

# Roboterdaten in der Google Cloud: Ein Praxisbeispiel mit Dataproc und BigQuery

Constantin Lehmann (inovex) & Oliver Frost (dmTECH)



# Wir sind dmTECH

Als IT-Tochter von dm-drogerie markt schaffen wir innovative Lösungen.

Für Kunden und Mitarbeitende in den dm-Märkten, den Verteilzentren und der Zentrale.



1.400



**Mitarbeiter:innen**

davon 80 Auszubildende  
und duale Studierende

94.000



**Kolleg:innen**

nutzen unsere IT in der  
gesamten dm-Gruppe



1988

**Gründung**

als eigenes Tochterunternehmen,  
um IT-Expertise zu bündeln



Wir freuen uns auf Ihren  
Besuch an **Stand 5** im Foyer!



# WE ARE INOVEX – INNOVATE. INTEGRATE. EXCEED.

Ihr Partner für die digitale Transformation  
mit dem Leistungsspektrum

**Data & AI**

**Application Development**

**Skalierbare IT-Infrastrukturen**

**Training & Coaching**



Ausschließlich festangestellte  
Mitarbeitende



Wachstum durch Innovationen  
und erfolgreiche Projekte



**inovex**

inovex.de





Wozu?



# Digitaler Marktzwilling

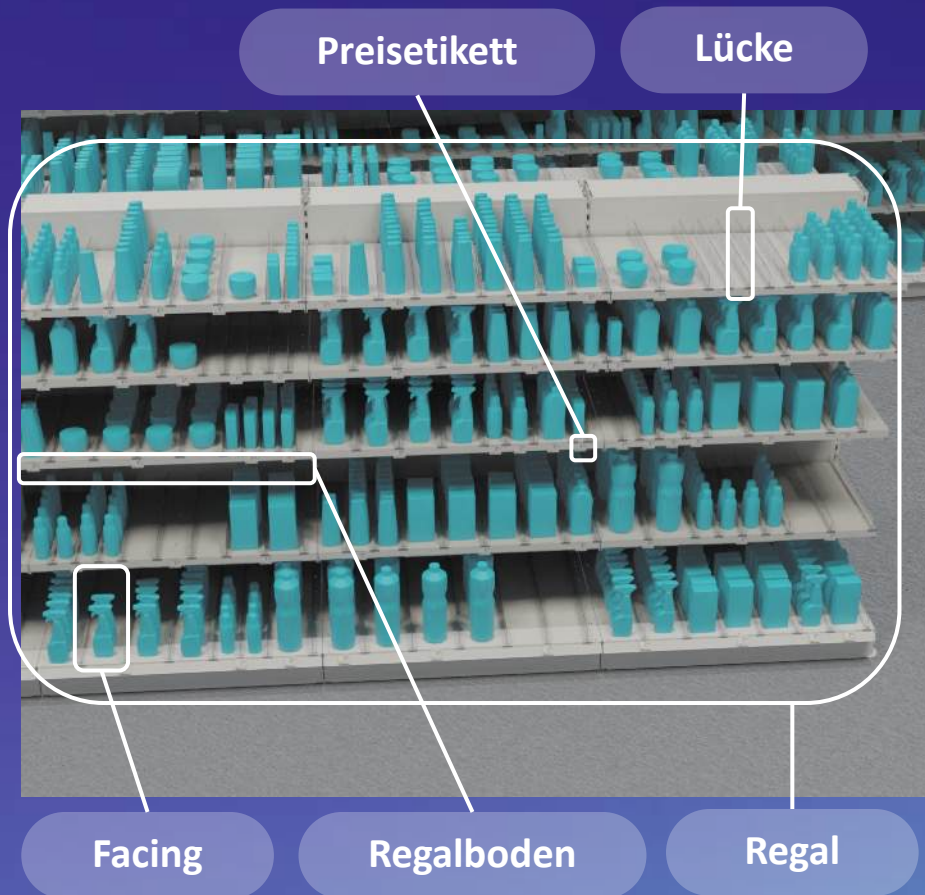


Welcher Artikel steht wo?

Wo habe ich Lücken in den Regalen?

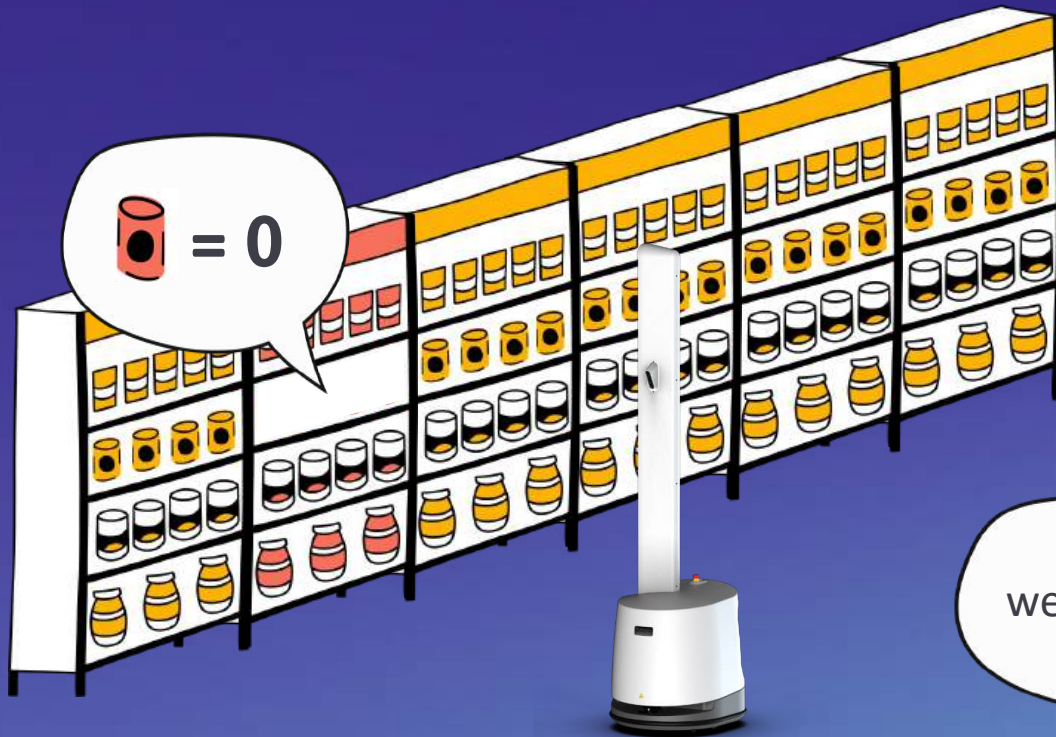
Gibt es noch Klopapier?

# Welche Daten liefert der Roboter?



**Der Roboter erkennt Artikel (noch) nicht direkt.** Welcher Artikel sich hinter einem Facing verbirgt, wird aus der relativen Position von Facing und Preisetikett abgeleitet.

# Inventur & Warenverfügbarkeit



Roboter erkennt Lücke im Regal.

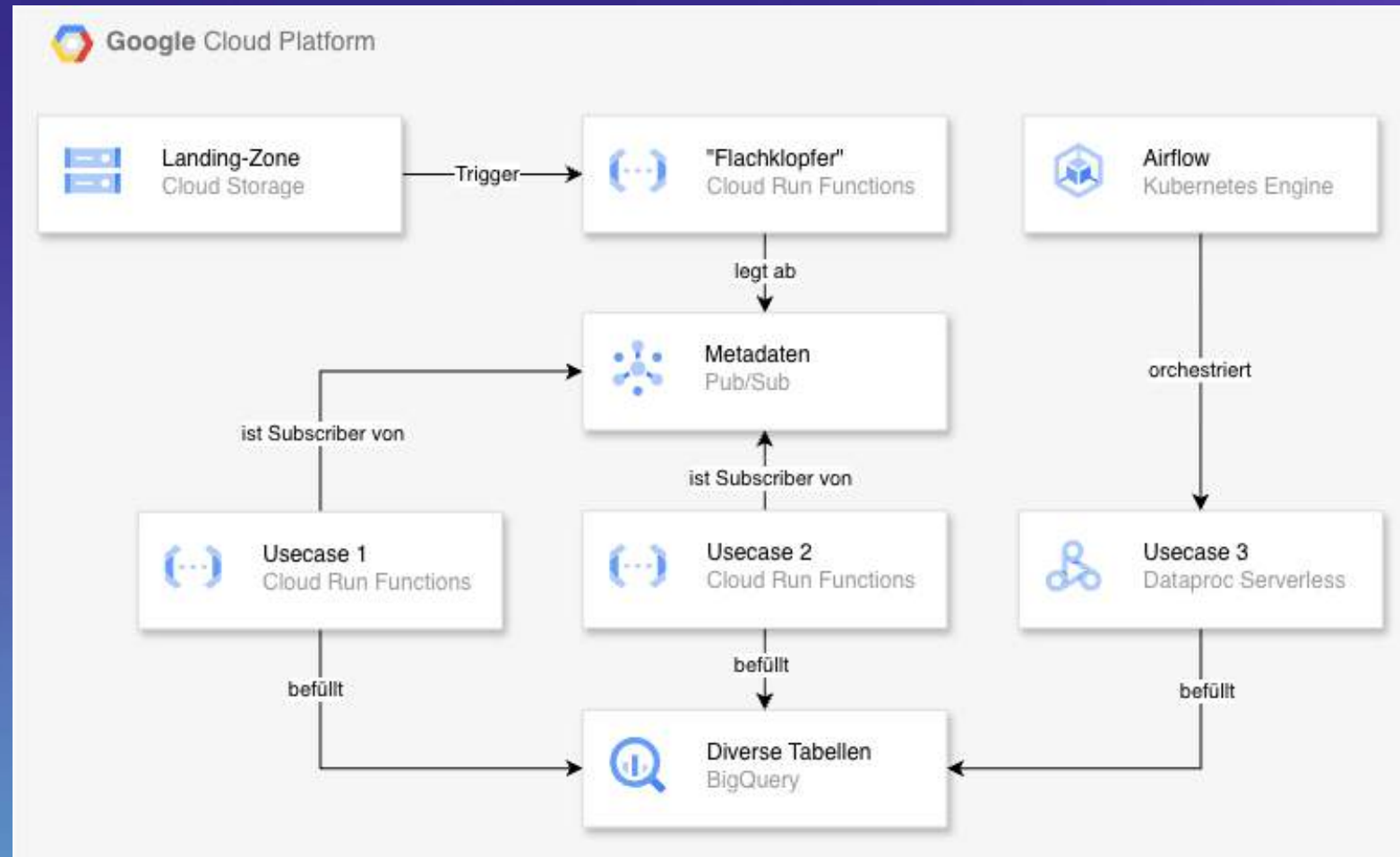
Automatische  
Inventur  
(Systembestand = 0)

Markt schließt  
Lücke.  
(Systembestand > 0)

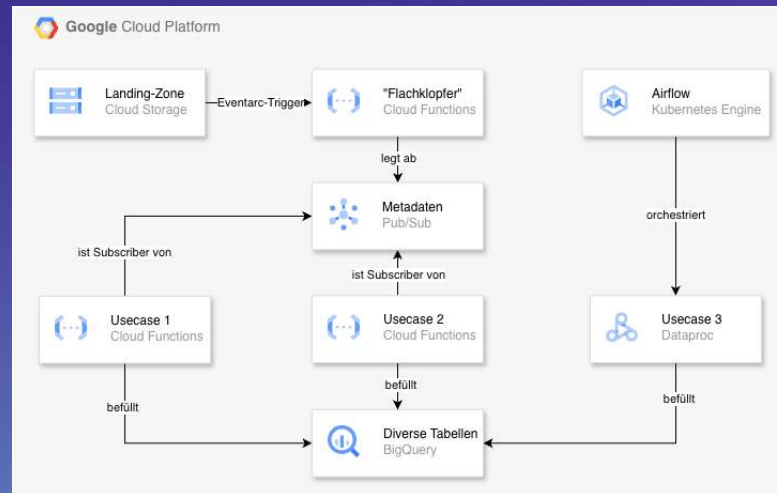
Ich muss  
weniger Artikel  
zählen.

Mein Lieblingsmüsli  
ist wieder da.

# Aktuelle Architektur



# Architektonische Altlasten



# BigQuery

Dreh- und Angelpunkt der Datenhaltung

Interne und externe Tabellen

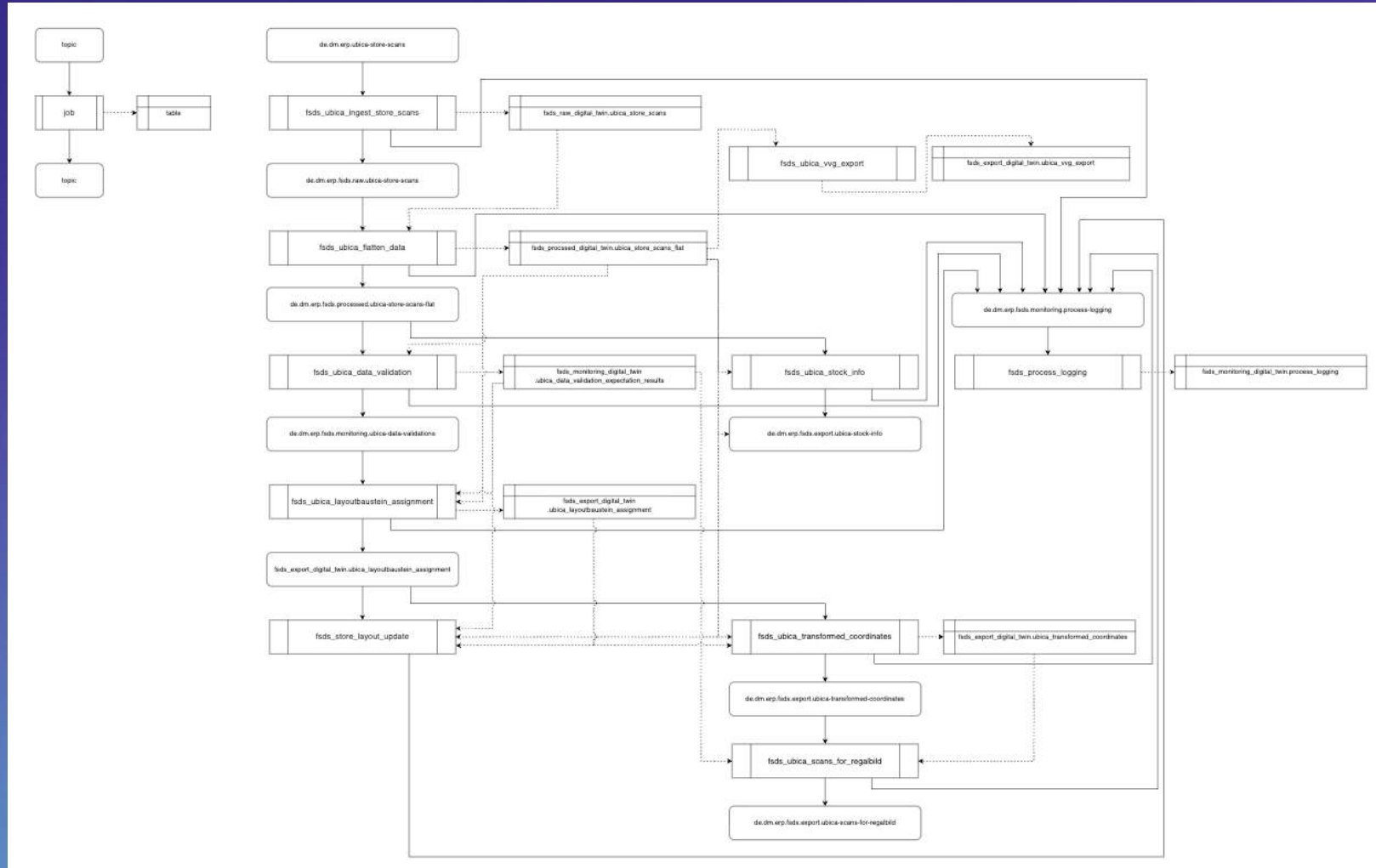
Berechtigung über Service Accounts und IAM-Gruppen

Scheduled Queries

Wie?



# Aktueller Datenfluss im OnPrem-Cluser



# Ursprünglicher Migrations-Plan

- Workshop mit Google-Engineers
- Lift and Shift (Rehosting) nach Dataproc Serverless
  - Schnell
  - Einfach
  - Günstig
- Batches anstatt von Streaming

# Erfahrungen mit Dataproc Serverless

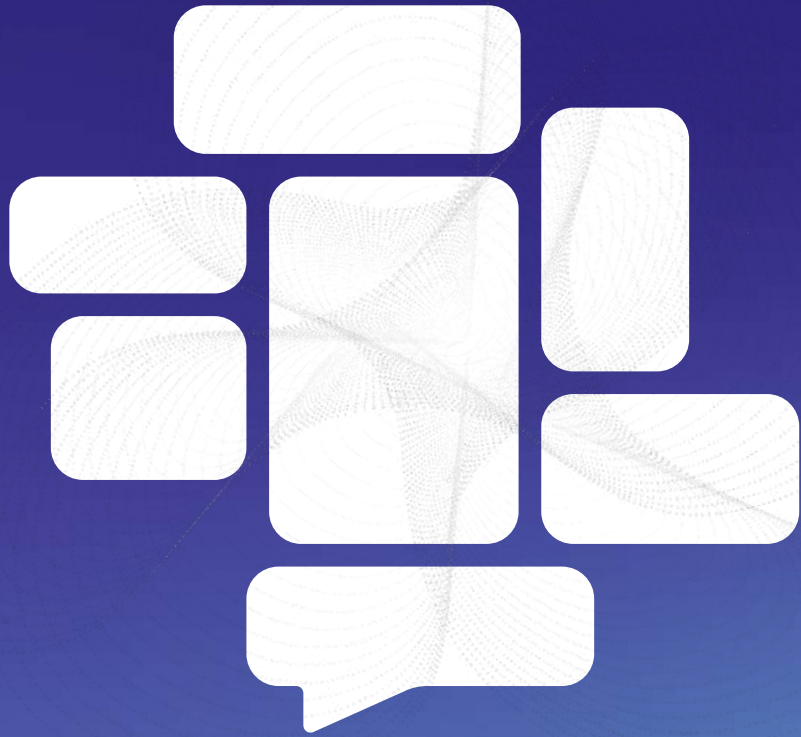
- Vorteile
  - Compute-On-Demand
  - Wenig Veränderungen am Code
  - Schnelle Migration
- Nachteile
  - Start-Up-Time (vor allem in Entwicklung)
  - Hohe Kosten (ca. 50€ pro Tag für einen Job)
  - Batches anstatt von Streams
  - Komplexe Orchestrierung der Jobs (via Airflow)
  - Unnötig viel theoretische Leistung

# Wechsel zu Cloud Run Functions

- Cloud-Nativ
- Ähnlich der Streaming-Architektur
- Mehrere Instanzen zur gleichen Zeit
- Code war einfach zu übersetzen (PySpark → Python Polars)
- Kommunikation zwischen CRFs via Pub/Sub
- Günstiger im Betrieb

# Unser Blick auf den Wechsel zu CRFs

- Wir halten die Entscheidung für sehr gut
- Vermutlich werden alle Streaming-Jobs als CRFs abgebildet
- Keine Einschränkungen im Bezug auf Use-Cases



# Q&A