

ML auf die Schiene gebracht

Thementag „Ist mein Zug pünktlich?“

Autor: Christoph Schmidt-Casdorff

16.08.2022



Agenda

- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML-Modelle zur Laufzeit – MLflow
- Monitoring
- Gesamtarchitektur
- Take Home

Agenda

- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML Modelle zur Laufzeit – MLflow
- Monitoring
- Gesamtarchitektur
- Take Home

Abfahrt <i>Departure / Départ</i>				 	
Zeit <i>Time / Temps</i>		Über <i>Via</i>	Ziel <i>Destination</i>	Gleis <i>Platform / Voie</i>	
10:35	MRB 80303	MarkkleebergMitte - Gaschwitz	Borna (b. Leipzig)	8	
10:35	S 37034	Leipzig-Gohlis - Leipzig-Wahren	Halle (Saale) Hbf	6	
10:47	RB 26417	Leipzig-Leutzsch - Markranstädt	Weißenfels	7	
10:51	ICE 1555	Riesa - Dresden-Neustadt	Dresden Hbf	10	
10:51	ICE 1612	Lu Wittenberg - BerlinSüdkreuz	Hamburg-Altona	11	
11:01	RE 26091	Leipzig-Plagwitz - Neukieritzsch	Werdau	8	-Lehndorf+++Schieneners
11:03	RE 17065	Borsdorf - Wurzen - Oschatz	Dresden Hbf	19	
11:03	IC 2038	MagdeburgHbf - BraunschweigHbf	Oldenburg(Oldb)	18	ten Fahrzeiten+++Heute in
11:04	MRB 80171	Mölkau - Liebertwolkwitz	Geithain	23	+++wonderer Zug+++
11:05	S 37036	Leipzig-Gohlis - Leipzig-Wahren	Halle (Saale) Hbf	6	
11:07	RE 18457	Eilenburg - Torgau - Falkenberg (E)	Cottbus	9	
11:07	BUS 26016	LeipzigMesse - Leipzig/HalleFlugh	Halle (Saale) Hbf	17	esse, Bussteig 5 Ostseite++

<https://www.bahnbilder.de/1200/die-grosse-abfahrtstafel-leipzig-hbf-732028.jpg>

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Abfahrtstafeln der Bahn

- ✿ Momentaufnahme der (in nächster Zeit) abfahrenden Züge
 - ◆ Es sind Verspätungen und allgemeine Informationen zu finden
 - ◆ ... es kann in einigen Minuten ganz anders aussehen
 - Es handelt sich immer nur um eine Momentaufnahme
- ✿ Sobald ein Zug abgefahren ist, verschwindet er von der Abfahrtstafel

Idee für unser Beispiel

- ✿ Wir laden Abfahrtstafeln der Deutschen Bahn
- ✿ Wir setzen ein ML-Modell auf,
 - ◆ welches die Verspätungen im Zielbahnhof vorhersagt
 - ◆ Das Modell liefert einen kontinuierlichen Strom an Vorhersagen
 - Zu jeder Abfahrt wird eine Vorhersage getroffen
- ✿ Wir setzen kein(e) Frontend/Oberfläche um
 - ◆ Endpunkt ist der Strom der Vorhersagen

Unsere Daten – Abfahrtstafeln

* Deutsche Bahn bietet eine Menge an öffentlichen Daten an

- ◆ <https://data.deutschebahn.com/dataset.html>
- ◆ Abfahrtsmonitor: <https://iris.noncd.db.de/wbt/js/index.html?bhf=EDG>

* Das Portal DBF ist ein (inoffizieller) Abfahrtsmonitor

- ◆ <https://finalrewind.org/projects/Travel-Status-DE-IRIS/>
- ◆ DBF stellt einen Open-Source-Client zur Abfrage von Abfahrtsstafeln bereit
 - <https://github.com/derf/db-fakedisplay/blob/master/README.md>
 - REST-API
 - Docker-Image verfügbar (<https://hub.docker.com/r/derfnull/db-fakedisplay>)
- ◆ DBF basiert auf den öffentlichen Schnittstellen der Bahn
- ◆ Die Schnittstellen DBF sind nicht beliebig stark zu frequentieren
 - Netiquette

Holzwickede, 20.6.2022 12:15

```
{ "departures": [  
  { "train": "RE 7", "trainClasses": ["N"], "trainNumber": "32557", "destination": "Krefeld Hbf", "platform": "3",  
    "delayArrival": 2, "delayDeparture": 2, "scheduledArrival": "12:18", "scheduledDeparture": "12:19", "scheduledPlatform": "3"  
  },  
  { "train": "RB 59", "trainClasses": ["N"], "trainNumber": "90331", "destination": "Soest", "platform": "4",  
    "delayArrival": 0, "delayDeparture": 0, "scheduledArrival": "12:26", "scheduledDeparture": "12:27", "scheduledPlatform": "4"  
  },  
  
  { "train": "RB 59", "trainClasses": ["N"], "trainNumber": "90330", "destination": "Dortmund Hbf", "platform": "5",  
    "delayArrival": 1, "delayDeparture": 1, "scheduledArrival": "12:31", "scheduledDeparture": "12:32", "scheduledPlatform": "5"  
  },  
  .....  
]}
```

Holzwickede, 20.6.2022 12:17

```
{ "departures": [
  { "train": "RE 7", "trainClasses": ["N"], "trainNumber": "32557", "destination": "Krefeld Hbf", "platform": "3",
    "delayArrival": 1, "delayDeparture": 1, "scheduledArrival": "12:18", "scheduledDeparture": "12:19", "scheduledPlatform": "3"
  },
  { "train": "RB 59", "trainClasses": ["N"], "trainNumber": "90331", "destination": "Soest", "platform": "4",
    "delayArrival": 1, "delayDeparture": 1, "scheduledArrival": "12:26", "scheduledDeparture": "12:27", "scheduledPlatform": "4"
  },
  { "train": "RB 59", "trainClasses": ["N"], "trainNumber": "90330", "destination": "Dortmund Hbf", "platform": "5",
    "delayArrival": 2, "delayDeparture": 2, "scheduledArrival": "12:31", "scheduledDeparture": "12:32", "scheduledPlatform": "5"
  },
  .....
]}
```

Agenda

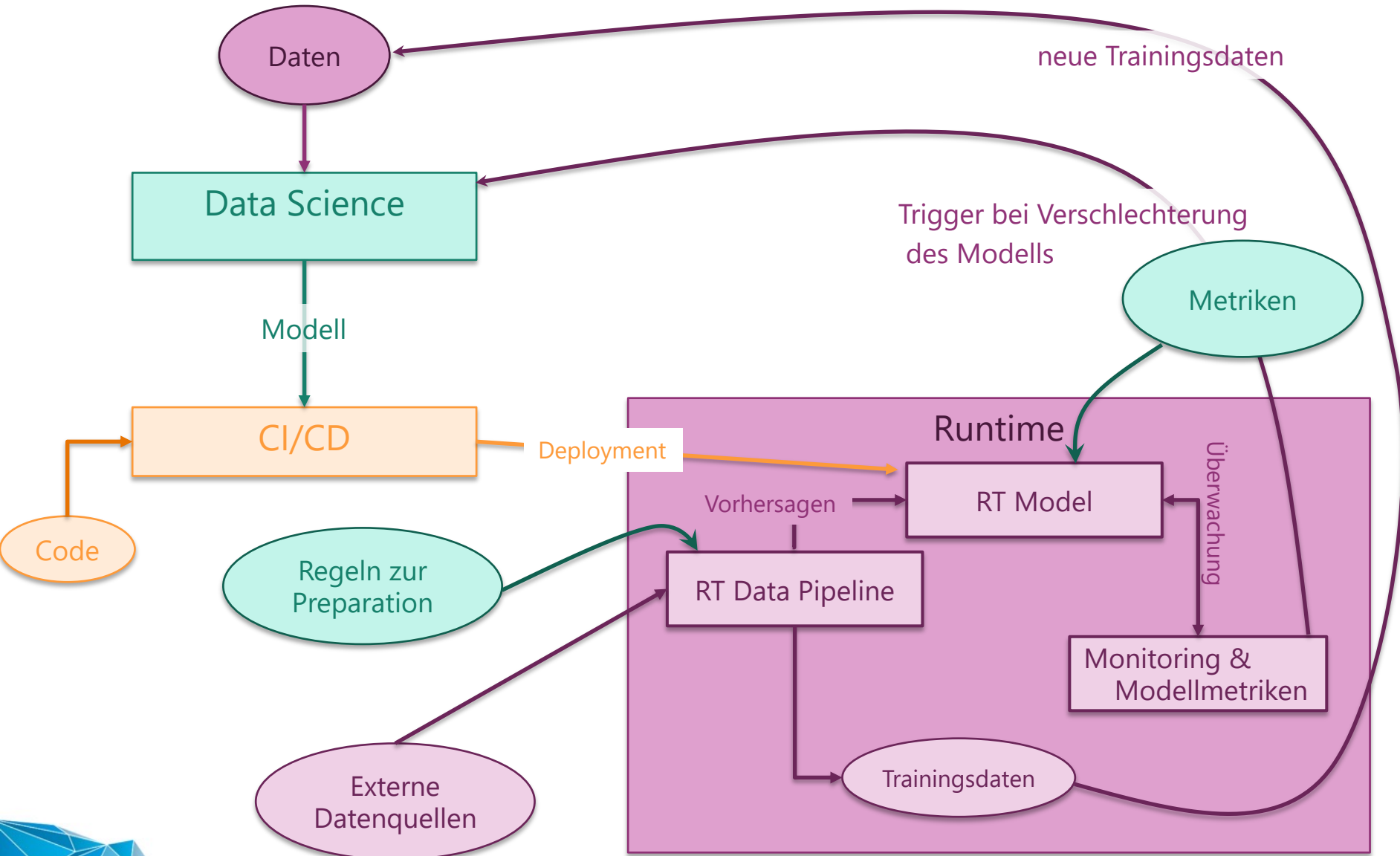
- Mit welchen Fragen beschäftigen wir uns?
- **Runtime Data Pipeline**
- ML Modelle zur Laufzeit – MLflow
- Monitoring
- Gesamtarchitektur
- Take Home

Was ist zu tun?

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Laufzeitumgebung eines Data-Science-Systems



Eigenschaften des Systems

* Stetig einkommende Daten

- ◆ Diese sollen sofort beim Eintreffen verarbeitet werden

=> Echtzeitverarbeitung und Streaming

Eigenschaften des Systems

- * Echtzeitverarbeitung und Streaming
- * Datenmenge ist nicht begrenzt und kann beliebig groß werden

=> (Horizontale) Skalierbarkeit

Eigenschaften des Systems

- ✧ Echtzeitverarbeitung und Streaming
- ✧ (Horizontale) Skalierbarkeit
- ✧ Daten kommen aus beliebig vielen und unterschiedlichen Datenquellen
 - ◆ Daten werden über REST-API und Filesystem/Datenbank eingespeist
 - Historische Daten befinden sich im Filesystem
 - ◆ Daten müssen zusammengeführt werden

=> Verteilte Verarbeitung der Daten

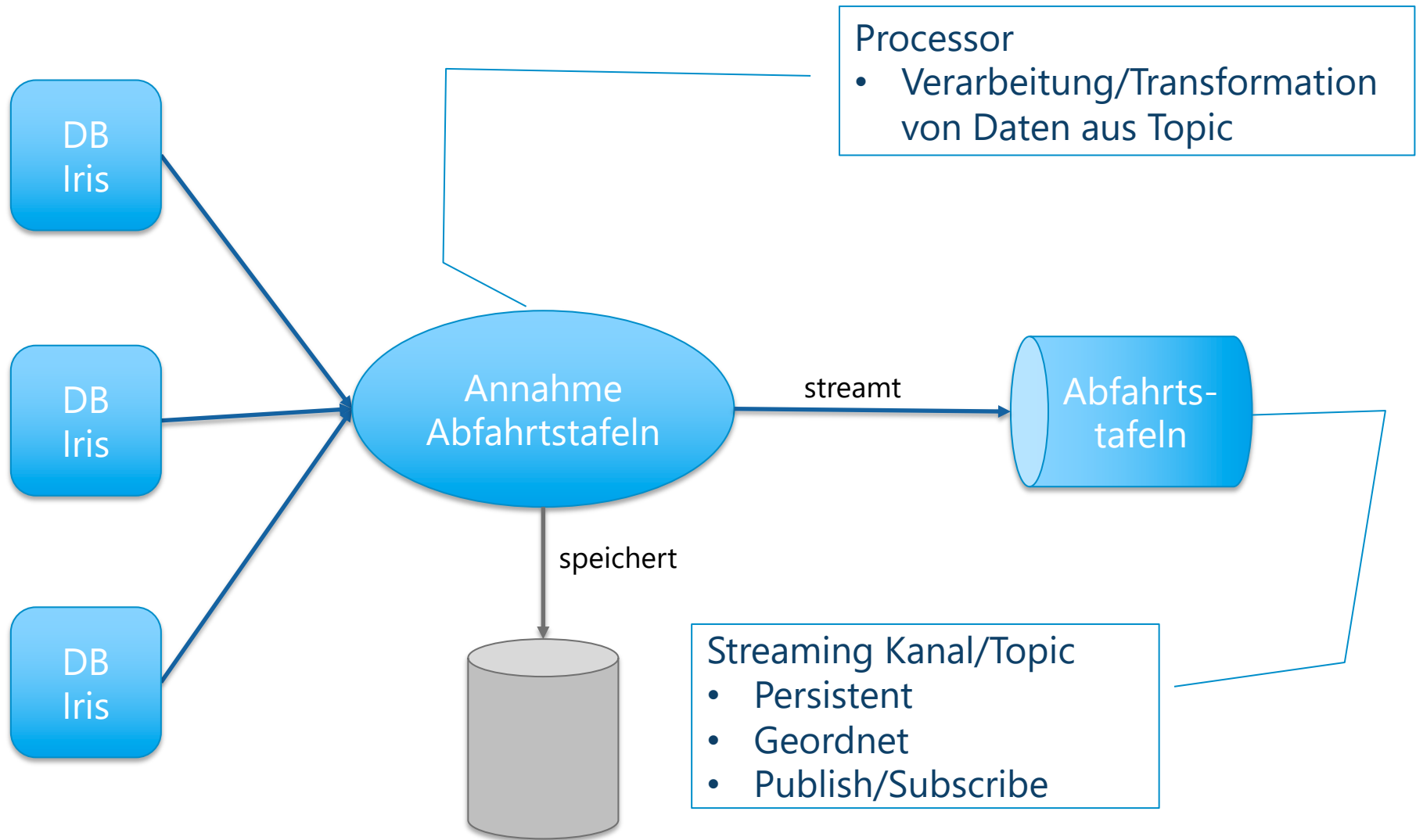
Eigenschaften des Systems

- ✧ Echtzeitverarbeitung und Streaming
 - ✧ (Horizontale) Skalierbarkeit
 - ✧ Verteilte Verarbeitung der Daten
 - ✧ System muss mit zeitweise starkem Datenaufkommen klarkommen
- => Pufferungsmechanismus (*Backpressure*)

Eigenschaften des Systems

- * Echtzeitverarbeitung und Streaming
- * (Horizontale) Skalierbarkeit
- * Verteilte Verarbeitung der Daten
- * Pufferungsmechanismus (*Backpressure*)

Schritt 1: Einlesen der Abfahrtstafeln



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | **Runtime Data Pipeline** | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Holzwickede, 20.6.2022 12:15

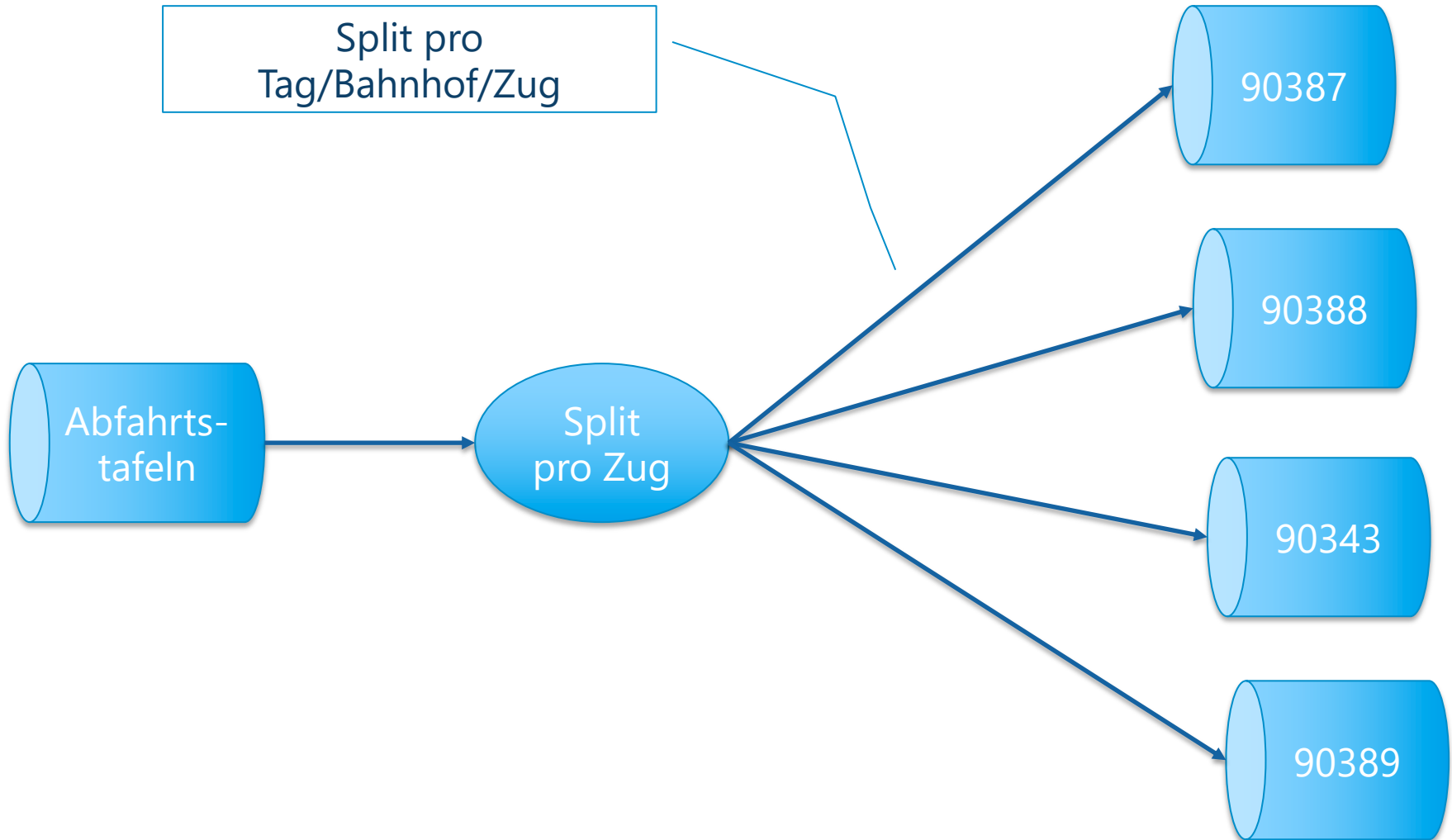
```
{ "departures": [  
  { "train": "RE 7", "trainClasses": ["N"], "trainNumber": "32557", "destination": "Krefeld Hbf", "platform": "3",  
    "delayArrival": 2,"delayDeparture": 2 "scheduledArrival": "12:18", "scheduledDeparture": "12:19", scheduledPlatform": "3"  
  },  
  { "train": "RB 59", "trainClasses": ["N"], "trainNumber": "90331", "destination": "Soest", "platform": "4",  
    "delayArrival": 0,"delayDeparture": 0, "scheduledArrival": "12:26","scheduledDeparture": "12:27","scheduledPlatform": "4"  
  },  
  
  { "train": "RB 59", "trainClasses": ["N"], "trainNumber": "90330", "destination": "Dortmund Hbf", "platform": "5",  
    "delayArrival": 1,"delayDeparture": 1,"scheduledArrival": "12:31", "scheduledDeparture": "12:32","scheduledPlatform": "5"  
  },  
  .....  
]}
```

Abfahrten pro Zug

```
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:15:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 1,
  "delayDeparture": 1,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:17:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
}
```

```
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:15:00",
  "train": "RB 59",
  "trainClasses": ["N"],
  "trainNumber": "90331",
  "destination": "Soest",
  "platform": "4",
  "delayArrival": 0,
  "delayDeparture": 1,
  "scheduledArrival": "12:26",
  "scheduledDeparture": "12:27",
  "scheduledPlatform": "4"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:17:00",
  "train": "RB 59",
  "trainClasses": ["N"],
  "trainNumber": "90331",
  "destination": "Soest",
  "platform": "4",
  "delayArrival": 1,
  "delayDeparture": 1,
  "scheduledArrival": "12:26",
  "scheduledDeparture": "12:27",
  "scheduledPlatform": "4"
},
...
```

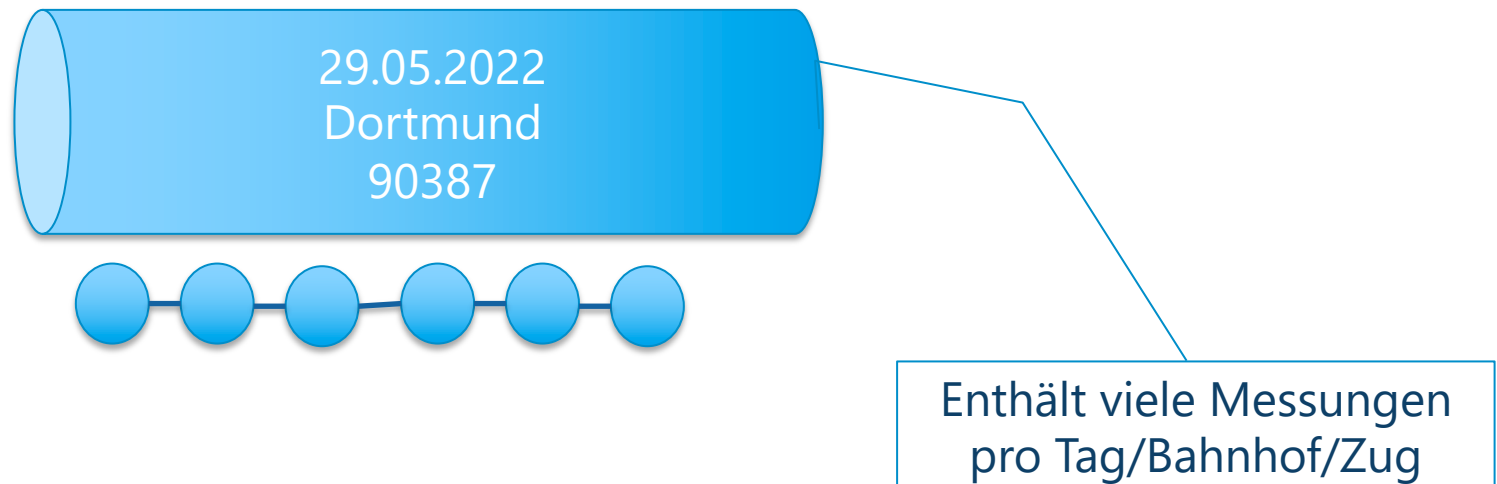
Schritt 2: Aufspalten pro Zug und Bahnhof



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | **Runtime Data Pipeline** | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Wann ist ein Zug abgefahren?



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | **Runtime Data Pipeline** | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Abfahrten pro Zug

```
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:13:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 1,
  "delayDeparture": 1,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:15:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 1,
  "delayDeparture": 1,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:17:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:19:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:21:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
}
```

Wie stellen wir fest, dass ein Zug den Bahnhof verlassen hat?

Abfahrten

✿ Wie stellen wir fest, dass ein Zug den Bahnhof verlassen hat?

✿ Idee:

Warte bis zu einem Zug und Bahnhof keine Meldung mehr kommt
=> dann gilt der Zug als abgefahren

Windowing

* Daten strömen stetig ein

- ◆ Es ist nicht klar, wann zusammengehörige Daten vollständig verarbeitet sind

Windowing

✿ Daten strömen stetig ein

✿ Welche Zeitpunkte sind zu beachten?

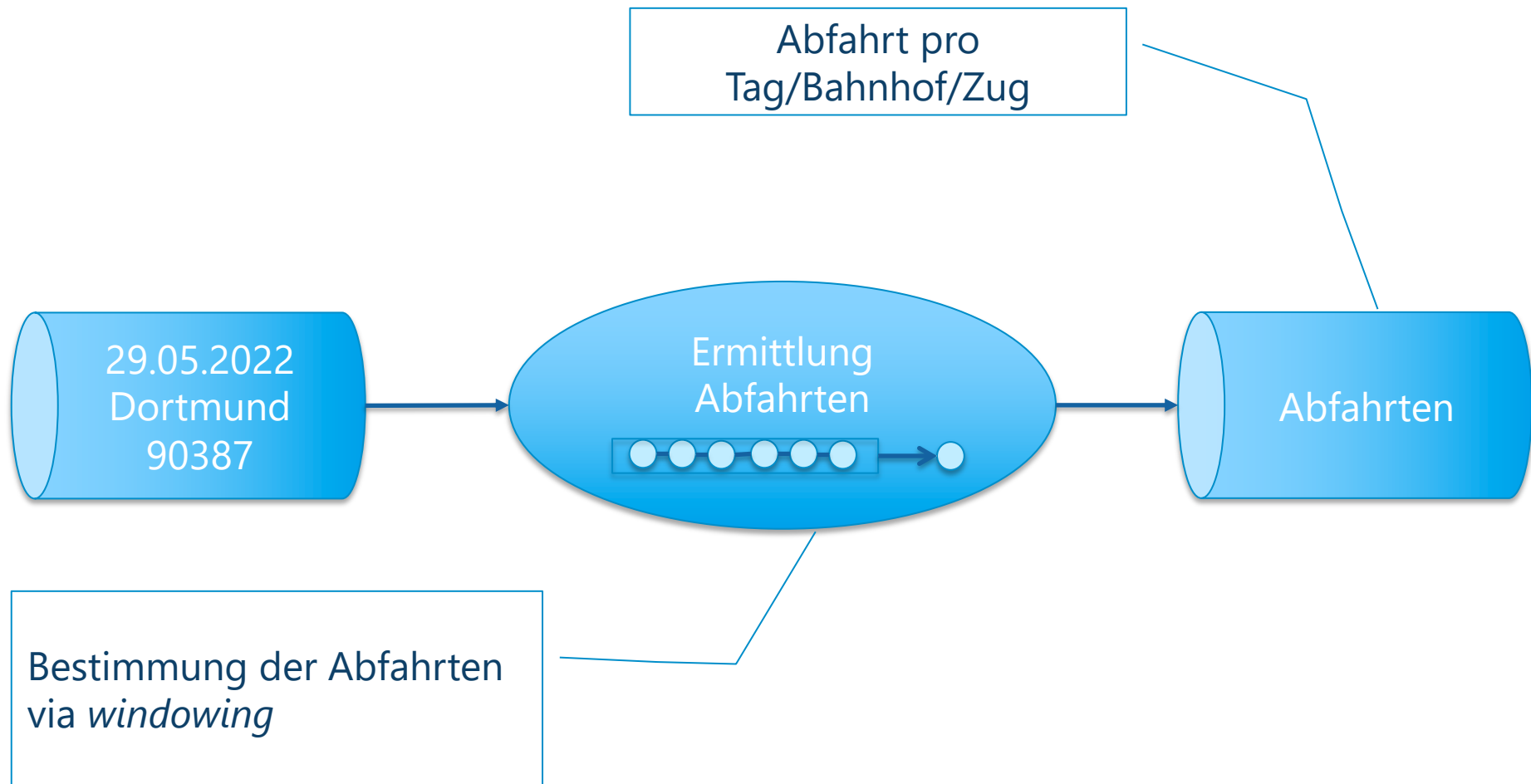
- ◆ Ereigniszeitpunkt – Wann hat das Ereignis tatsächlich stattgefunden?
- ◆ Verarbeitungszeitpunkt – Wann wurde das Ereignis im System verarbeitet?

Windowing

- * Daten strömen stetig ein
- * Ereigniszeitpunkt und Verarbeitungszeitpunkt
- * Wir benötigen Algorithmen, die
 - ◆ Abfahrten sammeln, bis n–Minuten keine Abfahrt für einen Zug und Bahnhof gemeldet werden
 - Ereigniszeitpunkt
 - ◆ eine Karenzzeit von x-Minuten berücksichtigen
 - Verarbeitungszeitpunkt kann sich verzögern

=> Entscheidung über Abfahrtszeitpunkt eines Zuges in einem Bahnhof

Windowing – Bestimmung einer Abfahrt



ML auf die Schiene gebracht

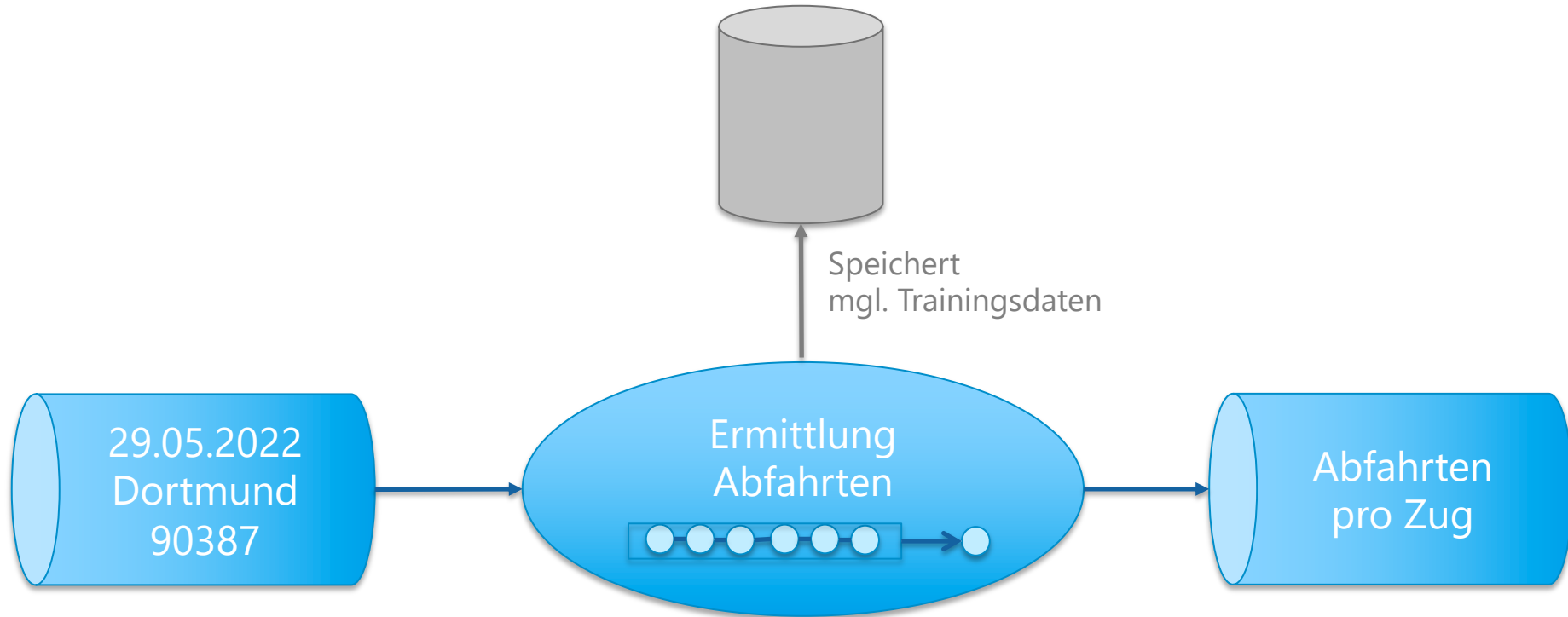
Mit welchen Fragen beschäftigen wir uns? | **Runtime Data Pipeline** | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Abfahrten pro Zug

```
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:13:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 1,
  "delayDeparture": 1,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:15:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 1,
  "delayDeparture": 1,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:17:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:19:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
},
{
  "station": "Holzwickede",
  "timestamp": "2022-06-20 12:21:00",
  "train": "RE 7",
  "trainClasses": ["N"],
  "trainNumber": "32557",
  "destination": "Krefeld Hbf",
  "platform": "3",
  "delayArrival": 2,
  "delayDeparture": 2,
  "scheduledArrival": "12:18",
  "scheduledDeparture": "12:19",
  "scheduledPlatform": "3"
}
```

12:28 wurde entschieden : Zug ist um 12:21 abgefahren

Abfahrten sind potentielle Trainingsdaten



Wir haben wir es gemacht - Data Pipeline



ML auf die Schiene gebracht

Wie haben wir es gemacht?

- ✱ Wir haben eine on-premise-Lösung umgesetzt.
 - ◆ Wir haben keine Cloud-Plattform wie AWS eingesetzt

Wie haben wir es gemacht?

- ✿ Wir haben eine on-premise-Lösung umgesetzt.

- ✿ Wir setzen Kafka als Plattform für die Data Pipeline ein
 - ◆ Kafka besitzt die geforderten Eigenschaften von Hause aus
 - Streaming
 - Skalierbarkeit
 - Backpressure
 - Windowing

Wie haben wir es gemacht?

- * Wir haben eine on-premise-Lösung umgesetzt.
- * Wir setzen Kafka als Plattform für die Data Pipeline ein
- * Wir setzen Kafka zur verteilten Verarbeitung
 - ◆ Sortierung / Gruppierung ist bei großen Datenbeständen anspruchsvoll
 - ◆ Kafka unterstützt die Gruppierung von Streamingdaten zur Laufzeit

Wie haben wir es gemacht?

- * Wir haben eine on-premise-Lösung umgesetzt.
- * Wir setzen Kafka als Plattform für die Data Pipeline ein
- * Wir setzen Kafka zur verteilten Verarbeitung
- * Wir setzen Kafka als Komponente zur Daten-Einspeisung
 - ◆ Optimal, um unterschiedlichen Datenquellen + Geschwindigkeiten zu koordinieren
 - ◆ Beliebig skalierbar

Wie haben wir es gemacht?

- ✿ Wir haben eine on-premise-Lösung umgesetzt.
- ✿ Wir setzen Kafka als Plattform für die Data Pipeline ein
- ✿ Wir setzen Kafka zur verteilten Verarbeitung
- ✿ Wir setzen Kafka als Komponente zur Daten-Einspeisung
- ✿ Kafka zur bietet *windowing-Funktionalität* an
 - ◆ Bestimmung der Abfahrten pro Zug und Bahnhof

Wie haben wir es gemacht?

- * Wir haben eine on-premise-Lösung umgesetzt.
- * Wir setzen Kafka als Plattform für die Data Pipeline ein
- * Wir heben Rohdaten auf
 - ◆ Vielleicht ergeben sich später andere Fragestellungen

Wie haben wir es gemacht?

- ✿ Wir haben eine on-premise-Lösung umgesetzt.
- ✿ Kafka als Plattform für die Data Pipeline ein
- ✿ Wir heben Rohdaten auf
- ✿ Wir speichern Ergebnisse der Data Pipeline als mögliche Trainingsdaten

Wie haben wir es gemacht?

- * Wir haben eine on-premise-Lösung umgesetzt.
- * Kafka als Plattform für die Data Pipeline ein
- * Wir heben Rohdaten auf
- * Wir speichern Ergebnisse der Data Pipeline als mögliche Trainingsdaten
- * Unsere Abfahrtszeiten sind so genau, wie die Abfragezyklen bei der DB
 - ◆ Derzeit wird ca. alle 3 Min ein Bahnhof angefragt
 - ◆ Besser sind unsere Abfahrtszeiten damit auch nicht

Agenda

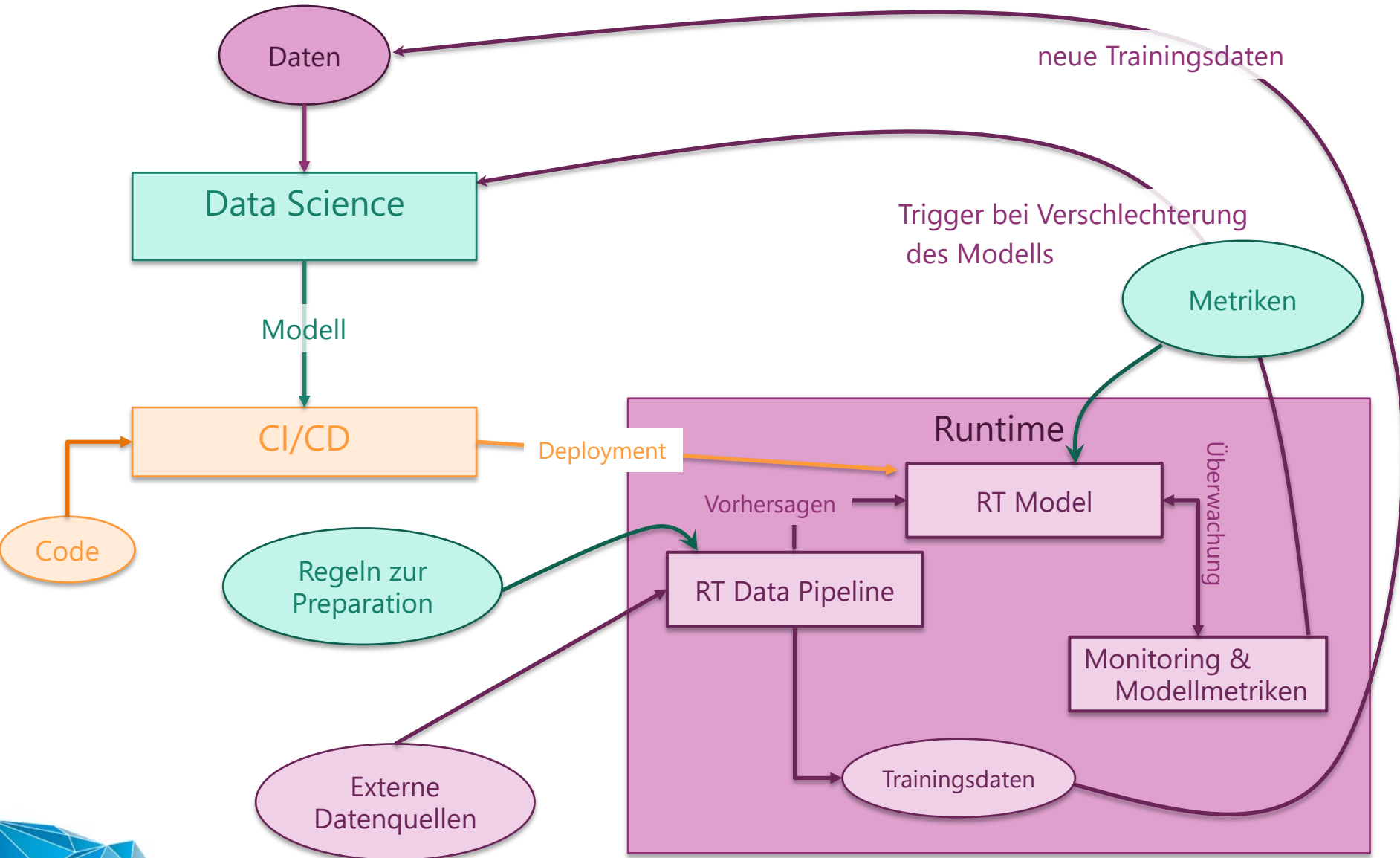
- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML-Modelle zur Laufzeit – MLflow
- Monitoring
- Gesamtarchitektur
- Take Home

Wie haben wir es gemacht - MLOps



<https://ml-ops.org/>

Laufzeitumgebung eines Data-Science-Systems



Problemstellung

- ✿ Nahtloser Übergang eines ML-Modells von Data Science bis zur Produktion
 - ◆ Wie wechselt ein ML-Modell die unterschiedlichen Umgebungen?



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Problemstellung

- ✿ Nahtloser Übergang eines ML-Modells von Data Science bis zur Produktion
- ✿ ML-Modell ist in einen CI/CD-Prozess zu integrieren



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Wie haben wir es gemacht?

- * Wir haben uns für eine Plattform namens MLflow entschieden
 - ◆ <https://mlflow.org/> von DataBricks
- * MLflow ist speziell für die Prozesse in MLOps entworfen worden
 - ◆ Nahtloses Zusammenspiel der einzelnen Aufgaben/Phasen
- * All das kann auch mit anderen Mitteln/Tools umgesetzt werden
 - ◆ Umsetzung mit docker
 - Die Integration der einzelnen MLOps-Phasen muss ausprogrammiert werden
 - ◆ Einsatz anderer MLOps Tools
 - DVC
 - ◆ Explizite Re-Programmierung der ML-Modelle
 - Spark, Kafka, ... besitzen eigene ML-Programmiermodelle

Konzepte von MLflow

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home



MLflow

* MLflow ist eine Open-Source-Plattform, ...

- ◆ um den ML-Lebenszyklus zu verwalten
 - Einschließlich von ML-Experimenten
- ◆ die die Reproduzierbarkeit von ML-Modellen gewährleistet
- ◆ die das Deployment von ML-Modellen ermöglicht

MLflow



✿ MLflow ist eine Plattform, um den Lebenszyklus eines ML-Modells zu verwalten

✿ MLflow Model & Projects

- ◆ definieren universelles Austauschformat für ML-Projekte
 - Modell inkl. Algorithmus
 - Metriken müssen geprüft werden
 - Trainings- und Testdaten
 - Software, in der das Modell geschrieben ist, ist verfügbar
 - Code zum Testen des Modells

MLflow



✿ MLflow ist eine Plattform, um den Lebenszyklus eines ML-Modells zu verwalten

✿ MLflow Model & Projects

- ◆ definieren universelles Austauschformat für ML Projekte
- ◆ Verstecken die Abhängigkeit zu ML Frameworks
- ◆ Ermöglichen das Deployment auf unterschiedlichen Ausführungsumgebungen

MLflow – take home



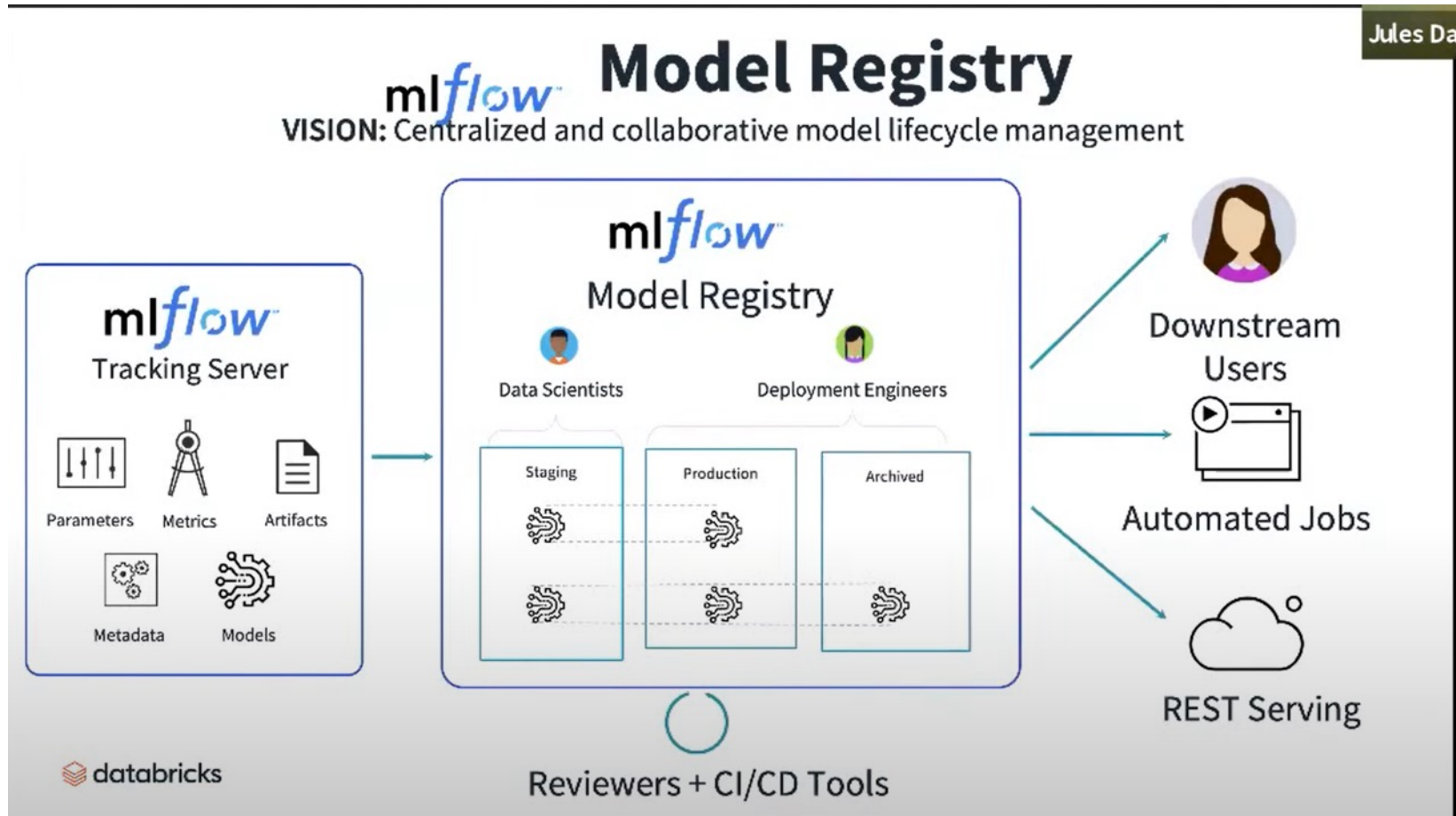
- * MLflow ist eine Plattform um den Lebenszyklus eines ML-Modells zu verwalten
- * ML-Modelle als Softwareprojekte: MLflow Model & Projects
- * MLflow Tracking-Server-Komponente
 - ◆ Verwaltet Experimente
 - Data Science kann Experimente auf den Tracking-Server laden
 - ◆ Experimente können veröffentlicht werden

MLflow – take home



- * MLflow ist eine Plattform um den Lebenszyklus eines ML-Modells zu verwalten
- * ML-Modelle als Softwareprojekte : MLflow Model & Projects
- * Verwaltung von Experimenten : MLflow Tracking Server
- * MLflow Registry Server
 - ◆ Server-Komponente
 - ◆ Verwaltet veröffentlichte ML-Modelle
 - ◆ Bietet *Build* und Ausführung der Modelle an
 - ◆ Bietet ‚staging‘-Konzept an

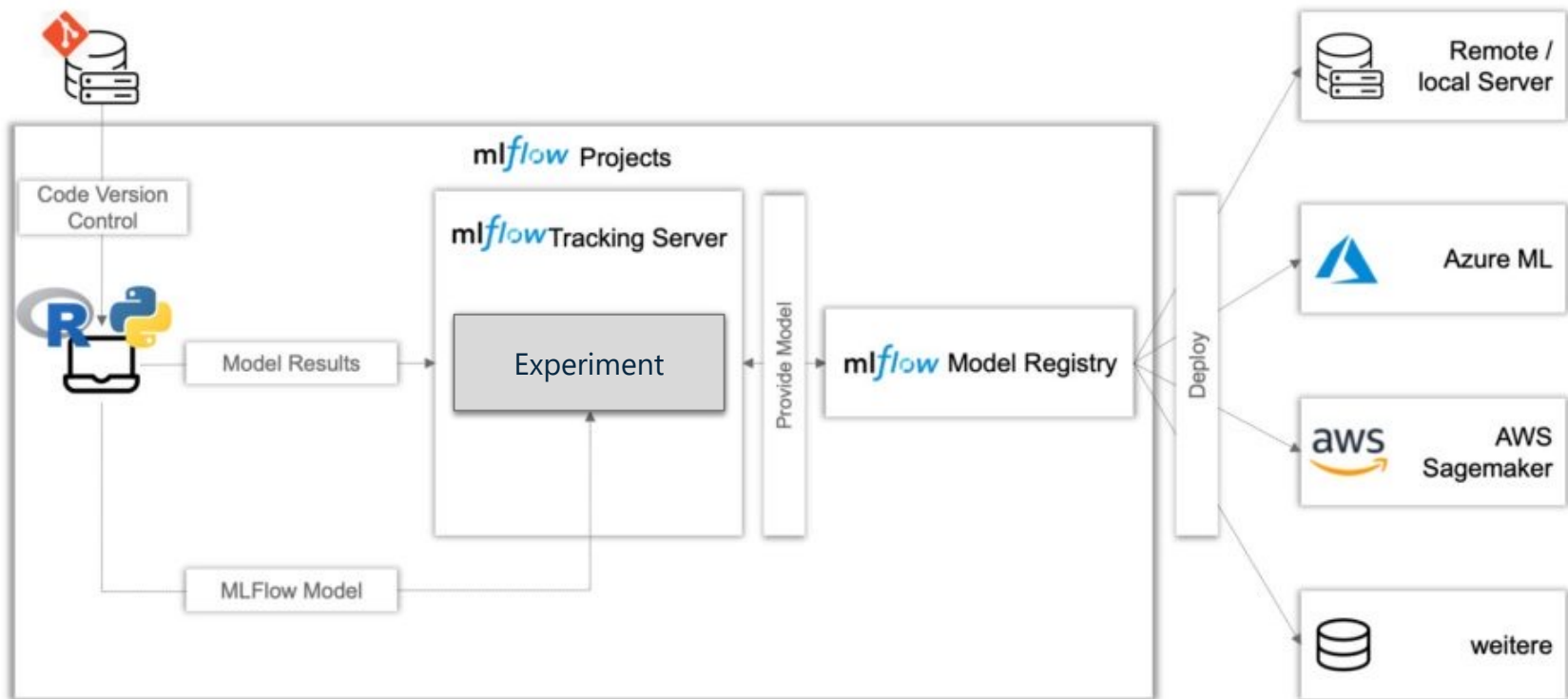
MLflow – Gesamtbild



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Deployment



https://www.statworx.com/wp-content/uploads/IC-24_MLflow_Folie3-1024x576.jpeg

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

MLflow – take home



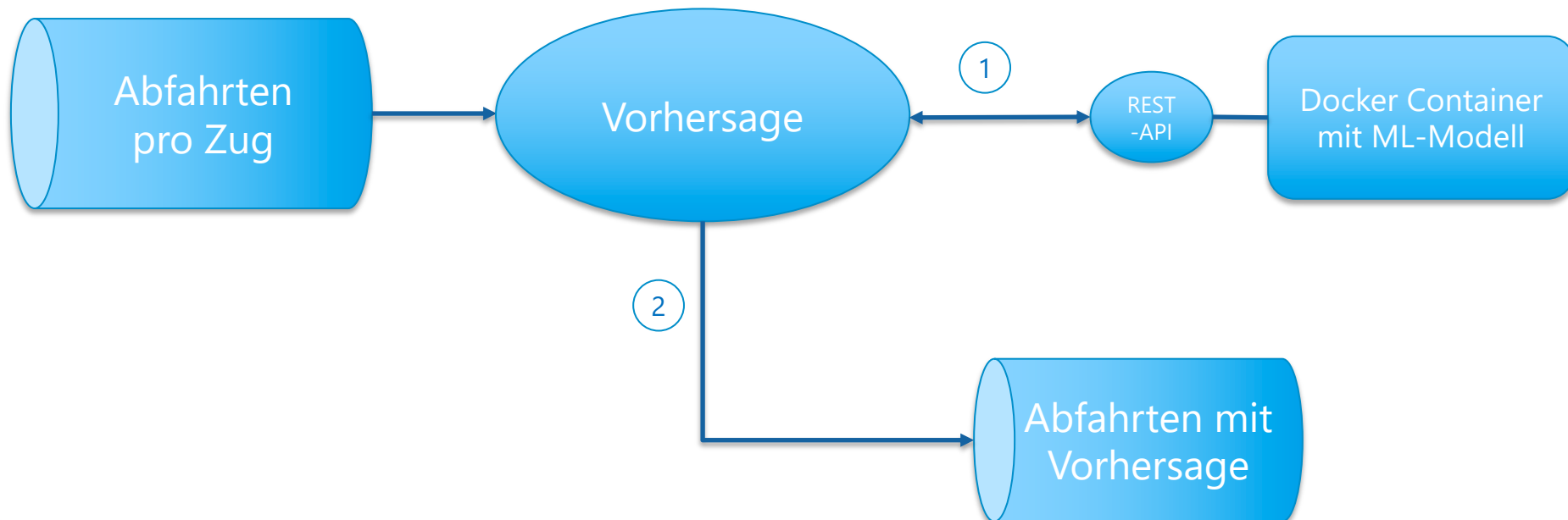
- ✿ MLflow ermöglicht eine nahtlose Entwicklung eines ML-Modells
 - ◆ ML Frameworks und Modellparameter werden durch Data Science definiert
 - ◆ Keine Re-Programmierung

- ✿ MLflow ermöglicht eine durchgängige Beschreibung eines ML-Modells
 - ◆ Keine Re-Programmierung
 - ◆ Definierte Laufzeitumgebung

- ✿ MLflow unterstützt einen CI/CD – Prozess für ML-Modelle
 - ◆ ML-Modelle als Software-Artefakte

Wie haben wir es gemacht?

Batch/Streaming Scoring – Vorhersagen aus Streaming-Daten



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Bemerkungen zu CI/CD

* MLflow Projects integrieren sich nahtlos in CI/CD-Tools

- ◆ Git-basiert
- ◆ API, um Modelle zu testen
- ◆ API, um Modelle nach Model Registry zu laden
- ◆ API, um Modelle zu in ‚stages‘ zu verscheiden
- ◆ API, um Modelle zu laden und deployen

* Strategien für Test und Auslieferung sind projekt-/unternehmensspezifisch

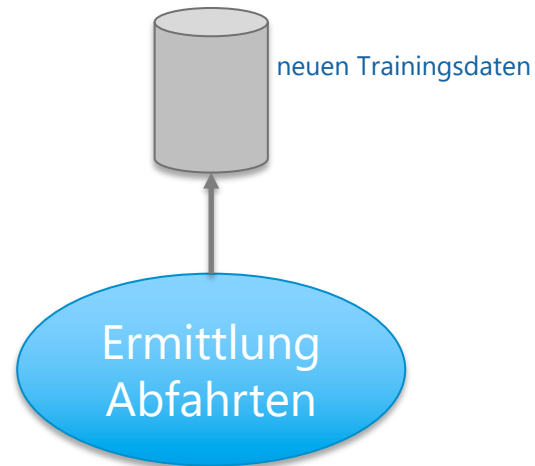
Agenda

- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML Modelle zur Laufzeit - MLflow
- **Monitoring**
- Gesamtarchitektur
- Take Home

Wie haben wir es gemacht - Monitoring



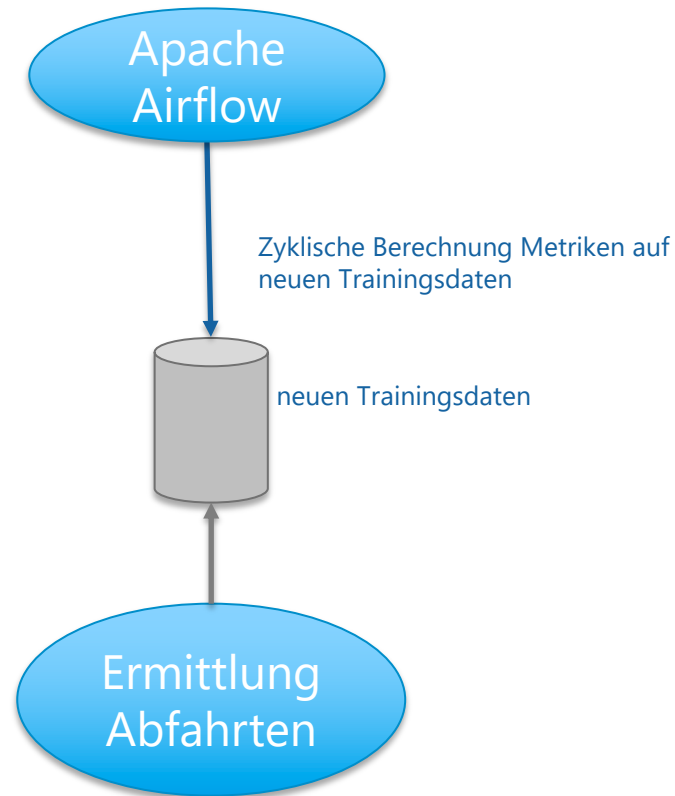
ML auf die Schiene gebracht



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | **Monitoring** | Gesamtarchitektur | MLOps in der Cloud | Take Home

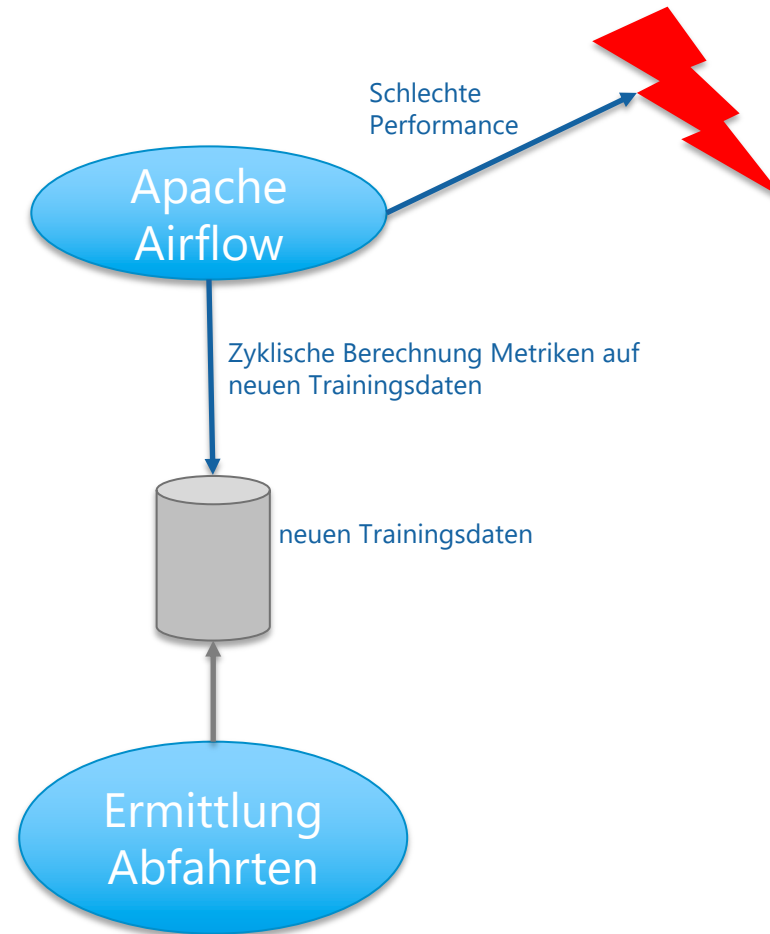
Monitoring



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | **Monitoring** | Gesamtarchitektur | MLOps in der Cloud | Take Home

Monitoring



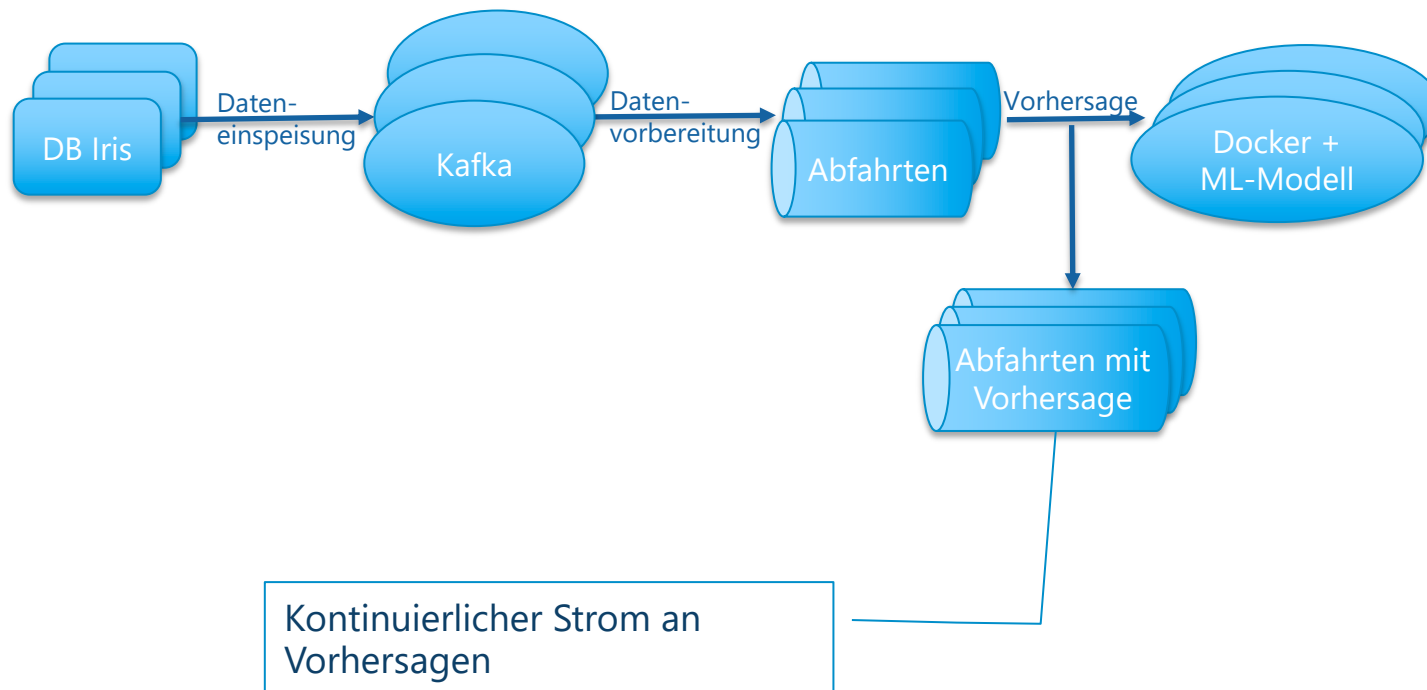
ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | **Monitoring** | Gesamtarchitektur | MLOps in der Cloud | Take Home

Agenda

- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML Modelle zur Laufzeit – MLflow
- Monitoring
- **Gesamtarchitektur**
- Take Home

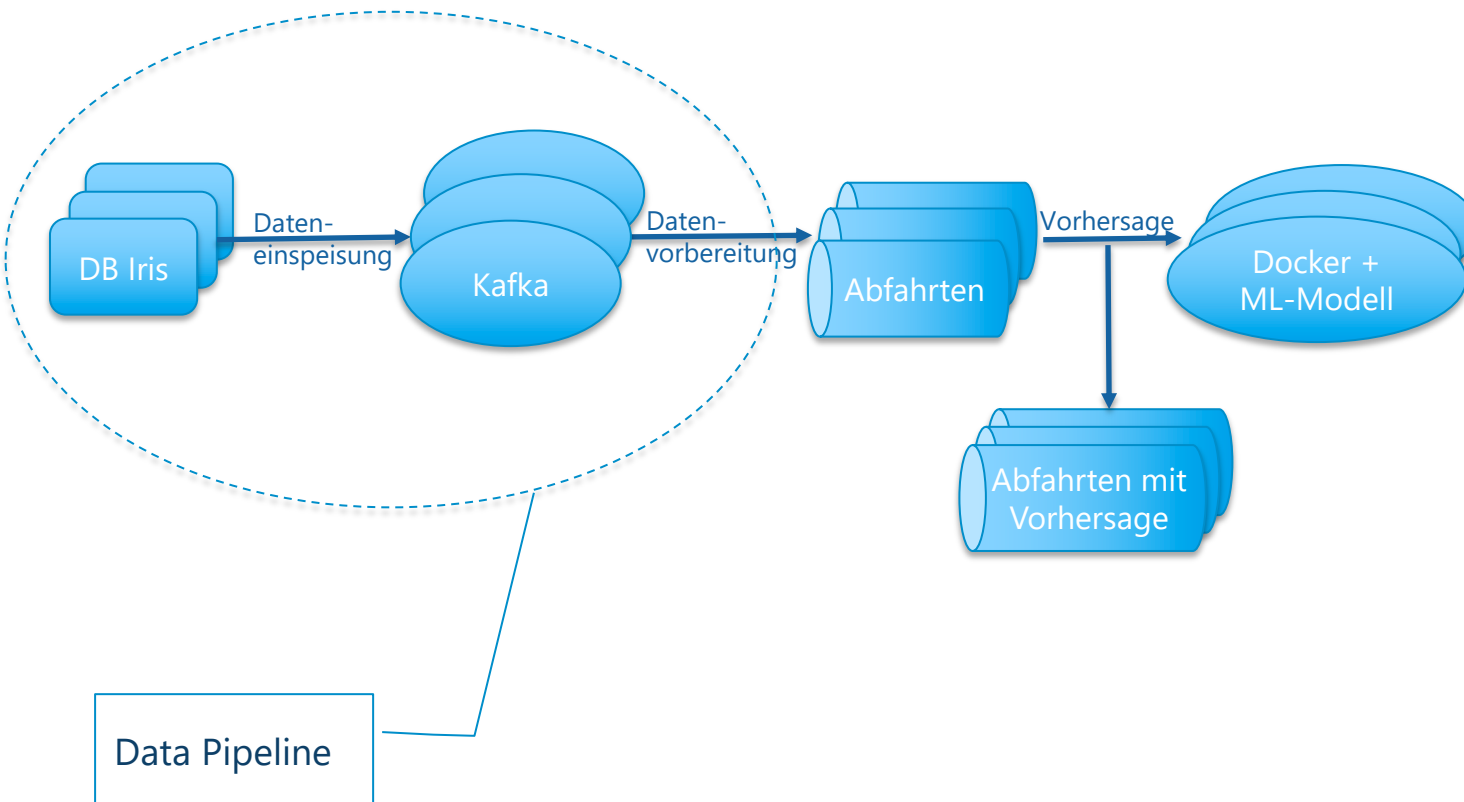
Gesamtüberblick



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | **Gesamtarchitektur** | MLOps in der Cloud | Take Home

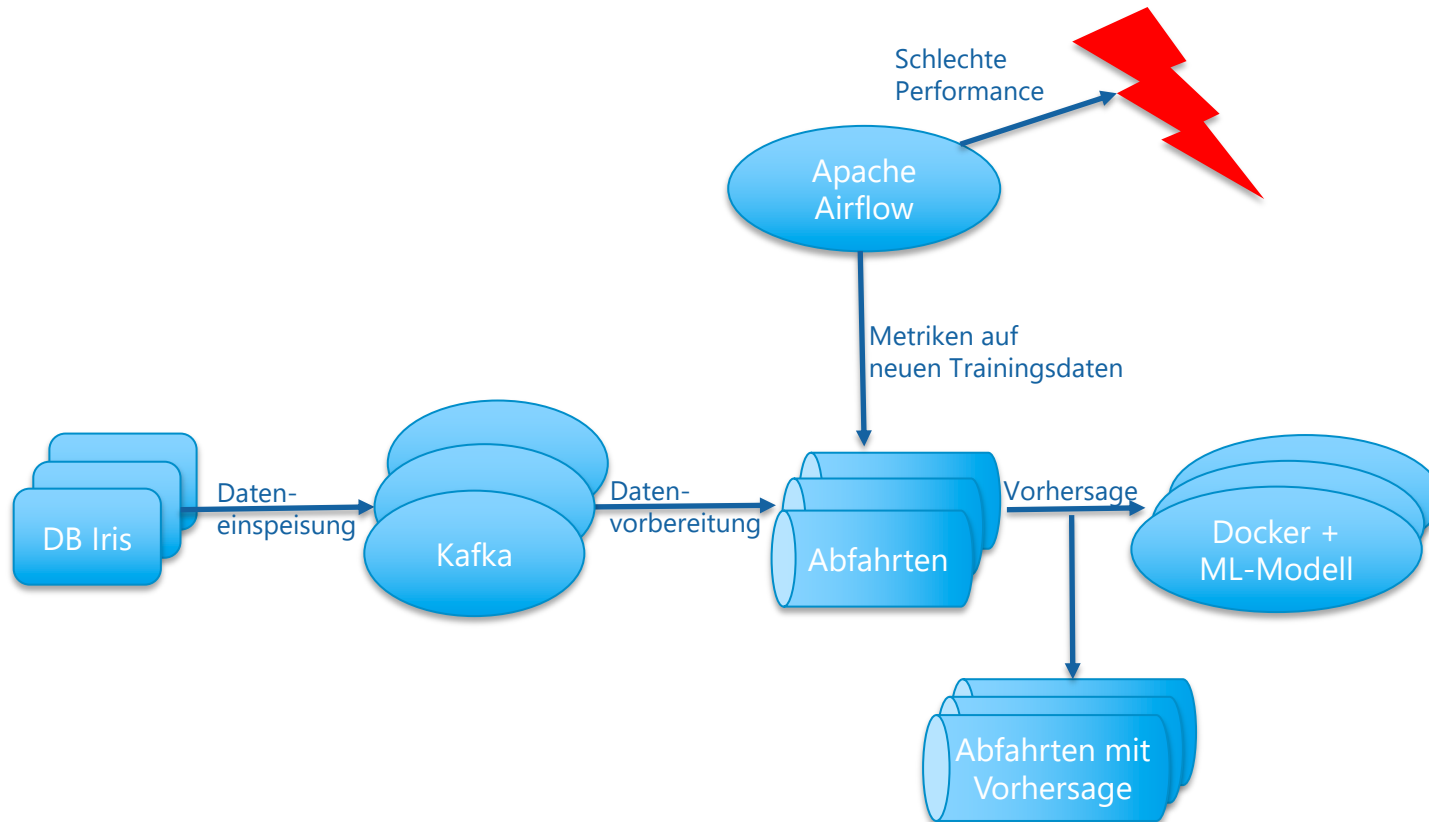
Gesamtüberblick



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | **Gesamtarchitektur** | MLOps in der Cloud | Take Home

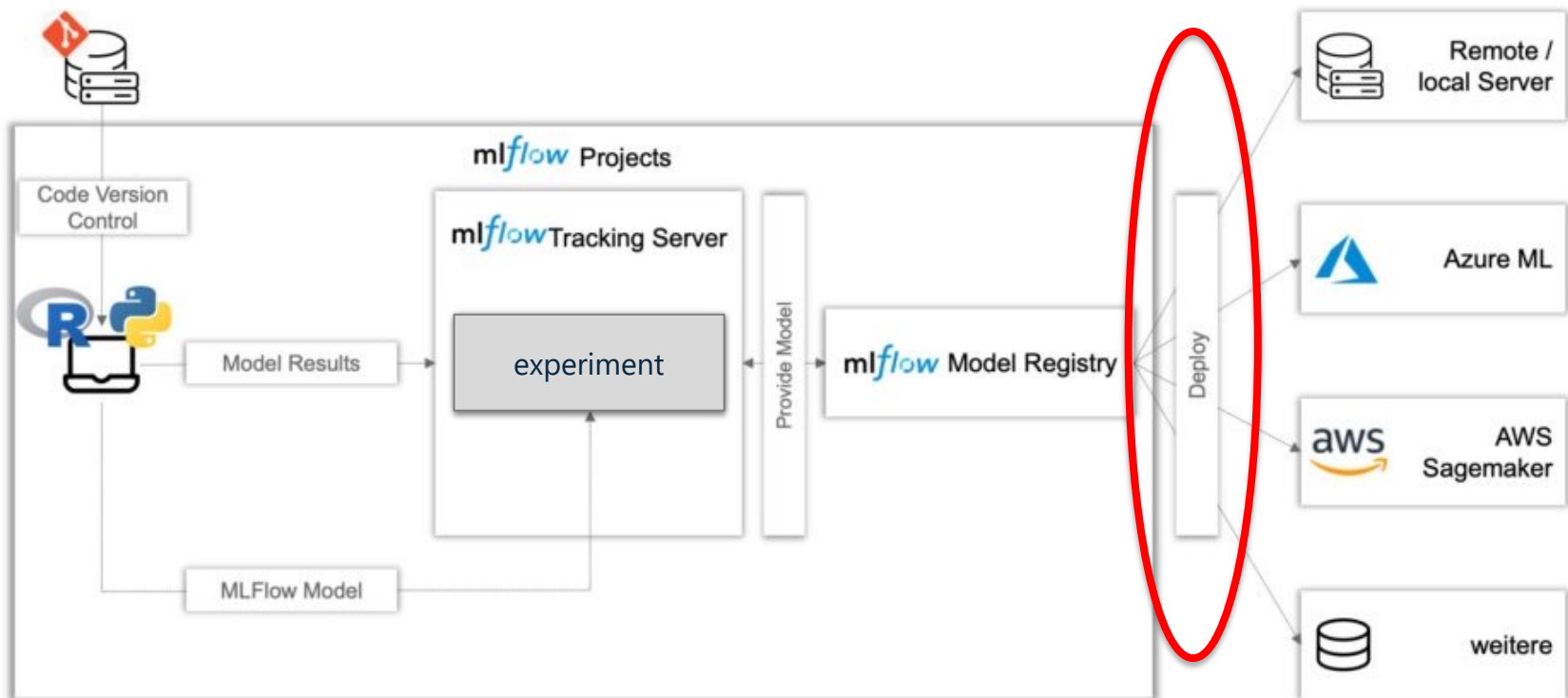
Gesamtüberblick



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | **Gesamtarchitektur** | MLOps in der Cloud | Take Home

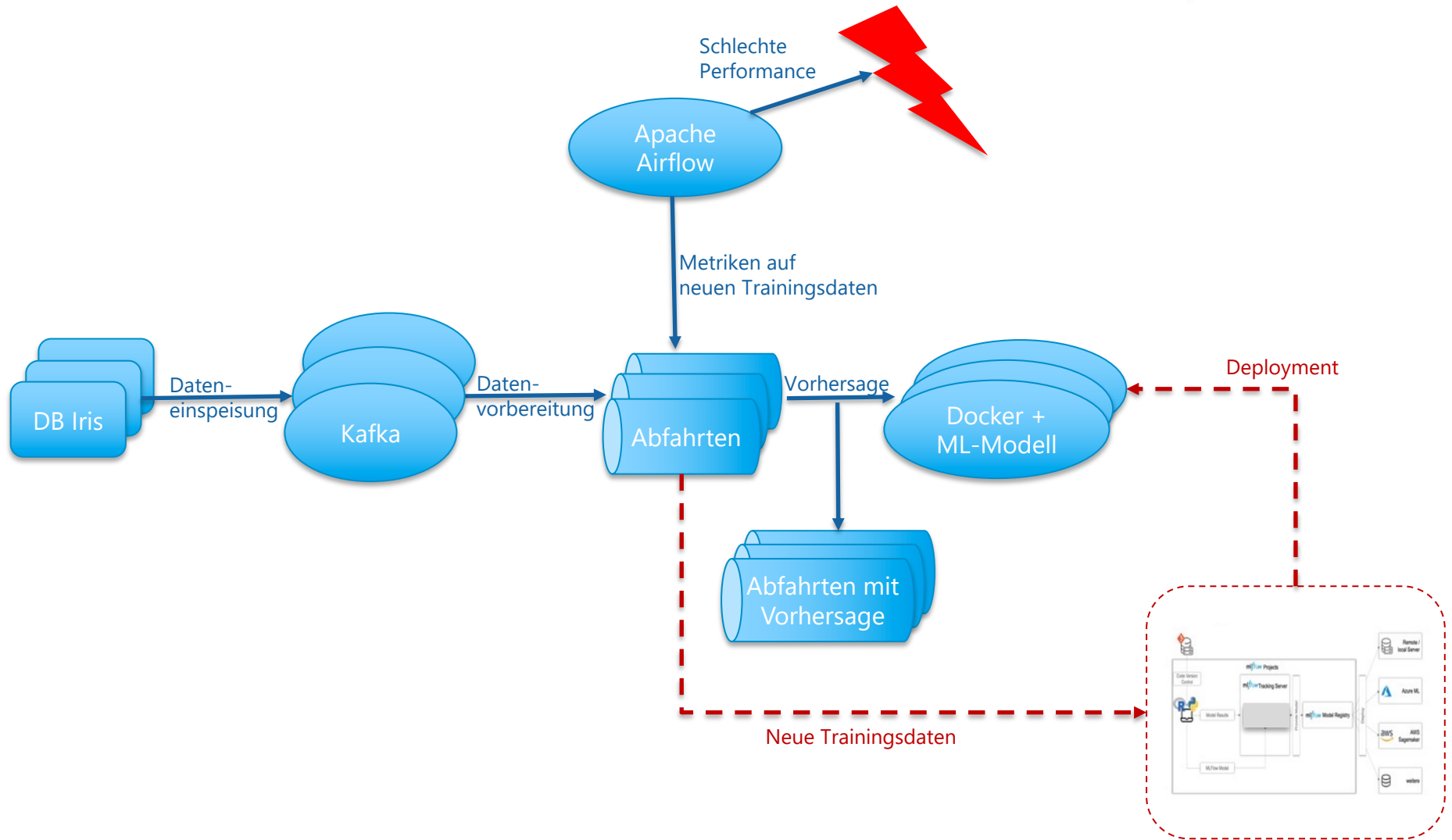
Deployment



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | **ML Modelle zur Laufzeit - MLflow** | Monitoring | Gesamtarchitektur | MLOps in der Cloud | Take Home

Gesamtüberblick



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | **Gesamtarchitektur** | MLOps in der Cloud | Take Home

Periodische Re-Training

Agenda

- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML-Modelle zur Laufzeit – MLflow
- Monitoring
- Gesamtarchitektur
- **MLOps in der Cloud**
- Take Home

Data Pipeline in der Cloud – Wie macht es AWS ?

- * AWS (Amazon Web Services) bietet unterschiedliche Streamingplattformen an
 - ◆ Apache Kafka
 - ◆ AWS Kinesis
- * Der Aufbau der *Data Pipeline* ist konzeptionell gleich

MLOps in der Cloud – Wie macht es AWS?

* AWS (Amazon Web Services) bietet die MLOps-Plattform *SageMaker* an

- ◆ ... *managed machine learning Service*
- ◆ ... unterstützt den vollständigen MLOps-Prozess

* SageMaker bietet

- ◆ SageMaker Studio/Notebook zur Exploration
- ◆ .. viele bereitgestellte Algorithmen, Notebooks, Lösungsschablonen
- ◆ ... Konzepte, um Modelle zu speichern und transferieren/deployen
- ◆ ... unterschiedliche Laufzeitmodelle

* MLflow lässt Modelle nach SageMaker deployen

- ◆ ... unterstützt das SageMaker-Format

Agenda

- Mit welchen Fragen beschäftigen wir uns?
- Runtime Data Pipeline
- ML-Modelle zur Laufzeit – MLflow
- Monitoring
- Gesamtarchitektur
- **Take Home**

Take Home

- * Wir haben MLOps mit Hilfe von Kafka und MLflow umgesetzt
 - ◆ ... der fehlenden MLOps-Prozess hemmte eine geschmeidige Entwicklung
- * MLflow hat sich bewährt
 - ◆ ... die Phasenübergänge sind nahtlos
- * Was fehlt noch?
 - ◆ Umsetzung des Monitoring mit Apache Airflow

Take Home

✳ Wir wollen die Umsetzung des Bahntrackers in einem WS vorstellen

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | **Take Home**

Backup

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | **Take Home**

Packaging format for ML Models

- Any directory with MLmodel file

Defines dependencies for reproducibility

- Conda environment can be specified in MLmodel configuration

Model creation and loading utilities

- `mlflow.<model_flavor>.save_model(...)` or `log_model(...)`
- `mlflow.<model_flavor>.load_model(...)`

Deployment APIs

- CLI / Python / R / Java
- `mlflow models [OPTIONS] COMMAND [ARGS]...`
 - `mlflow models serve [OPTIONS] [ARGS]`
 - `mlflow models predict [OPTIONS] [ARGS] ...`

MLflow Backend Registry Stores

MLflow Backend Registry Stores

Entity (Metadata) Store and Models

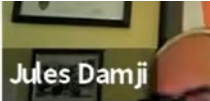
- SQLStore (via SQLAlchemy)
 - PostgreSQL, MySQL, SQLite
 - Default is `mlruns.db` file locally
- Set programmatically for locally
- `mlflow.set_tracking_uri("sqlite:///mlruns.db")`
- `sqlite3 ./mlruns.db` (on local host)
- Managed MLflow on Databricks
 - MySQL on AWS and Azure

Artifact Store

- Local Filesystem
 - `mlruns` directory
- S3 backed store
- Azure Blob storage
- Google Cloud Storage
- DBFS artifact repo
- `mlflow server --backend-store-uri /mnt/my-persistent-disk --default-artifact-root s3://my-mlflow-bucket/ --host 0.0.0.0`

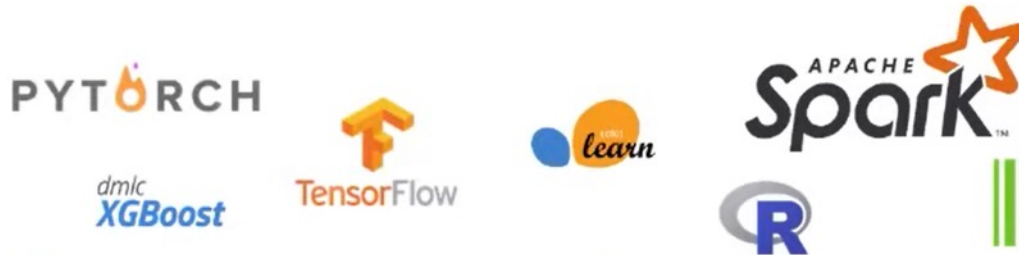
MLflow – ML Projects

mlflow™



MLflow Projects Motivation

Diverse set of tools



Diverse set of environments



mlflow™
Projects

Package data science
code in a format that
enables reproducible runs
on any platform

ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | **Take Home**

MLflow – ML Projects



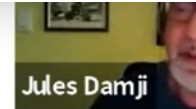
- * Tools unterscheiden sich in der Verarbeitung des Test-/Trainingsdaten
- * Ergebnisse hängen von Environment ab
 - ◆ Versionen der Tools/Libraries

MLflow

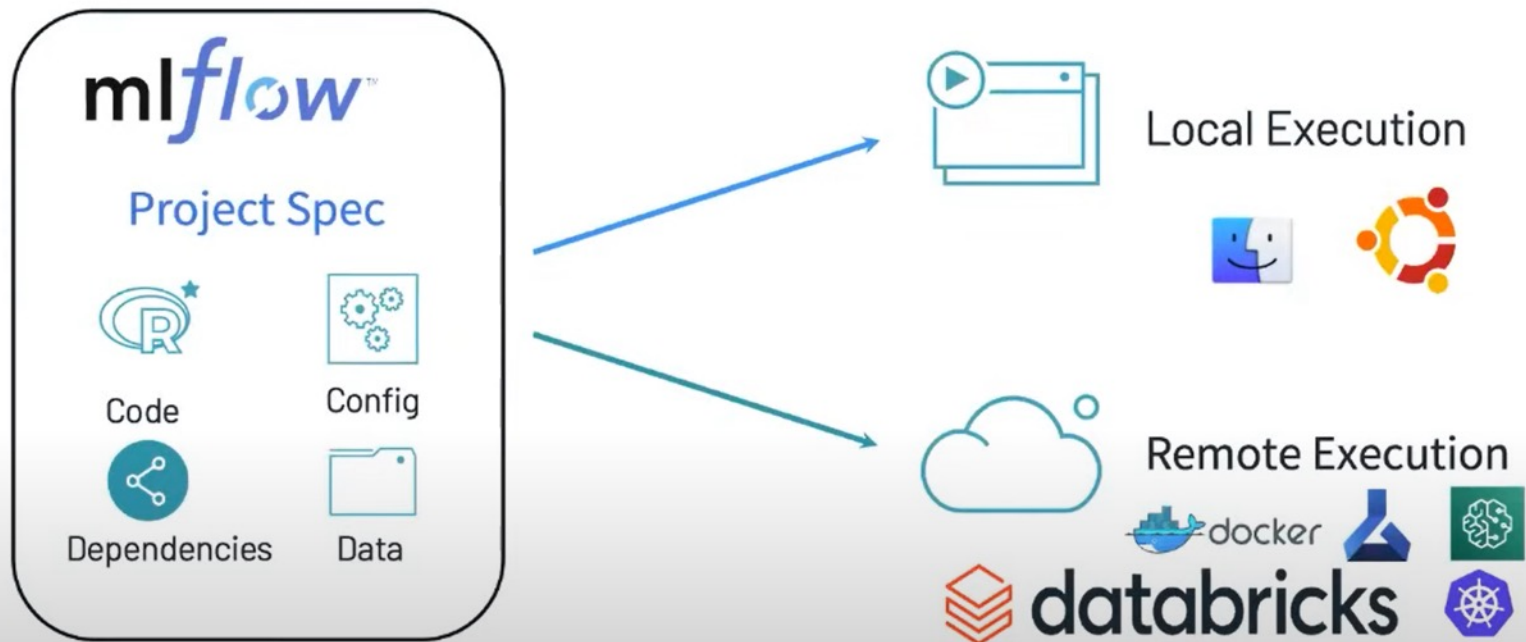


- * ML Projects ist ein Sub-Project von MLflow
- * ML Projects ist eine Format, um für Data Science Code
- * ML Projects ist *self-contained*
 - ◆ Bündelt alle Informationen, um das Tools und Environment wiederherzustellen
- * ML Project definiert, wie das ML-Modell paketiirt wird
 - ◆ Conda, Docker, R
- * ML Projects kann alle Abhängigkeiten im Ziel-Umgebung installieren
 - ◆ Sichert exakt gleiche Laufzeitumgebung zu
- * ML Projects ist i.d.R. ein Git-Repository
 - ◆ Integriert sich in CI/CD-Tools
- * MLflow übernimmt die Ausführung/Build eines ML Projects
 - ◆ Ein ML Project kann aus mehreren Experimenten bestehen
- * ML Projects ist Voraussetzung für
 - ◆ Vergleichbarkeit von ML-Modellen
 - ◆ Die Auslieferung/Deployment von ML-Modellen





MLflow Projects



ML auf die Schiene gebracht

Mit welchen Fragen beschäftigen wir uns? | Runtime Data Pipeline | ML Modelle zur Laufzeit - MLflow | Monitoring | Gesamtarchitektur | MLOps in der Cloud | **Take Home**