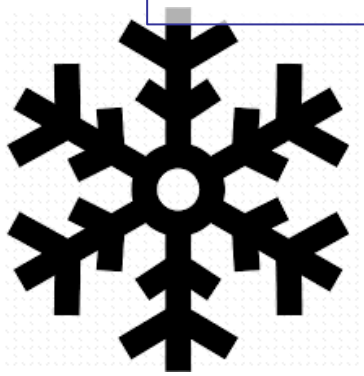
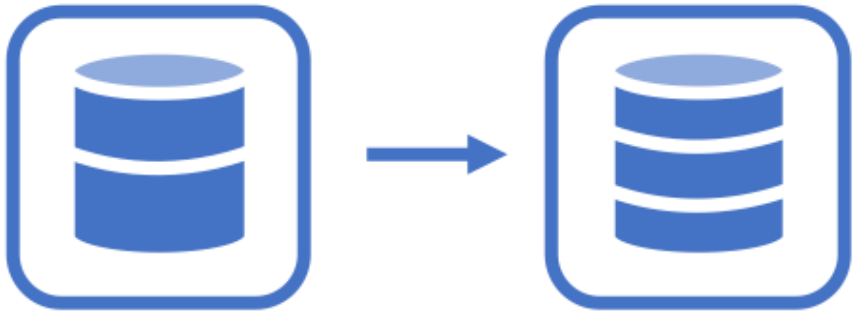
Abschlussprojekt Data Engineering - Gruppe B

Manuela Hebel, xxx, xxx, xxx

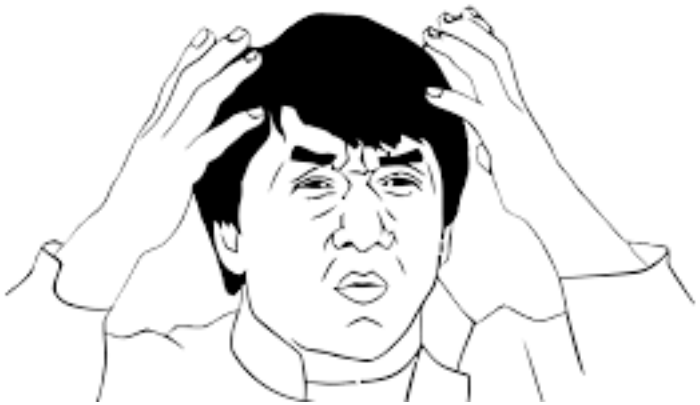




Projektumfang und Anforderungen des Kunden

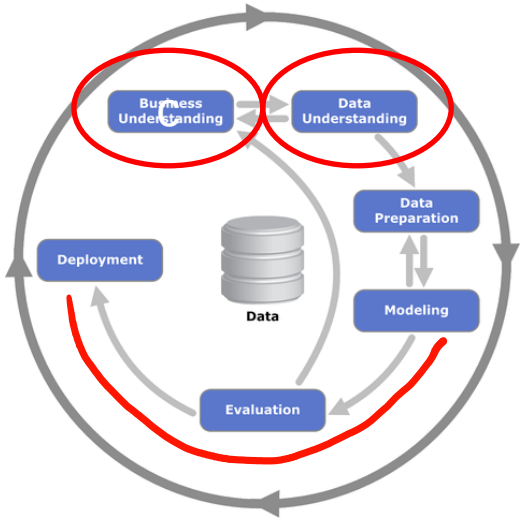
- Zweck:
  - privater Natur und Erschließung neuer Märkte (ggf. cross-selling, Teammanager, …)
  - Gutes Geschäftsmodell für kurzfristige Erfolge
- Kein genau definitierter Nutzer
- Wettarten: alles (Matches, Mannschaften, gelbe Karten, …)
- System muss erweiterbar sein um z.B. international Turniere & andere Sportarten

→ Keine Einschränkungen



Data Understanding

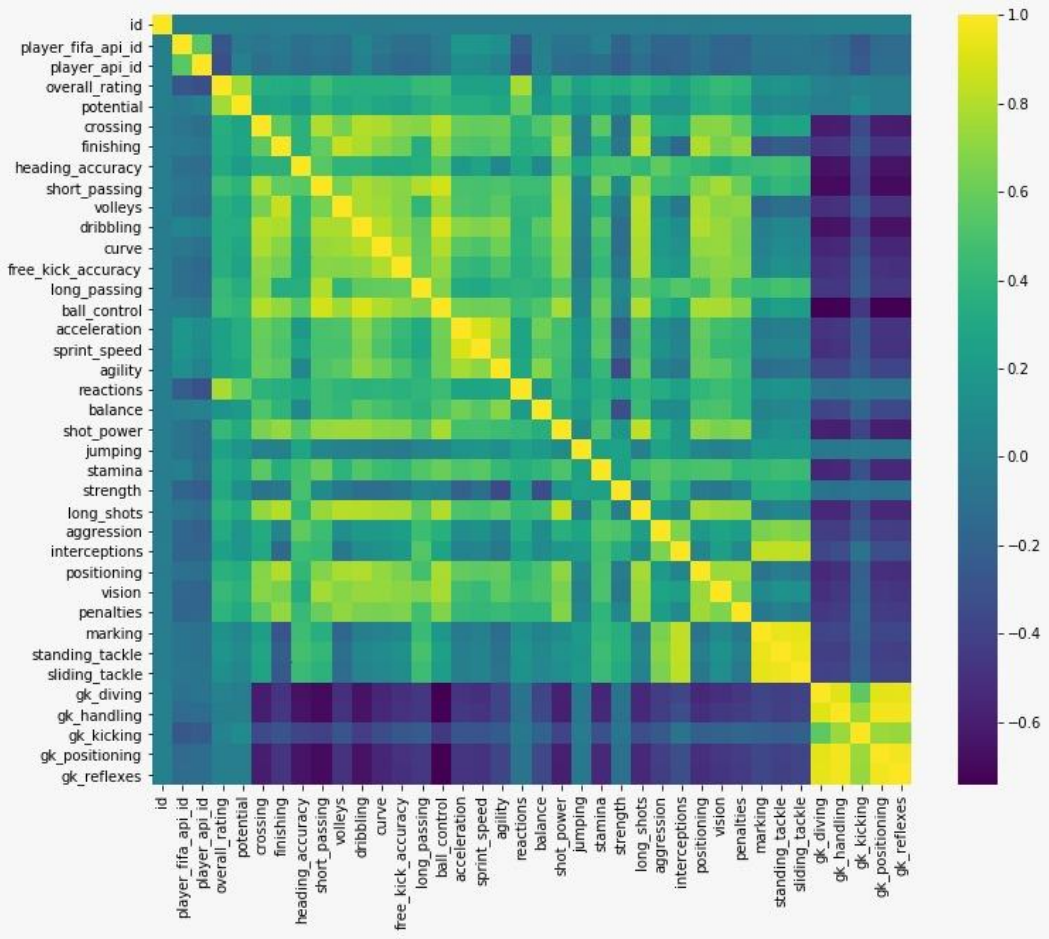
- Business Understanding:
  - Recherche zu Sportwetten (relevante Wettarten)
  - ➢ Ableitung von relevanten Spalten
- Data Understanding:
  - Im Quellsystem wird nicht über PKs referenziert, sondern über die Kombination aus api_id's und fifa_api_id's
  - Zum Teil Zeilen-Duplikate im Quellsystem → drop
  - home_player_X1 & home_player_Y1:
    - X = X-Koordinaten auf Spielfeld (1-9)
    - Y = Y-Koordinaten auf Spielfeld (1-11)
    - Torwart ist immer X=1, Y=1
  - B365H, B365D, B365A, … = Wett-"Odds" der verschiedenen Wettbüros

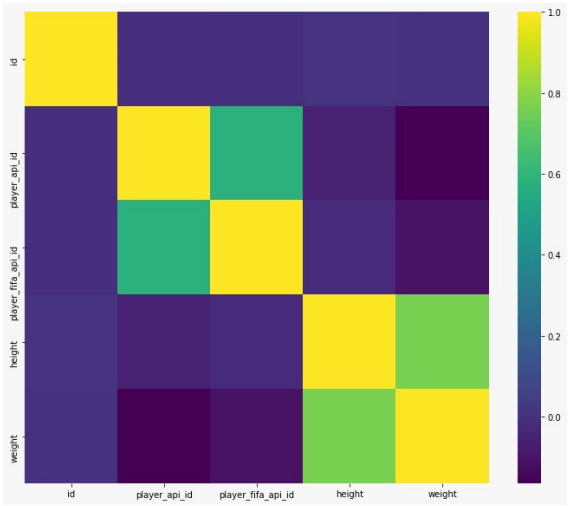


Datenbereinigung

- Null-Werte: drop (wenn signifikante Werte für uns)
- Duplikate: drop
- Falls bei Berechnungen keine Ergebnisse: 99999
  - bei Auswertung berücksichtigen!

Data Exploration

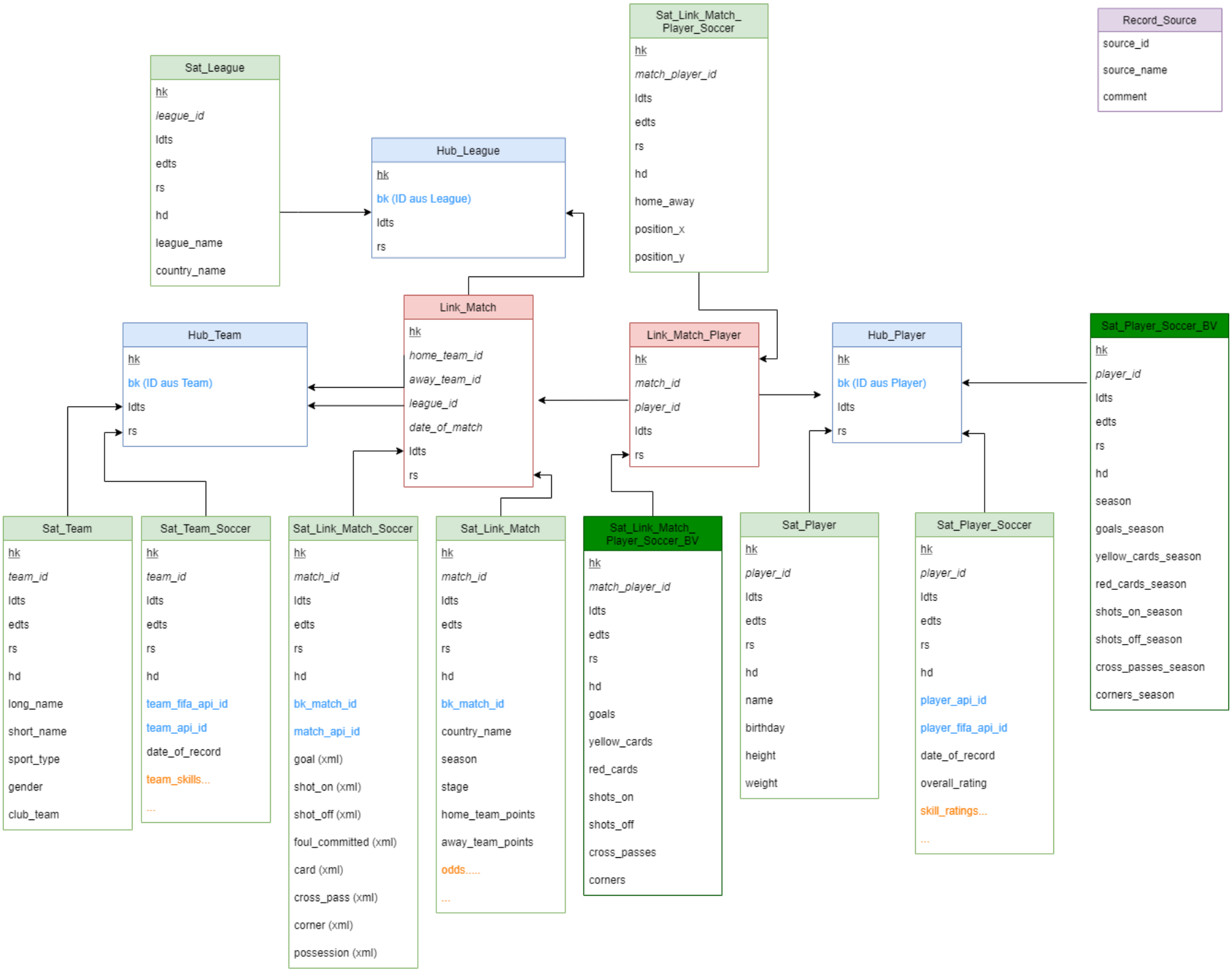




Architekturmodell

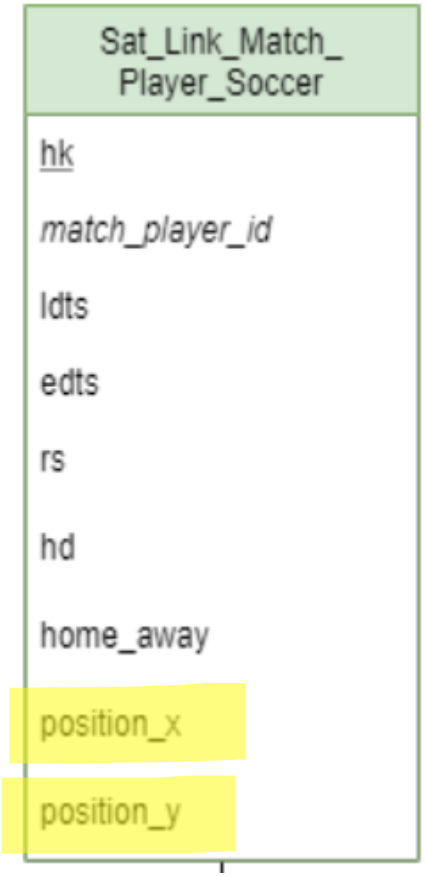
- Warum DataVault?
  - Historisierung
  - Flexible Struktur
  - Einfache Erweiterbarkeit
  - Parallele Beladung

➢ Star-Schema bzw. Snowflake-Schema würden in darauf aufgesetzten Data Marts verwendet werden.


Record_Source
source_id
source_name
comment
Sat_League
hk
league_id
ldts
edts
rs
hd
league_name
country_name
Hub_League
hk
bk (ID aus League)
ldts
rs
Sat_Link_Match_Player_Soccer
hk
match_player_id
ldts
edts
rs
hd
home_away
position_x
position_y
Link_Match
hk
home_team_id
away_team_id
league_id
date_of_match
ldts
rs
Hub_Team
hk
bk (ID aus Team)
ldts
rs
Link_Match_Player
hk
match_id
player_id
ldts
rs
Hub_Player
hk
bk (ID aus Player)
ldts
rs
Sat_Player_Soccer_BV
hk
player_id
ldts
edts
rs
hd
season
goals_season
yellow_cards_season
red_cards_season
shots_on_season
shots_off_season
cross_passes_season
corners_season
Sat_Team
hk
team_id
ldts
edts
rs
hd
long_name
short_name
sport_type
gender
club_team
Sat_Team_Soccer
hk
team_id
ldts
edts
rs
hd
team_fifa_api_id
team_api_id
date_of_record
team_skills...
...
Sat_Link_Match_Soccer
hk
match_id
ldts
edts
rs
hd
bk_match_id
match_api_id
goal (xml)
shot_on (xml)
shot_off (xml)
foul_committed (xml)
card (xml)
cross_pass (xml)
corner (xml)
possession (xml)
Sat_Link_Match
hk
match_id
ldts
edts
rs
hd
bk_match_id
country_name
season
stage
home_team_points
away_team_points
odds.....
...
Sat_Link_Match_Player_Soccer_BV
hk
match_player_id
ldts
edts
rs
hd
goals
yellow_cards
red_cards
shots_on
shots_off
cross_passes
corners
Sat_Player
hk
player_id
ldts
edts
rs
hd
name
birthday
height
weight
Sat_Player_Soccer
hk
player_id
ldts
edts
rs
hd
player_api_id
player_fifa_api_id
date_of_record
overall_rating
skill_ratings...
...

Datenmodell - Erweiterbarkeit

- Zerlegung in sportartspezifische und sportartübergreifende Satelliten (Zweck: Erweiterbarkeit ohne Redundanzen)
- Zerlegung in zwei Links für spielerspezifische Auswertungen
- Zu Link_Match_Player gibt es nur zwei Soccer-spezifische Satelliten:
  - RV für Soccer
  - BV für Soccer
  - → es gibt keine sportart*un*spezifischen Spielerdaten außer home_away pro Spiel. home_away als degradierte Dimension im RDV-Satelliten (Sat_Link_Match_Player)
- Erweiterbarkeit des Modells um:
  - Weitere Sportarten → Sat_Team_**Sportart**, Sat_Player_**Sportart**, usw.
  - Weitere Ligen (außer 1. Liga) → weitere Zeilen in Sat_League
  - Internationale und länderübergreifende Ligen/Turniere → Sat_League.country_name = “international” / ”EU” usw.



Datenmodell – Business Vault

- BV-Satellit für jeden Spieler pro Spiel (Soccer)
  - Extraktion der Spielverlaufsdaten aus xml-Code aus dem Spalten goal, shoton, shotoff, usw. aus der Tabelle Match
  - Auswertungen über z.B. durch einen spezifischen Spieler der Heimmannschaft geschossenen Tore in einer spezifischen Saison
- BV-Satellit für jeden Spieler pro Saison (Soccer)
  - Vergleichbarkeit der Spielerstatistiken über die Spieler hinweg
  - Grundlage für die Berechnung eigener Wett-“Odds“ aus den Spielerdaten
  - Grundlage für die Berechnung von Wett-“Odds“ auf Mannschaftsebene, da wir jeden Spieler einer Mannschaft zuordnen können → ggf. weiteren BV-Satelliten für Hub-Team kreieren.

Datenmodell – Business Vault: Extraktion der Infos aus XML-Spalten

```
root = ET.fromstring("<goal>
                        <value>
                            <comment>n</comment>
                            <stats>
                                <goals>1</goals>
                                <shoton>1</shoton>
                            </stats>
                            <event_incident_typefk>71</event_incident_typefk>
                            <elapsed>6</elapsed>
                            <player1>26392</player1>
                            <sortorder>0</sortorder>
                            <team>8689</team>
                            <id>1315151</id>
                            <n>30</n>
                            <type>goal</type>
                            <goal_type>n</goal_type>
                        </value>
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    <value><comment>o</comment><stats><owngoals>1</owngoals></sta
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    <value><comment>n</comment><stats><goals>1</goals><shoton>1</
                    </goal>")
```

→ lxml-Modul

Methoden, Software, Tools

- Tools/Software
  - Spyder / iphython
  - DB Browser für SQLite3
  - Jupyter-Notebook
- Module
  - Pandas
  - Lxml
  - Sqlite3
  - Hashlib
  - Datetime
- Methoden

Code / Live-Demo (inkl. DDL/DML)

Funktionen:

- hash_function
  - MD5 sollte bei Zeilenzahl >= 500,000 eine ausreichend niedrige p für HK-Colission erzeugen (hk VARCHAR(32))
  - Schwierigkeiten:
    - Link_Match_Player.hk setzt sich aus 5 (!) Spalten aus Quelldatenbank zusammen, + RS (viele JOINS!)
- fill_tables
  - Hubs
  - Links
  - Satelliten
- all_match_players
- get_goals_cards_per_player
- get_shots_on_off_crosses_corners_per_player
- get_first_goal_card_per_team
- prep_match_players_BV
- prep_match_first_goals

ToDos / Ausblick

- Vollständige Befüllung aller Tabellen
  - Aus Zeitgründen nur Befüllung von 500 Zeilen bei:
    - Link_Match_Player & seinen Satelliten
- Update-Routine:
  - Stand jetzt nur try-except für vergebene HKs
- Erstellung und Befüllung weiterer BVs:
  - Helper functions bereits kodiert (z.B. erstes Tor jeder Mannschaft)
- Debugging:
  - Sat_Player_Soccer behindert Join-Abfragen

Best Practices aus den Kundengesprächen

Always stapel low!

Always ask for more time than you get!

Nur Tauschgeschäfte anbieten!

Fragen

- Vielen Dank für eure Aufmerksamkeit!

- Fragen?