**Initial Concept:**

Der Data Lake enthält Fußball Rohdaten, die nach Bearbeitung im Database Server gespeichert werden.

Die jeweiligen Anfragen werden vom Load Balancer verteilt und an den Web Server weitergeleitet. Ist die gesuchte Information bereits im Cache geladen, wird sie von dort abgerufen, falls nicht, wird der Database Server angefragt.

Bei jeder Anfrage zu einem Spieler wird diese gezählt und dadurch der beliebteste Spieler (meistgesucht) ermittelt und auch diese Information in der Database abgespeichert.   
(siehe PPT Folie 2)

**Concept for Single Data Source:**

Da nur eine Datenquelle verfügbar ist und diese Daten bereits in Datenbank Format vorliegen, sind Data Lake und Database Server in unserem Beispiel ein und dasselbe.

Die Informationen zu den Spielern (Größe und Gewicht) werden aus dem Database Server (Tabelle Spieler) herausgezogen und verarbeitet. Der ermittelte Wert (BMI) wird zurückgespeichert in die Tabelle und ist sodann als zusätzliche Information verfügbar.

Die Möglichkeit zur Bestimmung des beliebtesten Spielers bleibt bestehen.

**Umsetzung:**

1. Erstellung der App auf Basis der Proprietäre App (Schaubild Aufgabenstellung) im Unterricht (Request, Load Balancer, Web Server(s), Cache Server(s).

Funktioniert vollumfänglich (Video 1)

1. Erstellung der zusätzlichen Zählung und Speicherung des beliebtesten Spielers

* Alternative eigenständige Lösung: Word Count der Kommentare zum WM Finale 2014  
  (zeigt den beliebtesten bzw. Spieler mit den meisten nennenswerten Aktionen)

(Video 2a)

Um Zugriffe von außen auf das System zu simulieren wurde ein python skript zum automatisierten abfragen von Spielern (Spieler ID) geschrieben. Hierdurch kann der beliebteste Spieler (meiste Anfragen) simuliert werden.

(Video 2b)

Der Ordner „data“ enthält neben dem im Video gezeigten WordCount WM Beispiel auch zahlreiche py Dateien in denen wir auf verschiedensten Wegen versucht haben die Daten aus dem Database Server mittels einer jdbc Verbindung herauszulesen (Hier wird deutlich, dass in unserem Beispiel der Database Server auch gleichzeitig der DataLake für diese Funktion hätte sein sollen). Darüber hinaus auch eine der Fehlermeldung(en) die beim Submit der Dateien angezeigt wurden. Leider hatten wir nicht die nötige Expertise zur Lösung im Team und konnten uns diese auch nicht aneignen.

1. Datenabfrage an Database Server, Datenverarbeitung und Datenrückspeicherung

* Alternative eigenständige Lösung: Zugriff auf die MYSQL Datenbank (Database Server) mit mysql terminal  
  (zeigt den Zugriff auf die Daten und ermöglicht die Bearbeitung, allerdings nicht wie geplant mit Apache Spark (py Code) sondern lediglich in mysql und dadurch direkt auf dem Server.

(Video 3)

Video 3 soll zeigen, dass es wir in der Lage waren, den Database Server erfolgreich zu adressieren und diese mit einem alternativen Tool auch anzeigen zu lassen.

Hätte die Erstellung eines dataframe mittels jdbc Anbindung funktioniert, würden wir die Daten (Größe und Gewicht) in der py. Datei verarbeiten und anschließend zurückspeichern in die Datenquelle (Database Server = Data Lake).

Zusammenfassung:

Die Unterrichtsinhalte können reproduziert und in Artgleichen Beispielen angewendet werden. Der Zugriff auf die Daten des mysql Servers (Database Servers) mittels pyspark wurde nicht erreicht. Die aufgekommenen Fehlermeldungen konnten nicht zur Lösungsfindung beitragen. Zahlreiche Versuche und Tipps aus Foren haben nicht zum Erfolg geführt. Als Gruppe bestehend aus reinen Anfängern konnten wir keine zusammenhängende Gesamtlösung erzielen, haben allerdings Alternativen erarbeitet die einen Use Case wie oben beschrieben darstellen.