

# Einsatz von PL/SQL am Beispiel des Reisendeninformationsservers der Deutschen Bahn

**DB** BAHN

**DB Bahn Fernverkehr**



**DB Bahn Regio**



**DB Arriva**



**DB Bahn Vertrieb<sup>1</sup>**



---

DB Systel GmbH

---

Norbert Schaffitzel

---

I.LVP 21

---

Universität Frankfurt, 03.02.2016

1. Einführung in die Welt der Deutschen Bahn
2. Übersicht der Reisendeninformation
3. Details zum Datenbankserver Reisendeninformation

# Referenten



## Norbert Schaffitzel

Engagementmanager  
ICT Solutions Passenger Transport  
(I.LVP 21)

Tel. +49 069 265-18340  
Mobil: 0176 54192137  
Norbert.Schaffitzel@deutschebahn.com

DB System GmbH  
Jürgen-Ponto-Platz 1  
60329 Frankfurt am Main  
www.dbsystem.de



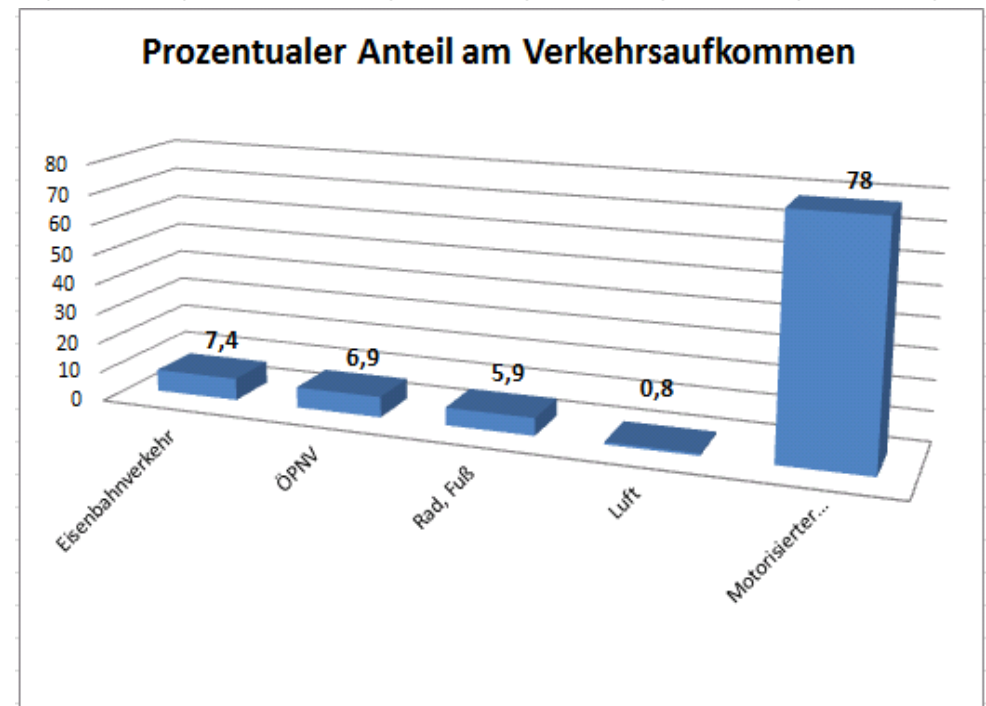
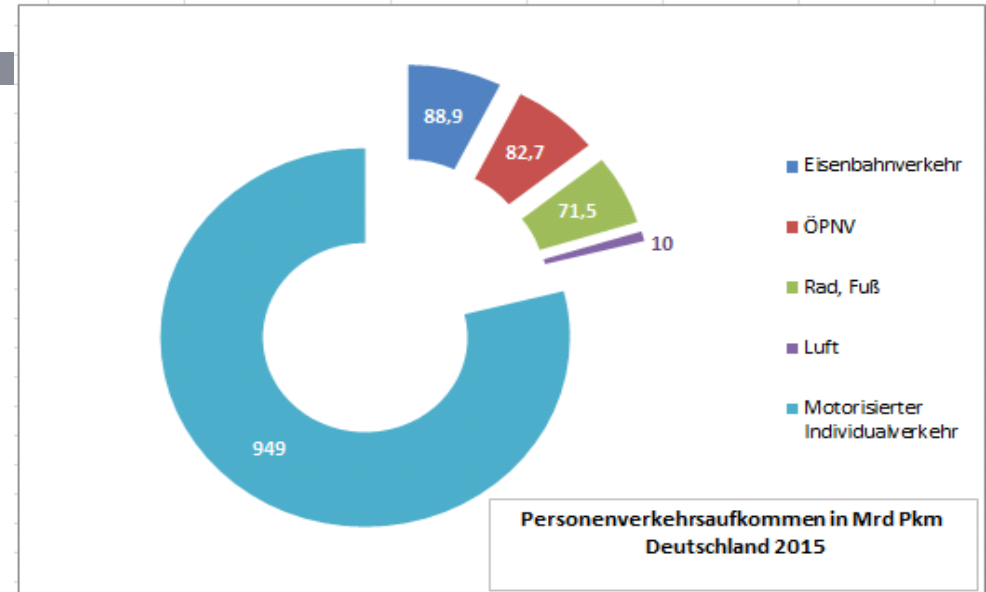
## Jörg Kunze

Projektmanager  
externe Fachkraft im  
RIS Solution Center (I.LPA 15)

Tel. +49 069 265-14736  
Mobil: 0151 53588786  
Joerg.kunzel@deutschebahn.com

DB System GmbH  
Jürgen-Ponto-Platz 1  
60329 Frankfurt am Main  
www.dbsystem.de

# Der Mobilitätsmarkt auf dem die Deutsche Bahn AG ein Player im Markt darstellt



# Fakten zu Deutschen Bahn

## Zahlen und Daten

- Europas zweitgrößter Anbieter im gesamten Personenverkehrsmarkt
- 33.600 km Schienennetz - das längste in ganz Europa
- 2,7 Milliarden Reisende pro Jahr über 5 Mio. Reisende pro Tag
- Ca. 89 Mrd. Pkm Verkehrsleistung im Jahr 2015
- 25.000 Personenzüge pro Tag
- 260 Züge umfasst die ICE-Flotte der DB
- 39,7 Mrd. Euro Umsatz in 2014
- 306.000 Mitarbeiter weltweit, 195.000 in Deutschland

## Fakten

- 1835 erste Eisenbahnfahrt von Nürnberg nach Fürth - Eisenbahn in Deutschland seit 180 Jahre
- 1886 übernimmt der preußische Staat die „Rheinische Eisenbahngesellschaft“, womit fast alle Eisenbahnstrecken verstaatlicht sind
- 1920 wird die Deutsche Reichsbahn gegründet und das Zeitalter der Länderbahnen geht zu Ende
- 1945 Trennung der Eisenbahnen in DR und Bundesbahn
- 1994 Bahnreform - die deutsche Bahn wird eine AG und DR und Bundesbahn werden zusammengeführt
- 2015 Initiativen zur Digitalisierung aller Bereiche sowie der Beginn der Fernverkehrsoffensive

# Fakten zur DB System

## Wir sind:

- 3.100 Mitarbeiter an den drei Standorten Frankfurt/Main, Berlin und Erfurt
- Umsatz im Jahr 2014 lag bei 800 Mio. Euro

## Wir betreiben:

- 3 Rechenzentren mit über 3.800 Servern
- Datennetz mit rund 340.000 IP-Anschlüssen von DSL bis Breitband-Glasfaser mit 2 Netzwerkleitständen
- Rund 600 produktive IT-Verfahren
- 3 Petabyte Plattenspeicher / 7 Petabyte Backup-Kapazität

## Wir betreuen bei der Bahn:

- 96.000 Nutzer der bahnweiten einheitlichen Büro- und Sprachkommunikation
- 93.000 VoIP-Anschlüsse

- Bei uns arbeiten hochmotivierte Experten in interdisziplinären Teams kundenorientiert zusammen.
- Dabei sind gegenseitiger Respekt, Raum für Kreativität und flache Hierarchien die Grundlagen für das Miteinander.



Kreative Teams in einer facettenreichen Unternehmenskultur

# Marktentwicklung vor dem Hintergrund zunehmender Digitalisierung

## Disruptoren auf Basis digitaler Technologien

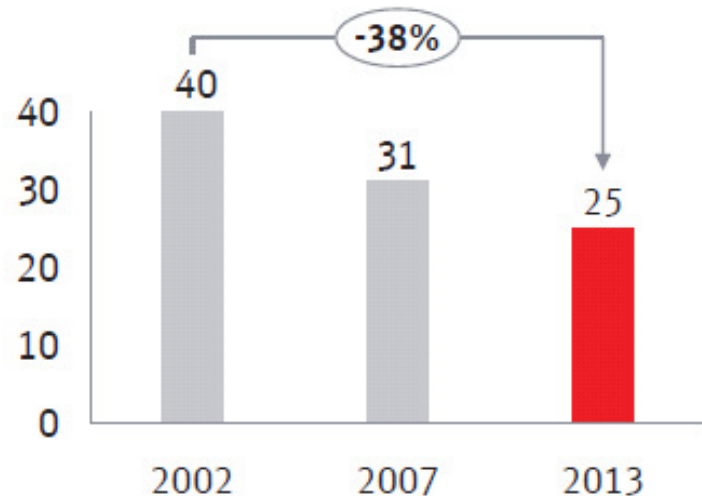
### Neue Technologien



# Die Zukunft des Mobilitätsmarktes liegt in der Vernetzung der Mobilitätsträger

## Autobesitz bei junger Generation um 38% zurückgegangen

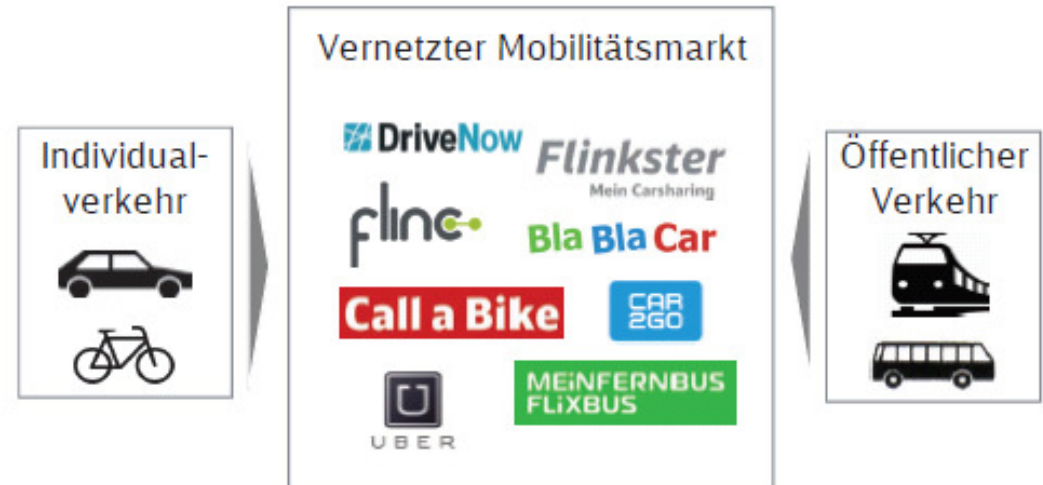
[Autobesitzquote bei 18-29jährigen in Deutschland, in %<sup>1</sup>]



- Pragmatischer und vernetzender **Lebensstil spiegelt zunehmend Mobilitätsverhalten** wider
- **Auto** bei junger Generation **seltener Statussymbol**

## Branche reagiert mit neuen Angeboten auf verändertes Mobilitätsverhalten

[Ausgewählte Beispiele neuer Angebote im Mobilitätsmarkt]



- Vernetzter Mobilitätsmarkt setzt auf „**Nutzen statt Besitzen**“
- **Smarte Devices** und **übergreifende Informationsplattformen** machen vernetzte Mobilitätsangebote einfach zugänglich und nutzbar



# Reisendeninformation wird damit zu einem zentralen Schlüssel für die Unterstützung vernetzter Mobilität

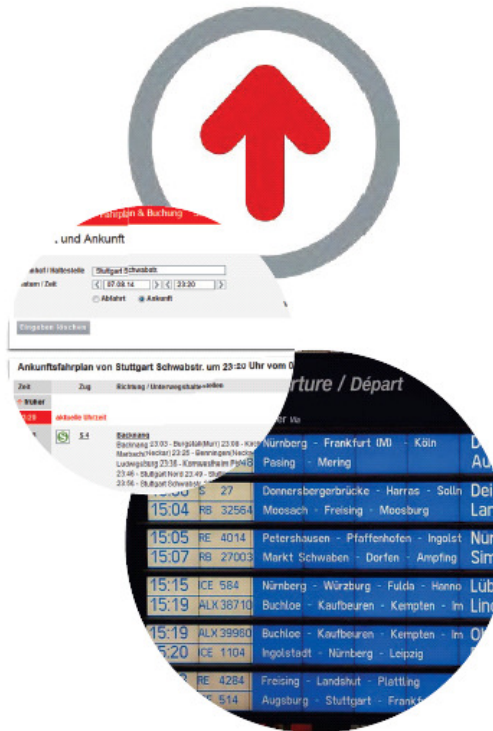


# Die Anforderungen an Reisendeninformationen werden bestimmt durch deren Informationsgüte und einfache Bereitstellung

## Kundenzufriedenheit



## Qualität der Information



## Kosten



# Was muss das Ziel von Reisendeninformation sein?

**Die einzige deutschlandweite Datendrehscheibe für Reisendeninformation**

- Sammeln, Konsolidieren, Aufbereiten
- Über 50 Schnittstellenpartner
- Unterstützt VDV-Standard
- RIS-http

**Das modernste Fahrgastinformationssystem in Zügen**

- live-Daten
- Updates over the air
- Ansage +Anzeige, innen +außen
- Modular

**Die innovativste App für das Zugpersonal**

- 20.000 Zub/KiN/Tf
- Monatliche Updates
- Android-Plattform

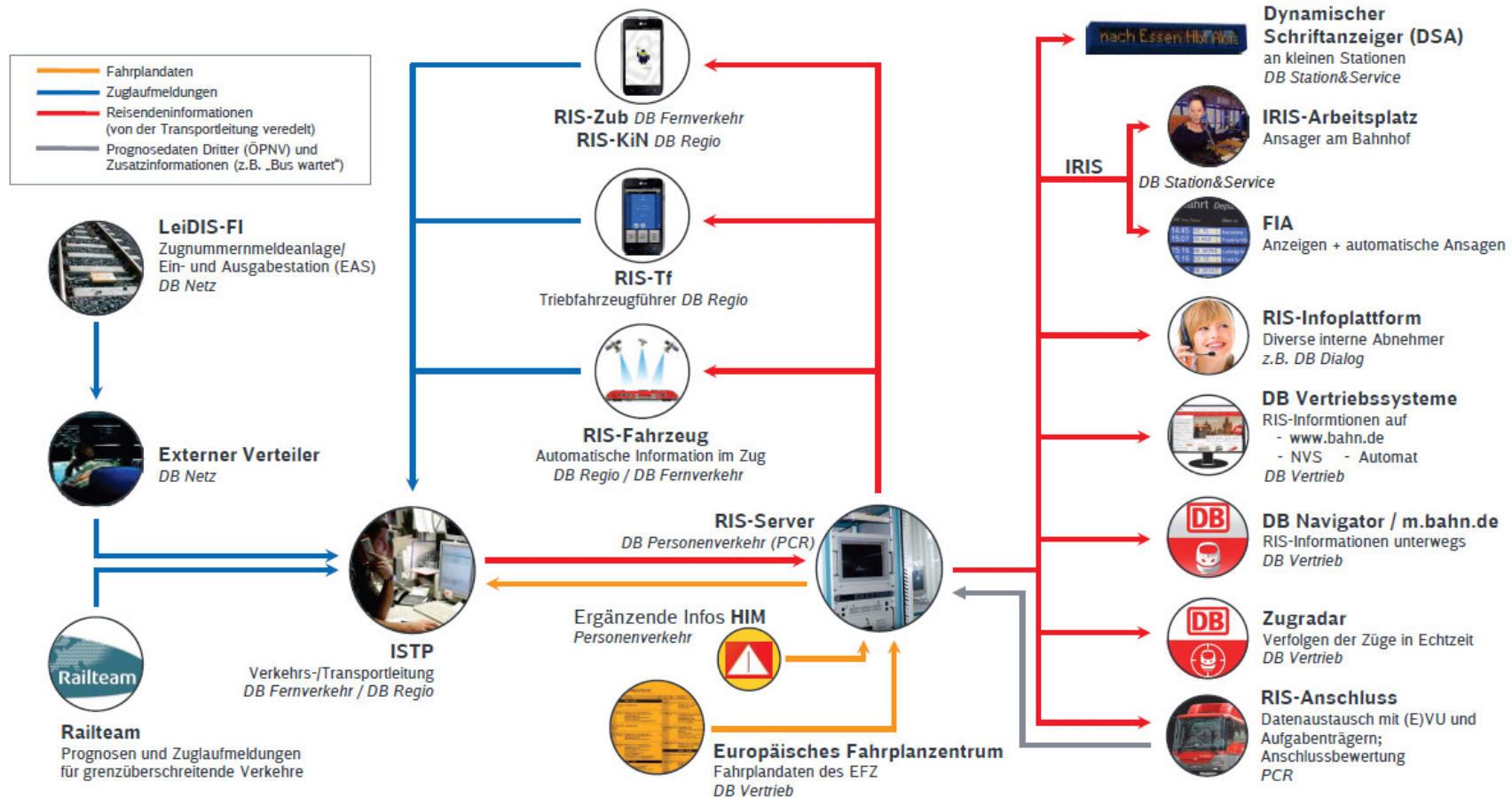
**Business Intelligence: Aus RI-Daten werden Erkenntnisse**

- Berichte
- Analysen
- Value added services

**Schnelle kleine Lösungen rund um die Reisendeninformation**

- ESF-Hinweise
- Forum RIS-Infoplattform
- Eingabetool

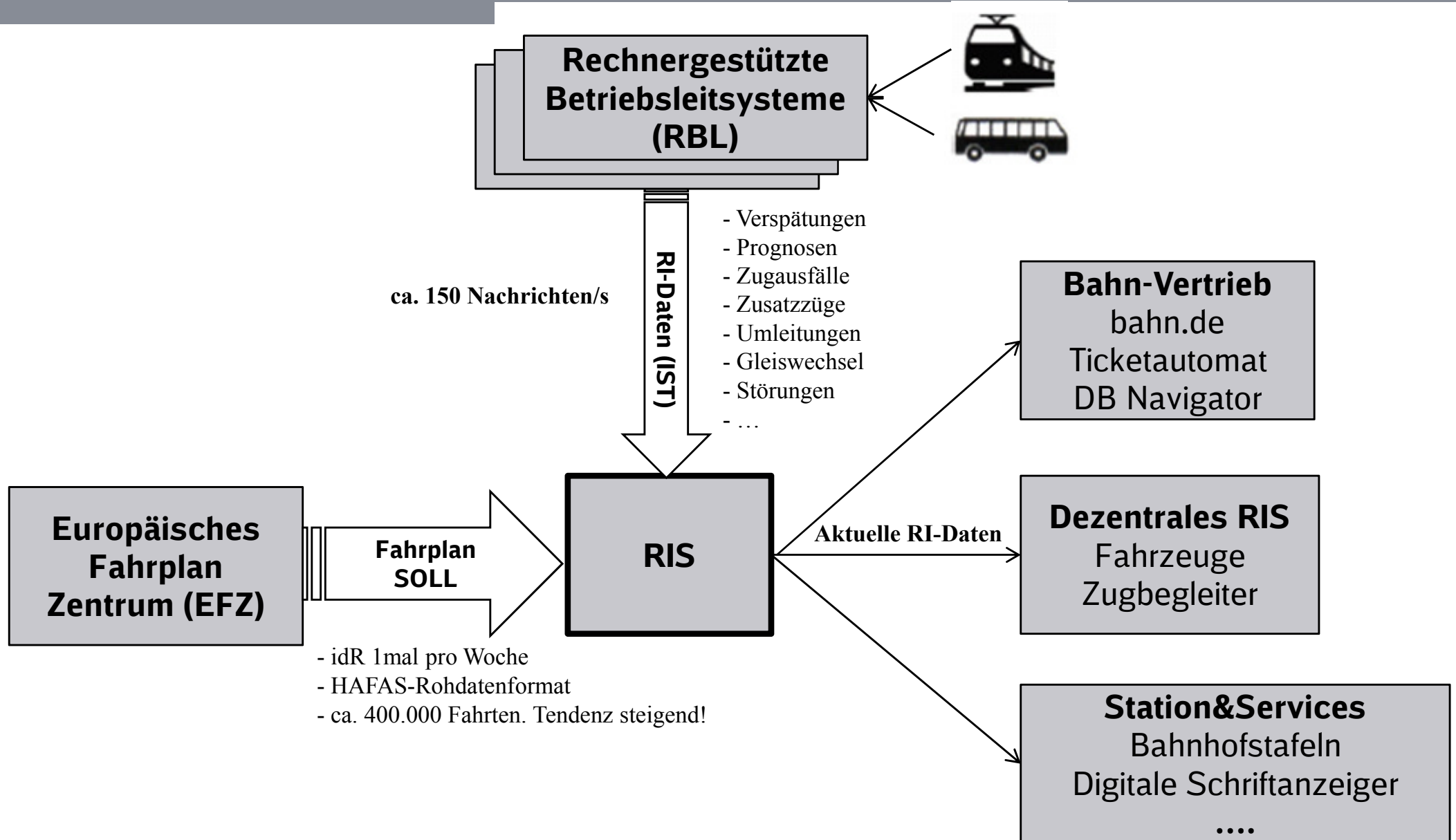
# Aus betrieblichen Rohdaten werden erst über deren Verknüpfung kundengerechte Reisendeninformationen



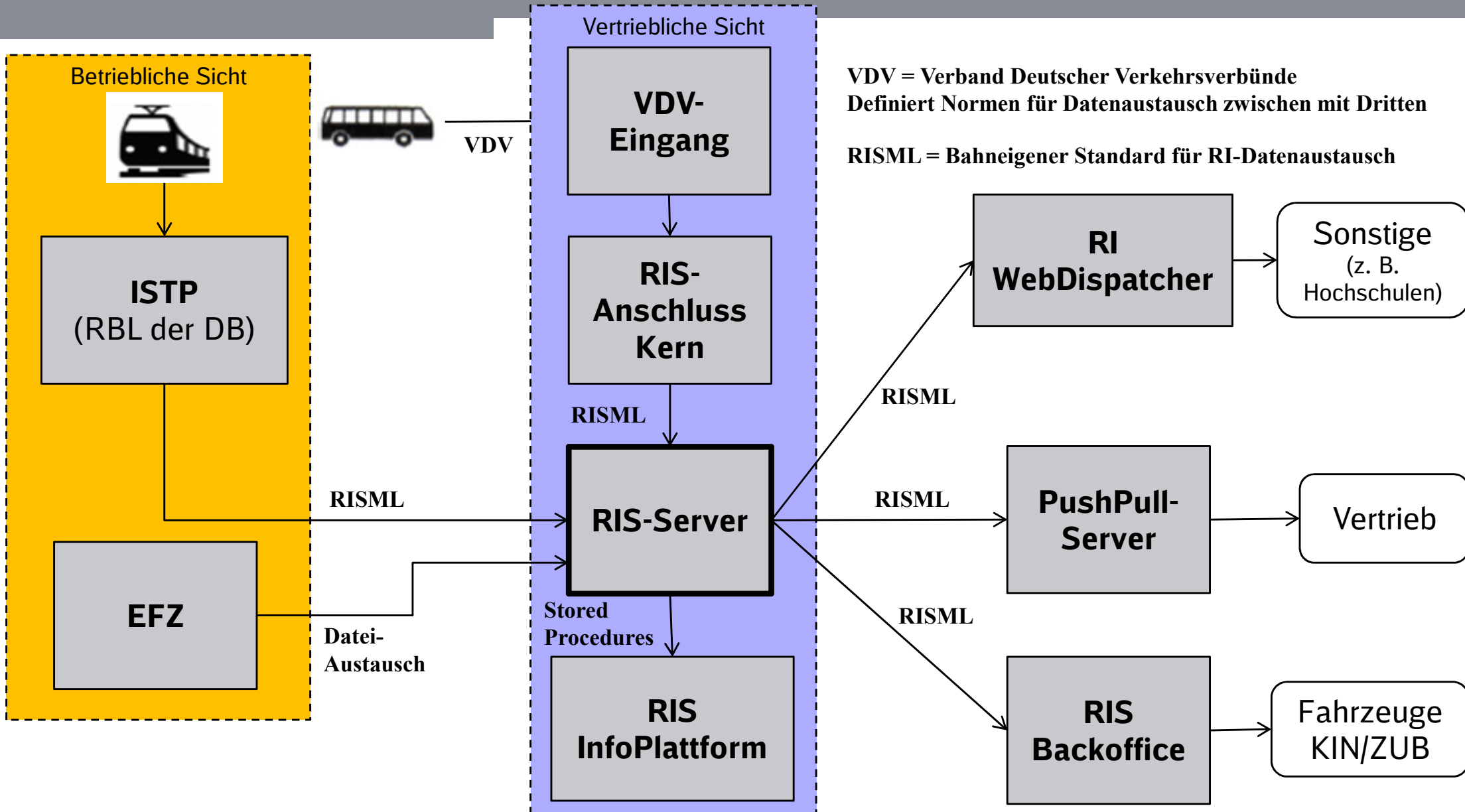
# Der RIS-Verbund im Überblick

- Die technische Umsetzung der ReisendenInformation basiert auf einer Reihe von Applikationen (Bahnausdruck: „Verfahren“), die über eine Vielzahl von Schnittstellen miteinander verbunden sind.
- Der RIS-Verfahrensverbund basiert auf folgendem Technologie-Stack:
  - ORACLE Rel. 11.2.0.4 (bis Herbst 2016 Migration auf ORACLE 12 geplant)
  - ORACLE Advanced Queuing (AQ) als Messaging-Middleware
  - Java J2EE, JBoss, Apache, Tomcat
  - Frontends überwiegend mit Webtechnologien (HTML, JavaScript, AJAX, Jason)
  - Linux als Serverbetriebssystem (SLES 11)
- Entwicklungsumgebungen:
  - Java: Eclipse
  - Oracle: TOAD, ORACLE SQL Developer
- Die Anforderungen an die Ausfallsicherheit sind sehr hoch (Betrieb 365/24, kurze Wartungsfenster)
- Definiertes Antwortzeitverhalten
- Starke Schwankungen im Lastprofil (Hauptverkehrszeiten, Sonderfälle wie z. B. Unwetter, Streik)
- Der Verfahrensverbund wird in Rechenzentren der DB Systel GmbH gehostet und betriebsgeführt

# RIS – Die zentralen Datenströme



# Verfahren im RIS-Verbund (Auswahl)



## Zahlen und Daten

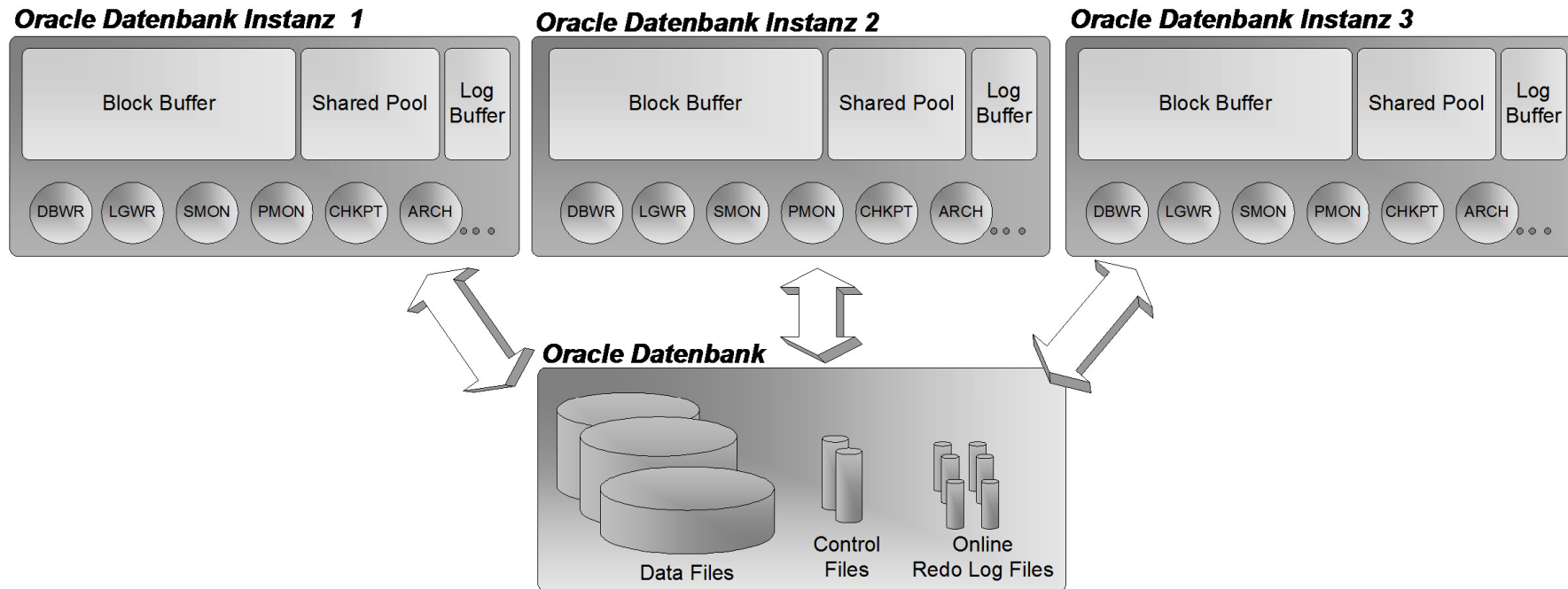
- Der RIS-Server (RISS) basiert auf ORACLE. Die Business Logik ist (fast) ausschließlich in PL/SQL entwickelt worden und umfasst ca. 120 kLOC.
- Das Datenmodell umfasst rund 400 Tabellen
- Die erste Version von RIS-Server ging 2003 in Betrieb
- Das letzte, vollständige Redesign (RISS 7.0) wurde von 2011-2015 durchgeführt
- Hierbei wurde der komplette RISS-Server vollständig neu implementiert.  
Zielsetzung: Skalierbarkeit und Redundanz auf Basis einer Oracle RAC-Architektur (Real Application Cluster)
- Fachlicher Nutzen der Skalierbarkeit: Anbindung des deutschen ÖPNV, Anbindung / Datenaustausch mit anderen europäischen Bahnen. Mittelfristig bis zu 900.000 Fahrten/Tag (vgl. Zugverkehr Deutsche Bahn: ca. 30.000 Fahrten/Tag)



# Gründe für PL/SQL als Sprache für die Business-Logik

- **PL/SQL: (Procedural Language/Structured Query Language)**  
Proprietäre Sprache von ORACLE, welche die Abfragesprache SQL mit einer prozeduralen Programmiersprache verbindet.
  
- **Vorteile:**
  - PL/SQL wird direkt auf dem Datenbankserver ausgeführt. Die Daten müssen nicht über ein Netzwerk zur Weiterverarbeitung z. B. an einen Application-Server transportiert, dort verarbeitet und das Ergebnis wieder zurücktransportiert werden
  - PL/SQL ist optimiert auf die Verarbeitung von relationalen Daten Objektorientierte Sprachen benötigen normalerweise zunächst ein Mapping auf die Objektstrukturen, dies führt häufig zu Performance-Problemen und „logischen Brüchen“
  - Einheitliche Skill-Anforderung an Mitarbeiter, im Vergleich zu heterogenen Mehrschicht-Architekturen
  
- **Nachteile**
  - Objektorientierung nur sehr eingeschränkt, komplexe Logik kann unübersichtlich werden
  - Auswahl an Tools und Bibliotheken nur eingeschränkt (z. B. im Vergleich zu Java)
  - Proprietär, damit eine defacto-Festlegung auf ORACLE

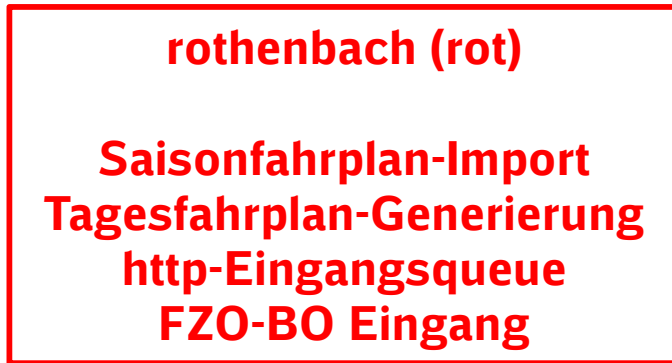
# Technischer Aufbau: Oracle RAC mit 3 Knoten



- Skalierbarkeit kann vertikal (mehr CPU im bestehenden Knoten) oder horizontal (mehr Knoten) erfolgen
- Begrenzender Faktor im RIS-Verbund ist die CPU

# Verteilung der Dienste im RAC (vereinfacht)

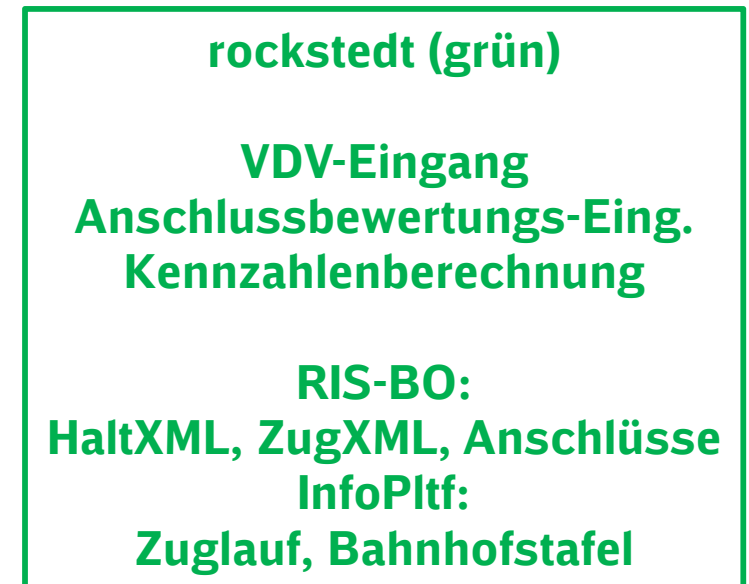
## RIS1



## RIS2



## RIS3



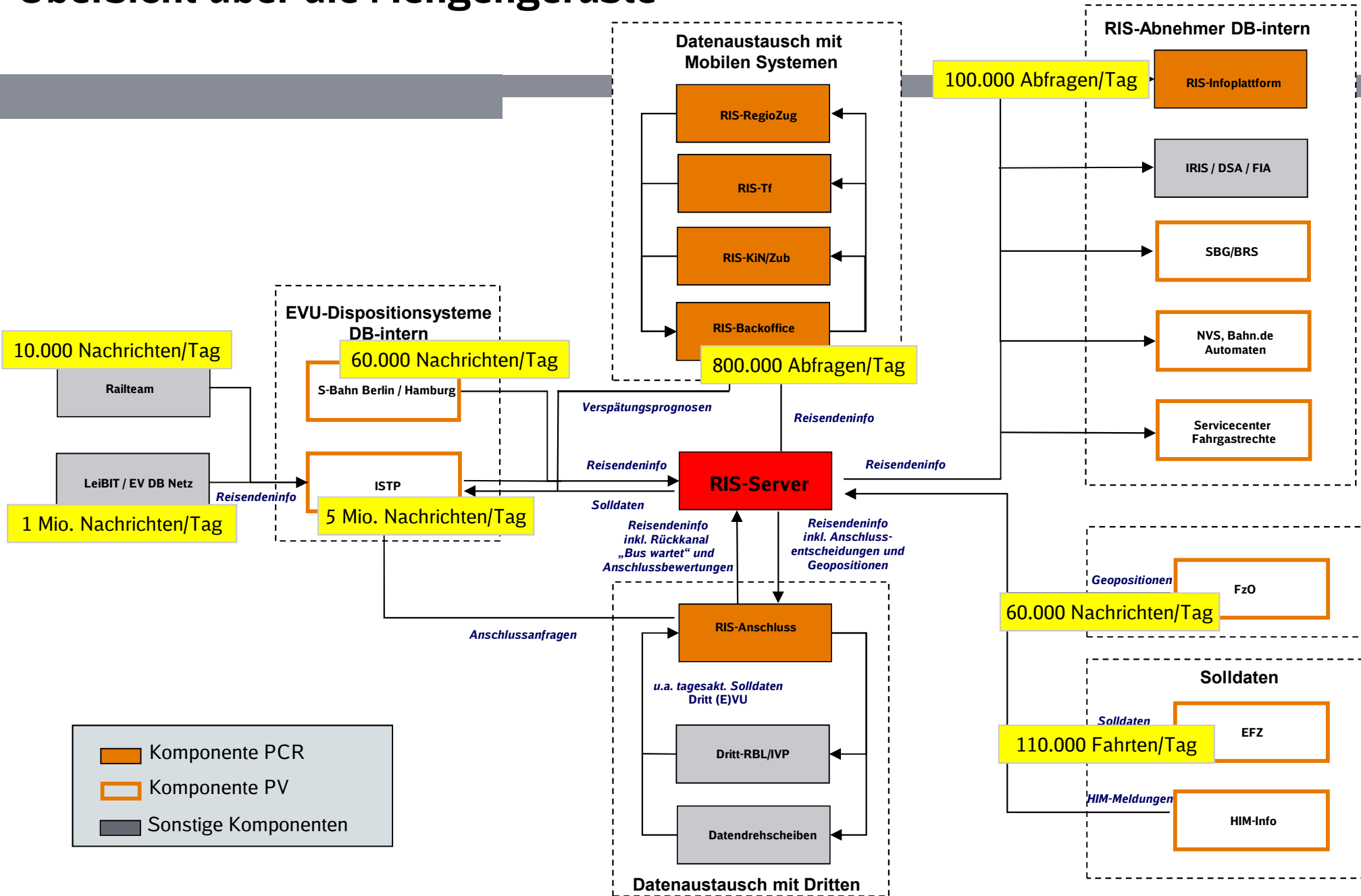
- Die Verteilung kann zu jedem Zeitpunkt rein konfigurativ, ohne Systemunterbrechung geändert werden
- Bei Ausfall eines Knoten im RAC übernehmen die verbleibenden Knoten dessen Dienste. In Überlastsituation können bei Bedarf einzelne Dienste mit geringerem Servicelevel abgeschaltet werden.

# Monitoring der Performancewerte



- Um Auffälligkeiten zu erkennen, werden wichtige Performance-Kennzahlen aufgezeichnet und ausgewertet
- Beim Überschreiten definierter Schwellwerte werden automatische Benachrichtigungen an die Betriebsführung ausgeführt.
- Auffälligkeiten können durch Analysen frühzeitig erkannt werden, so dass Zeit für Gegenmaßnahmen bleibt.
- Unterstützung durch die PL/SQL-Logik

# Übersicht über die Mengengerüste



# Randbedingungen für das Datenmodell

- Performance geht vor Reduktion der Datenmenge: An vielen Stellen werden bewusst Normalformen verletzt und Datenredundanz in Kauf genommen. Dadurch können Abfragen einfacher gestaltet werden, da Joins zwischen Tabellen reduziert werden. Einfachere Abfragen erleichtern die Arbeit des Optimizers, optimale Ausführungspläne zu ermitteln
- Das Datenmodell muss zum RAC passen und Cluster-Interconnects vermeiden, bei denen sich Knoten untereinander blockieren
- Neben Indizes optimieren Partitionierung und Subpartitionierung den Datenzugriff
- Alle eingehenden und ausgehenden Daten müssen bis zu 360 Tage archiviert werden (z. B. wegen Fahrgastrechten)
- Zeitzonen müssen unterstützt werden (europäischer Bahnverkehr, z. B. „EuroStar“)
- Es existiert keine eindeutige Fahrtidentifikation: Komplexe fachliche Schlüssel müssen gebildet werden.
- Datenqualität nicht immer optimal (z. B. bei kleinen Verkehrsverbänden)

# Ausschnitt aus dem Datenmodell - Tagesfahrplan

DAILYTT_ACTUALTRIP	
P	START_DATE DATE
P	SERVICEID INTEGER
	STARTTIMESTAMP_TZ TIMESTAMP (0) WITH TIME ZONE
	STARTTIMESTAMP TIMESTAMP (0) WITH LOCAL TIME ZONE
	TRIP_STATUS_ID INTEGER
	ISADHOC NUMBER (1)
	IS_SERVICE_CLOCKED NUMBER (1)
	NAME_LONG VARCHAR2 (50 CHAR)
	IS_SPRINTER NUMBER (1)
	START_ORDINAL_INDEX NUMBER
	START_SCHEDULED_HOUR_TZ NUMBER (2)
	DEST_ORDINAL_INDEX NUMBER
	DEST_SCHEDULED_HOUR_TZ NUMBER (2)
	CARRIER_ADMINISTRATION_ID VARCHAR2 (6 CHAR)
	LAST_SCHEDULED_TS_TZ TIMESTAMP (0) WITH TIME ZONE
	LAST_FOA_TS_TZ TIMESTAMP (0) WITH TIME ZONE
	LAST_ORDINAL_INDEX NUMBER
	LAST_UPDATE_TS TIMESTAMP WITH TIME ZONE
	LAST_PACKAGE_TS TIMESTAMP WITH TIME ZONE
	LAST_JOB_SESSIONID INTEGER
	STT_PARTITION_ID INTEGER
	TRIPNO_OPR_SHORT_START NUMBER (7)
	SCHEDULED_STOP_ID_START NUMBER (7)
	START_DAY_SHIFT INTEGER
	POS_MAP_STATUS NUMBER (1)
	LAST_POS_LAT NUMBER (9,6)
	LAST_POS_LONG NUMBER (9,6)
	LAST_POS_TS TIMESTAMP WITH TIME ZONE
	LAST_POS_LAT_OLD NUMBER (9,6)
	LAST_POS_LONG_OLD NUMBER (9,6)
	LAST_POS_TS_OLD TIMESTAMP WITH TIME ZONE
	DEST_SCHEDULED_TS_TZ TIMESTAMP (0) WITH TIME ZONE
	DEST_FOA_TS_TZ TIMESTAMP (0) WITH TIME ZONE
	SCHEDULED_STOP_ID_DEST NUMBER (7)
	SCHEDULED_STOP_NAME_DEST VARCHAR2 (50 CHAR)
	SCHEDULED_STOP_DS100_DEST VARCHAR2 (5 CHAR)
	SCHEDULED_STOP_NAME_START VARCHAR2 (50 CHAR)
	SCHEDULED_STOP_DS100_START VARCHAR2 (5 CHAR)
	MAIN_CATEGORY_CODE_START VARCHAR2 (3 CHAR)
	TRIP_REF_TYPE_ID INTEGER
	LAST_FOA_TRIP_EVENT_TYPE_ID INTEGER
	LAST_FOA_STOP_DS100 VARCHAR2 (5 CHAR)
	LAST_FOA_STOP_ID NUMBER (7)
	LAST_FOA_CREATE_TS TIMESTAMP WITH TIME ZONE
	LINE_NAME_START VARCHAR2 (50 CHAR)
	MAIN_CATEGORY_EXTERNAL_START VARCHAR2 (50 CHAR)
	LIMLINE_NAME VARCHAR2 (100 CHAR)
	SUB_NET_ID INTEGER
	TRIPNO_IS_ARTIFICIAL NUMBER (1)
	TRIPNO_CARRIER_START VARCHAR2 (8 CHAR)
	NO_OF_CYCLES INTEGER
	MSG_STATUS INTEGER
	RAILTEAM_RIS_FLAG INTEGER
	TRIP_PK (START_DATE, SERVICEID)
	TRIP_IDX1 (CARRIER_ADMINISTRATION_ID)

DAILYTT_ACTUALTRIP_STATUS	
P	TRIP_STATUS_ID INTEGER
	NAME VARCHAR2 (32 CHAR)
	DESCRIPTION VARCHAR2 (1000 CHAR)
	TRIP_STATUS_PK (TRIP_STATUS_ID)

DAILYTT_ACTUALTRIP_EVENT_STATUS	
P	TRIP_EVENT_STATUS_ID INTEGER
	NAME VARCHAR2 (32 CHAR)
	DESCRIPTION VARCHAR2 (1000 CHAR)
	TRIP_EVENT_STATUS_PK (TRIP_EVENT_STATUS_ID)

DAILYTT_ACTUALTRIP_EVENT	
P	START_DATE DATE
P	SCHEDULED_HOUR_TZ NUMBER (2)
P	SERVICED INTEGER
P	STARTTIMESTAMP_TZ TIMESTAMP (0) WITH TIME ZONE
P	STARTTIMESTAMP TIMESTAMP (0) WITH LOCAL TIME ZONE
P	ORDINAL_INDEX NUMBER
	EVENT_CLASSIFICATION VARCHAR2 (32 CHAR)
	SCHEDULED_TS_TZ TIMESTAMP (0) WITH TIME ZONE
	SCHEDULED_TS TIMESTAMP (0) WITH LOCAL TIME ZONE
	FORECAST_OR_ACTUAL_TS_TZ TIMESTAMP (0) WITH TIME ZONE
	FORECAST_OR_ACTUAL_TS TIMESTAMP (0) WITH LOCAL TIME ZONE
	TIME_SPECIFICATION_TYPE_ID INTEGER
	FOA_CREATION_TIMESTAMP TIMESTAMP WITH TIME ZONE
	FOA_IS_GUARANTEED NUMBER (2)
	FOA_SOURCE VARCHAR2 (64 CHAR)
	FOA_SOURCE_IP VARCHAR2 (64 CHAR)
	EXIT_DIRECTION_ACTUAL CHAR (1)
	EXIT_DIRECTION_PLAN CHAR (1)
	CHANGE_OF_PASSENGERS_ALLOWED NUMBER (1)
	TRIPNO_CARRIER VARCHAR2 (8 CHAR)
	TRIPNO_OPR VARCHAR2 (28 CHAR)
	TRIPNO_OPR_SHORT NUMBER (7)
	STOPPING_REASON CHAR (1)
	IS_WEAKENING_STOP NUMBER (1)
	IS_STRENGTHENING_STOP NUMBER (1)
	IS_SEPARATION_STOP NUMBER (1)
	REGIONAL_DIVISION_NO INTEGER
	IS_SPRINTER NUMBER (1)
	REGULARITY_STATUS_ID INTEGER
	TRIP_EVENT_TYPE_ID INTEGER
	TRIP_EVENT_TYPE_ID_ANAB INTEGER
	IS_CONNECTION_SCHEDULING NUMBER (1)
	SEASON_STOP_ID NUMBER (7)
	SEASON_TRACK VARCHAR2 (50 CHAR)
	SCHEDULED_STOP_ID NUMBER (7)
	SCHEDULED_TRACK VARCHAR2 (50 CHAR)
	SCHEDULED_STOP_NAME VARCHAR2 (50 CHAR)
	SCHEDULED_STOP_DS100 VARCHAR2 (5 CHAR)
	ACTUAL_STOP_ID NUMBER (7)
	ACTUAL_TRACK VARCHAR2 (50 CHAR)
	ACTUAL_TRACK_STOP_ID NUMBER (7)
	ACTUAL_TRACK_STOP_DS100 VARCHAR2 (5 CHAR)
	ACTUAL_TRACK_STOP_NAME VARCHAR2 (50 CHAR)
	ACTUAL_TRACK_SOURCE VARCHAR2 (64 CHAR)
	ACTUAL_TRACK_SOURCE_IP VARCHAR2 (64 CHAR)
	TRACK_MSG_TIMESTAMP TIMESTAMP WITH TIME ZONE
	ACTUAL_STOP_NAME VARCHAR2 (50 CHAR)
	ACTUAL_STOP_DS100 VARCHAR2 (5 CHAR)
	DIRECTION VARCHAR2 (3 CHAR)
	MAIN_CATEGORY_CODE VARCHAR2 (3 CHAR)
	MAIN_CATEGORY_EXTERNAL VARCHAR2 (50 CHAR)
	LINE_NAME VARCHAR2 (50 CHAR)
	LIMLINE_NAME VARCHAR2 (100 CHAR)
	SUB_NET_ID INTEGER
	CARRIER_ADMINISTRATION_ID VARCHAR2 (6 CHAR)
	REFERENTIALTRIPNO INTEGER
	REFERENCING_STATUS_ID INTEGER
	REFERENTIALTRIP_CAT VARCHAR2 (3 CHAR)
	UPDATE_TS TIMESTAMP WITH TIME ZONE
	PACKAGE_TS TIMESTAMP WITH TIME ZONE
	JOB_SESSIONID INTEGER
	STT_PARTITION_ID INTEGER
	NAME_LONG VARCHAR2 (50 CHAR)
	EFZ_KEY NUMBER (10)
	NO_OF_CYCLES INTEGER
	SCHEDULING_PERIOD INTEGER
	VT_ID NUMBER (6)
	IS_SERVICE_CLOCKED NUMBER (1)
	START_DAY_SHIFT INTEGER
	REFTRIP_ID NUMBER
	TRIPNO_IS_ARTIFICIAL NUMBER (1)
	TRIP_EVENT_STATUS_ID INTEGER
	TRIP_ID NUMBER (10)
	TRIP_EVENT_PK (START_DATE, SCHEDULED_HOUR_TZ, SERVICEID, ORDINAL_INDEX)
	TRIP_EVENT_IDX1 (MAIN_CATEGORY_CODE)

- Jeden Abend werden aktuelle Tagesfahrpläne für 6 Folgetage generiert.
- Im Mittelpunkt stehen Fahrten (TRIP) und Fahrereignisse (TRIP\_EVENT)
- Es gibt derzeit 5 Typen von Fahrereignissen:
  - Start
  - Ankunft
  - Abfahrt
  - Ziel
  - Durchfahrt
- RI-Daten verändern laufend die Ist-Lage

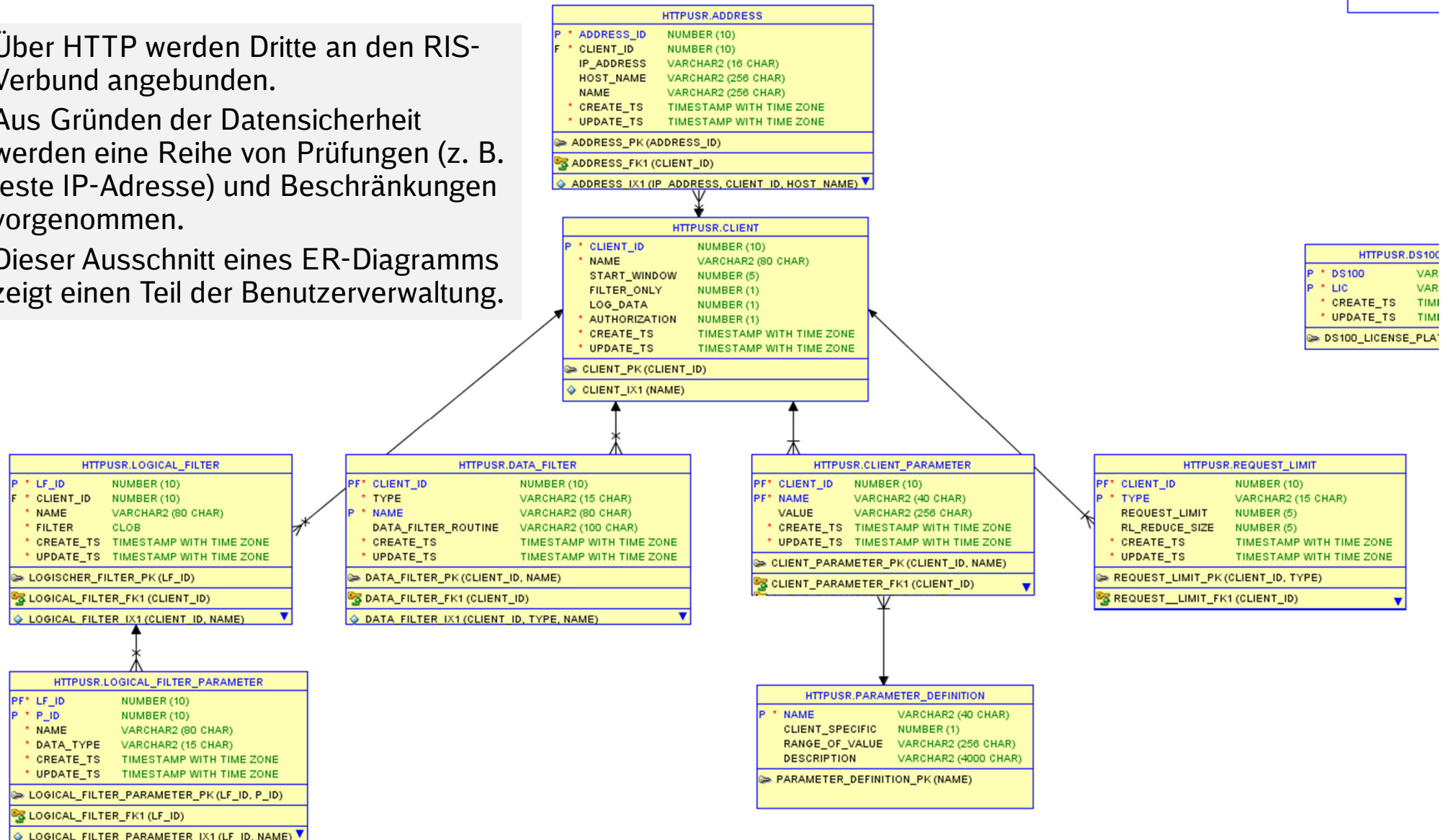
DAILYTT_ACTUALSTOPPING_POINT	
P	STOPPING_POINT_ID NUMBER (7)
	DS100_CANONICAL VARCHAR2 (5 CHAR)
	NAME VARCHAR2 (50 CHAR)
	VALID_FROM TIMESTAMP WITH TIME ZONE
	VALID_UNTIL TIMESTAMP WITH TIME ZONE
	LATITUDE NUMBER (9,6)
	LONGITUDE NUMBER (9,6)
	UIC_COUNTRY_ID NUMBER (8)
	COUNTRY_TIMEZONE VARCHAR2 (64 CHAR)
	TRANSFER_TIME_PRIOR_HIGH INTEGER
	TRANSFER_TIME_PRIOR_LOW INTEGER
F	CARRIER_ADMINISTRATION_ID VARCHAR2 (6 CHAR)
	IS_META_STATION NUMBER (1)
	PRIORITY INTEGER
	TYPE VARCHAR2 (20 CHAR)
	LIC VARCHAR2 (5 CHAR)
	STT_PARTITION_ID INTEGER

DAILYTT_ACTUALCARRIER_ADMINISTRATION	
P	CARRIER_ADMINISTRATION_ID VARCHAR2 (6 CHAR)

DAILYTT_ACTUALCARRIER	
P	CARRIER_ID NUMBER (10)

# Ausschnitt aus dem Datenmodell – HTTP-User

- Über HTTP werden Dritte an den RIS-Verbund angebunden.
- Aus Gründen der Datensicherheit werden eine Reihe von Prüfungen (z. B. feste IP-Adresse) und Beschränkungen vorgenommen.
- Dieser Ausschnitt eines ER-Diagramms zeigt einen Teil der Benutzerverwaltung.





# Austausch von RI-Daten mit RISS

- Ein großer Teil der Datenkommunikation innerhalb des RIS-Verbunds erfolgt über RISML. Für jeden RI-Geschäftsvorfall (Verspätung, Ausfall, Zusatzzug....) existiert eine passende Definition
- Als Messaging-Middleware wird Oracle Advanced Queuing (AQ) eingesetzt (Streaming).
  - Aus der Eingangs-Queue können sich ein- oder mehrere DB-Jobs bedienen und die RISMLs zur Verarbeitung nach dem FiFo-Prinzip entnehmen.
  - Zwischenspeichern in Hochlastphasen oder zur Überbrückung von Störungen
- Herausforderung: RISML ist als XML-Format nicht ideal für ein klassisches, relationales Datenbanksystem. Als „gewachsener Standard“ enthält RISML viele Spezialfälle und Sonderbehandlungen. Die Verarbeitung ist rechenzeitintensiv. Das System ist darauf ausgelegt, bis zu 300 RISML-Nachrichten/s zu verarbeiten, ohne dass es zu einem Rückstau in den Eingangs-AQ kommen darf
- Zur Verarbeitung von RISML wird in PL/SQL eine Verarbeitung in mehreren Schritten durchgeführt:
  1. Paketweise Entnahme aus Eingangs-AQ (6-12 Jobs parallel)
  2. Vorverarbeitung / Normalisierung durch XSLT-Stylesheets
  3. Abbildung der XML-Elemente in Oracle Object Types
  4. Anwendungsfallbezogene Verarbeitung, je nach Nachrichtentyp
  5. Erzeugen von Ausgangs-RISML-Nachrichten für Abnehmer mit entsprechender Subscription

# RISML – Beispiel „IstProg“

```
?xml version="1.0"?>
<Paket Version="1.2" SpezVer="1" TOut="20160122000000713" KNr="444190291">
  <ListNachricht>
    <Nachricht>
      <IstProg>
        <Service Id="139607129310" IdZGattung="S" IdZGattungInt="s" IdBf="KKER" IdBfEvaNr="8000208" IdZeit="20160121235400" IdZNr="33280"
        ZielBfCode="KHEN" ZielBfEvaNr="8002753" Zielzeit="20160122003500" IdVerwaltung="03" SourceZNr="EFZ" IdLinie="12">
          <ListZug>
            <Zug Nr="33280" Gattung="S" GattungInt="s" Linie="12" Name="S 33280" Verwaltung="03">
              <ListZE>
                <ZE Typ="An">
                  <Bf Code="KK" EvaNr="8000207" Name="Köln Hbf"/>
                  <Zeit Soll="20160122000000" Prog="20160122000100" Dispo="NEIN"/>
                </ZE>
                <ZE Typ="Ab">
                  <Bf Code="KK" EvaNr="8000207" Name="Köln Hbf"/>
                  <Zeit Soll="20160122000100" Prog="20160122000200" Dispo="NEIN"/>
                </ZE>
                .....
                <ZE Typ="Ziel">
                  <Bf Code="KHEN" EvaNr="8002753" Name="Hennef(Sieg)"/>
                  <Zeit Soll="20160122003500" Prog="20160122003600" Dispo="NEIN"/>
                </ZE>
              </ListZE>
            </Zug>
          </ListZug>
        </Service>
      </IstProg>
    <ListQuelle>
      <Quelle Sender="10.35.205.12:7213/13" Typ="UIC 102" TIn="20160121235917" Esc="dui0v0k9v3"/>
    </ListQuelle>
  </Nachricht>
</ListNachricht>
</Paket>
```

# Weiterentwicklung von RIS in 2016

- Verbesserte Prognosen für Fahrtverläufe
  - Statistische Berechnungen anstelle von Verspätungs-Fortschreibungen
  - BigData-Ansätze in Erprobung
  
- Wagenreihung
  - Fahrgast soll bis zu seinem Sitzplatz elektronisch geführt werden
  
- Enterprise Service Bus (ESB) als technische Lösung für Zug/Land-Kommunikation
  
- Umstieg auf agile Entwicklungsprozesse (KANBAN, Scrum) für schnellere Entwicklungszyklen

# Fazit: Was sind die Herausforderung bei der Entwicklung großer, datenzentrierter Anwendungen

- Architektur-Entscheidungen müssen gut überlegt werden
  - Business-Logik die große Mengen strukturierte Daten verarbeitet, kann häufig gut in PL/SQL abgebildet werden. Für unstrukturierte Daten sollten auch andere Ansätze geprüft werden
  - Skalierbarkeit: ORACLE RAC ist ein interessantes Konzept, skaliert aber in der Praxis nicht immer linear. Auch andere Ansätze prüfen (z. B. ORACLE (Active) Dataguard oder Mehrschicht-Architekturen)
  - Folgekosten für Hardware und Lizenzen in die Entscheidungen mit einbeziehen
  
- Definierter Entwicklungsprozess:
  - Besser agil als Wasserfall. Kleinere Inkremente, schnelles Feedback des Kunden einholen. Märkte und Anforderungen können sich heute innerhalb weniger Monate verändern!
  - Automatisierte Tests sind bei PL/SQL häufig nicht vorgesehen. Bei größeren Logikanteilen sind diese aber unbedingt zu berücksichtigen
  - Regelmäßige Installation testen!
  
- Anforderungsmanagement
  - Build-to-Run-Anforderungen (B2R) unbedingt frühzeitig klären und bewerten
  - Entwickler müssen die Anforderungen wirklich verstehen (auch Offshore-Teams!)

**Vielen Dank für Ihre Aufmerksamkeit**