



mip - mit Vertrauen rechnen

DB2 Aktuell

DB2 BLU mit Dynamic Cubes für Ad Hoc Reporting

19.09.2017
Jörg Kremer, Andreas Deutscher mip GmbH, München

Steckbrief DWH Mapis Daimler AG

Inhalt

- Das zentrale Data Warehouse des Automobilherstellers für stückzahlenbezogene Vertriebsdaten des Konzerns
 - Berechnete Lieferungen, Konzernabsatz, Retail
 - Auftragseingänge, Auftragsbestände
 - Lagerbestände, Produktionsdaten
 - Produktionsprogramme
 - Sowie Zulassungsdaten der eigenen Fahrzeuge, seiner Wettbewerber und CO2-Emissionswerte der Produkte des Konzerns.
- Über 70 teilweise hochkomplexe Input- und 30 Outputschnittstellen
- Datenbank: DB2 10.5 (produktiv) / DB2 11 (im Aufbau) (auf INTEL Maschine mit SuseLinux 64bit, 12 Kerne, 128 GB RAM, 3 TB Platte)
- ETL-Tool: Informatica Powercenter 9.1 Jobsteuerung: BMC Control-M
- Präsentationswerkzeuge: Cognos 11.x / Cognos 10.x , J2EE Portal
- Über 2000 User weltweit

→ DB-Server (DB2 10.5 FP 8)

- Intel-Maschine, mit 12 Kernen (E5-2643 v3 @ 3.40GHz)
- 128 GB RAM
- Hardware RAID 10 (12 Platten) + 1 Hotspare + 1 interne Platte (OS)
- DB-Größe: ca. 1 TB, Wachstum ca. 300 GB / Jahr

→ Cognos Server (Cognos 10.2.2)

- 1 Gateway / 1 Dispatcher (Cognos Standard) / 1 Dispatcher (Dynamic Cubes) (64 GB RAM)
(Intel, Windows 2012 Server)

Aufgaben

- Ablösung eines bisher fremdbezogenen Frontends (Zulassungsdaten) mittels Integration in das bestehende DWH Maxis.
- Nutzung der vorhandenen Standard-Komponenten
- Ad Hoc Prüfmöglichkeit der eingespielten Daten mit geeignetem Konzept
- Benutzerfreundliche Oberfläche für die Präsentation der Daten
- Integration untertägig aktualisierter interner Daten mit Ad Hoc Auswertungsmöglichkeiten

- Erstellung neue Auswertung-Schicht für Zulassungsdaten und Datenbewirtschaftung
 - Storage Layer Tabellen (row based)
 - Datamart: Zieltabellen (sowohl Fakten als auch Dimensionen) BLU
 - ETL-Prozesse mit delete bezogen auf eine Spalte (TIME_ID), danach normale inserts
 - Deletes sind in der eingesetzten Version dann sehr schnell, wenn nur auf eine Spalte in der where clause zugegriffen wird → schneller als bei row based tables!
 - Inserts sind mittlerweile ähnlich schnell, wie bei einer row based table
 - Load from cursor war demnach nicht notwendig
 - Bereitstellung von untertäglichen Datenscheiben mit ca. 2 Millionen Fakten inklusive aller Stammdatenprüfungen und Transformationen in ca. 1 Minute
 - Dynamic Cube auf die Auswertungsschicht

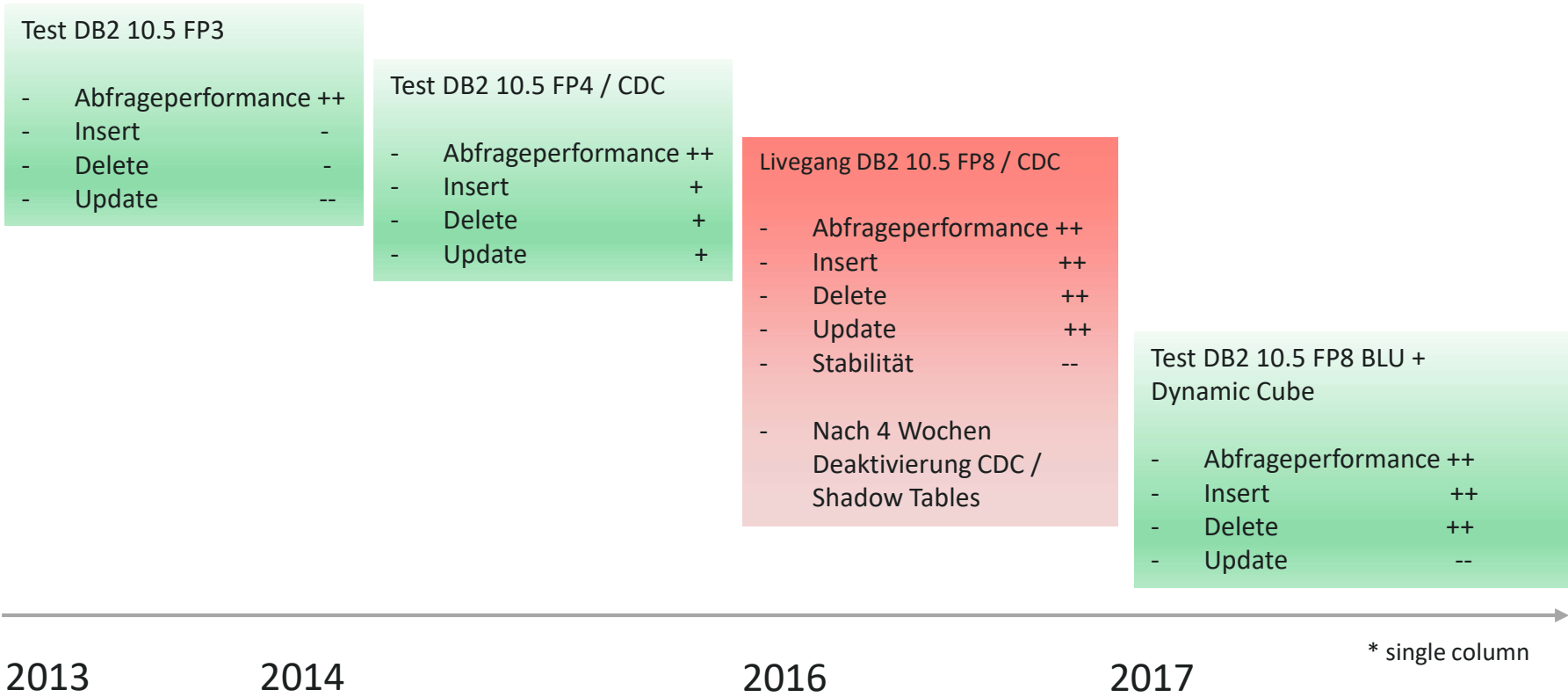
→ Erstellung von Berichten

- Frameworkmanagermodell auf Dynamic Cube als auch auf relationale Tabellen
- Berichte sowohl auf den Cube als auch auf BLU-Tabellen

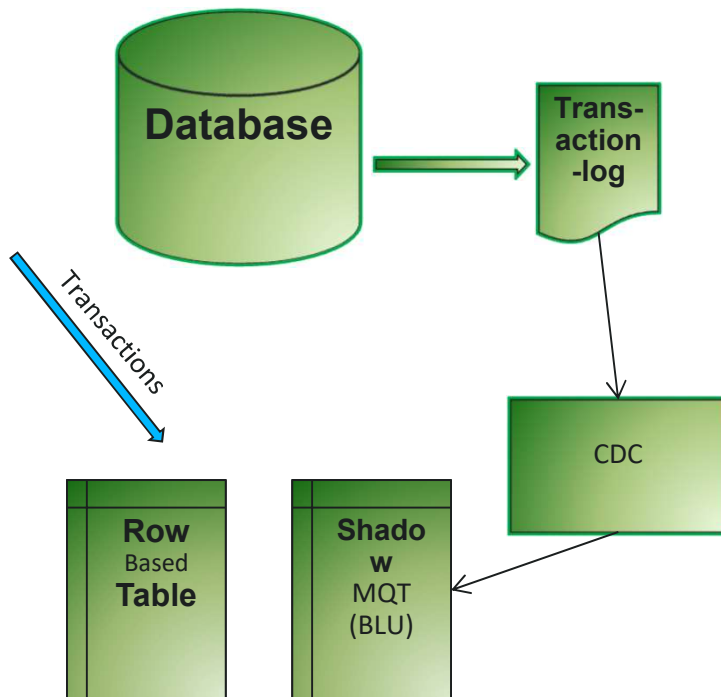
→ Herausforderungen

- Dimensionen mit einer sehr hohen Zahl an Members
- Ausbalancierung, so dass die Cubes optimal trainiert werden können
- Gezielte Nutzung von BLU Tabellen, um eine ganzheitlich optimale Performance zu gewährleisten, ohne auf Performance-Probleme bei der Befüllung der Quelltabellen zu stoßen
- Abbildung der Security Anforderungen

BLU Technik Erfahrungen aus dem Projekt (1/2)



2016: Livegang mit DB2 10.5 FP7 BLU / CDC / Shadow Tables / Findings



Tests waren erfolgreich aber:

- Der Optimizer nutzte die BLU Shadow Tables bei Cognos Abfragen nicht, weil diese „teurer“ waren als die Abfrage auf die row based table. → Produktbug (in 10.5. nach wie vor vorhanden)
→ Lösung (Optimierungsprofil anlegen)
- Eine view mit komplexen analytischen Funktionen und speziellen Zeitfunktionen funktionierte nicht in BLU
- ETL-Abläufe stürzten unregelmäßig und nicht nachvollziehbar ab. Die DB2 war danach nur noch mit einem db2_kill und einem disaster recovery wiederbelebbar
- DAHER: Rückbau und zunächst Nutzung der DB2 10.5 ohne BLU / CDC
- DANN: Nutzung von BLU / Dynamic Cube für Auswertungs-Data-Mart (Rest row based)

Optimierungsprofil zur Nutzung Shadow Tables (1/2)

Create SYSTOOLS.OPT_PROFILE table

db2 connect to \$LOGNAME

db2 "call sysinstallobjects('OPT_PROFILES', 'C', '', 'MAPIS') "

db2 "COMMIT "

load file

Content of MQTENFORCE.del:

MAPIS;OPTMQTENFORCE;MQTENFORCE.xml

XML file:

Content of MQTENFORCE.xml:

```
<?xml version="1.0" encoding="UTF-8"?><OPTPROFILE>
```

```
<OPTGUIDELINES> <MQTENFORCE TYPE='ALL' />
```

```
</OPTGUIDELINES>
```

```
</OPTPROFILE>
```

load command:

db2 "LOAD FROM MQTENFORCE.del OF DEL LOBS FROM . MODIFIED BY COLDEL; INSERT INTO MAPIS.OPT_PROFILE (SCHEMA , NAME , PROFILE) ;

Optimierungsprofil zur Nutzung Shadow Tables (2/2)

Clinet setting

```
db2 "SET CURRENT OPTIMIZATION PROFILE = MAPIS.OPTMQTENFORCE "
```

You have successfully deployed the optimization profile. To enable it, please follow the below steps:1.For JDBC application using the JCC Universal Driver Embed the statement

```
SET CURRENT OPTIMIZATION PROFILE = "SYSTOOLS"."OPT1472793401083"
```

in the Java application at session level.2.

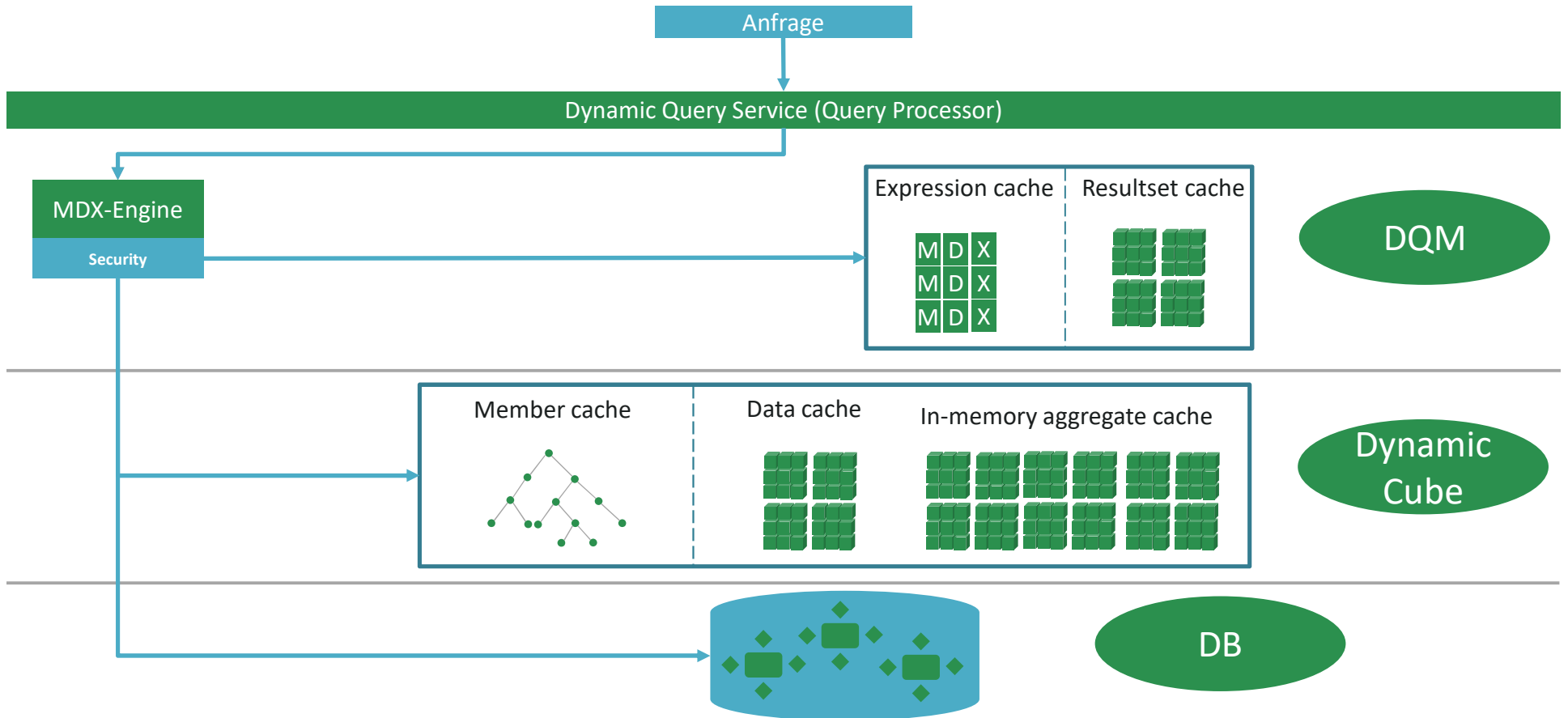
For embedded SQL in C/C++ applications Use the bind file with OPTPROFILE option to the database, for example bind prog1.bnd OPTPROFILE SYSTOOLS.OPT14727934010833.For SQLJ applications with static SQL statements

Use the BINDOPTIONS parameter in the customize phase to associate the profile, for example sqlj prog1.sqlj db2sqljcustomize -url jdbc:db2://SERVER:PORT/SAMPLE -user USER -password PASSWORD -bindoptions

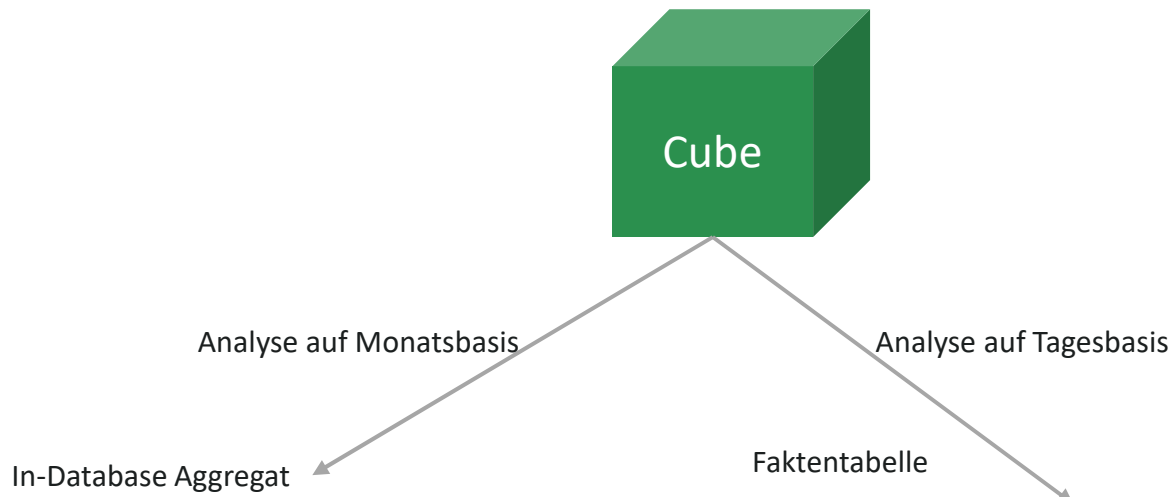
```
"OPTPROFILE SYSTOOLS.OPT1472793401083" -storebindoptions prog1_SJProfile0 4.For CLI applications Force database connections using the CONNECT_PROC database configuration parameter, for example CONNECT TO SHADOW@ CREATE OR REPLACE PROCEDURE SCHEMA.OPTPROF_PROC() BEGIN DECLARE APPLNAME VARCHAR(128); SET APPLNAME = (SELECT APPLICATION_NAME FROM TABLE(MON_GET_CONNECTION(MON_GET_APPLICATION_HANDLE(), -1))); IF (APPLNAME = 'db2bp' OR APPLNAME = 'db2batch') THEN SET CURRENT OPTIMIZATION PROFILE = "SYSTOOLS"."OPT1472793401083"; END IF; END@ GRANT EXECUTE ON PROCEDURE SCHEMA.OPTPROF_PROC TO PUBLIC@ UPDATE DB CFG USING CONNECT_PROC SCHEMA.OPTPROF_PROC IMMEDIATE@ CONNECT RESET@
```

- Dynamic Cubes verfolgen einen hybriden Ansatz.
 - D.h.: mehrere Caching Möglichkeiten (in Memory auf App-Server)
 - Falls der Cache die Daten nicht hergibt Durchgriff auf die Datenbank. Dort sind die Daten auch wieder in memory (BLU)
- Ideal für große Datenmengen, die eine lückenhafte Indizierung zulassen (nicht alle Dateninhalte müssen gecached werden)
 - Hot data
 - Cold data

Dynamic Cubes



In-Database Aggregate

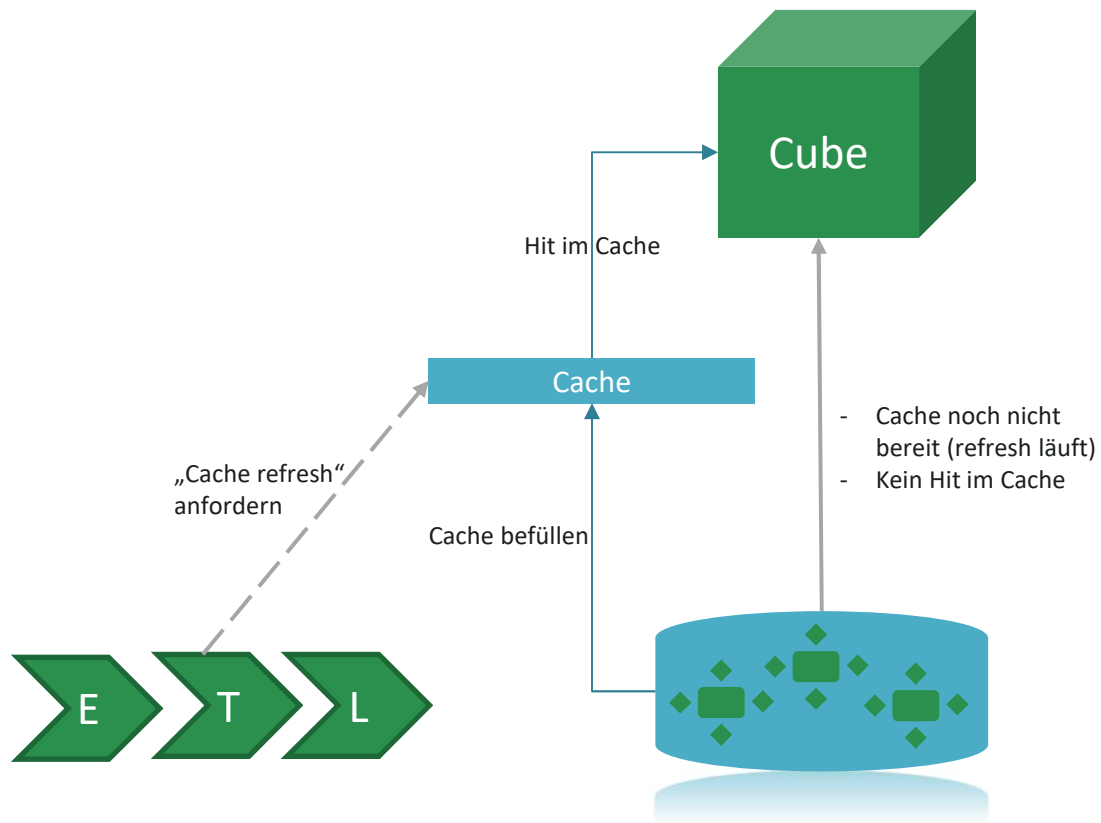


TIME_ID	CUST_ID	VOLUME
201701	1	620
201702	2	300
201702	1	320

TIME_DAY_ID	CUST_ID	VOLUME
20170101	1	500
20170102	1	120
20170204	2	300
20170202	1	150
20170202	1	170

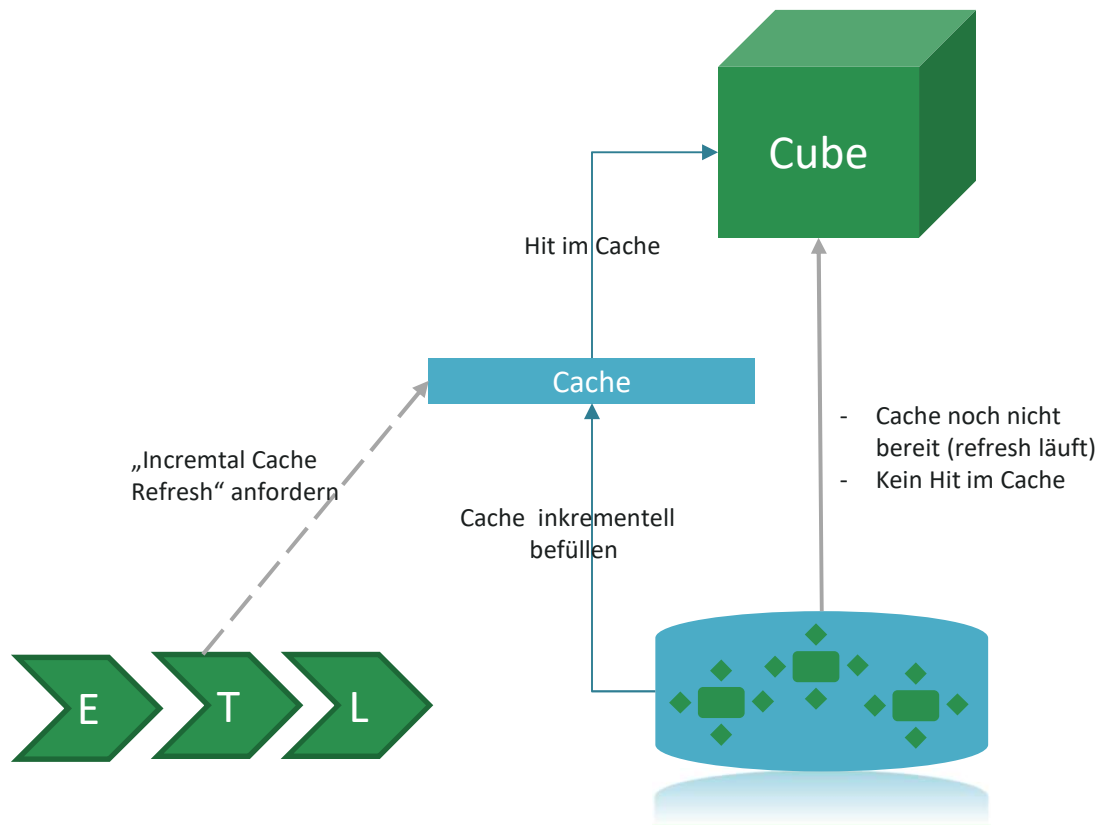
- Eine weitere Methode zum Beschleunigen von Abfragezeiten
- Hierbei wird die Faktentabelle auf eine globaleren Ebene dupliziert
- Hierfür bietet sich z. B. die Zeitebene an
- Die MDX Engine entscheidet zur Laufzeit welche Table für die Analyse angezogen wird
- Gerade mit BLU eine interessante Möglichkeit
- Nachteil: Tables müssen synchron gehalten werden

Nutzungsszenarien – Standard



- Cube wird in weniger häufigen Zeitintervallen aktualisiert (monatlich/evtl. täglich abhängig von der Datenmenge)
- Caches müssen bei jedem Beladen der Fakten/Dimensionen, neu gebildet werden
- Gut geeignet für mittlere Datenmengen (cache refresh kann von der DB einigermaßen abgefangen werden)
- Evtl. auch größere Datenmengen mit eingebauten Aggregationsstufen in der DB und „lückenhaftem“ Caching

Nutzungsszenarien – Inkrementell

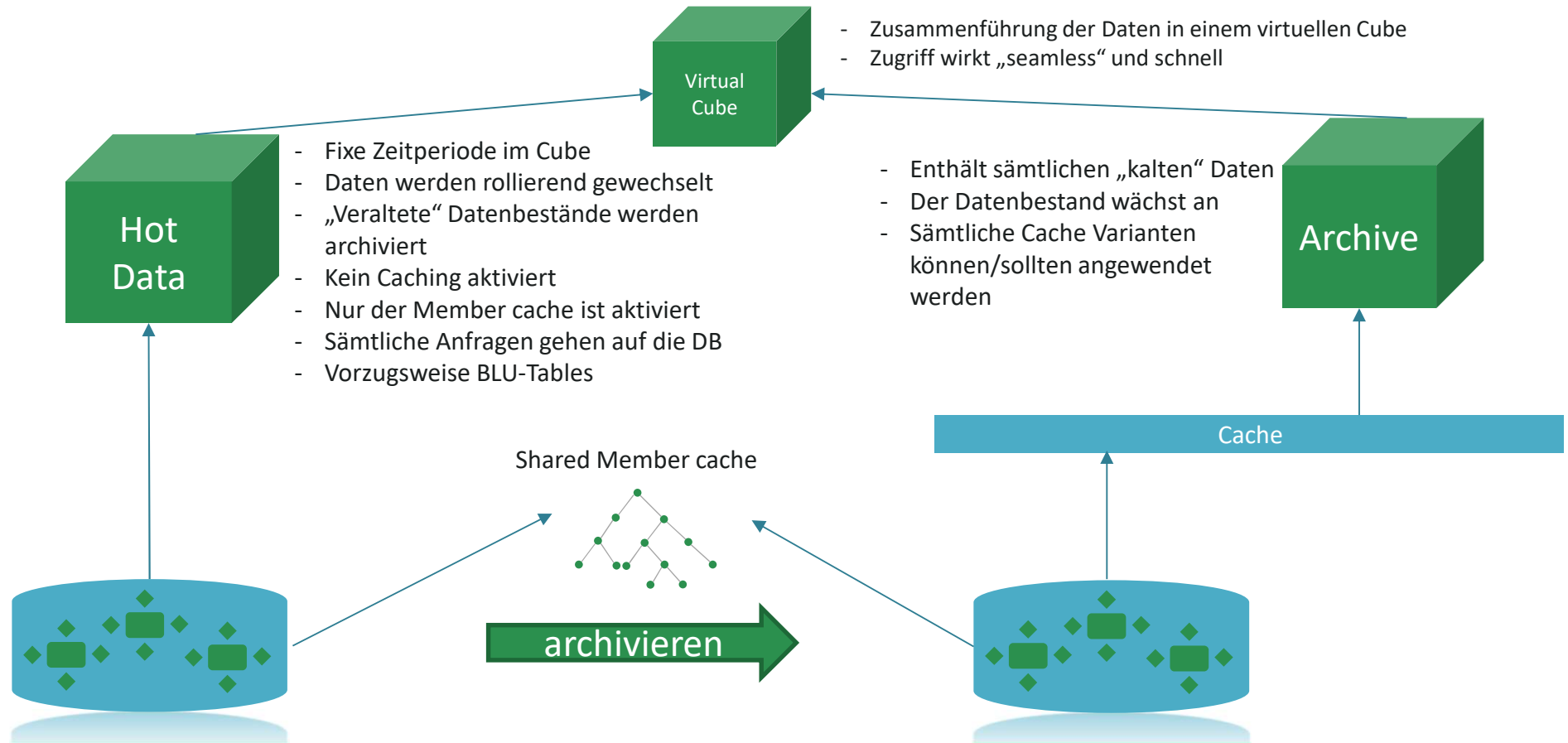


- Gerade bei nahe beeinanderliegenden Updateintervallen interessant
- Der Cube wird beim Beladen der Fakten inkrementell geupdated
- Über die TID (Transaction ID – laufender numerischer Wert) wird der letzte Aktualitätsstand gehalten
- Beginnend ab der letzten TID werden INSERTS durchgeführt
- Caches müssen daher nicht neu gebaut werden

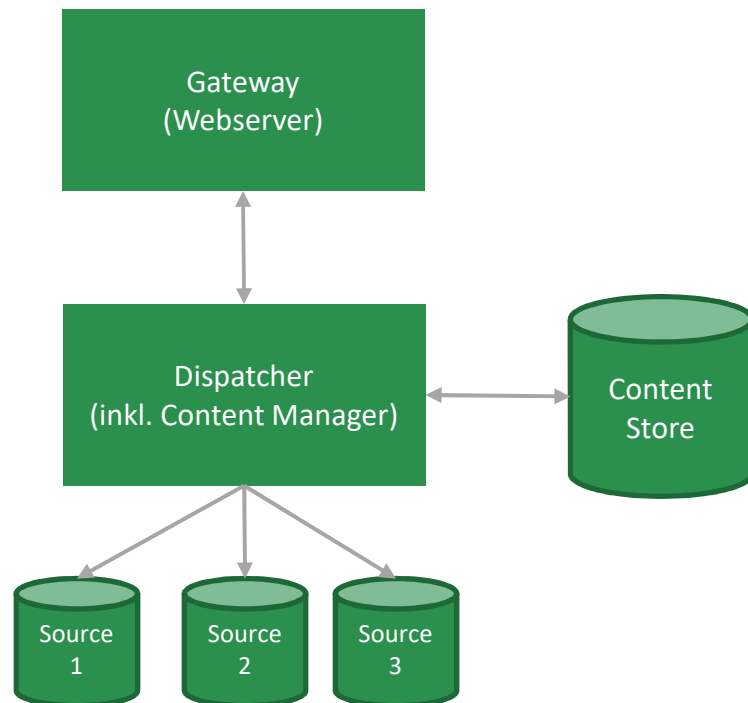
Einschränkungen:

- Momentan nur für INSERTS auf Faktentabellen
 - Dimensionen werden momentan gar nicht unterstützt
- Daher nur für nicht volatile Modelle geeignet

Nutzungsszenarien – Near Realtime



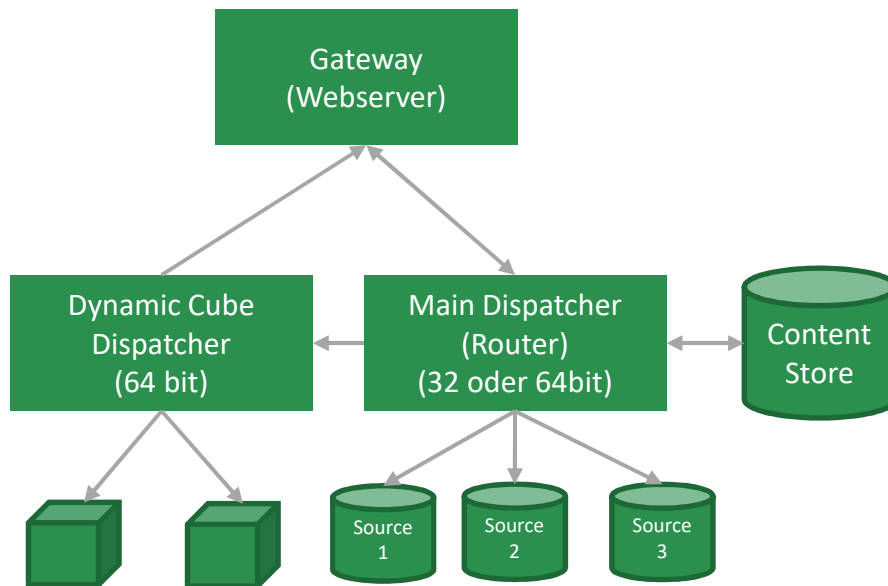
Standard Setup



- Standardaufbau mit einer 3- Tier Architektur
- CPU: 4 Kerne
- Ram: min 8 GB

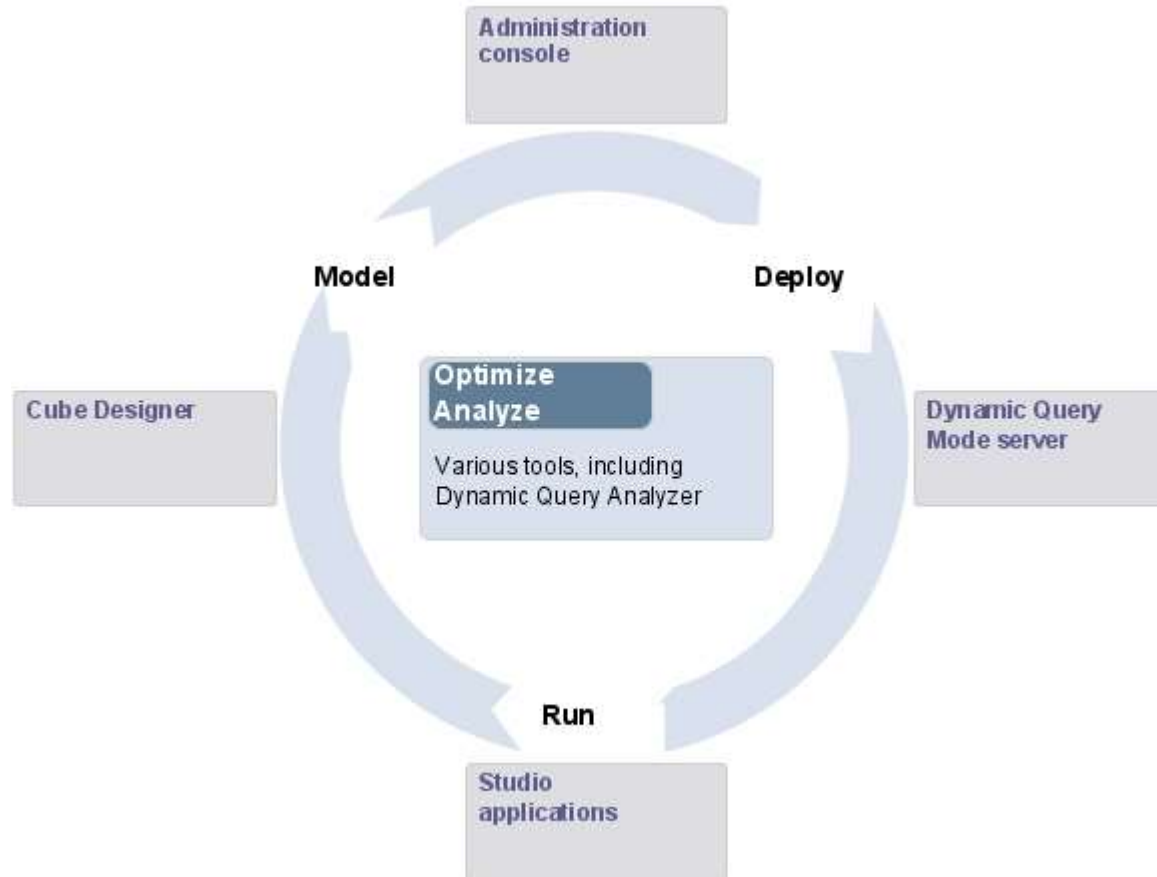
- Natürlich kann der Aufbau je nach Anforderung mit mehreren Gateways und Dispatchern erweitert werden, um über Load-Balancing die Last zu verteilen

Dynamic Cube Setup



- Der Main Dispatcher dient als Router und leitet die jeweiligen Anfragen basierend ihrer Art auf die jeweiligen Dispatcher um
- Der Cube Dispatcher ist ausschließlich für Analyseanfragen auf die Cubes vorgesehen
- Cube Dispatcher Anforderungen:
 - CPU: Min. 4 Kerne
 - RAM: min 64 GB
- Die erforderlichen Ressourcen für Cubes kann sehr gut über den Cube Designer analysiert werden lassen
- Natürlich kann der Aufbau je nach Anforderung mit mehreren Gateways und Dispatchern erweitert werden, um über Load-Balancing die Last zu verteilen

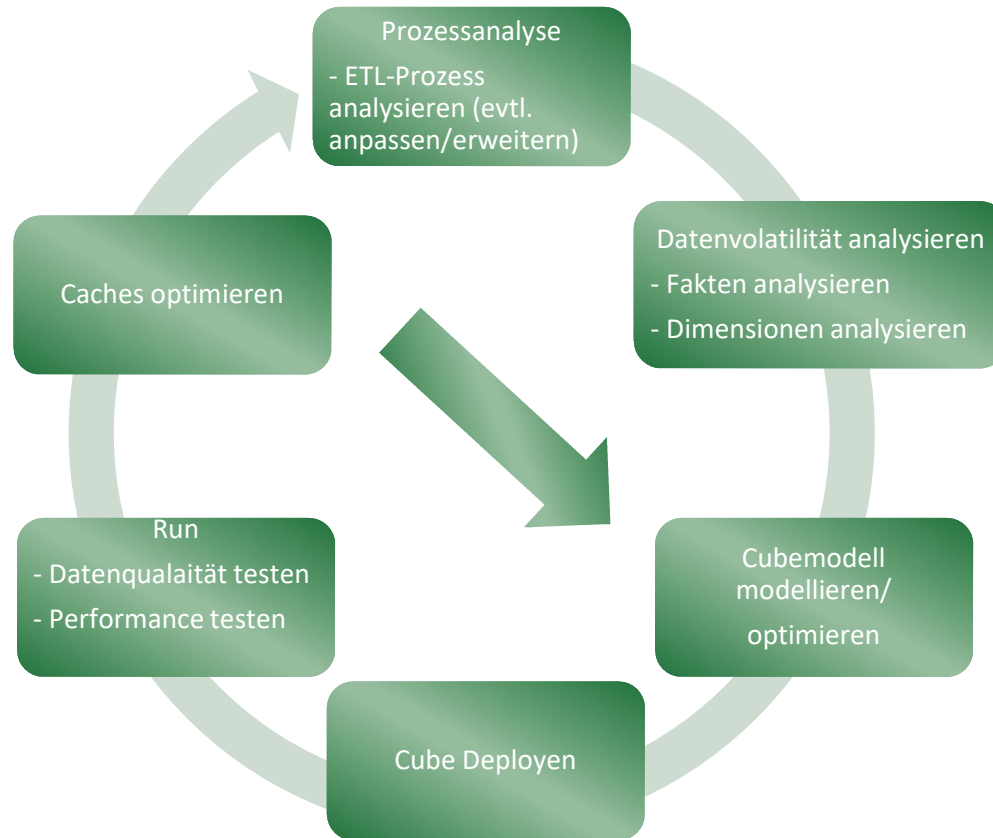
Dynamic Cube workflow (aus IBM Knowledge Center)



Diese Darstellung vernachlässigt Aspekte der Datenmodellierung, Datenbewirtschaftung.

Wenn man zusätzlich DB2 BLU nutzt, wird das Ganze Modell noch um eine weitere Komplexität erweitert. Daher leitet sich als Empfehlung eine erweiterter Prozess ab!

Erweiterter Entwicklungsprozess



Ihr Kontakt



Jörg Kremer
Leiter Geschäftsstelle Stuttgart
mail: joerg.kremer@mip.de

| Tel. +49-89-589394-40

Andreas Deutscher
Leitung Entwicklung
mail: Andreas.Deutscher@mip.de

| Tel. +49-89-589394-41

mip Management Informationspartner GmbH
Fürstenrieder Straße 267 | 81377 München
Niederlassung
Rosensteinstraße 22-24 | 70191 Stuttgart