2150 GmbH

Datavault BUILDER

DDVUG ONLINE: WIE DIE ZEIT VERGEHT (BI- UND TRI-TEMPORALITÄT IM DATA VAULT)

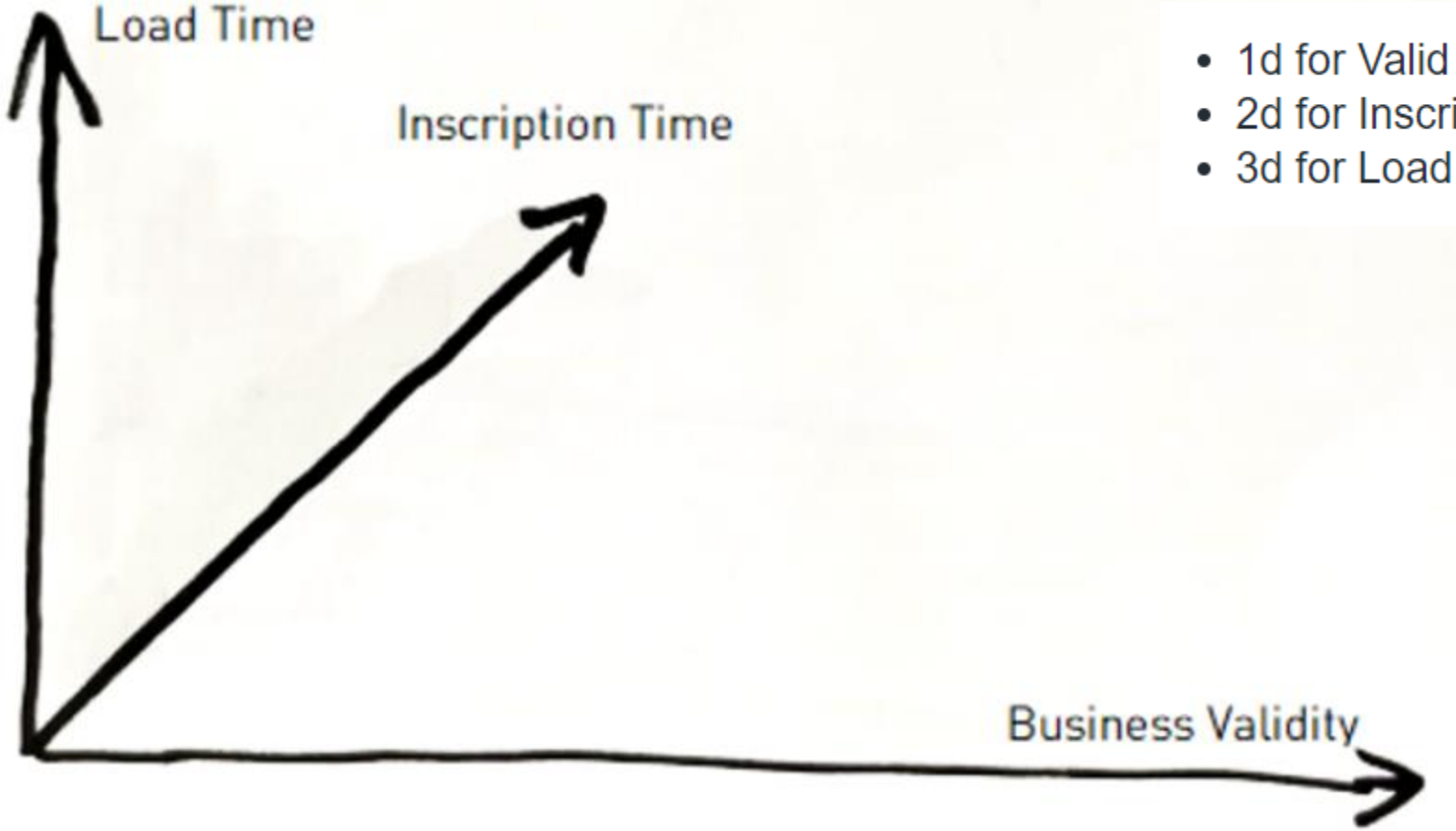
HERZLICH WILLKOMMEN

Petr Beles

- Gestartet im Finance Umfeld
- Data Management im Telco Bereich
- BI Consulting
- Aktuell Pre- und Post-Sales bei 2150 Datavault Builder
- https://www.linkedin.com/in/petr-beles-8a49531/

Datavault BUILDER

ZEIT IM DATA VAULT MEINE TERMINOLOGIE



- 1d for Valid Time
- 2d for Inscription Time
- 3d for Load Time

Begriffe für Zeitlinien: Christian Kaul "O tempora, o mores":
https://datavaultusergroup.de/wp-content/uploads/DDVUG-%E2%80%93-Christian-Kaul-%E2%80%93-O-tempora-o-mores.pdf

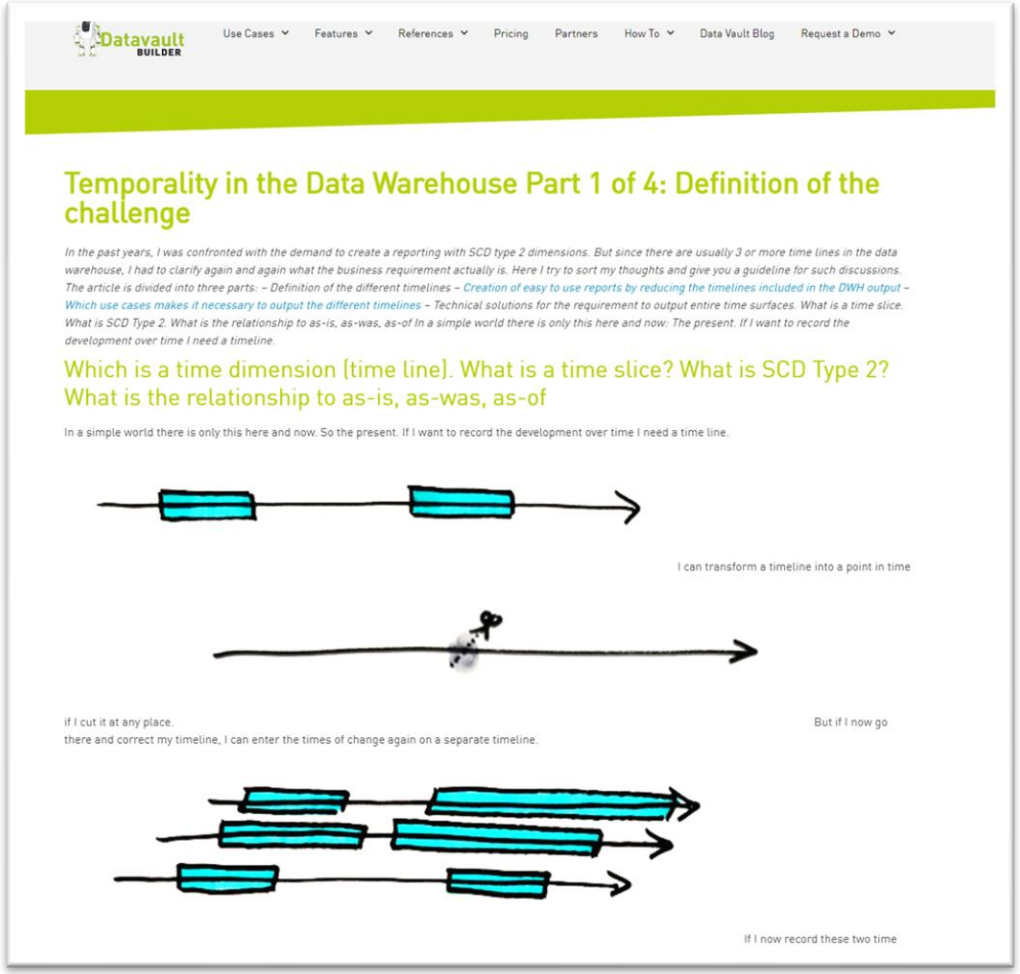
Datavault BUILDER

ZEITEN IM DATA VAULT

Als Ausgangslage
https://datavault-builder.com/data-vault-blog/

Wir haben festgestellt, dass gemäss Business Anforderungen, die “Inscription Zeit” für uns relevant ist.

Wir wollen einen Prozess etablieren, der im Betrieb viele Fälle korrekt abhandeln kann und Audit-Fähig ist



Datavault BUILDER

EINSCHRÄNKUNGEN

- Reduktion von Zeitlinien / Zeitflächen im Blog behandelt

- Ich betrachte nicht, wenn zwei technische Zeitlinien relevant sind
  - Beispiel: Bi-Temporale Daten aus dem Data-Lake in Tri-Temporal Satelliten zu laden

- Ich betrachte nicht zeitliche Abfragen zum Berichtzeitpunt

Datavault
BUILDER

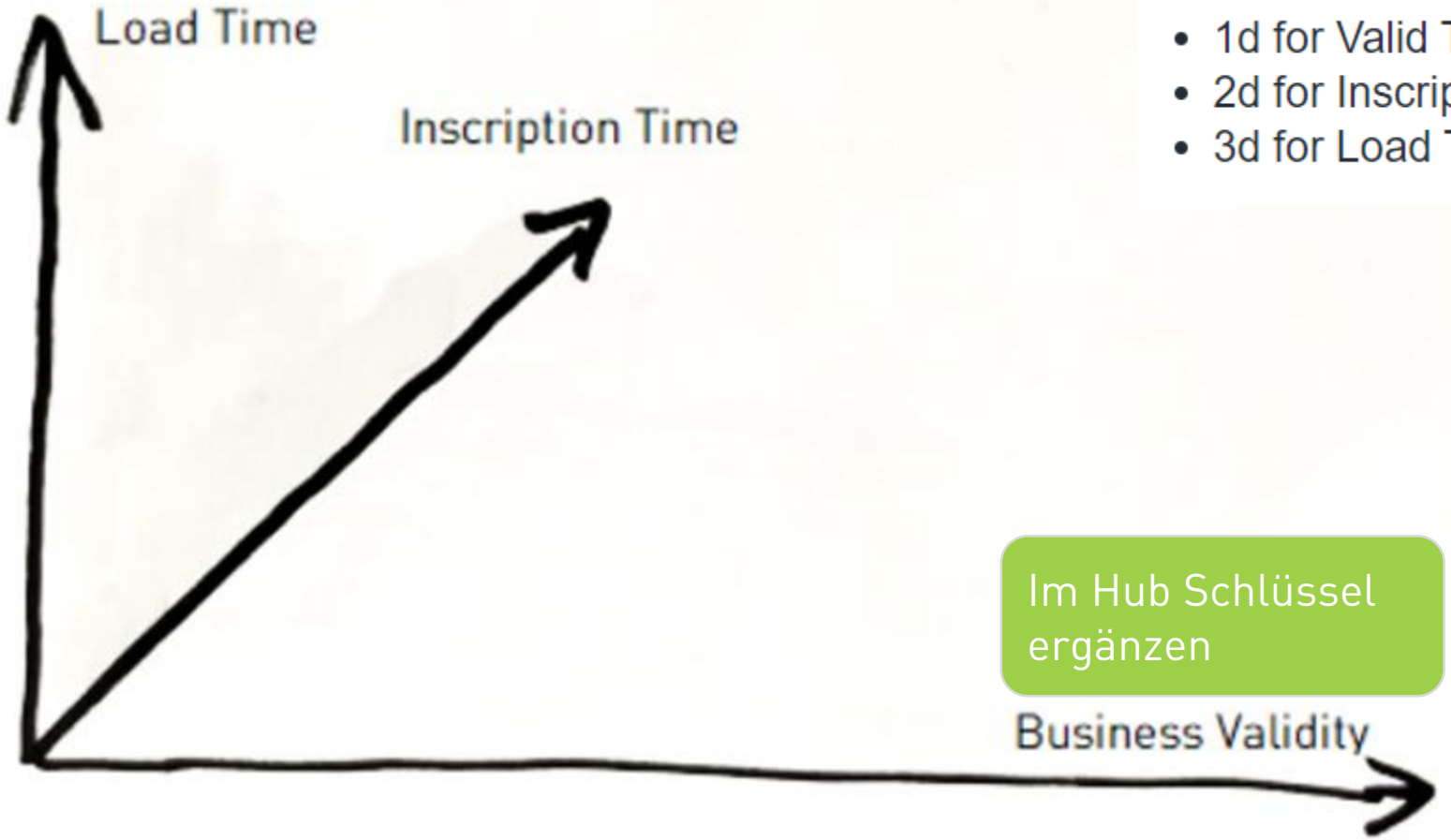
RELEVANZ

- Banken und Versicherungsumfeld (TEV / EOB / EOD)
- Andere regulierte Unternehmen
- Unternehmen mit CDC Datenquellen
- Unternehmen mit File-Transfers
  - Verzögerung
  - Korrekturen

Datavault BUILDER

GRUNDKOMPLEXITÄT

TRITEMPORAL

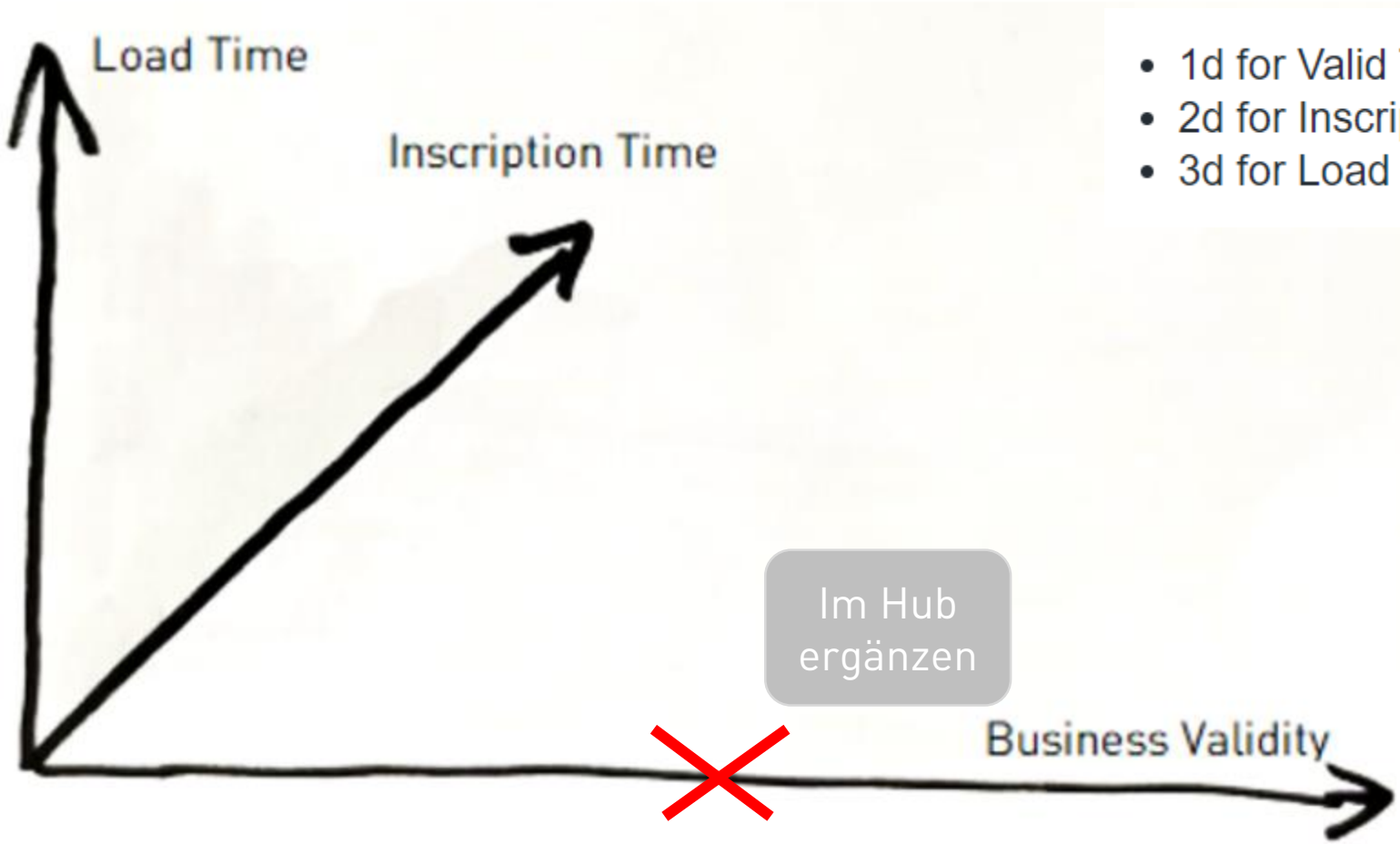


- 1d for Valid Time
- 2d for Inscription Time
- 3d for Load Time

Im Hub Schlüssel ergänzen

Datavault BUILDER

HEUTE IM FOCUS



- 1d for Valid Time
- 2d for Inscription Time
- 3d for Load Time

Datavault BUILDER

HEUTE IM FOCUS: INSCRIPTION AND LOAD TIME

Export als Quelle

Vorhistorisierte Daten

CDC Stream als Quelle

Real Time Stream als Quelle

- Das Quellsystem setzt eine verbindliche “technische” Zeit wie eine Tagesendverarbeitung (auch EOD, TEV, EOB)
- Liest man eine bestehende Historie ein, muss man diese externe Zeitlinie auch verarbeiten können.
- Man erhält die Daten mit grosser Verzögerung um eine Zwischensystem
- Ist die externe Zeitlinie z.B. eine CDC Quelle, kann es aber sein, dass man in einem Batch mehrere Changes erhält
- Out of Order Daten müssen verarbeitet werden

Datavault BUILDER

LÖSUNGANSÄTZE ÜBERSICHT

- Load Time als Proxy für Inscription Time – unter umständen durch Streaming
- PSA / Data Lake – Change ID als Schlüssel
- Inscription Time als Load Time
- Bi-Temporale Satelliten* - Als Sonderfall des Multiaktiven Satelliten

* Ich glaube mich zu erinnern, dass dieses Konzept von Thomas Herzog an einer DDVUG Tagung aufgebracht wurde: https://www.linkedin.com/in/thomasherzogcubicon/

LÖSUNGANSÄTZE LOAD TIME ALS PROXY

- Sind Inscription Time und Load Time nicht weit auseinander kann man die Load Time als Proxy für die Inscription Zeit nehmen

- Man kann die Inscription Zeit dennoch als normales Attribut ablegen

- man kann nicht mehrere Datensätze in einem Batchverarbeiten, solange man kann Load Time “Magie” hinzufügt*

- Was geschieht bei Load Verzögerungen / Unterbrechungen

* Load Time “Magie”: Millisekunden auf der Load Time hinzufügen

Datavault BUILDER

LÖSUNGSANSÄTZE PSA / DATA LAKE

+ kann alle Datenspeichern
+ löst das Problem mit mehreren Changes / Batch

- nur die letzte Version pro Batch wird in den Raw Vault übernommen oder “Load-Time Magie” muss angewendet werden
- externe Zeitlinie wird nicht automatisch ausgewertet
- keine Lösung für Out-of-Order Daten

- Loading Interval in den Raw Vault kann sehr kurz sein

Datavault BUILDER

LÖSUNGSANSÄTZE INSCRIPTION ALS LOAD TIME

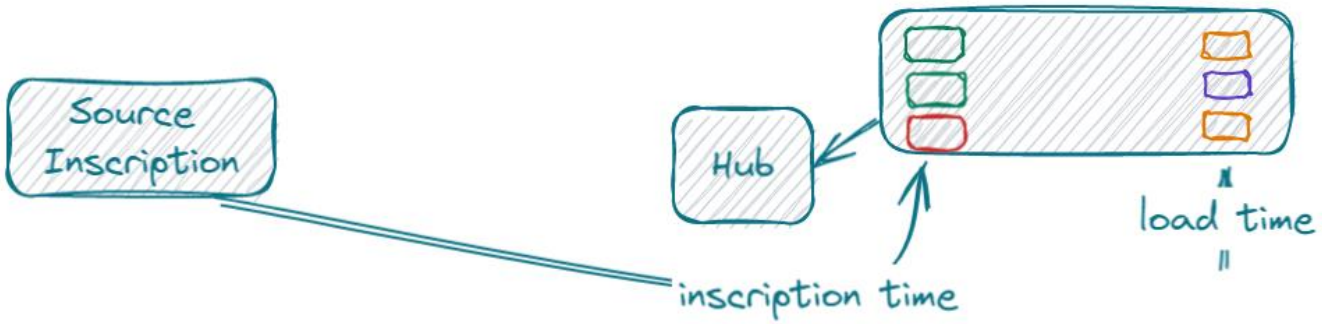
+ Funktioniert, wenn die Daten immer korrekt sind und niemals nachträglich korrigiert werden
+ Kein Problem, wenn Zeitdifferenz zwischen Inscription Time und Load Time gering ist.
+ Patterns für das befüllen nur minim anders. Abfrage Patterns identisch

- Wir ein Problem, sobald mal korrigierte Informationen für die gleiche Inscription Time bekommt → hier müsste man bereits geladene Daten überschreiben, oder die neue Version ignorieren
- Wir können nicht sagen, wann die Daten effektiv für die Reports bereitgestanden sind (wenn z.B. ein CDC Stream für wenige Tage nicht liefert)
- In der Summe nicht Audit-Fähig

Kann das eine Lösung sein bei CDC Loads? Vielleicht – aber bei unseren Testszenarien konnten wir die Annahmen, dass keine Korrekturen geliefert werden nicht halten

Datavault BUILDER

LÖSUNGSANSÄTZE BI-TEMPORALE SATELLITEN



+ Können Inscription Time und Load Time abbilden
+ Sind eine Sonderform eines Multi-Activen Satelliten und somit vom Standard abgedeckt
+ Wenn man das Inscription Zeit als führende Zeitschiene festlegt, können die Abfragen gleich sein

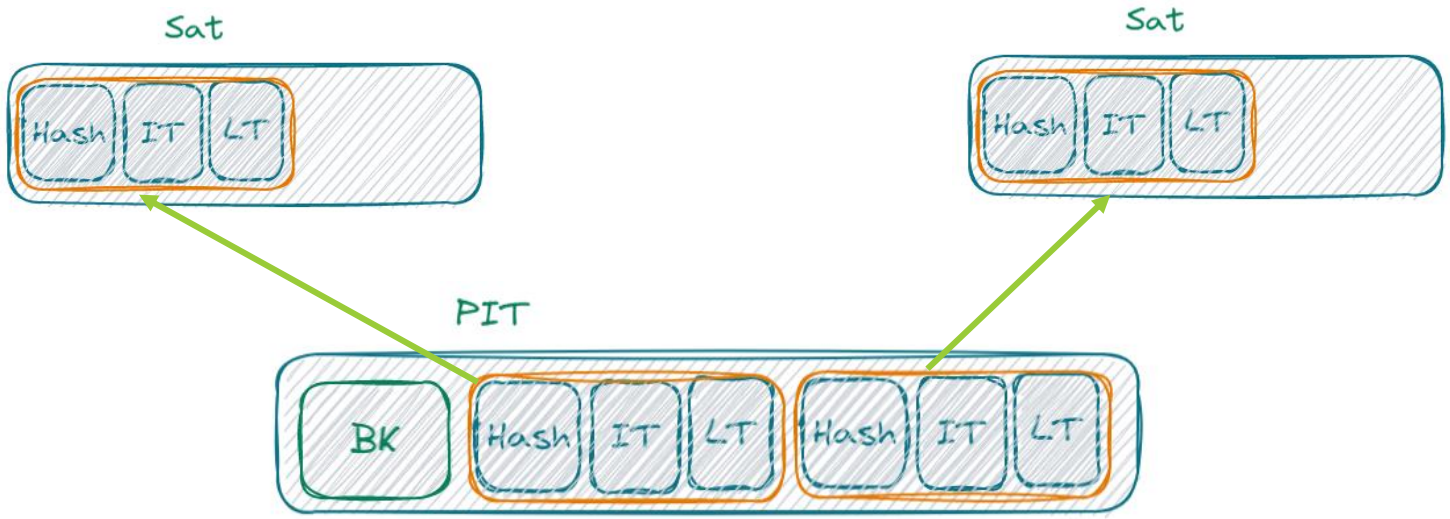
- Wir brauchen andere Patterns zum befüllen
- Unser Schlüssel setzt sich jetzt aus BK (repräsentiert durch den Hash), Inscription Time und Load Time zusammen
- Wir haben unter umständen ein Performance Issue, weil wir jetzt 3 Sichten auf den Satelliten brauchen: Bi-Temporal, Uni-Temporal (entweder Load oder Inscription Time) und As-of-Now

Datavault BUILDER

LÖSUNGSANSÄTZE BI-TEMPORAL LOAD

- BK (repräsentiert durch den Hash), Inscription Time und Load Time
- Kann als Multiaktiver Satellit verstanden werden (Hash + IT)

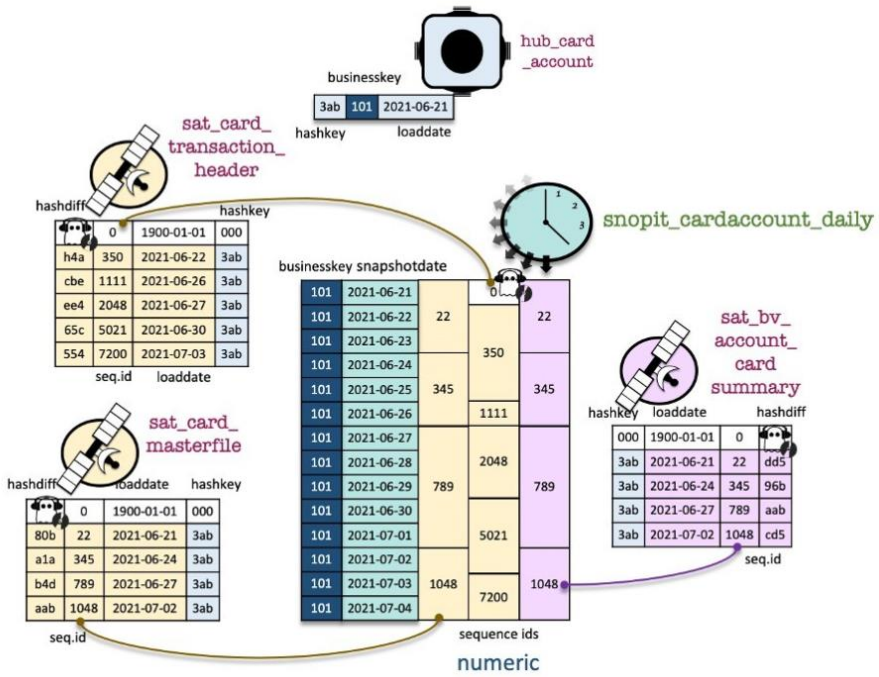
- Dies führt zu extrabreiten PIT Tabellen
- Das Joinen über mehrere Attribute ist auf vielen Datenbank ineffizient



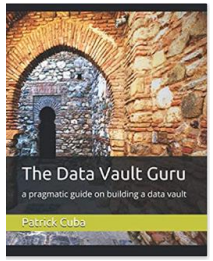
Datavault BUILDER

LÖSUNGSANSÄTZE SNOPIT

- Patrick Cuba hat ein Konzept entwicklet, was er SnoPIT nennt:
- Annahme: nur ein Prozess befüllt den Satelliten
- Man kann den Satelliten somit mit einer Indexspalte versehen, welche über eine Sequenz / Identity befüllt wird
- Man kann diese Indexspalte in die PIT übernehmen



- Das löst das Problem des JOINS über verschiedene Spalten. Zudem kann man jetzt auch den Ghost Record über die gleiche Spalte verlinken und muss in die PIT nicht mehr Hash / Satellit aufnehmen

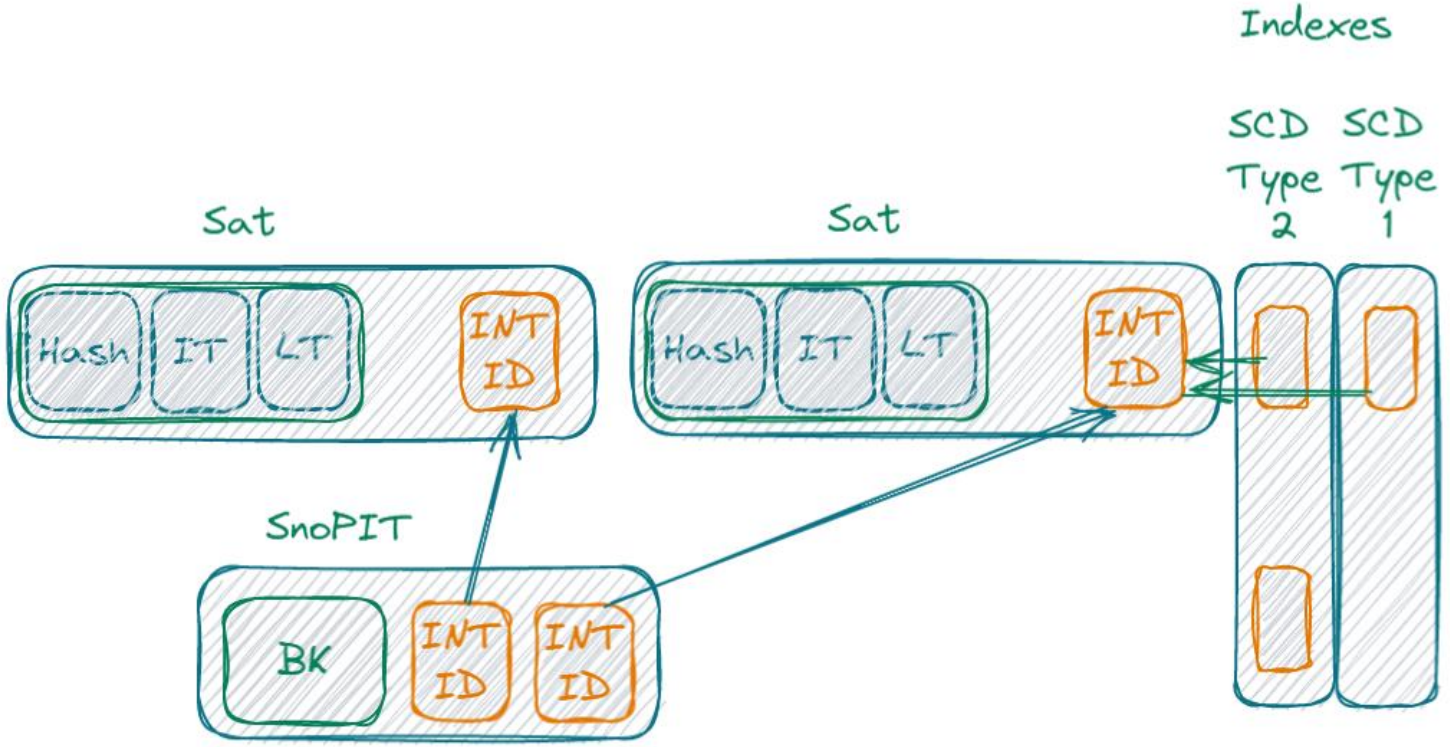


https://medium.com/snowflake/data-vault-2-0-on-snowflake-5b25bb50ed9e
https://www.linkedin.com/pulse/why-equijoins-matter-patrick-cuba/

Datavault BUILDER

LÖSUNGSANSÄTZE BI-TEMPORAL LOAD

- 1 Hash und 2 Datumsfelder werden durch INT ersetzt
- Es können auf den Satelliten sehr schlanke Indexes für verschiede Zeitsichten angelegt werden
- Die Indexe können die PIT Erstellung zusätzlich beschleunigen



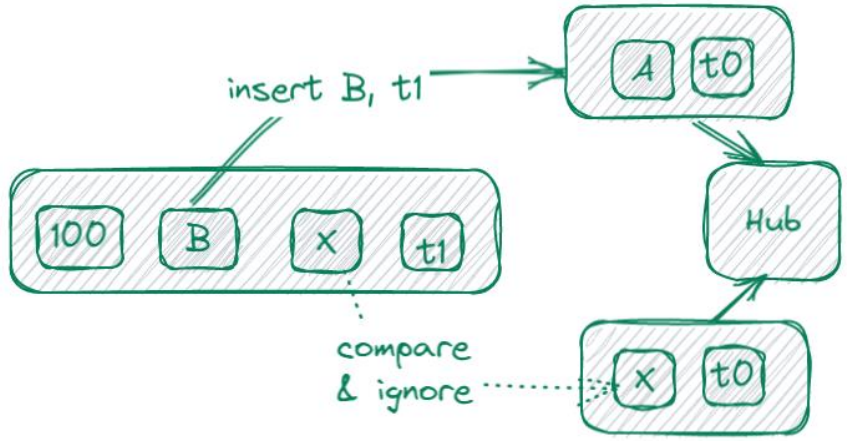
* Die Idee von Indexes für andere SCD Sichten wurde mir an einer DDVUG Tagung von Martijn Evers vorgestellt https://www.linkedin.com/in/thefullscaledataarchitect/

Datavault BUILDER

WEITER HERAUSFORDERUNGEN COMPRESSION

Man könnte annehmen, dass wenn Daten als CDC Stream geliefert werden nur effektive Änderungen beinhalten = nur ein neuer Datensatz geliefert wird, wenn effektiv etwas geändert hat. Somit muss man keine Change Detection mehr durchführen.
Leider stimmt diese Annahme nicht, weil:

- Gewisse Banksysteme Stammdaten an jedem COB/TEV liefern, auch wenn nichts geändert hat
- Bei einer Aufteilung in mehrere Satelliten evtl. nichts an den Spalten geändert hat, was im Satellit vorhanden ist



→ Man könnte also grundsätzlich das gleiche Pattern, welches man für einen normalen Satelliten verwendet wiederverwenden und nur Zeilen schreiben, falls sich ein Wert für den Satelliten geändert hat....

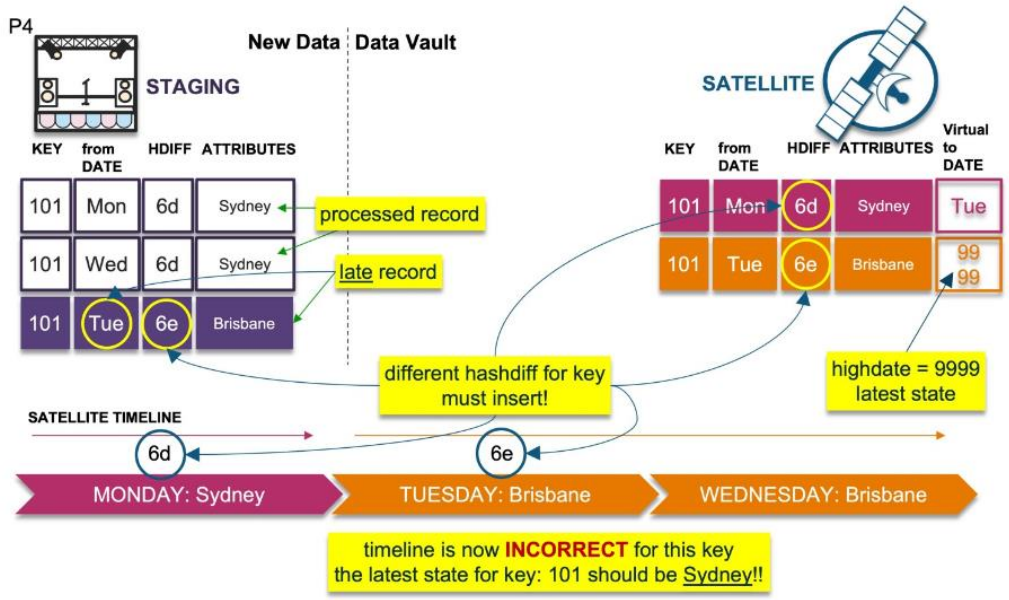
Datavault BUILDER

WEITER HERAUSFORDERUNGEN OUT-OF-ORDER

Daten kommen nicht in der richtigen Reihenfolge an, weil

- Kafka
- Ein Export fehlerhaft war und reproduziert werden muss
- Im Quellsystem auf der technischen Zeitschiene Daten rückwirkend korrigiert werden
- Ein CDC Stream unterbrochen worden ist und bereits wieder läuft, aber zur Sicherheit ein Full Load zum Zeitpunkt des Stream Restarts nachgeladen wird

Haben wir “Compression” angewendet beim Satellit schreiben erzeug das ein Problem

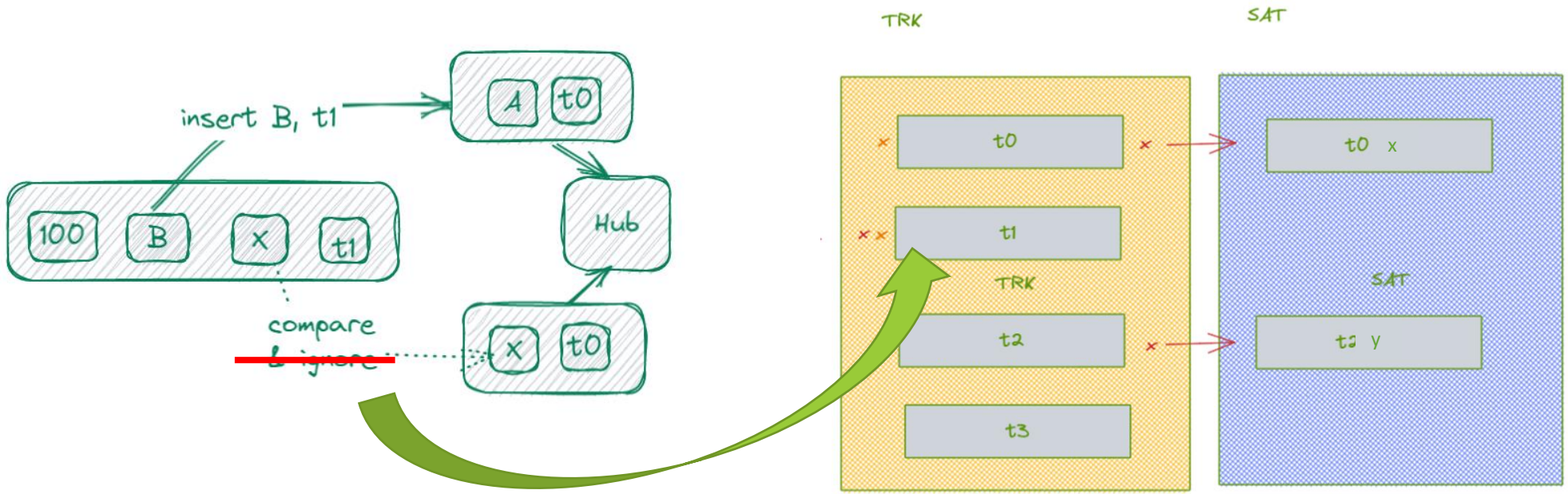


https://www.linkedin.com/pulse/data-vault-snowflake-out-of-sequence-patrick-cuba/

Datavault BUILDER

OUT-OF-ORDER UND COMPRESSION

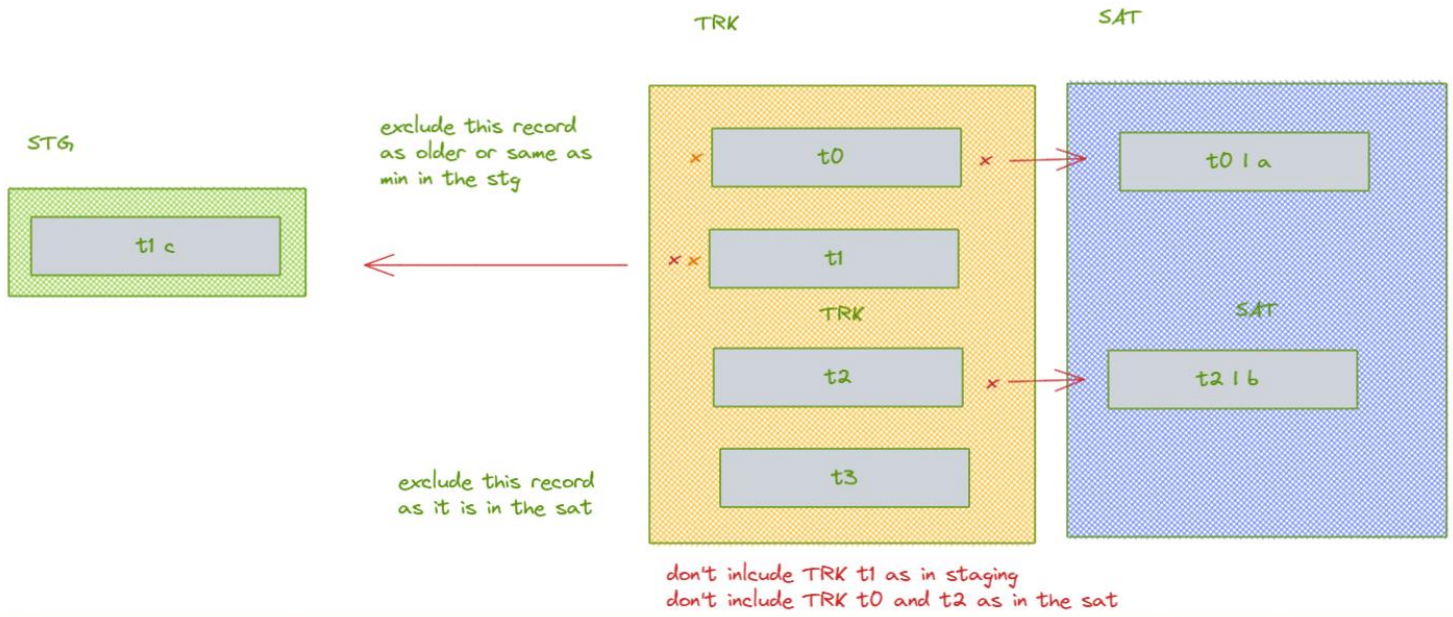
- Die Lösung ist die Daten zu komprimieren, ABER im Tracking Satelliten zu notieren, welche Zeilen man nicht geschrieben hat
- Kommt nun eine rückwirkende Änderung, muss man den wegoptimierten Zeitschnitz unter umständen wieder einschiessen



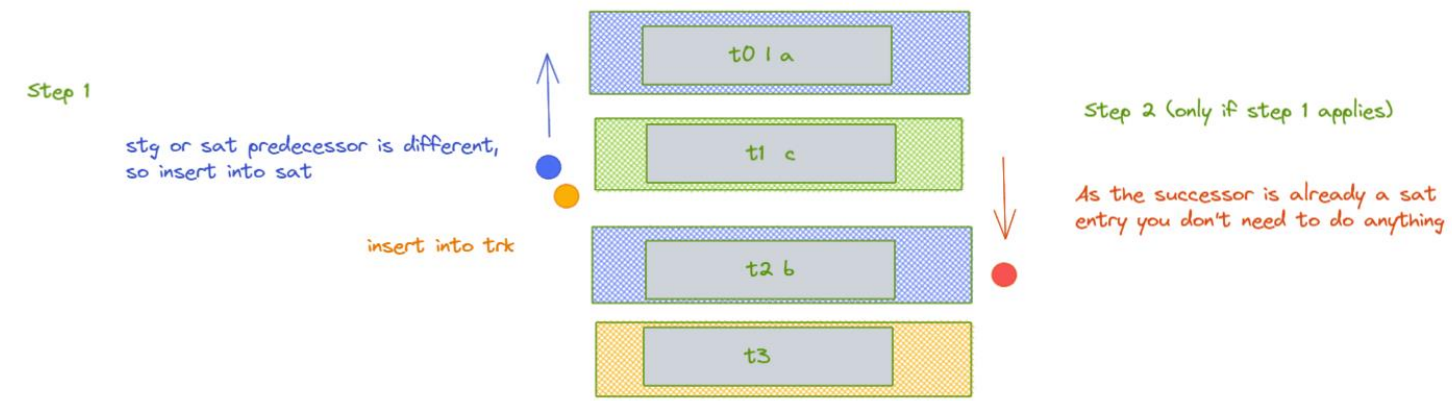
Datavault BUILDER

Use Case 000 without previous compresion

To be processed



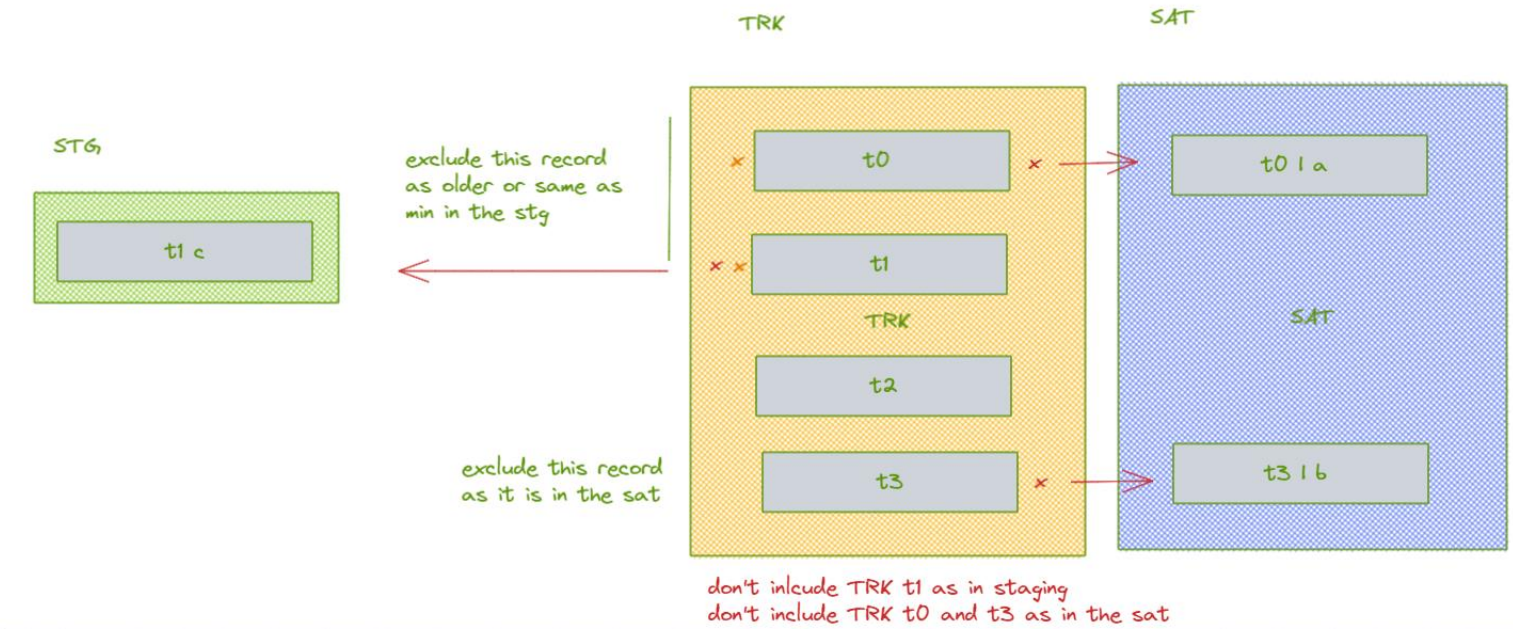
Comparison



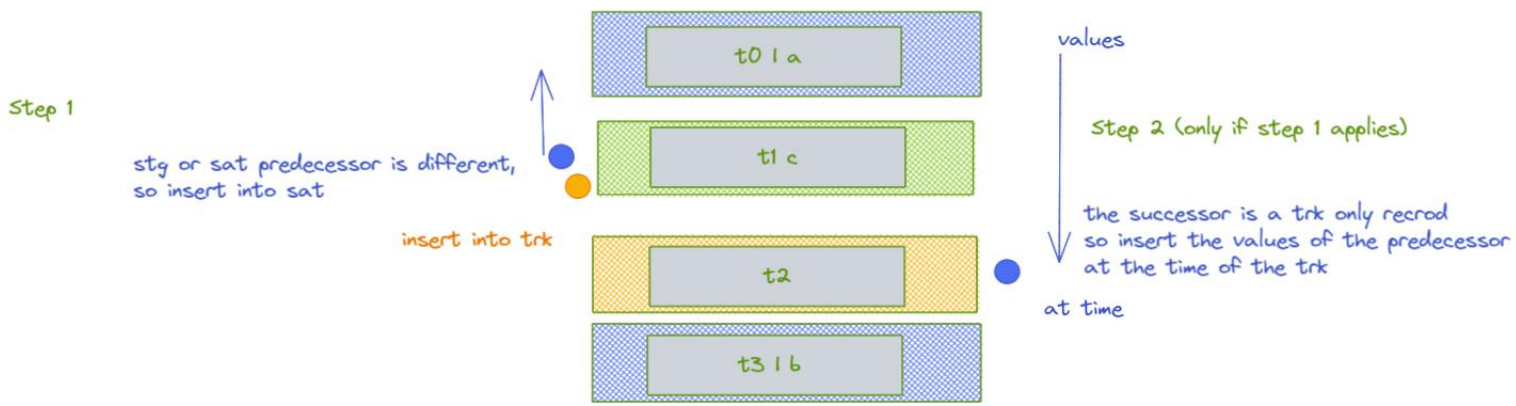
Datavault BUILDER

Use case 000 with previouse compression

To be processed



Comparison



Datavault BUILDER

OUT-OF-ORDER UND COMPRESSION

- Kompression anwenden
- Beim weglassen von Satelliten-Einträgen Tracking Satellit schreiben
- Bei Rückwirkender Änderung auf IT Zeitschiene prüfen, ob ein nachfolgender Satelliten Eintrag ergänzt werden muss

https://www.linkedin.com/pulse/data-vault-snowflake-out-of-sequence-patrick-cuba/

ZUSAMMENFASSUNG

- Prüfe zuerst, ob eine Zeitlinie für die Datenempfänger relevant ist
- Prüfe ob man durch die Reduktion von Zeitlinien Komplexität reduzieren kann
- Falls die Inscription Time Zeitlinie relevant ist, setze sie richtig um

ZUSAMMENFASSUNG

- Ist die Inscription Relevant und müssen alle möglich Fälle abgedeckt werden kann dies durch Bi-Temporal Satelliten abgehandelt werden

- Um die Performance zu verbessern kann eine Index Spalte im Satellit eingeführt werden

- Man kann diese Indexspalte verwenden, um verschiedene temporale Sichten zu beschleunigen

- Die Indexspalte verschlankt die PIT Tabellen

- Da Daten auch Out-of-Order angekommen können und komprimiert werden sollen, muss man Tracking-Satelliten verwenden, welche die wegoptimierten Zeitschnitze repräsentieren

- Kommt eine Out-of-Order Änderung an, muss man evtl. Rückwirkend einen Satelliten Eintrag einschießen

Datavault
BUILDER