



Historie à la Carte

TDWI Konferenz 2025 (Mi 9.3)

Timo Cirkel & Markus Burger-Scheidlin
München | 25.06.2025 11:25 – 12:10

HDI



Timo Cirkel
Team CoreCrafters@HDI



✉ timo.cirkel@hdi.de

BI

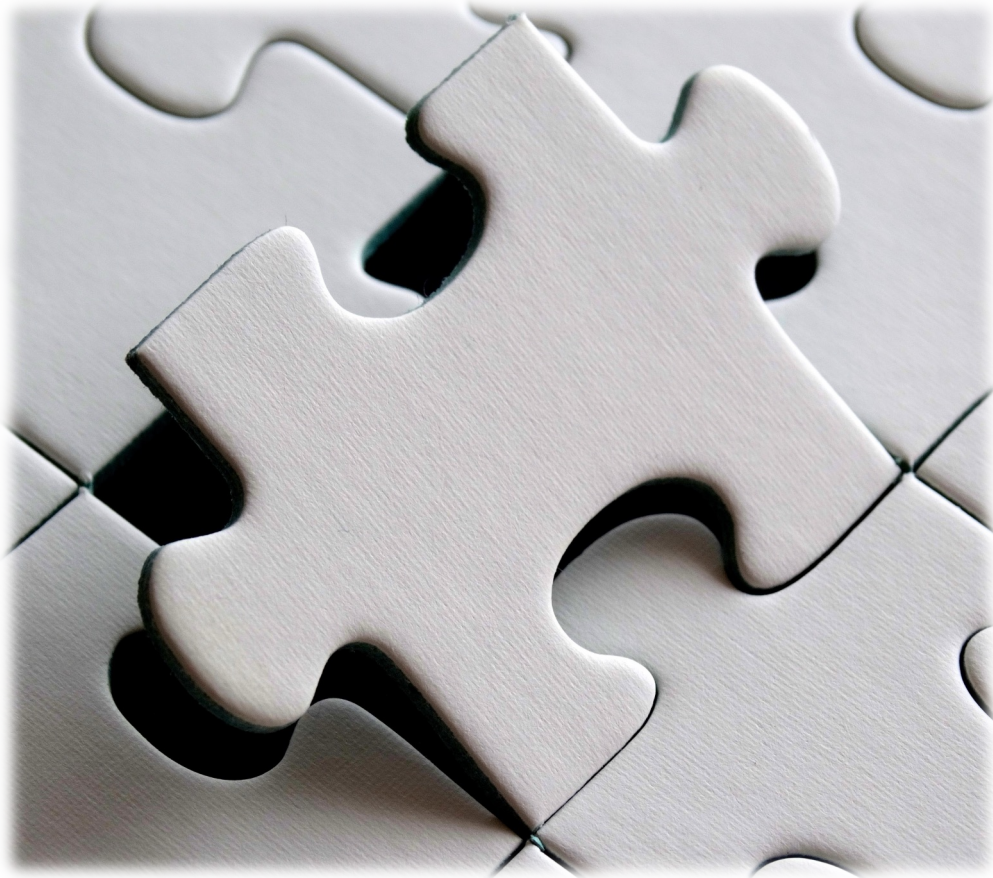


Markus Burger-Scheidlin
Burger-Scheidlin BI Beratung



✉ mbs@busch-bi.de

Sie sind hier richtig, wenn



Herausfordernde
Quellenlandschaft

Komplexe
Historisierung



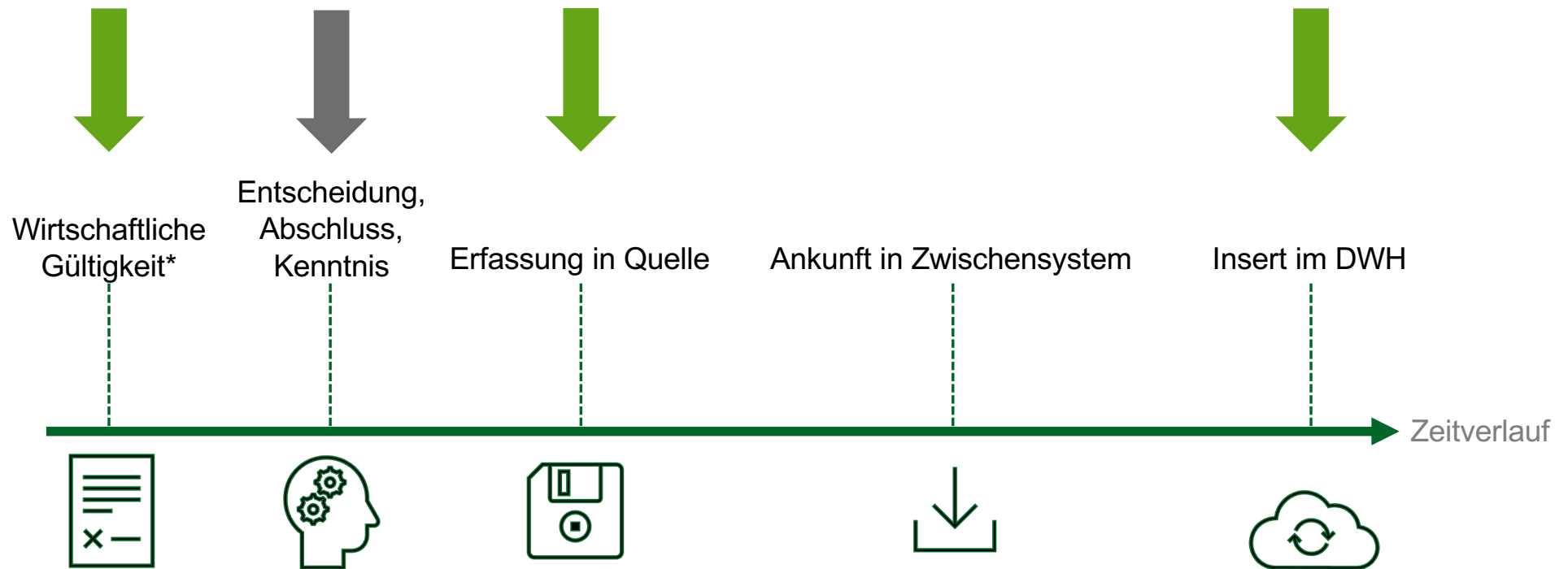
Agiles Vorgehen

Vielfältige
historisierte Sichten

1

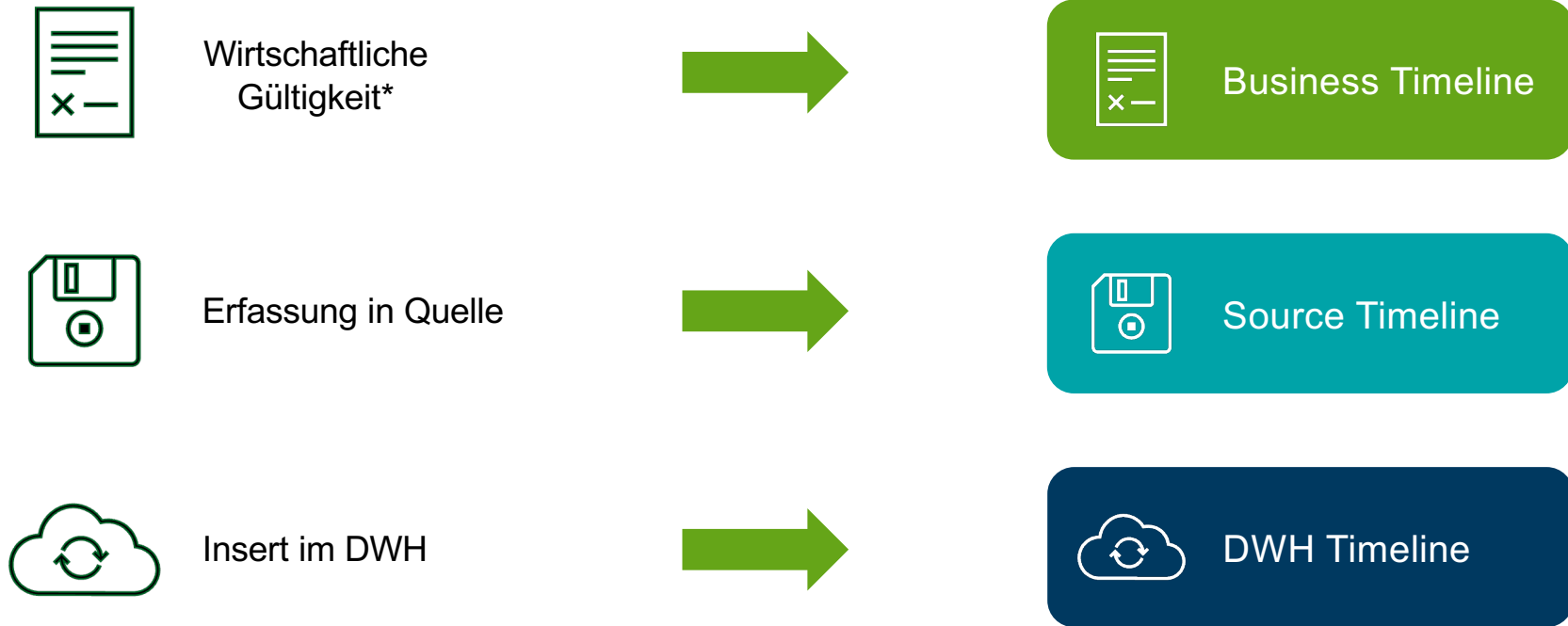
Die Problemstellung

Versicherungsverträge sind kompliziert: Relevante Zeitpunkte für das Data Warehouse



* Falls das Quellsystem eine eigene Gültigkeit aniefert, kann dieser Zeitpunkt beliebig auf dem Zeitstrahl liegen

Mögliche Zeitlinien im Data Warehouse



Vielfältige Benutzergruppen und Fragestellungen



Vertrieb

Aktuelle Sicht
z.B. anstehende Prolongation



Underwriting

Verschiedene Wissensstände
z.B. Technical Pricing



Controlling

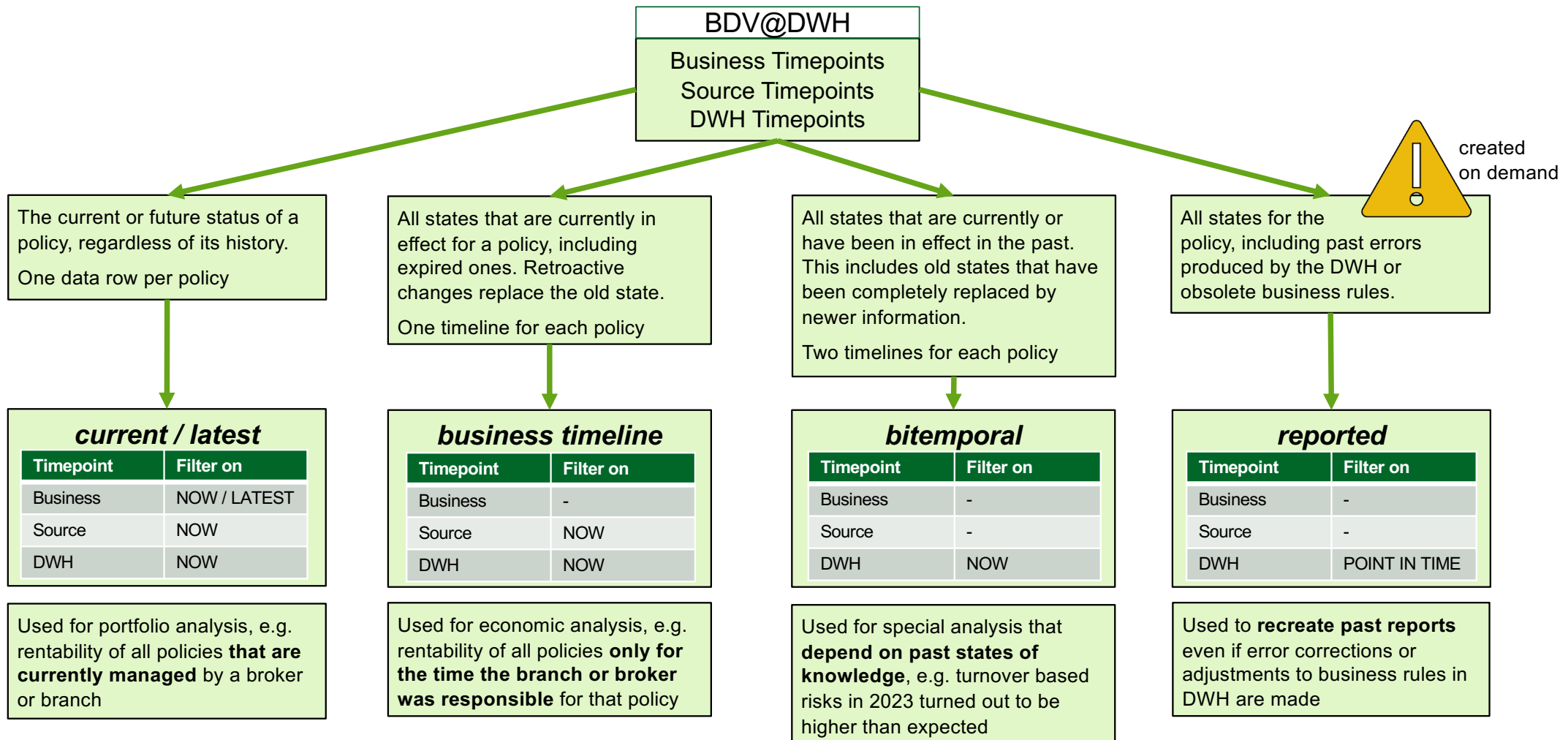
Wirksame Historie
z.B. Rentabilität von Verträgen



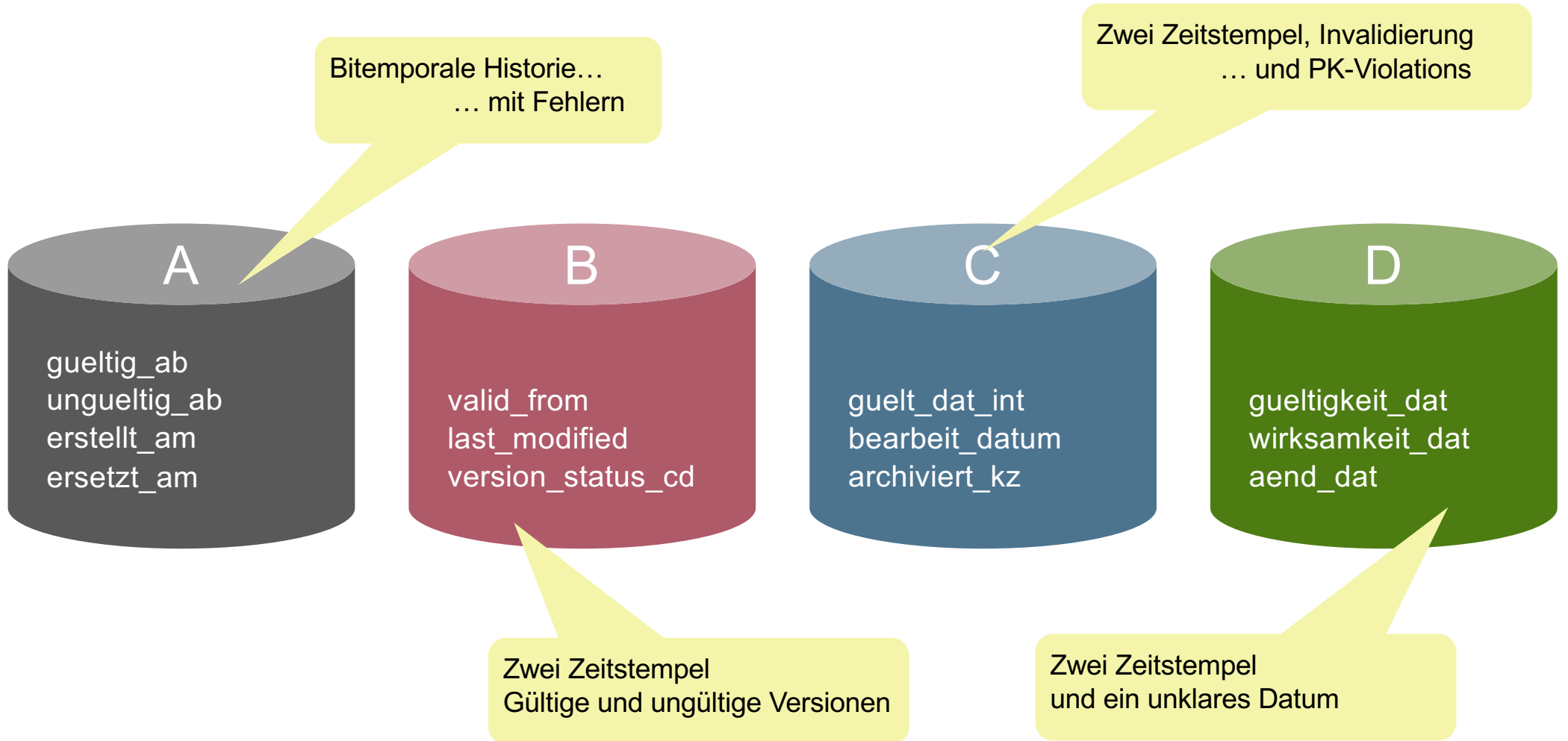
Rechnungswesen

Bilanzerstellung
Reproduktion von Berichten

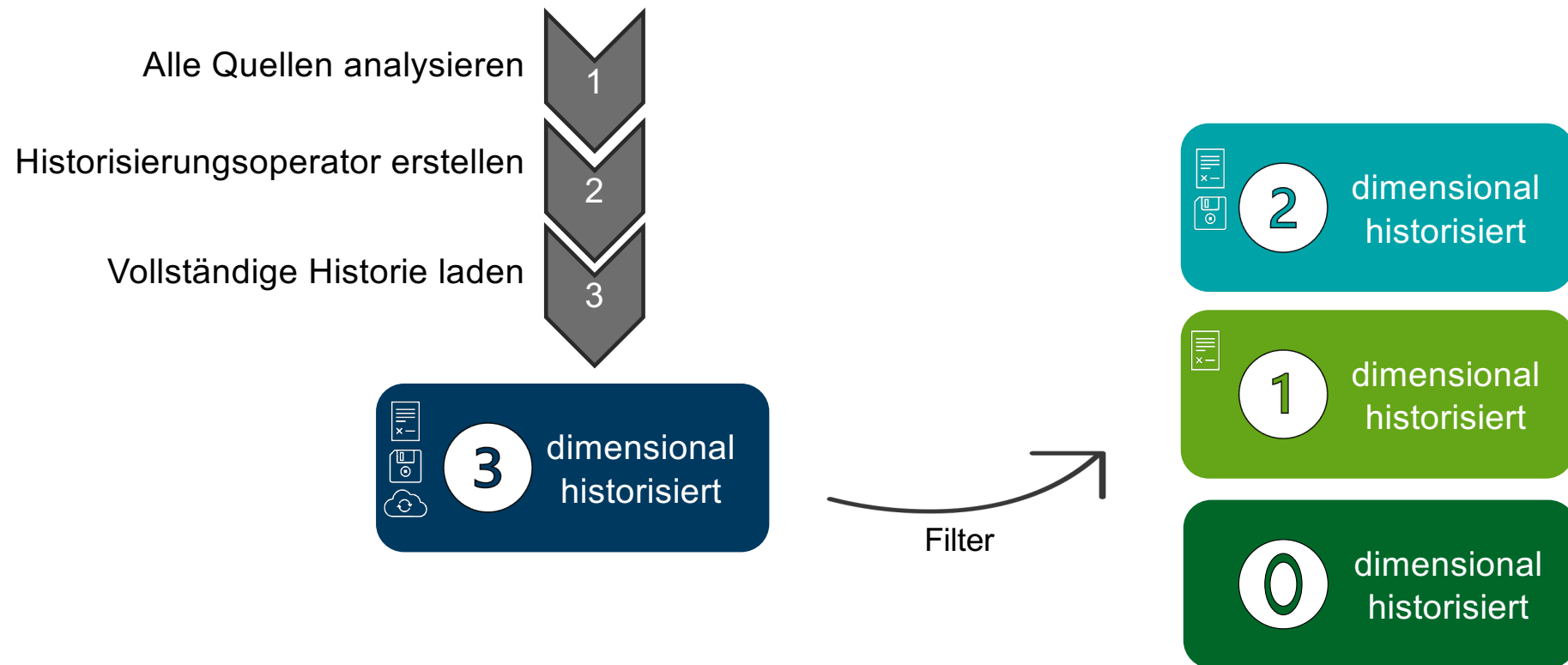
Projektbeispiel: „Which timelines can be made available?“



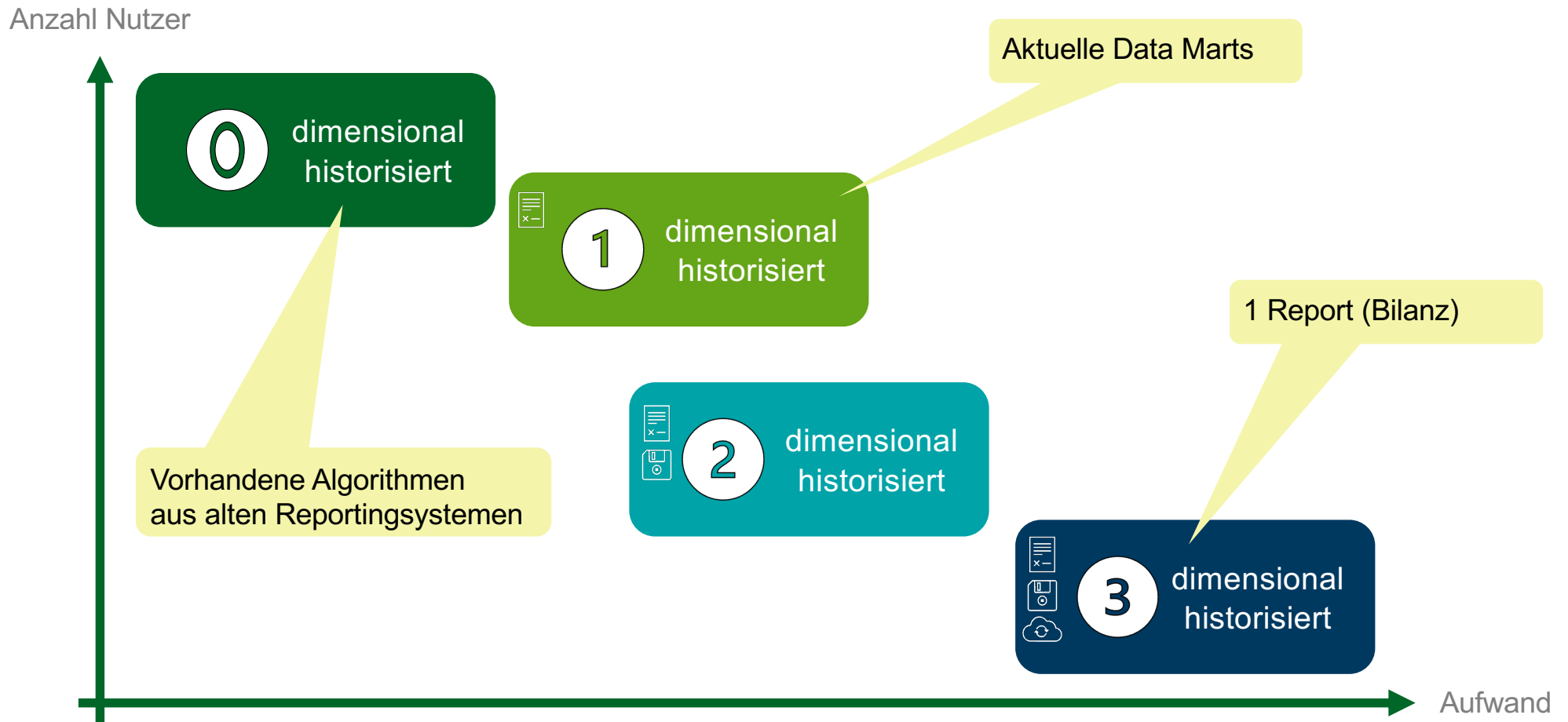
Jedes Quellsystem liefert anders



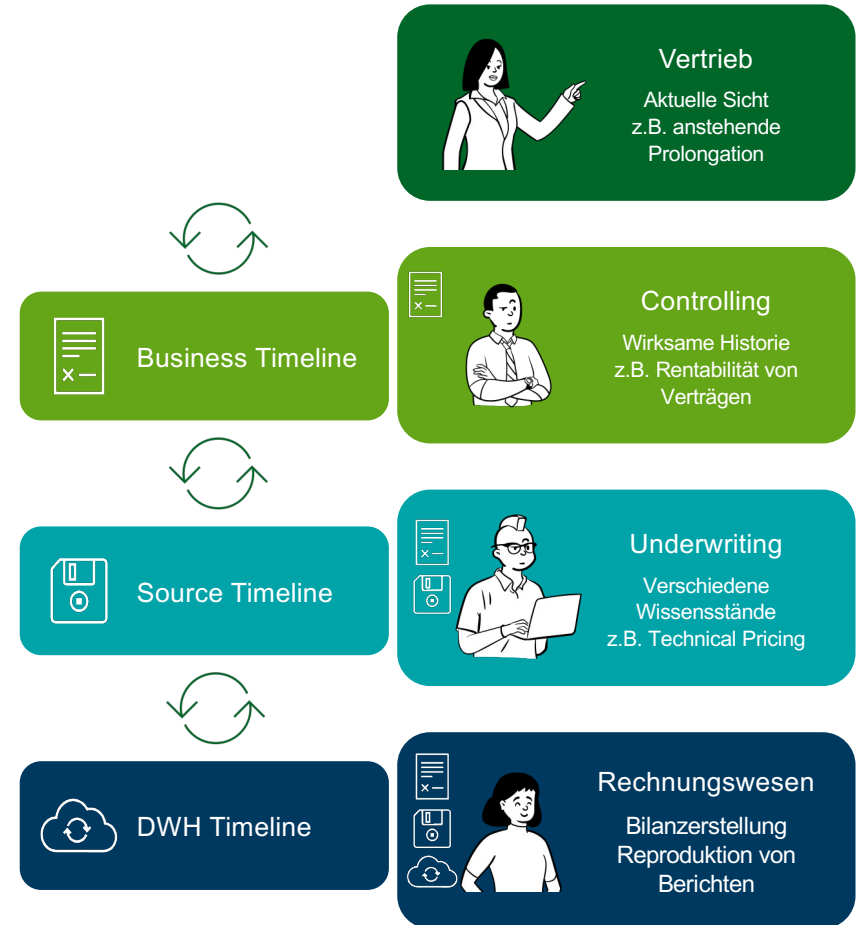
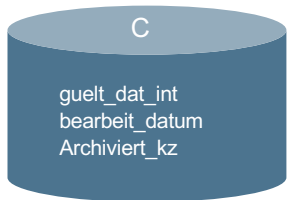
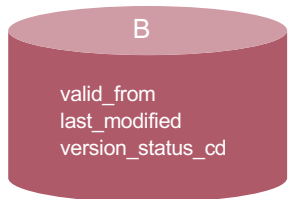
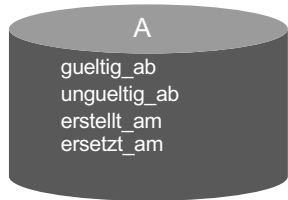
Klassisches Vorgehen benötigt viel Zeit



Sinnvoll ist ein Start bei der aktuellen Sicht



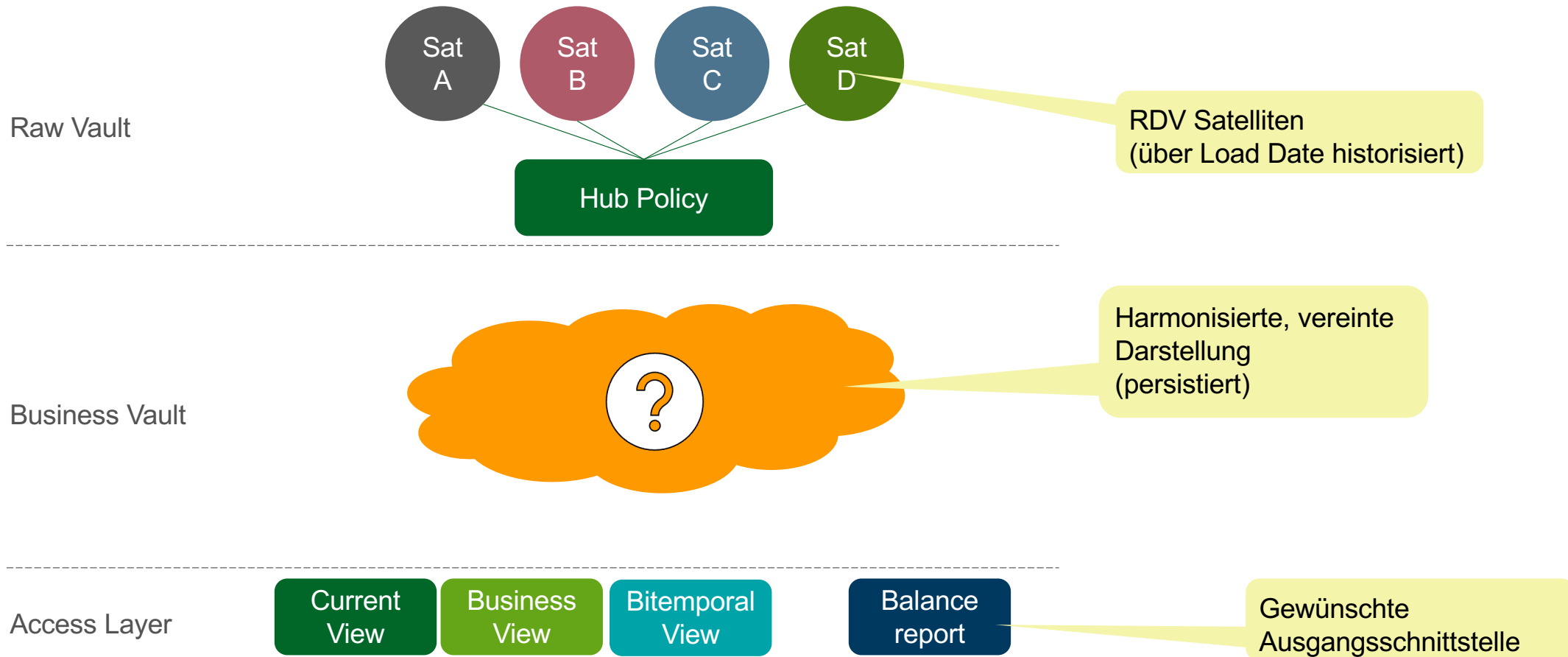
Geht das auch agil?



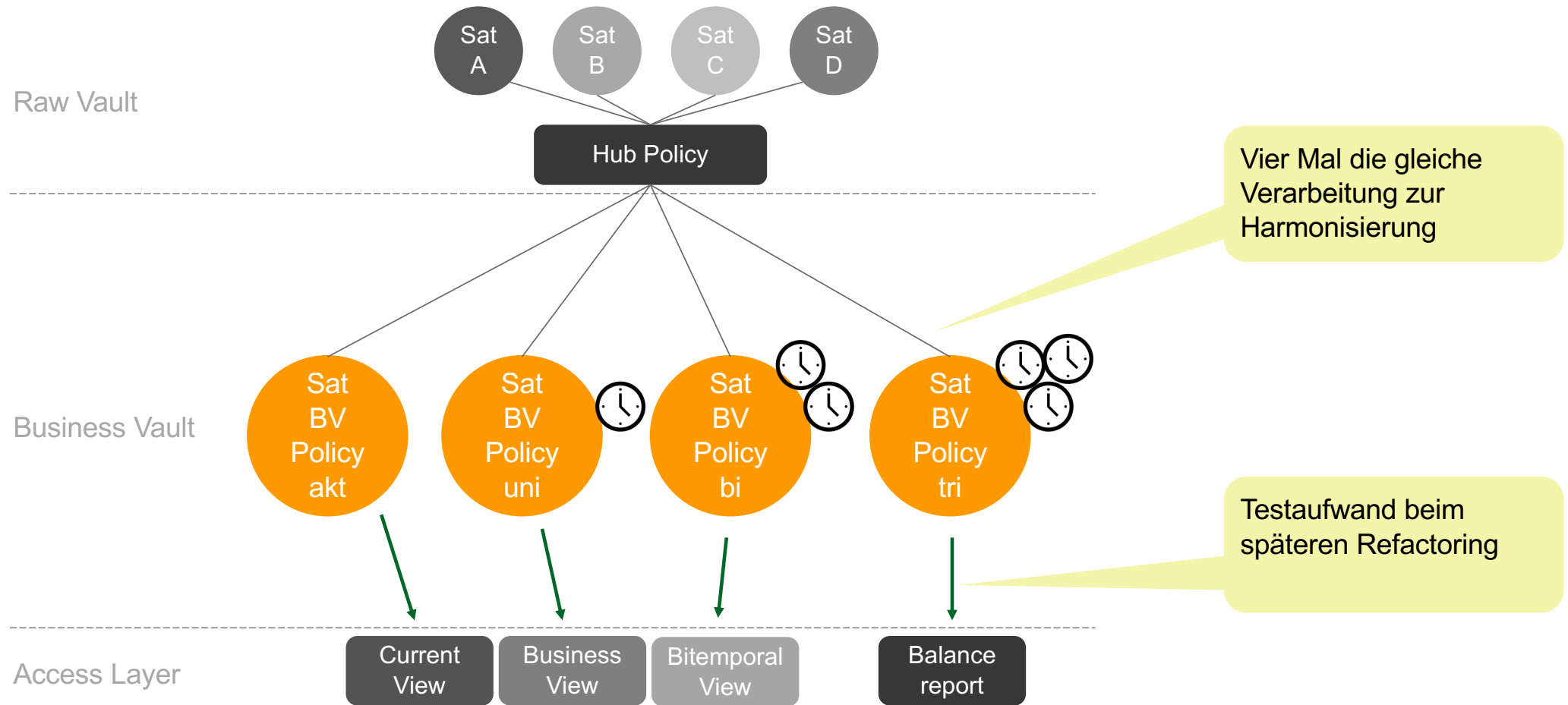
2

Unser Lösungsansatz

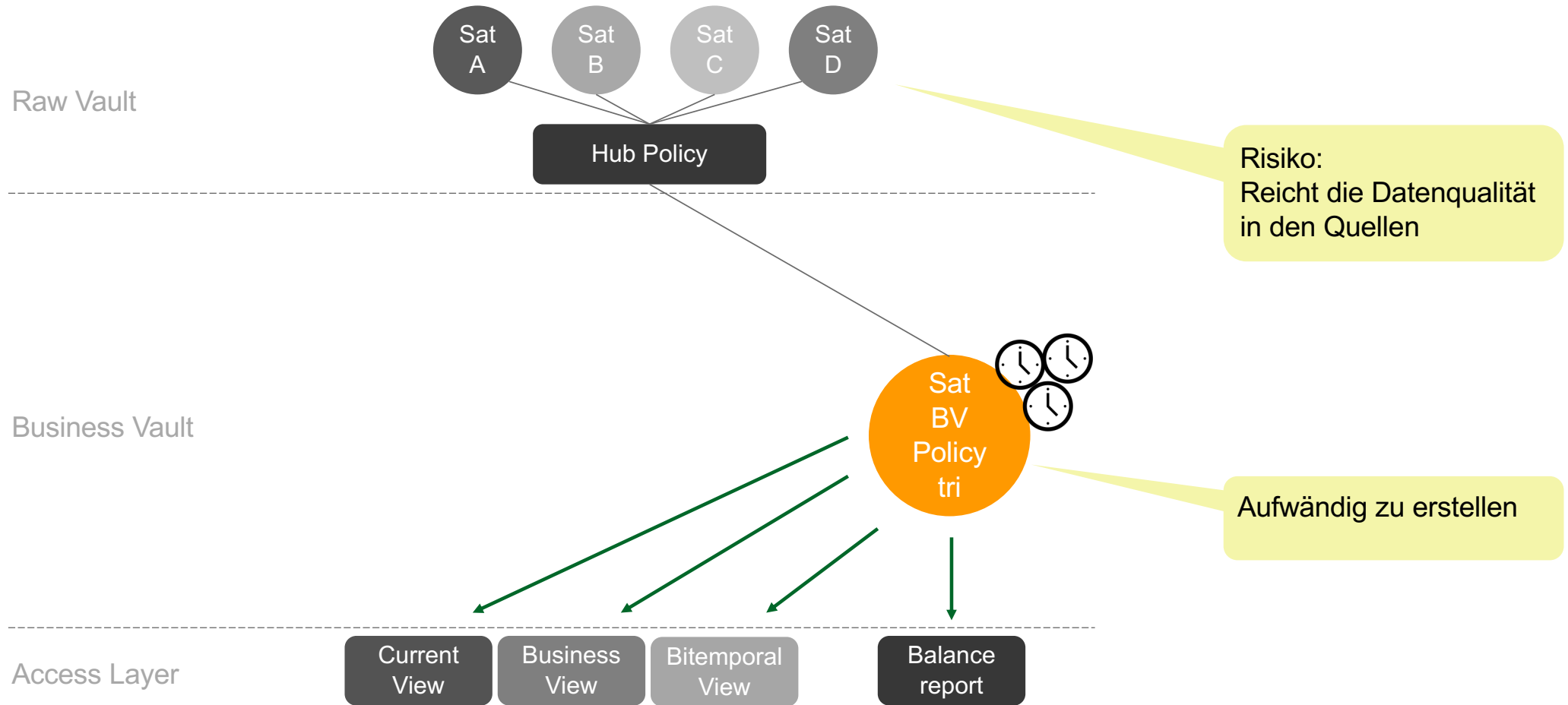
Technische Ausgangssituation im Business Vault



Iterativer Aufbau verursacht Probleme



Gleich dreidimensional entwickeln verursacht andere Probleme



Ein weiteres Teil im Data Vault Puzzle

*Verknüpfung
(Link)*

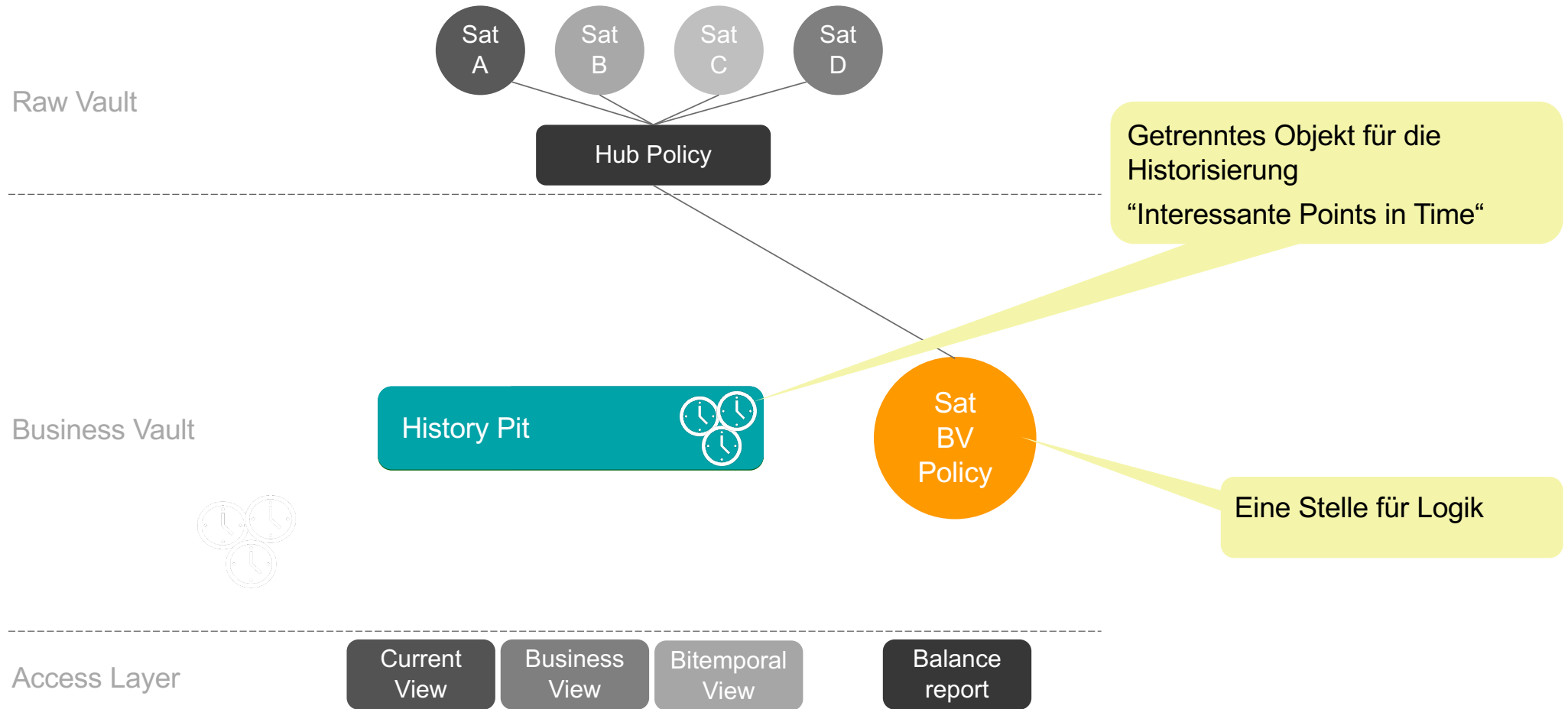
*Identifikation
(Hub)*

*Ableitungen
(Csat)*

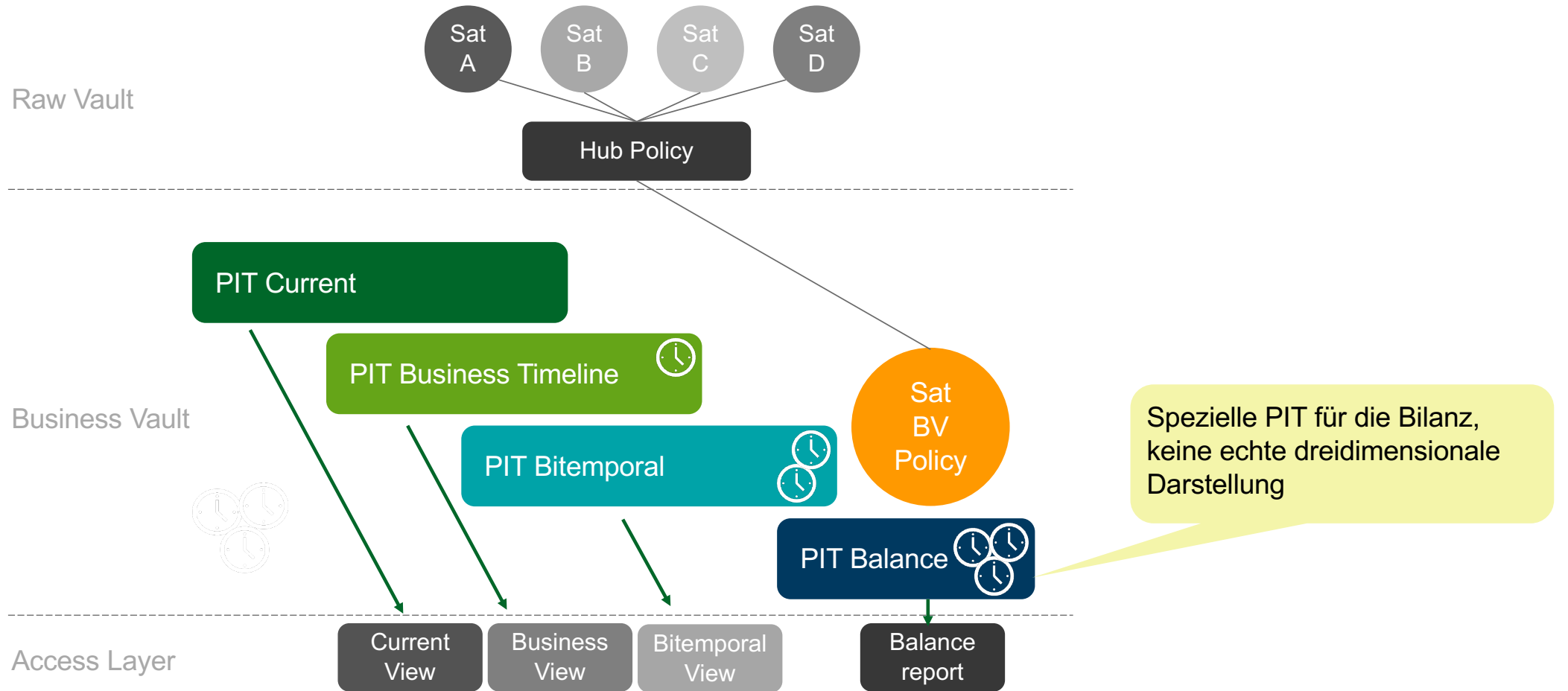
*Rohdaten
(Sat)*

Historisierung

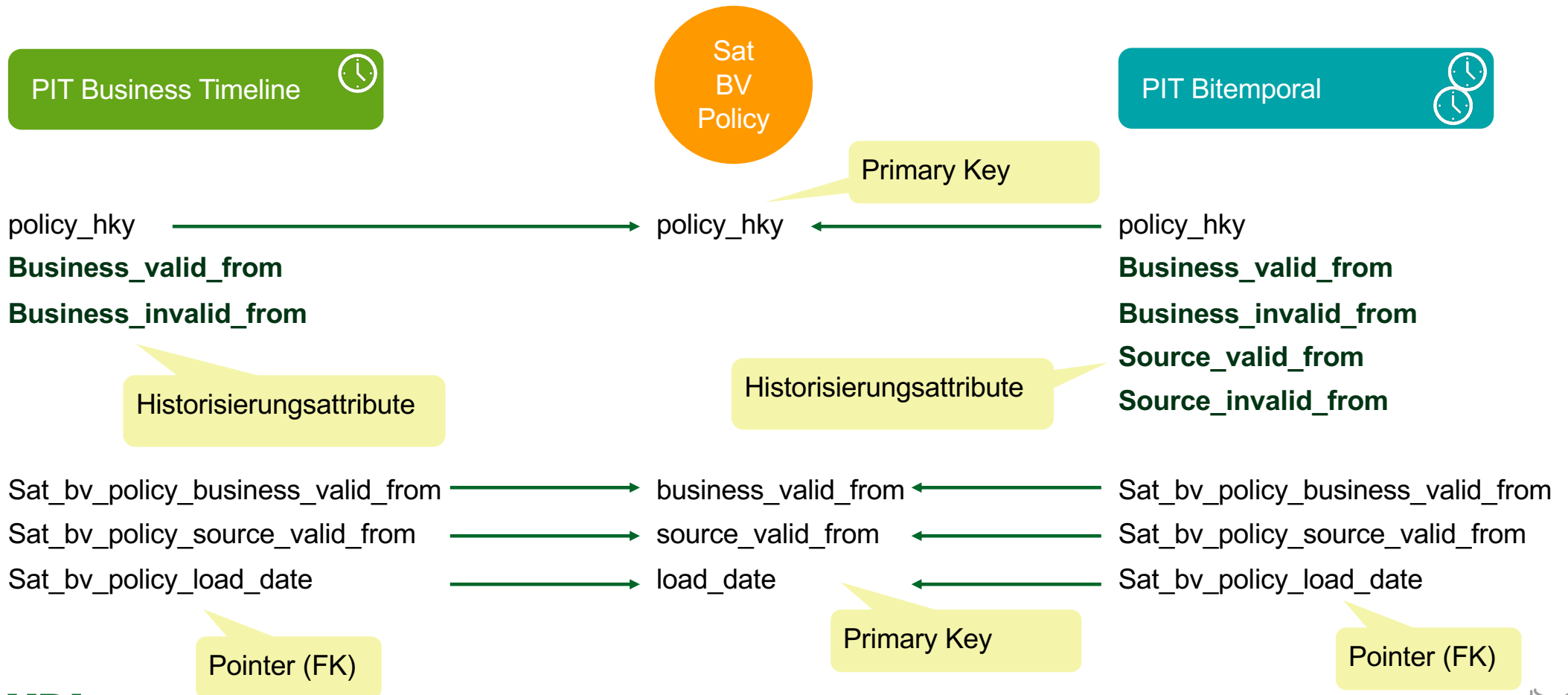
Eigene PIT Tabelle für Historisierung



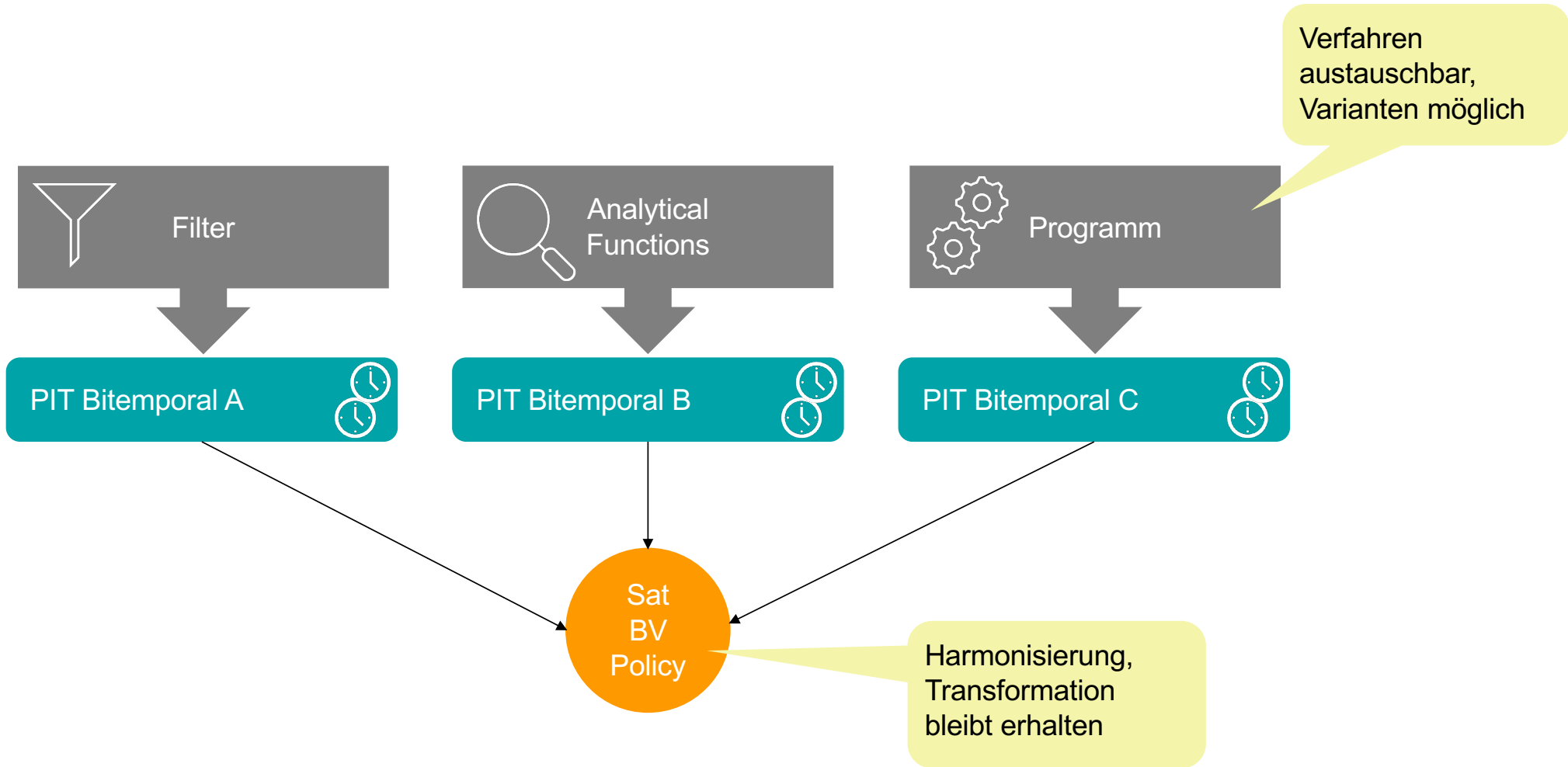
Unterschiedliche Historien bei gleicher Harmonisierungslogik



Historien-PITs zeigen auf den PK des SATs



Experimente erlaubt



Abschließende Bewertung

Einsatz empfehlenswert



Komplexe Historisierungssituation



Differierende Quellsysteme



Iterativer Aufbau gewünscht



Eher am Projektanfang

Verfahren eher ungeeignet



Eindimensionale Historisierung



Einfache Quellenlage (1 System)



Time to „market“ nicht relevant



Nicht nachträglich implementieren

3

Informationstypen in der Historisierung

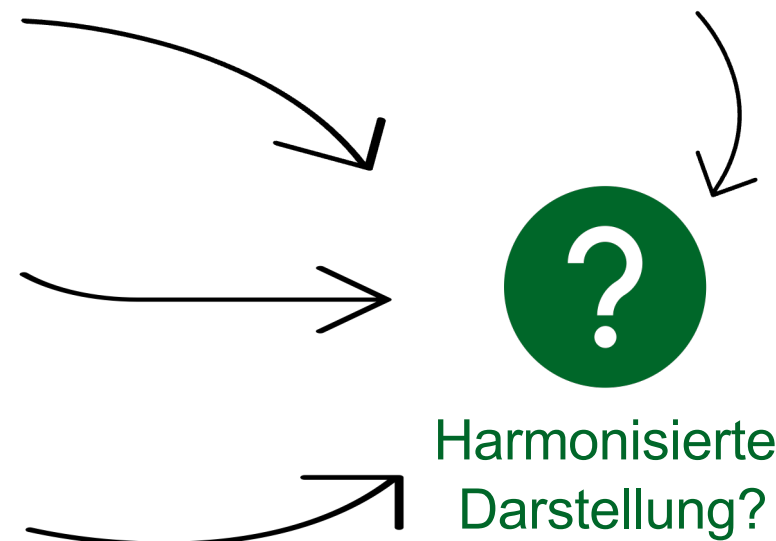
Motivation: Vereinheitlichung verschiedener Verfahren zur Historienführung

Policy	gueltig_ab	ungueltig_ab	erstellt_am	ersetzt_am
A	01.01.2024	31.12.9999	23.11.2023	17.12.2024
A	01.01.2024	01.01.2025	30.08.2024	05.12.2024
A	01.01.2025	31.12.9999	30.08.2024	05.12.2024
A	01.01.2024	31.12.9999	05.12.2024	31.12.9999

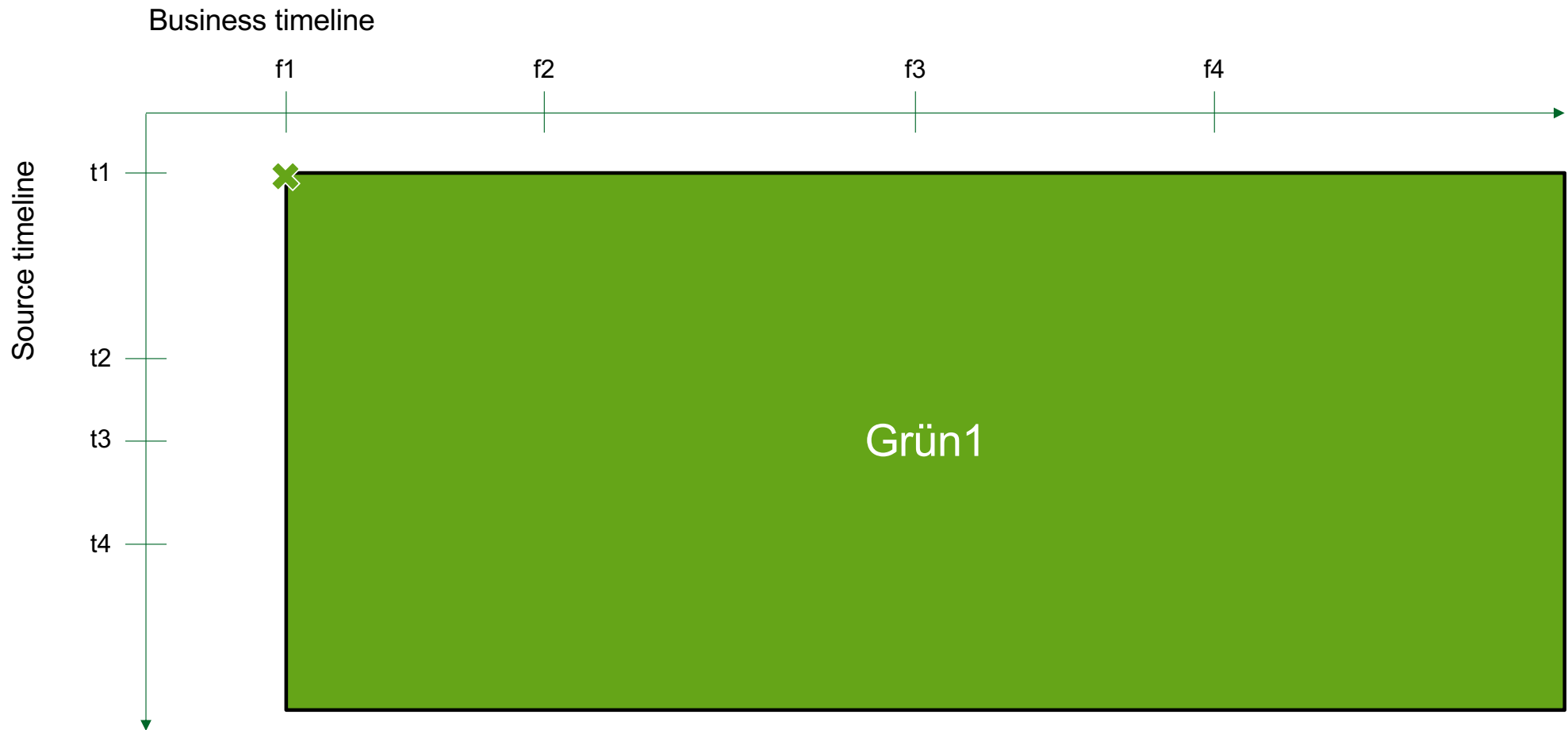
Policy	valid_from	last_modified	version_num	version_status
B	01.01.2024	23.11.2023	1V	
B	01.01.2025	30.08.2024	2V	
B	01.01.2024	05.12.2024	1I	
B	01.01.2025	05.12.2024	2I	
B	01.01.2025	05.12.2024	3V	

Policy	guelt_dat	bearbeit_dat	Nachtrag	archiviert
C	01.01.2024	23.11.2023	101	1
C	01.01.2025	30.08.2024	102	1
C	01.01.2025	05.12.2024	103	0

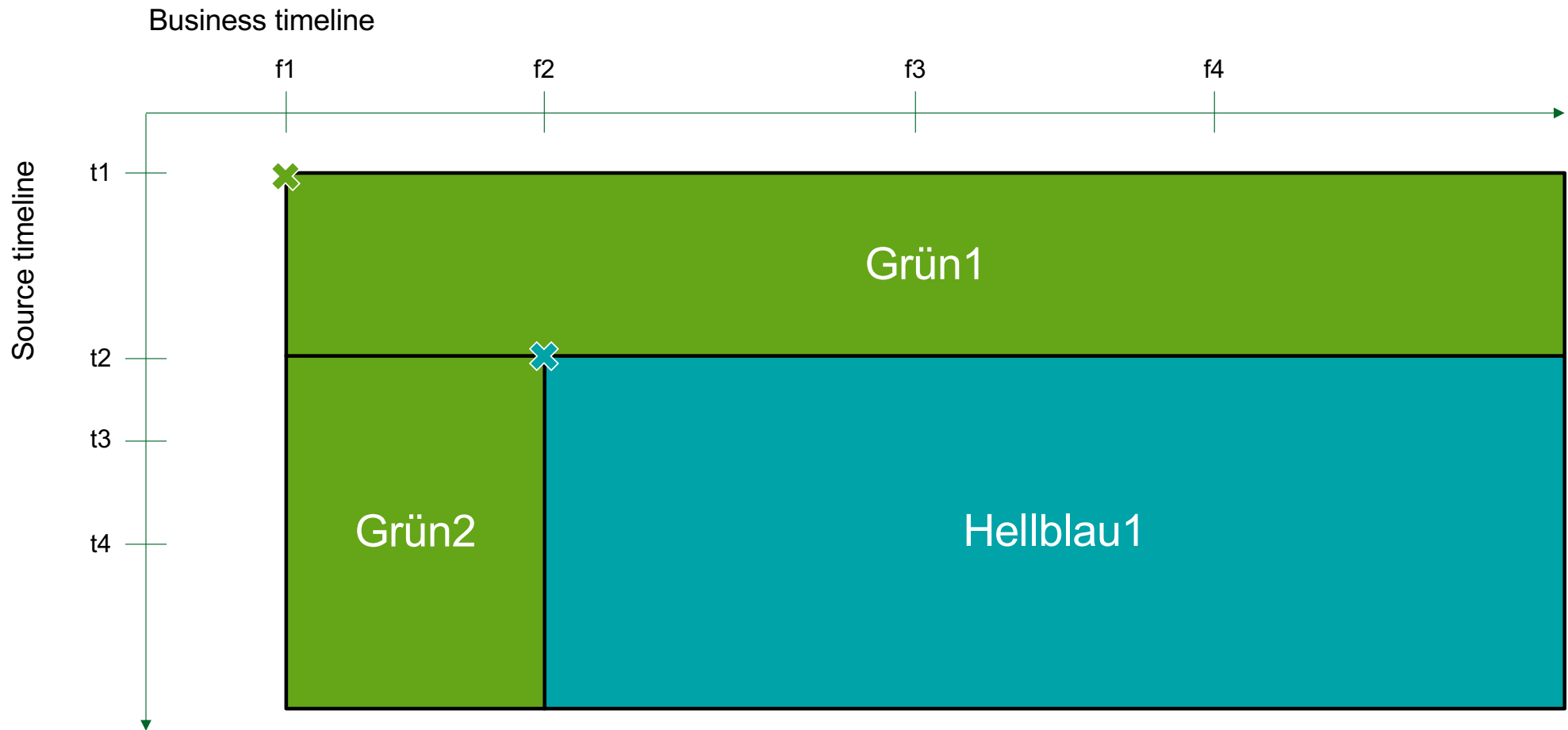
Policy	guel_dat	bearbeit_dat
D	01.01.2024	23.11.2023
D	01.01.2025	30.08.2024
D	01.01.2025	05.12.2024



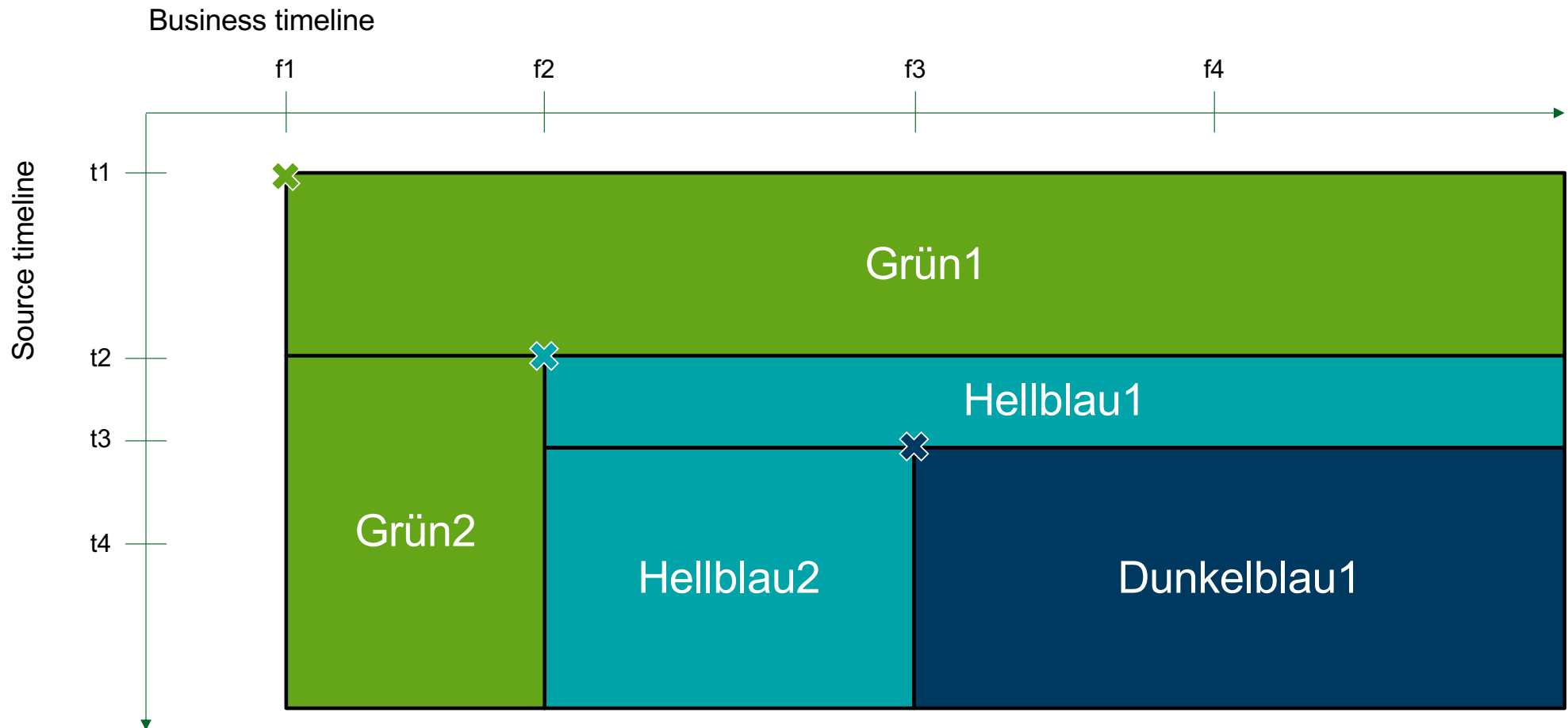
Bitemporale Historisierung



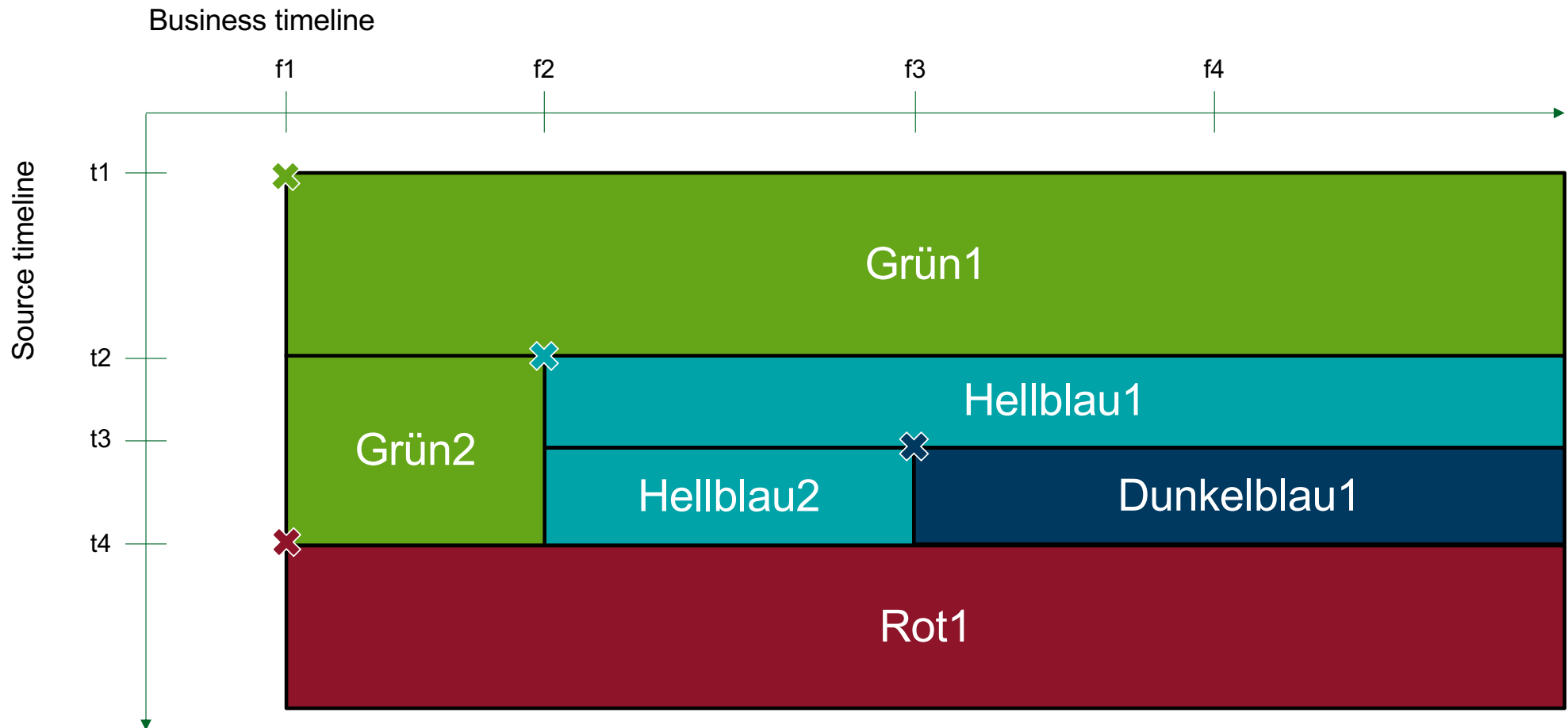
Bitemporale Historisierung



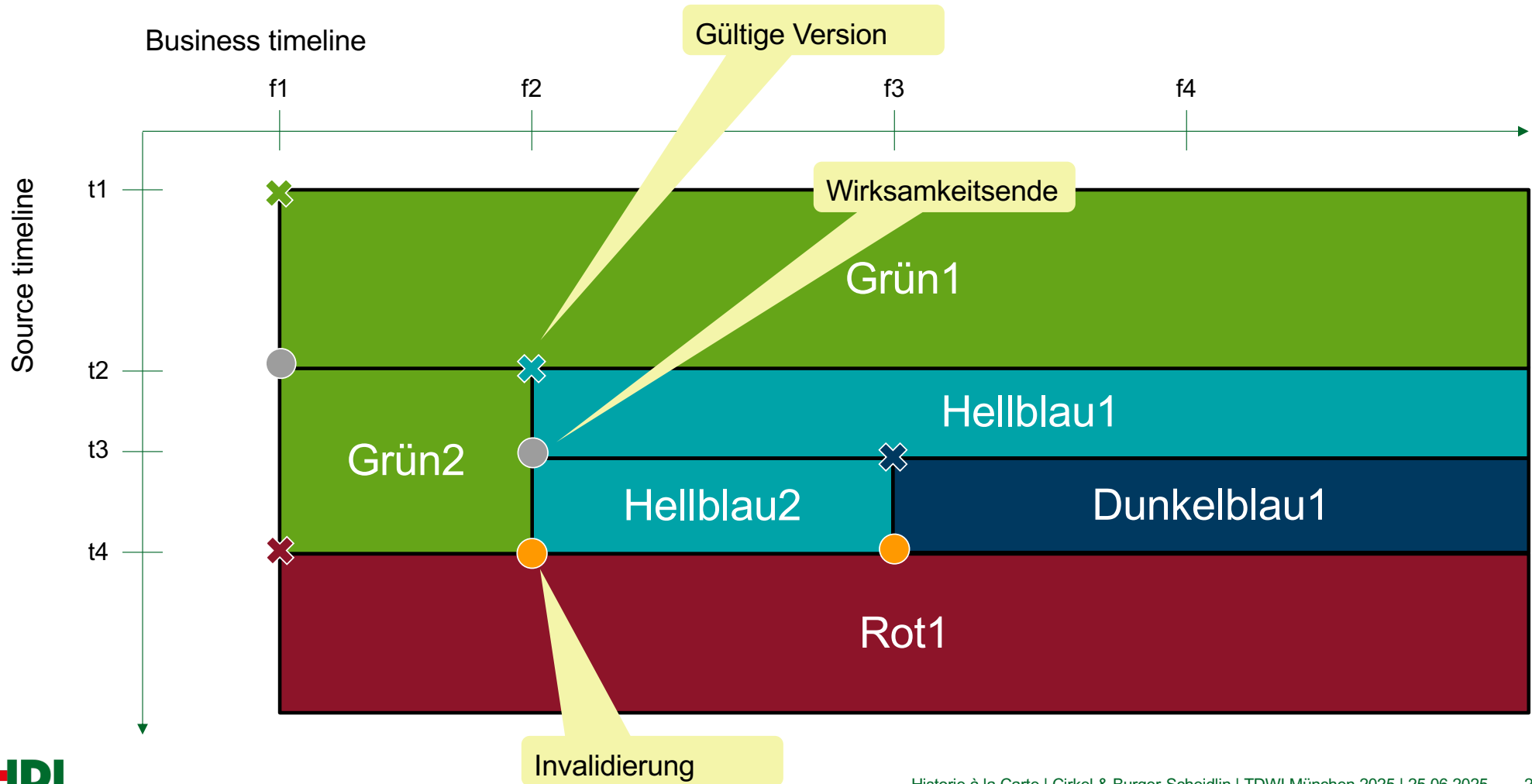
Bitemporale Historisierung



Bitemporale Historisierung



Was sind relevante Zeitpunkte?



Ziel: Historisierung als Basisinformationen speichern

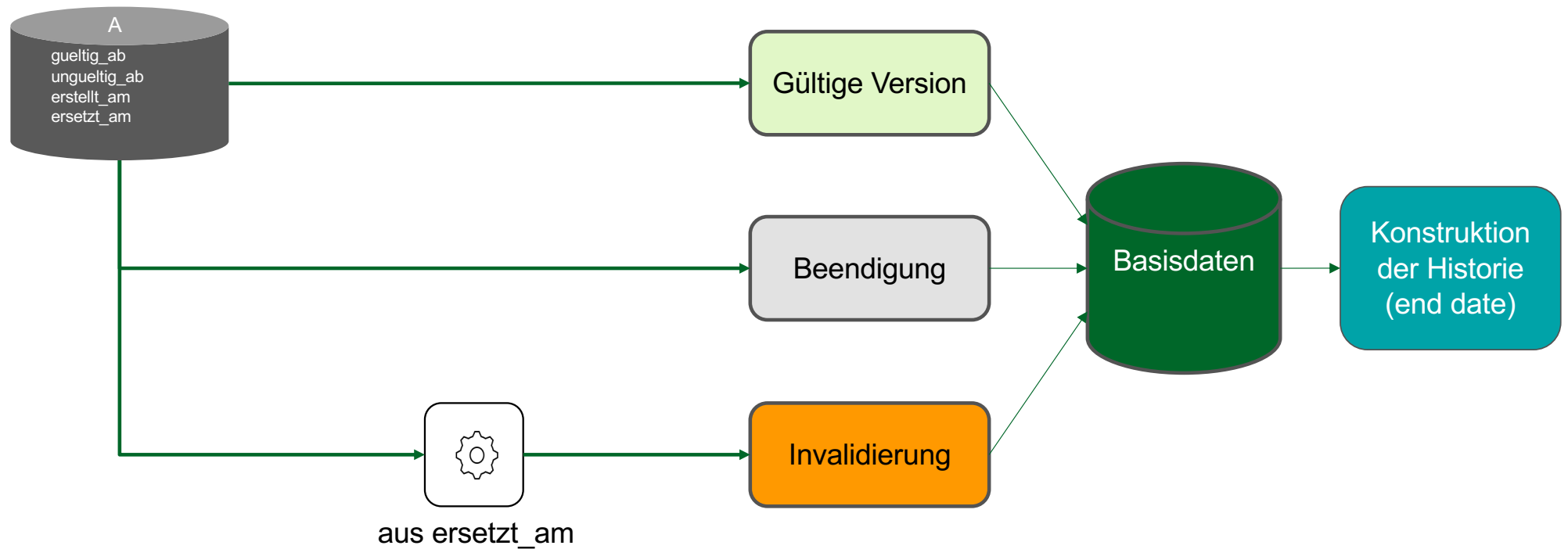
Policy	valid_from	modified_at	version_status
4711	01.01.2024	23.11.2023	Created
4711	01.01.2024	30.08.2024	Ended
4711	01.01.2025	30.08.2024	Created
4711	01.01.2024	05.12.2024	Invalidated
4711	01.01.2025	05.12.2024	Invalidated
4711	01.01.2025	05.12.2024	Created

Erste Version

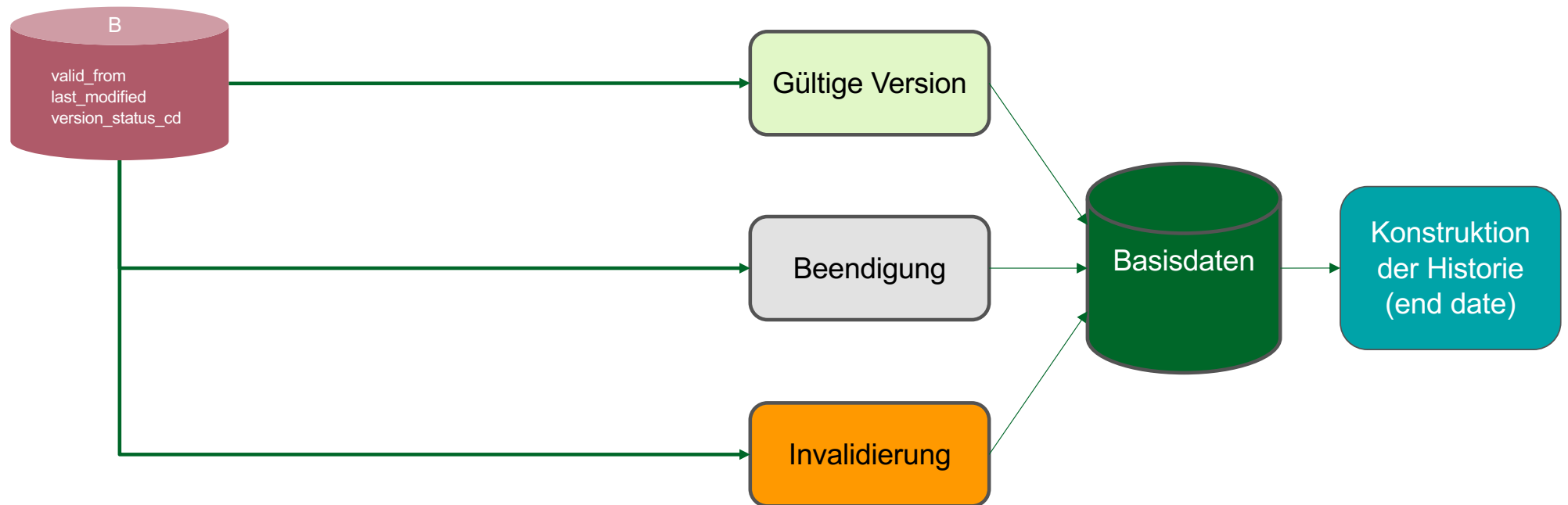
Klassischer Historisierungsvorgang

Rückwirkende Änderung

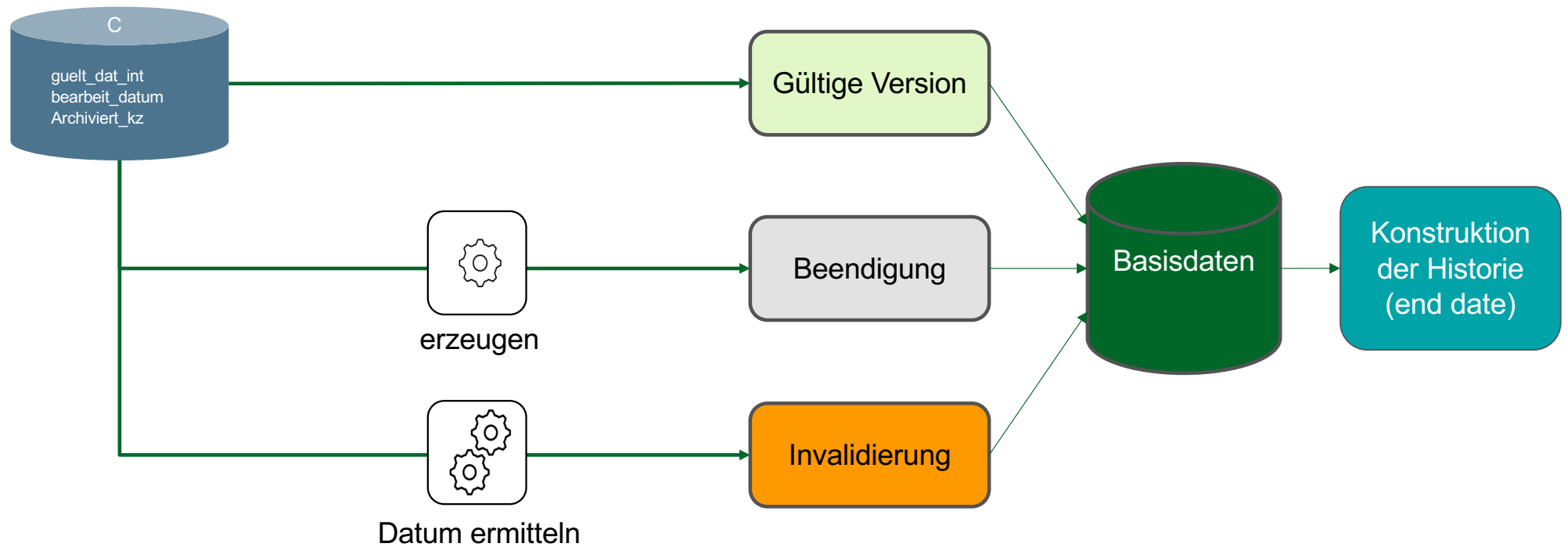
Beispiel Quelle A: Bitemporal historisiert



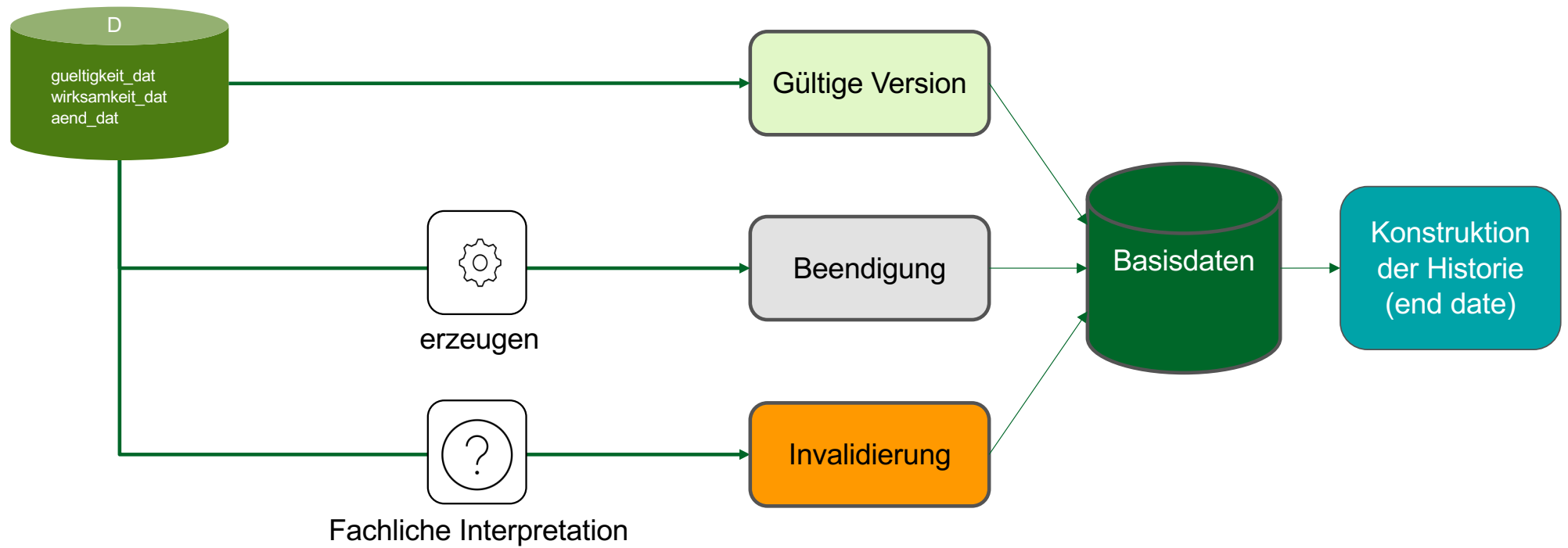
Beispiel Quelle B: Mit Versionsstatus



Beispiel Quelle C: Archivierte Sätze



Beispiel Quelle D: Zwei Zeitstempel



Konstruktion der end dates

```
WITH
  datenbasis as (
    ...
  ),
  source_timeline as (
    select
      d.*
      , lead(d.source_valid_from) over (partition by d.hky, d.business_valid_from
                                       order by d.source_valid_from asc, d.version asc) as source_invalid_from
    from datenbasis d
  ),
  business_timeline as (
    select
      hky
      , version
      , status
      , source_valid_From
      , source_invalid_from
      , business_valid_from
      , lead(business_valid_from) over (partition by hky, source_valid_from
                                       order by business_valid_from asc, version asc) as business_invalid_from
      , attribute
      , kommentar
    from source_timeline
    where coalesce(source_invalid_from, '9999-12-31') > source_valid_from and status <> 'Invalidated'
  )
select * from business_timeline
```

Schritt 1: Source Timeline

Schritt 3: Business Timeline

Schritt 2: Invalidierungen
entfernen

Feedback zu Historie à la Carte (Mi 9.3)

Timo Cirkel & Markus Burger-Scheidlin
TDWI-Konferenz München | 25.06.2025 11:25 – 12:10

