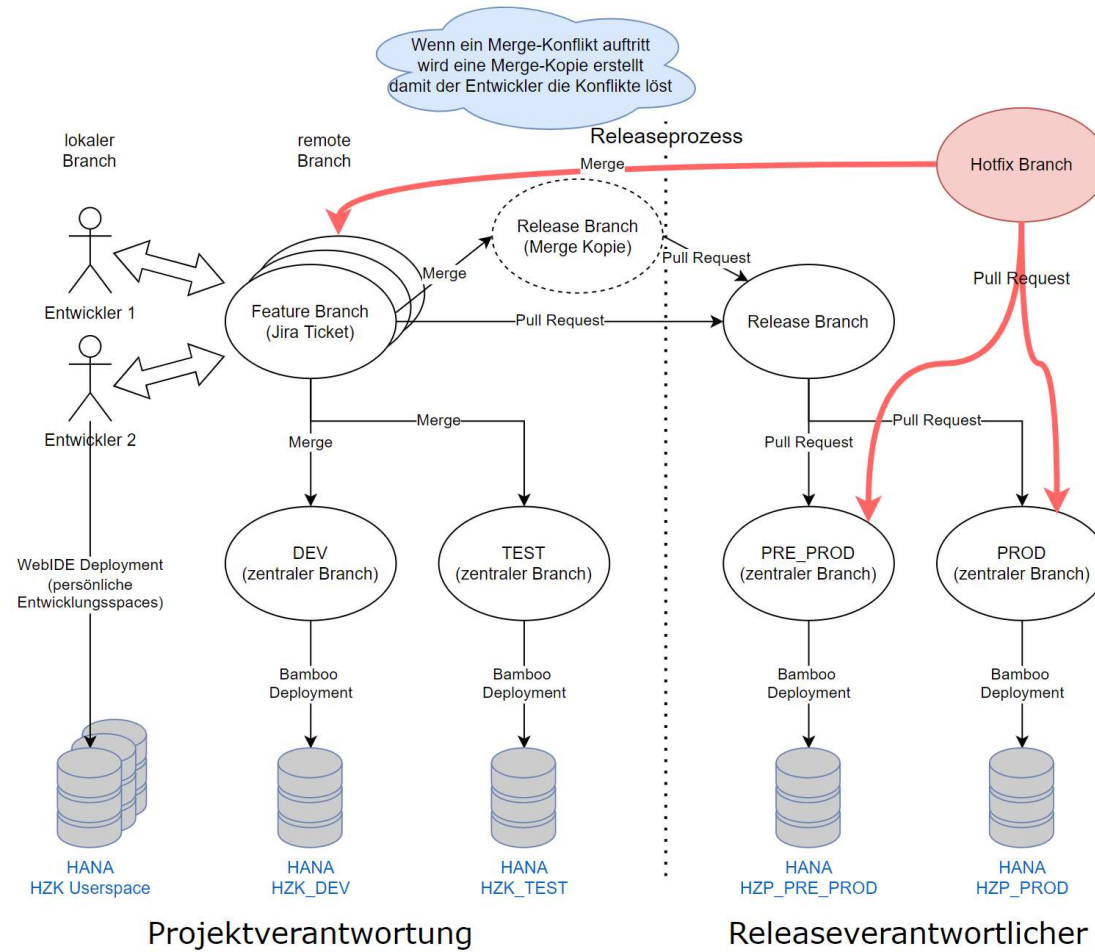
CDA automation with HDI Containers – a little complex but running fine

Unified Core Layer – an example

Lessons learned

▸ Technical preconditions must be fulfilled like Cross Database Access. SDA as great restrictions for a Mixed Scenario

▸ XSA and BW developers must "trust" each other → Unified Core-Layer (at best one developer for both technologies)

▸ Architects has to understand BW and XSA/HANA to decide objective where to implement a data flow (HANA, BW or hybrid approach)

▸ XSA is oriented on software development processes which is not easy to understand for colleagues from the business

▸ The native modelling strategy has to be aligned with the requirements of the overall model. So e.g. a straight data vault approach is not possible as it is not compatible with BW modelling

▸ Most important: reserve time for SAP incidents and actual missing functions like automatic revalidation of artefacts

SW//M

What you get with Mixed Scenario Data Warehouse by SAP

A highly flexible Data Warehouse covers most of the customer needs like standardreporting, SelfService-Reporting (OLAP) up to data wrangling (SQL) with less implementation effort. Supporting classic and agile projects best.

SW//M