

Ein Modell ist trainiert – und jetzt?

Thementag „Ist mein Zug pünktlich?“

Autor: Philipp Steinweg / Christoph Schmidt-Casdorff

16.08.2022



Agenda

- Data-Science-Modelle als Softwaresysteme
- MLOps und Data Science
- MLOps und CI/CD
- MLOps und produktive Laufzeitumgebung
- Take Home

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Agenda

- Data-Science-Modelle als Softwaresysteme
- MLOps und Data Science
- MLOps und CI/CD
- MLOps und produktive Laufzeitumgebung
- Take Home

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

ML-Modelle unter Laborbedingungen

- ML-Modelle werden auf **begrenztem** und **statischem** Datenbestand trainiert
 - ◆ Insbesondere können keine Echtzeitdaten verarbeitet werden
- Training findet typischer Weise mit explorativen Werkzeugen statt
- ML-Modelle basieren auf Testdaten

Im Labor trainierte ML-Modelle sind ein erster Schritt ...

... aber wie wird aus ihnen lauffähige Software?

ML-Modelle als Softwaresysteme

- Softwaresysteme nutzen ML-Modelle und leiten Entscheidungen ab
 - ◆ ML-Modelle sind Teil eines Softwaresystem
 - ◆ Die Entscheidungen werden auf Basis von Realdaten getroffen

ML-Modelle müssen zu einem Teil eines Softwaresystems werden ...

... was bedeutet das?

ML-Modelle als Teil von Softwaresystemen

- **ML-Modelle müssen automatisiert ausgeführt werden**
 - ◆ Wie erreicht man ML-Modelle programmatisch?
 - ◆ ML-Modelle müssen Schnittstellen anbieten

ML-Modelle als Teil von Softwaresystemen

- ML-Modelle müssen automatisiert ausgeführt werden
- ML-Modelle müssen in Produktion überführt werden
 - ◆ Wie integrieren sich ML-Modelle in Deployment und CI/CD-Pipelines?

ML-Modelle als Teil von Softwaresystemen

- ML-Modelle müssen automatisiert ausgeführt werden
- ML-Modelle müssen in Produktion überführt werden
- ML-Modelle müssen versioniert werden
 - ◆ Was gehört zu einem ML-Modell?
 - ◆ Was wird versioniert?

ML-Modelle als Teil von Softwaresystemen

- ML-Modelle müssen automatisiert ausgeführt werden
- ML-Modelle müssen in Produktion überführt werden
- ML-Modelle müssen versioniert werden
- ML-Modelle müssen systematisch getestet und verglichen werden
 - ◆ Wie stelle ich Vergleichbarkeit von ML-Modellen her?
 - ◆ Wie bewerte ich die Qualität von ML-Modellen?

Was unterscheidet ML-Modelle von Softwaresystemen

- **Die Qualität von ML-Modellen ändert sich im Laufe der Zeit**
 - ◆ ML-Modelle sind nur so gut wie die Realdaten
 - ◆ Realdaten und Trainingsdaten laufen i.d.R. auseinander
 - ◆ Bleibt mein ML-Modell über die Zeit so gut wie zum Trainingszeitpunkt?
 - Kann es sogar besser werden?
 - ◆ ML-Modelle können falsch sein/werden, ohne dass ein Fehler geworfen wird !!!!!

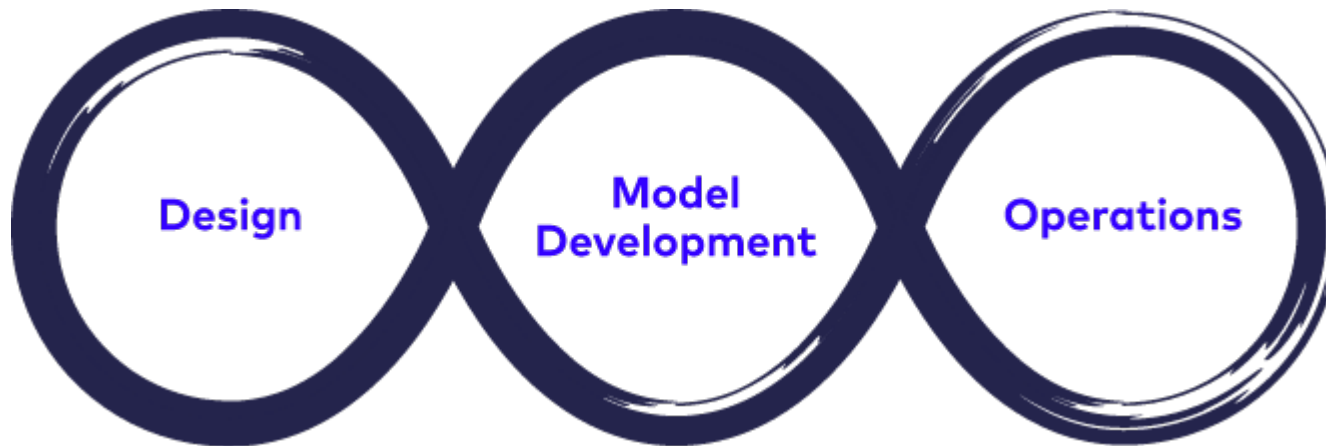
Was unterscheidet ML-Modelle von Softwaresystemen

- Die Qualität von ML-Modellen ändert sich im Laufe der Zeit
- ML-Modelle entstehen im Labor
 - ◆ Wie kann ich ML-Modelle unterschiedlicher Entwickler*innen vergleichen?

Was unterscheidet ML-Modelle von Softwaresystemen

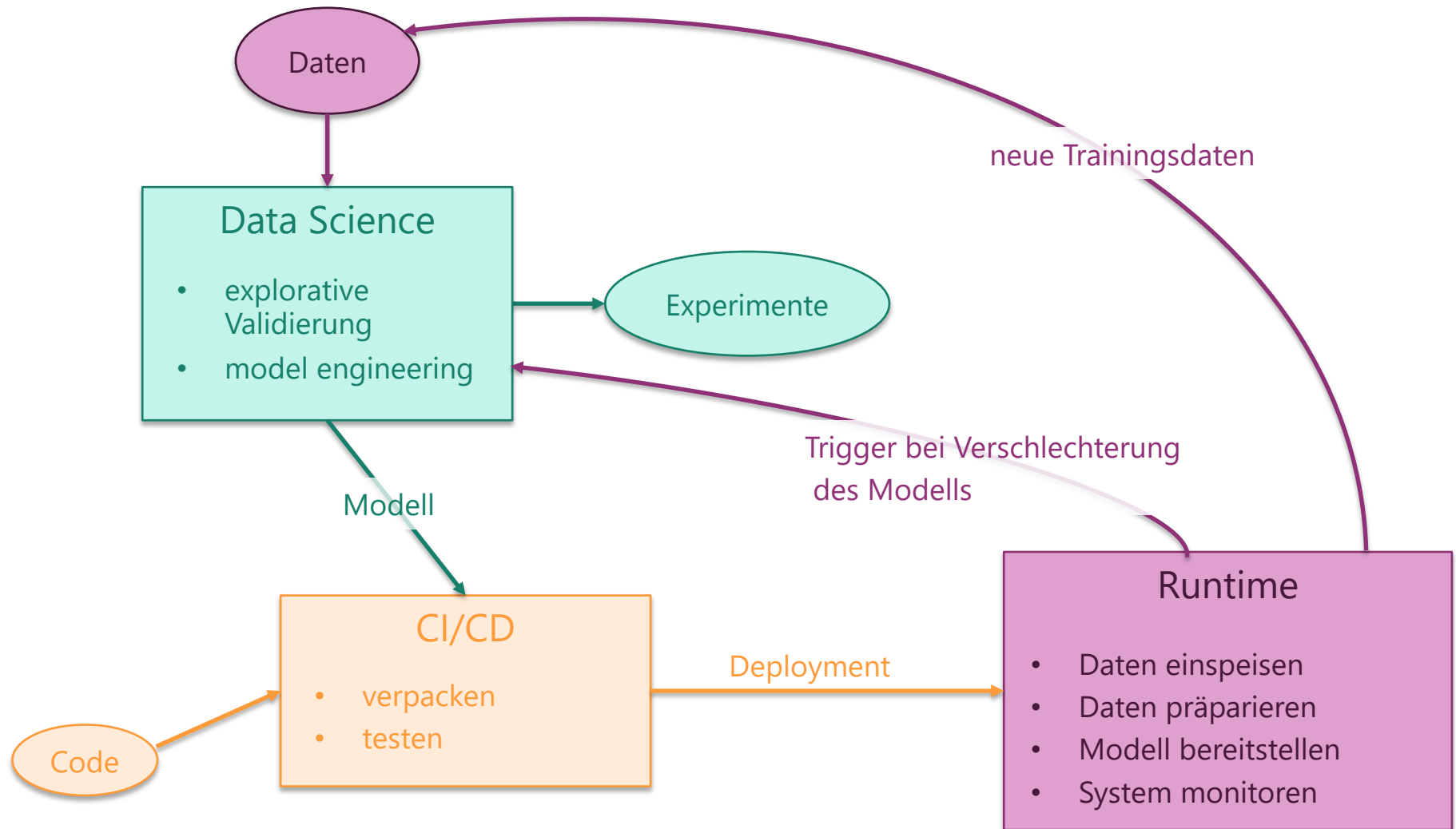
- Die Qualität von ML-Modellen ändert sich im Laufe der Zeit!!!
- ML-Modelle entstehen im Labor
- Wie bringe ich ein ML-Modell in Produktion
 - ◆ Wie bietet ein ML-Modell Schnittstellen an?
 - ◆ Wie verpacke ich ein ML-Modell?
 - ◆ Warum kann ich mein ML-Modell nicht einfach in Produktionsumgebung kopieren?
 - ◆ Wie automatisiere ich Schritte wie *Cleaning*, *Feature Extraction*, etc. ... ?

With Machine Learning Model Operationalization Management (MLOps), we want to provide an end-to-end machine learning development process to design, build and manage reproducible, testable, and evolvable ML-powered software.



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

Überblick über den gesamten MLOps-Prozess



Ein Modell ist trainiert – und jetzt?

Agenda

- Data-Science-Modelle als Softwaresysteme
- MLOps und Data Science
- MLOps und CI/CD
- MLOps und produktive Laufzeitumgebung
- Take Home

Ein Modell ist trainiert – und jetzt?

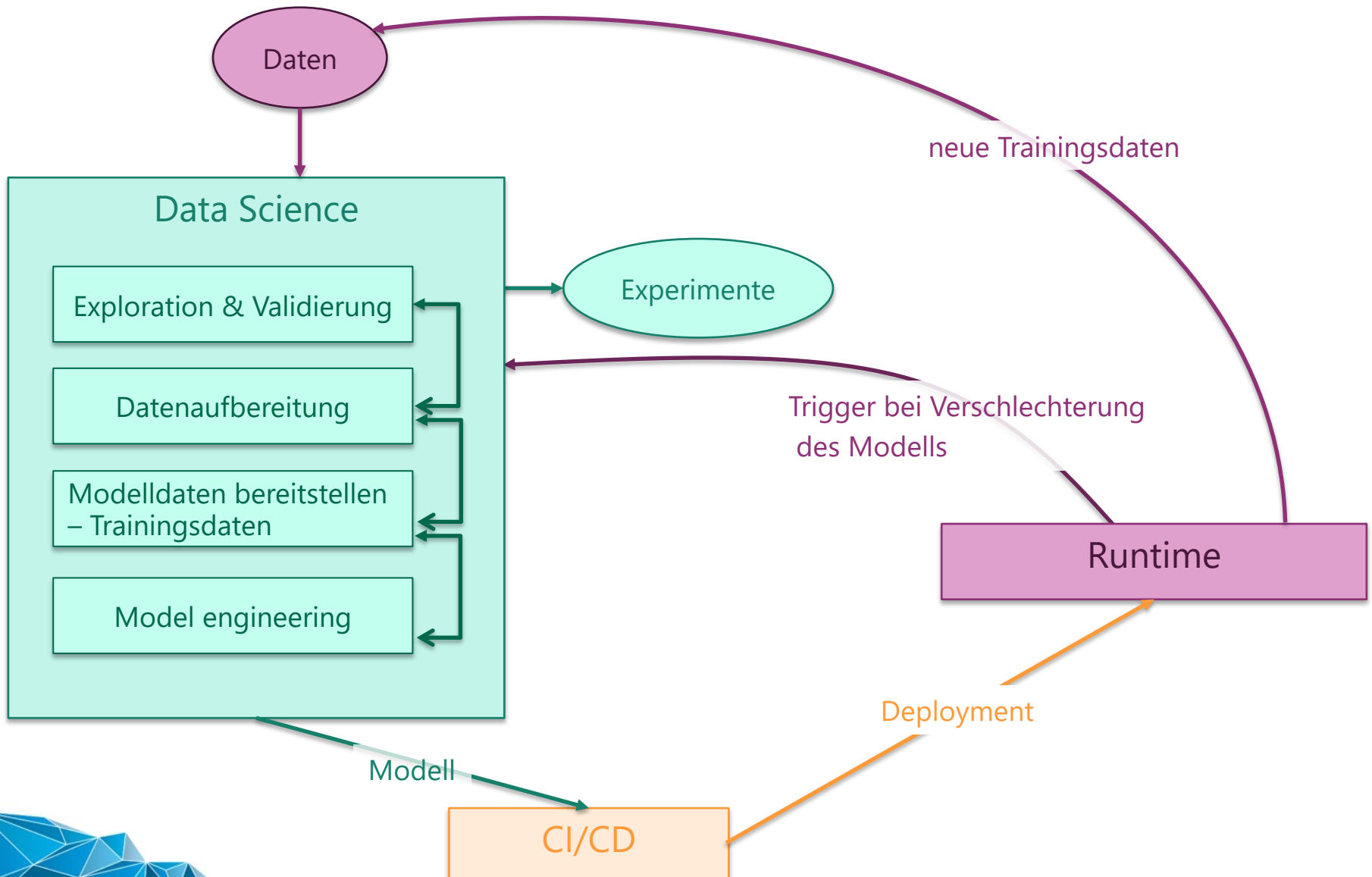
Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Data Science unter Laborbedingungen



Ein Modell ist trainiert – und jetzt?

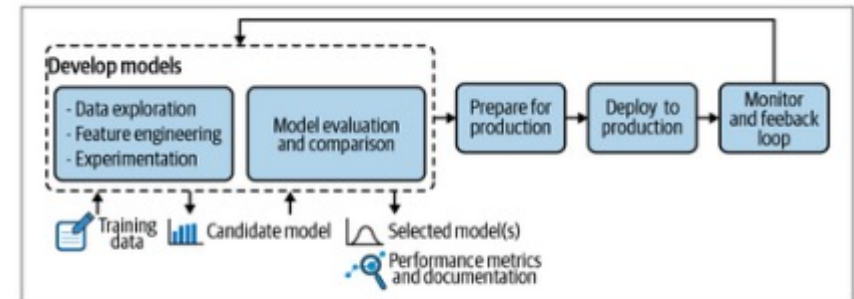
Tieferer Einstieg in den Data-Science-Teil



Modelle entwickeln

* Entwicklung durch Experimente, z.B.:

- ◆ Daten bereinigen
- ◆ Hinzufügen neuer Features
- ◆ Ausprobieren verschiedener Algorithmen
- ◆ Hyperparameter-Tuning
 - Wie gehe ich mit diesen Experimenten um?



Bausteine eines ML-Modells

* Änderung einer Achse

- ◆ Ggf. neues Modell
- ◆ Ggf. neues Deployment



Ein Modell ist trainiert – und jetzt?

Experimente

- ✧ Experiment = „Neuanordnung“ der Bausteine
 - ✧ Experimente sind idealerweise
 - ◆ transparent
 - ◆ nachvollziehbar
 - ◆ reproduzierbar
 - ◆ vergleichbar
- > Experimente müssen versioniert/getrackt werden
- > Experimente müssen zentral zur Verfügung gestellt werden

Transparenz von Experimenten

* Durchgeführte Experimente sollten an zentraler Stelle einsehbar sein

- ◆ für andere Data Scientisten
- ◆ für andere Entwickler
- ◆ für Business User
 - Data Governance
 - Process Governance
 - Risiko Management



Nachvollziehbarkeit von Experimenten

- * „Anordnung“ der Bausteine sollte von außen nachvollziehbar sein
- * Wünschenswert:
Nachvollziehbarkeit von Vorhersagen eines deployten Modells
 - ◆ [Explainable/Interpretable AI](#)



Reproduzierbarkeit von Experimenten

✱ Experimente sollen reproduzierbar sein

- ◆ um später etwa erneut trainieren zu können
- ◆ auch durch andere Teammitglieder (z.B. für Code Reviews)
 - „Bausteine“ müssen versioniert werden

Vergleichbarkeit von Experimenten

- ✱ Durchgeführte Experimente müssen miteinander vergleichbar sein
 - ◆ z.B. durch vorher definierte Metriken

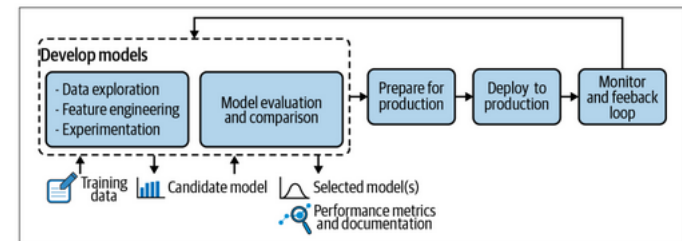


Deployment von Experimenten

Ein Modell ist trainiert – und jetzt?

Deployment von Experimenten

- ❖ Wann ist ein Experiment „produktionsreif“?
- ❖ Wie überwache ich in Produktion deployte Modelle?



Ein Modell ist trainiert – und jetzt?

Modelle in Produktion

- * ML-Modelle trainieren mit Daten -> Realität/Daten ändert sich aber stetig
- * ML-Modelle „veralten“ mit der Zeit -> *Model Drift*
 - ◆ Concept Drift
 - ◆ Data Drift
- * MLOps muss Drifts erkennen & idealerweise Neutraining des Modells anstoßen

Concept Drift

✿ Zusammenhang Features & Zielvariable ändert sich im zeitlichen Verlauf

✿ Beispiele

- ◆ Es ändert sich die Nutzung des Nahverkehrs
- ◆ Kunden wollen vermehrt Fahrräder mitnehmen

Data Drift

* Daten „in Produktion“ ändern sich im Vergleich zu Trainings-/Testdaten

* Beispiele:

- ◆ Es kommen neue Bahnlinien hinzu
- ◆ Der Fahrplan/Taktung ändert sich
- ◆ Kurzfristige Trends entstehen durch z.B. Baustellen oder 9-€-Ticket
- ◆ Die Schnittstelle der Bahn-API ändert sich und liefert weitere Daten

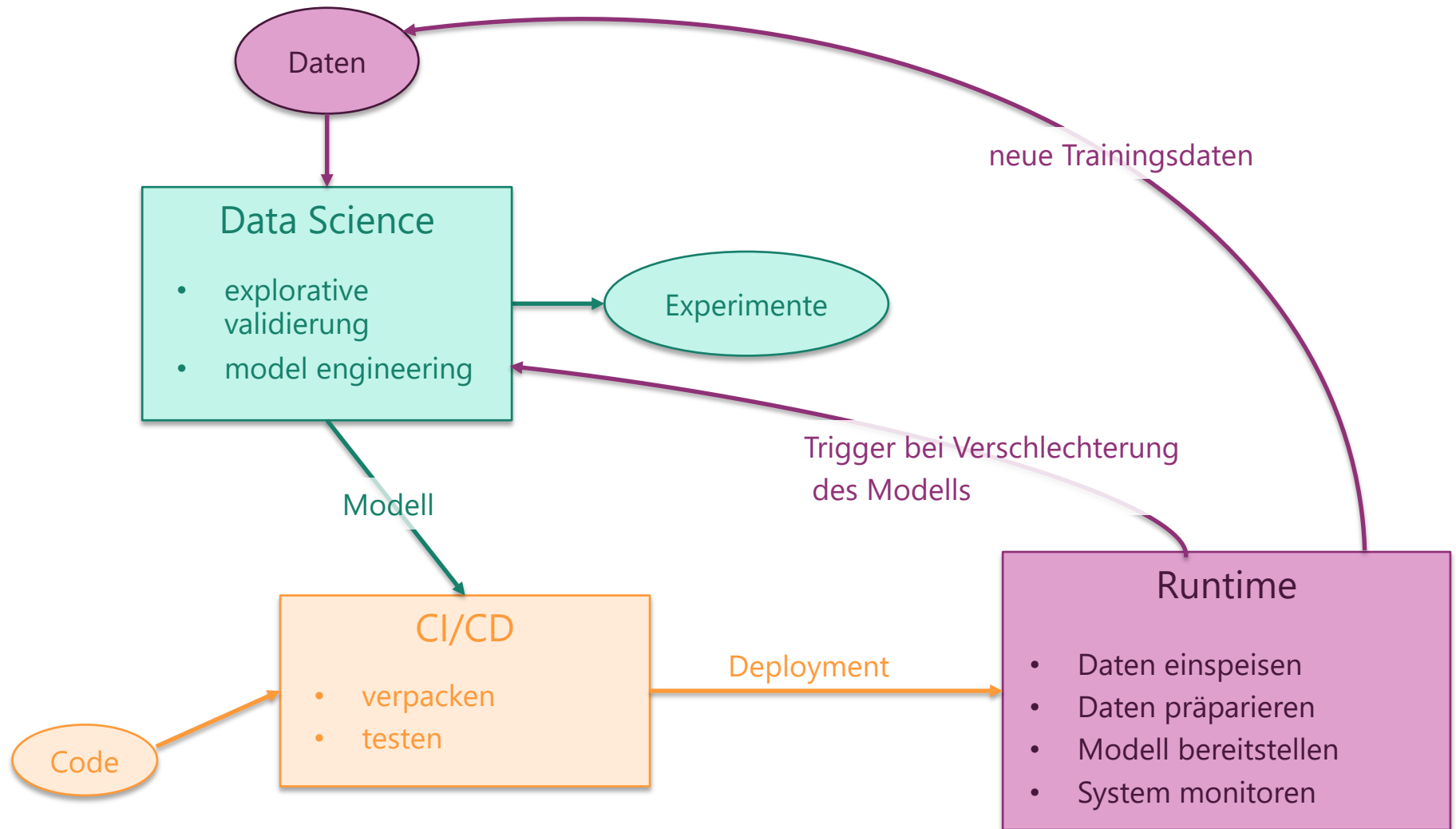
Agenda

- Data-Science-Modelle als Softwaresysteme
- MLOps und Data Science
- MLOps und CI/CD
- MLOps und produktive Laufzeitumgebung
- Take Home

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Tieferer Einstieg in den CI/CD-Teil



Ein Modell ist trainiert – und jetzt?

Continuous Integration / Continuous Delivery

✱ CI/CD ist ein kontinuierlicher Softwareentwicklungsprozess, der ...

- ◆ das Systeme (und seine Teile) automatisiert baut, testet und ausliefert
- ◆ dies stetig und wiederholbar tut
- ◆ auf Versionierungen der einzelnen Teile achtet
- ◆ auch CI/CD Pipeline genannt wird

✱ CI/CD -> *Continuous Integration / Continuous Delivery*

Continuous Integration / Continuous Delivery

- * CI/CD ist ein kontinuierlicher Softwareentwicklungsprozess
- * CI/CD beschleunigt
 - ◆ Die Zyklen zwischen Entwicklung und Auslieferung werden kürzer
 - ◆ Automatisierung entlastet Entwicklung

Continuous Integration / Continuous Delivery

- * CI/CD ist ein kontinuierlicher Softwareentwicklungsprozess
- * CI/CD beschleunigt die Auslieferung
- * CI/CD erhöht die Qualität des Systems
 - ◆ Auslieferung ist stabiler und weniger fehleranfällig
 - ◆ Automatisierte Tests sind integraler Bestandteil des Prozesses
 - ◆ Ausgelieferte Softwarestände sind nachvollziehbar und transparent

Was gehört zu einer Data-Science-Plattform?

* Zu einer produktiven Data-Science-Plattform gehören

- ◆ ML-Modelle
- ◆ Infrastruktur, um Modell auszuführen
- ◆ weitere unterstützende Software
 - Untersuchen wir später...

All das spielt zusammen

... und sollte automatisiert getestet, gebaut und ausgeliefert werden

CI/CD und Data-Science-Plattform

- ✳ Wer testet die einzelnen Komponenten?
 - ◆ auch das ML-Modell muss getestet werden !!!

CI/CD und Data-Science-Plattform

- * Wer testet die einzelnen Komponenten?
- * Wer baut/verpackt die einzelnen Komponenten?
 - ◆ *ready to deploy*
 - ◆ Versionierung auch für ML-Modelle

CI/CD und Data-Science-Plattform

- * Wer testet die einzelnen Komponenten?
- * Wer verpackt die einzelnen Komponenten?
- * Wer testet das Zusammenspiel der einzelnen Komponenten?

CI/CD und Data-Science-Plattform

- * Wer testet die einzelnen Komponenten?
- * Wer verpackt die einzelnen Komponenten?
- * Wer testet das Zusammenspiel der einzelnen Komponenten?
- * Wer *deployed* das gesamte System?

=> CI/CD-Pipeline

Continuous Integration / Continuous Delivery



CI/CD – ein Paketband für Software

Ein Modell ist trainiert – und jetzt?

Deployment von Modellen



Deployment bedeutet, dass ein ML-Modell (standardisiert) verpackt wird

Ein Modell ist trainiert – und jetzt?

Deployment von Modellen

- **Deployment meint, dass**
 - ◆ ... ein ML-Modell verpackt wird
 - ◆ ... und in unterschiedlichen Umgebungen reaktiviert/ausgeführt werden kann

Deployment von Modellen

- Deployment meint, dass ein ML-Modell verpackt werden kann
- ML Modell muss so „verpackt“ werden, dass
 - ◆ ... dem ML-Modell eine eindeutige und feste Version zugeordnet werden kann

Deployment von Modellen

- Deployment meint, dass ein ML-Modell verpackt werden kann
- ML Modell muss so „verpackt“ werden, dass
 - ◆ ... dem ML-Modell eine eindeutige und feste Version zugeordnet werden kann
 - ◆ ... jederzeit Test und Ausführung des Modells möglich ist
 - Metriken zur Prüfung der Performance müssen vorhanden sein
 - Trainings- und Testdaten müssen verfügbar sein
 - Software/ML Frameworks, in der das Modell geschrieben ist, ist verfügbar
 - Code zum Testen des Modells ist verfügbar
 - ◆ „Paket“ muss all dies umfassen

Deployment von Modellen

- Deployment meint, dass ein ML-Modell verpackt werden kann
- ML-Modell muss so „verpackt“ werden, dass
 - ◆ ... dem ML-Modell eine eindeutige und feste Version zugeordnet werden kann
 - ◆ ... jederzeit ein Test des Modells möglich ist
 - ◆ ... das ML-Modell in der Produktions-Infrastruktur in Betrieb genommen werden kann
 - ML-Modell muss eine Schnittstelle anbieten

Verpackungsmaterial

> Containerisierung

- ◆ Docker ist eine Format, welches alle Vorrausetzungen erfüllt
- ◆ Spezielles Konfiguration/Dockerfile für jedes ML Framework (+version)
- ◆ Es ist zusätzlicher Code notwendig, um ...
 - das ML-Modell zugänglich zu machen

Verpackungsmaterial

➤ Containerisierung

➤ Universelle Paketierungsformate

- ◆ ... wie in MLflow/DVC
- ◆ Paketierung ist unabhängig vom ML Framework
- ◆ Paketierung umfasst alles, was für Ausführung und Test notwendig ist
- ◆ Universelle Paketierung ermöglicht
 - ... die Vergleichbarkeit von Modellen
 - ... den Austausch und Installation von Modelle *,as it is'*

Verpackungsmaterial

- Containerisierung
- Universelle Paketierungsformate
- Sehr gut: Kombination von Docker und universellem Paketierungsformat
 - ◆ alle Vorteile von Docker
 - ◆ gleichförmiger Build- und Deploymentprozess

Ausführung von Modellen

- **Toolkette unterscheidet sich zwischen Labor und Produktion**
 - ◆ Häufig : Re-Implementierung der ML Modelle
 - Apache Spark, Apache Kafka, Apache Flink ,
 - ◆ Dies macht den Prozess träge und fehleranfällig
 - ◆ Durchgängigkeit vom Labor bis in die Produktion fehlt

... möglichst vermeiden

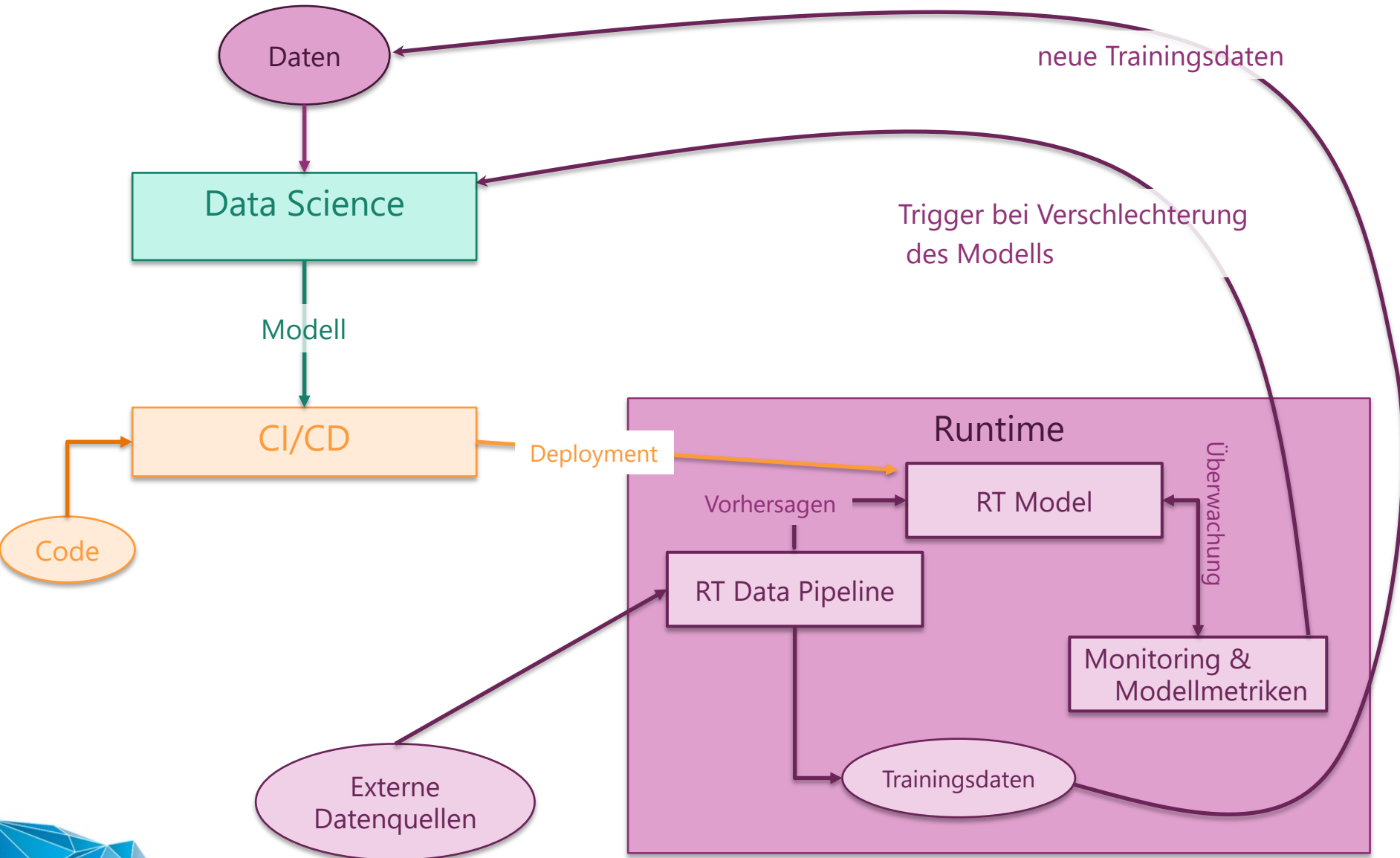
Agenda

- Data Science Modelle als Softwaresysteme
- MLOps und Data Science
- MLOps und CI/CD
- MLOps und produktive Laufzeitumgebung
- Take Home

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Tieferer Einstieg in den Runtime-Teil



Welche Aufgaben erfüllt die Laufzeitumgebung?

- ✱ Daten müssen aus unterschiedlichen Quelle eingesammelt werden
 - ◆ Daten müssen an das Eingabe-Format des ML-Modells angepasst werden
 - ◆ i.d.R. werden Daten nach Vorgaben der Data Science aufbereitet

- ✱ Es müssen stetig neue Trainingsdaten gesammelt werden
 - ◆ Data Science benötigt Trainingsdaten, um zu re-trainieren

Welche Aufgaben erfüllt die Laufzeitumgebung?

- * Daten müssen aus unterschiedlichen Quelle eingesammelt werden
- * Es müssen stetig neue Trainingsdaten gesammelt werden
- * Das ML-Modell muss Vorhersagen treffen können
 - ◆ Es muss erreichbar sein z.B. als Webservice
 - ◆ Es muss eine definierte Schnittstelle für die Daten anbieten

Welche Aufgaben erfüllt die Laufzeitumgebung?

- * Daten müssen aus unterschiedlichen Quelle eingesammelt werden
- * Es müssen stetig neue Trainingsdaten gesammelt werden
- * Das ML-Modell muss erreichbar sein
- * Die Qualität des ML-Modells muss überwacht werden
 - ◆ Modell Degradation

Welche Aufgaben erfüllt die Laufzeitumgebung?

- * Daten müssen aus unterschiedlichen Quelle eingesammelt werden
- * Es müssen stetig neue Trainingsdaten gesammelt werden

Runtime Data Pipeline

- * Das ML-Modell muss erreichbar sein
- * Die Qualität des ML-Modells muss überwacht werden
 - ◆ Modell Degradation

Data Pipeline



RT Data Pipeline gibt den Takt in einem Data Science Systems vor

Ein Modell ist trainiert – und jetzt?

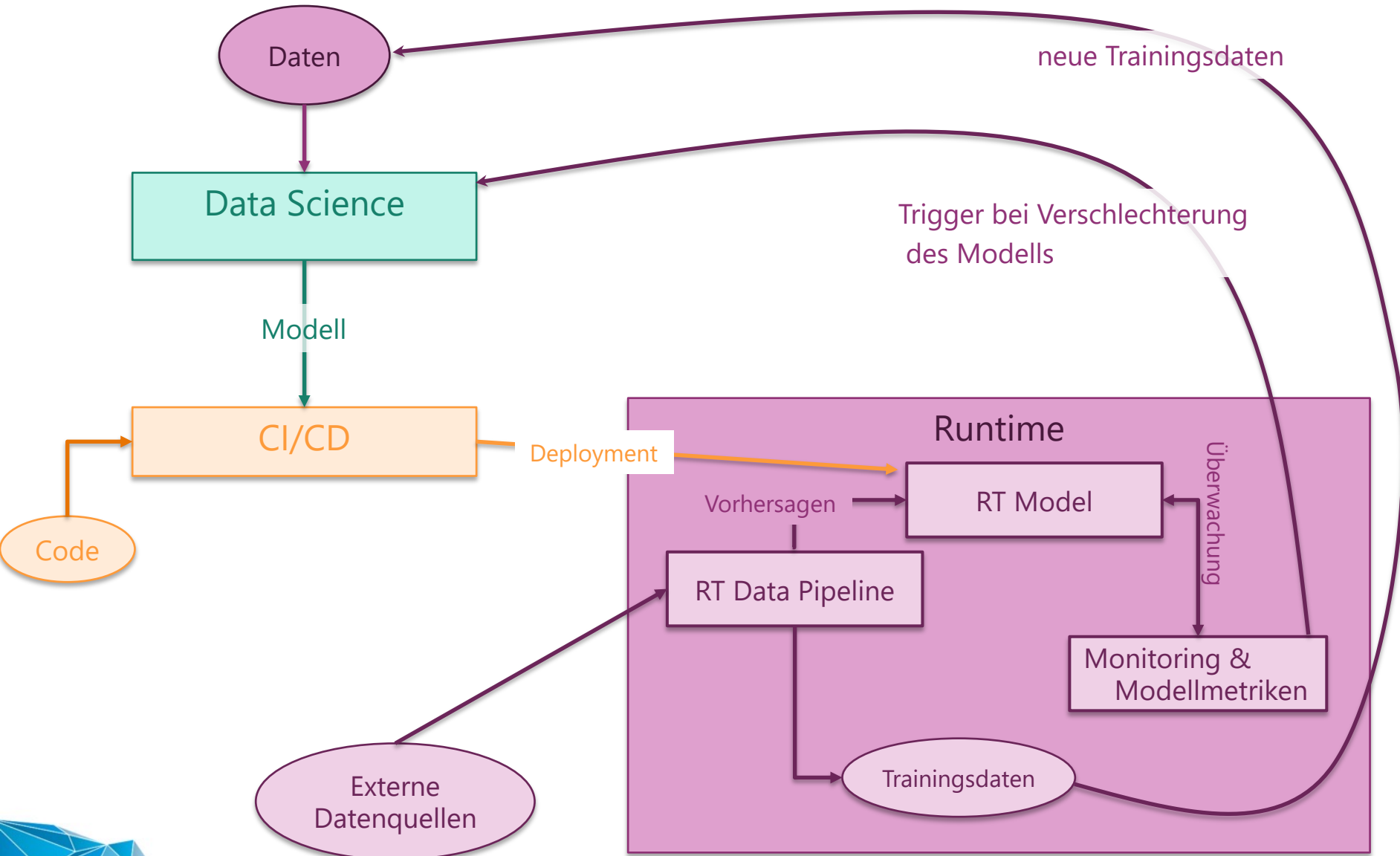
RT Model – Mittelpunkt des Data Science Systems



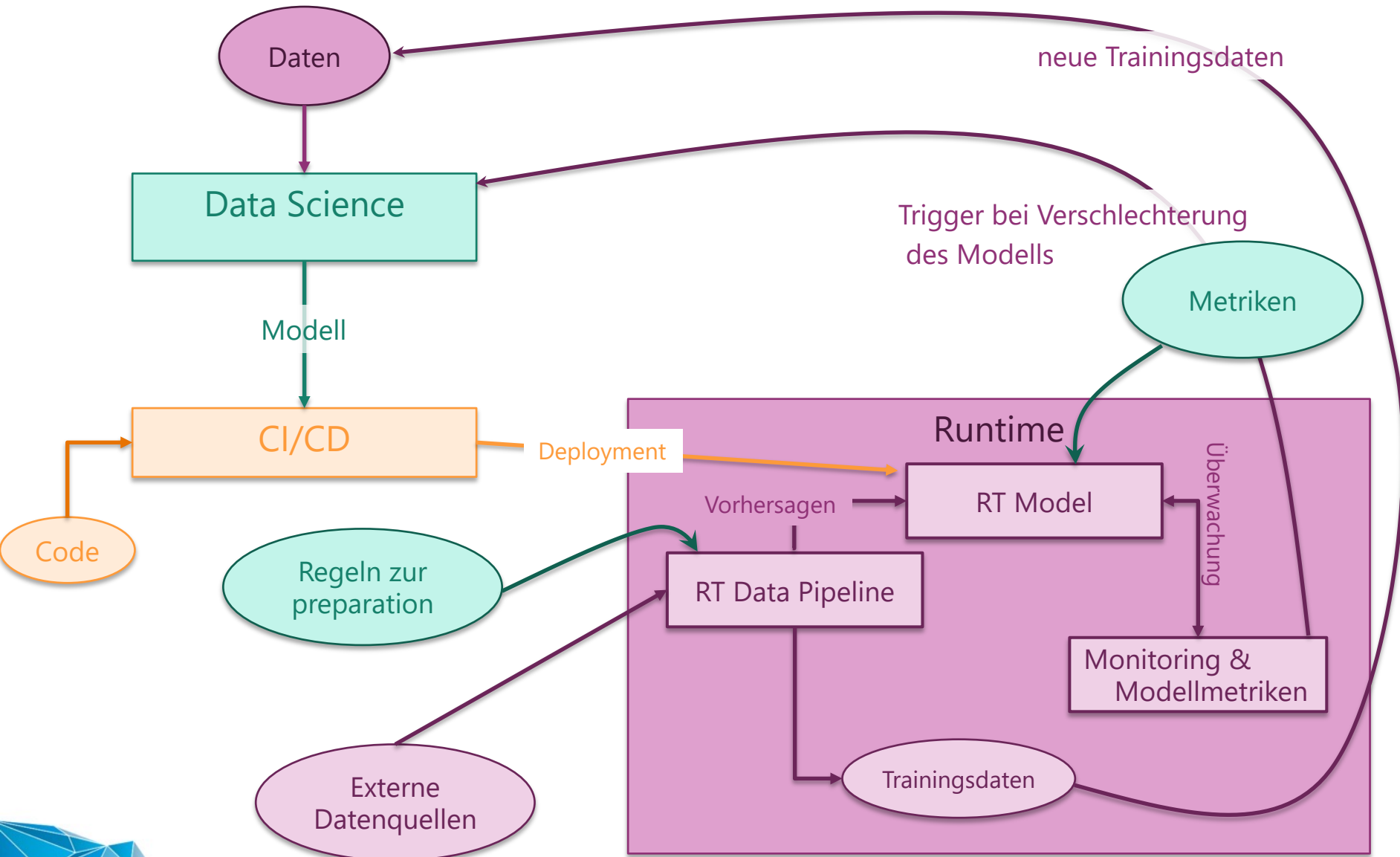
... aber im Mittelpunkt steht das ML Modell

Ein Modell ist trainiert – und jetzt?

Tieferer Einstieg in den Runtime-Teil



Tieferer Einstieg in den Runtime-Teil



Vorbereitung der Data Pipeline

- **Adaption der vorbereitenden Schritte der Data Pipeline**
 - ◆ Weg der Daten von der Einspeisung bis zum ML-Modell
 - ◆ Jede Art von Transformation wie Cleaning
 - ◆ Werden durch Data Science vorgegeben

- **In Echtzeitsystemen oder Systemen mit echt großen Datenmengen,**
 - ◆ ... kommen Tools wie Apache Hadoop, Apache Cassandra, S3 zur Datenhaltung ins Spiel
 - ◆ ... kommen Tools wie Apache Kafka, Apache Spark zur Verarbeitung ins Spiel
 - ◆ ... müssen ggfs. diese Schritte der Data Pipeline reprogrammiert werden

- **Data Pipeline bringt die Daten zum ML-Modell**

Qualität muss überwacht werden



Ein Modell ist trainiert – und jetzt?

Feedback Loop – Monitoring und re-Training

- Die Qualität des ML-Modells muss überwacht werden
 - ◆ Modell Degradation
- Zyklische Bewertung Modellqualität mittels einer Metrik
 - ◆ Eine Metrik zur Qualitätsmessung
 - nutzt Informationen, welche zur Laufzeit erhoben werden
 - ◆ Prozess, der zyklisch die Modellqualität misst
 - ◆ Regel, die besagt,
 - wann ein Modell zu schlecht ist
 - ... und re-trainiert werden muss
- Reaktion auf Model Degradation
 - ◆ Stößt ggfs. eine neue Version des Modells an
 - entweder über Data Science oder vollautomatisch
 - ◆ ... basierend auf den neuen Trainingsdaten aus der Laufzeitumgebung

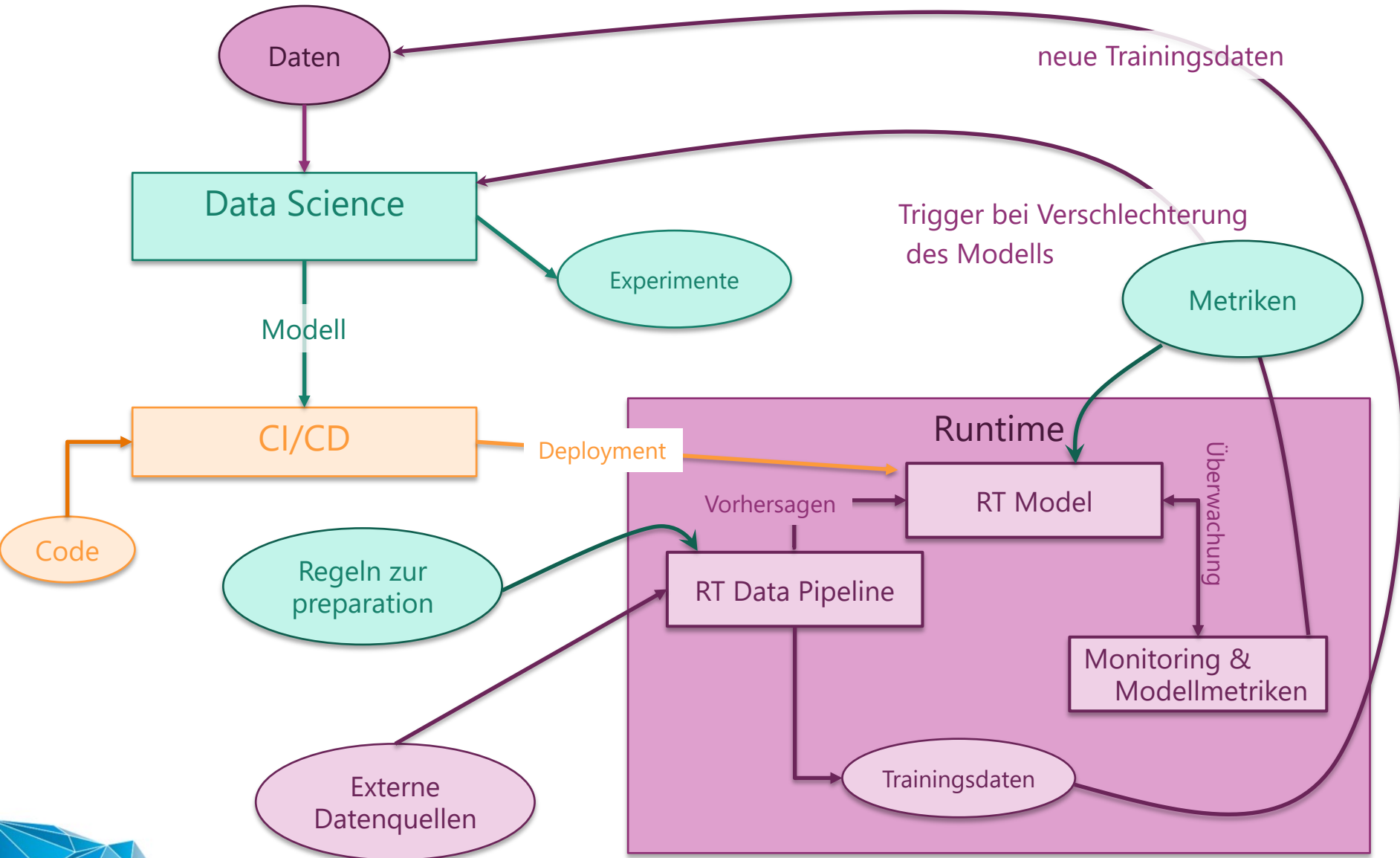
Agenda

- Data Science Modelle als Softwaresysteme
- MLOps und Data Science
- MLOps und CI/CD
- MLOps und produktive Laufzeitumgebung
- Take Home

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Tieferer Einstieg in den Runtime-Teil



Take Home – MLOps

* Durchgängiger Entwicklungsprozess für ML-Modelle

- ◆ Vergleichbarkeit von ML-Modellen in der Entwicklungsphase
- ◆ Integration in CI/CD
- ◆ Unterstützung von ML-Modellen in unterschiedlichen Laufzeitumgebungen
- ◆ Monitoring von Modellen

* Voraussetzung dazu sind ...

- ◆ ... universelles Modell-Paketierungsformat

Take Home – MLOps

- ✿ Durchgängiger Entwicklungsprozess für ML-Modelle
- ✿ Der MLOps-Entwicklungsprozess integriert sich in den Bestehenden
 - ◆ ... Integration in bestehender CI/CD-Tools
 - ◆ ... Integration in bestehende Laufzeitumgebungen
- ✿ Der MLOps-Entwicklungsprozess ergänzt den Bestehenden
 - ◆ ... Umgang mit ML-Modellen in Data Science
 - ◆ ... Paketierung von ML-Modellen
 - ◆ ... Monitoring von ML-Modellen

Ein integrierter, automatisierter Entwicklungsprozess ist
in „normalen“ Softwareprojekten Standard

... dazu muss es in Data-Science-Systemen auch kommen

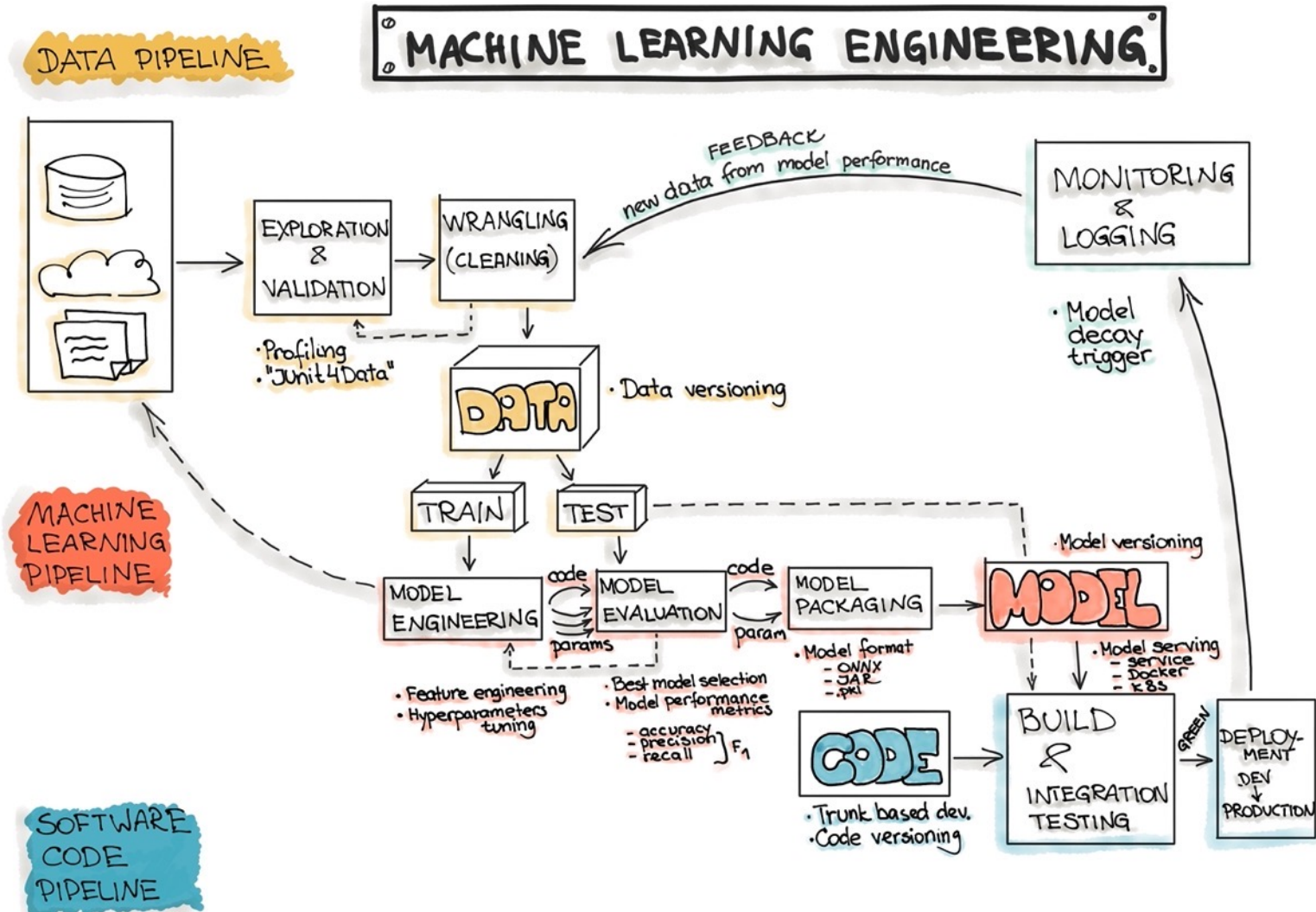
MLOps

Backup

Ein Modell ist trainiert – und jetzt?

WWW.IKS-GMBH.COM

ML Engineering



Was will MLOps

- **MLOps aims to unify the release cycle for machine learning and software application release.**
- **MLOps enables automated testing of machine learning artifacts (e.g. data validation, ML model testing, and ML model integration testing)**
- MLOps enables the application of agile principles to machine learning projects.
- MLOps enables supporting machine learning models and datasets to build these models as first-class citizens within CI/CD systems.
- MLOps reduces technical debt across machine learning models.
- **MLOps must be a language-, framework-, platform-, and infrastructure-agnostic practice.**

Rollen

Business Experte

- Provide business questions, goals, or KPIs
- Continually evaluate and ensure that model performance aligns with or resolves the initial need.

Data Scientist

- Build models that address the business question
- Deliver operationalizable models
- Assess model quality

Data Engineer

- Optimize the retrieval and use of data to power ML models.

Software Engineer

- Integrates ML-Model and ensure that ML models work seamlessly with other non-machine-learning-based applications

DevOps

- Continuous Integration/Continuous Delivery (CI/CD) pipeline management.

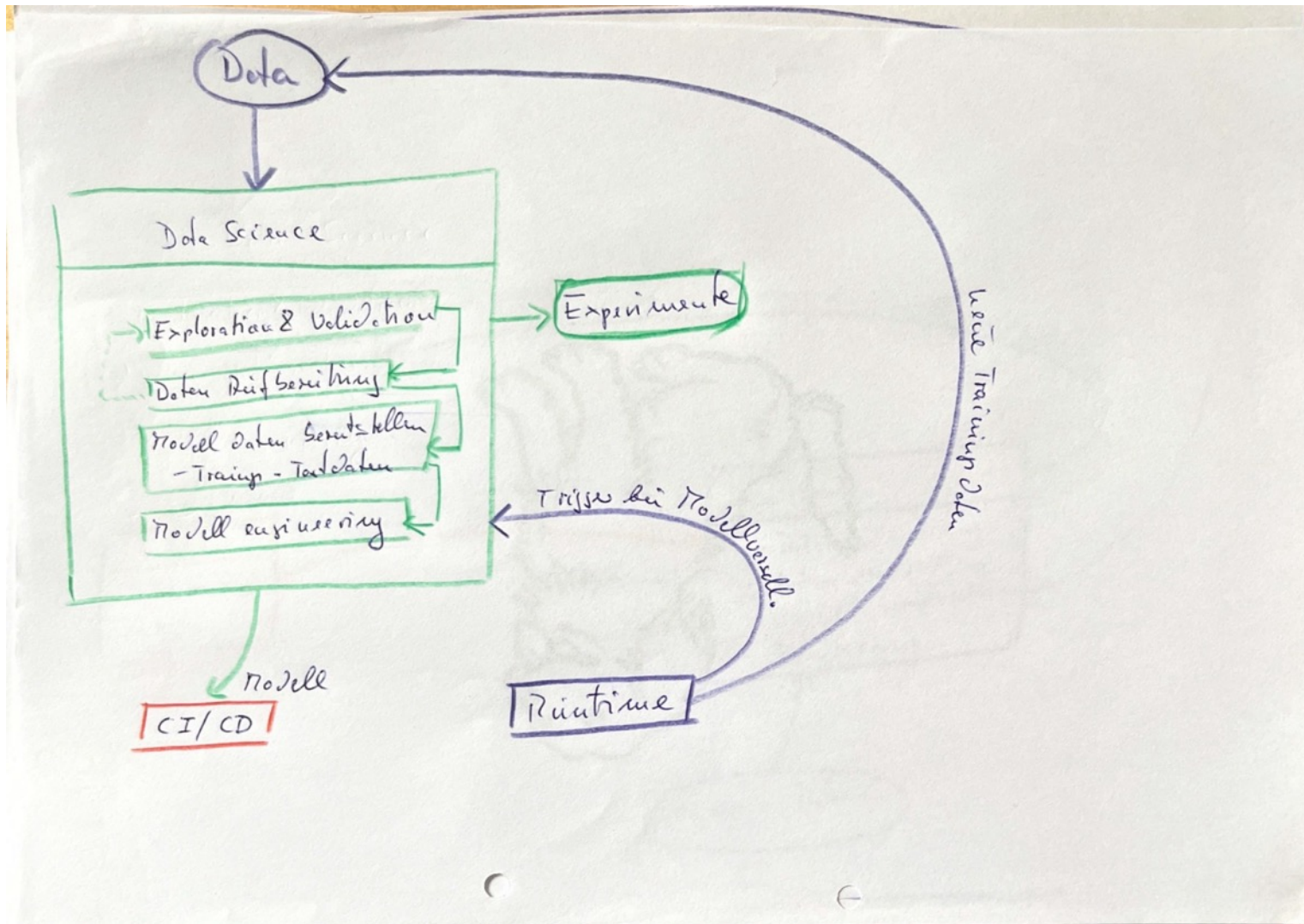
Model Risk Mngmt

- Minimize overall risk to the company as a result of ML models in production.
- Ensure compliance with internal and external requirements before pushing ML models to production.

Maschine Learning Architect

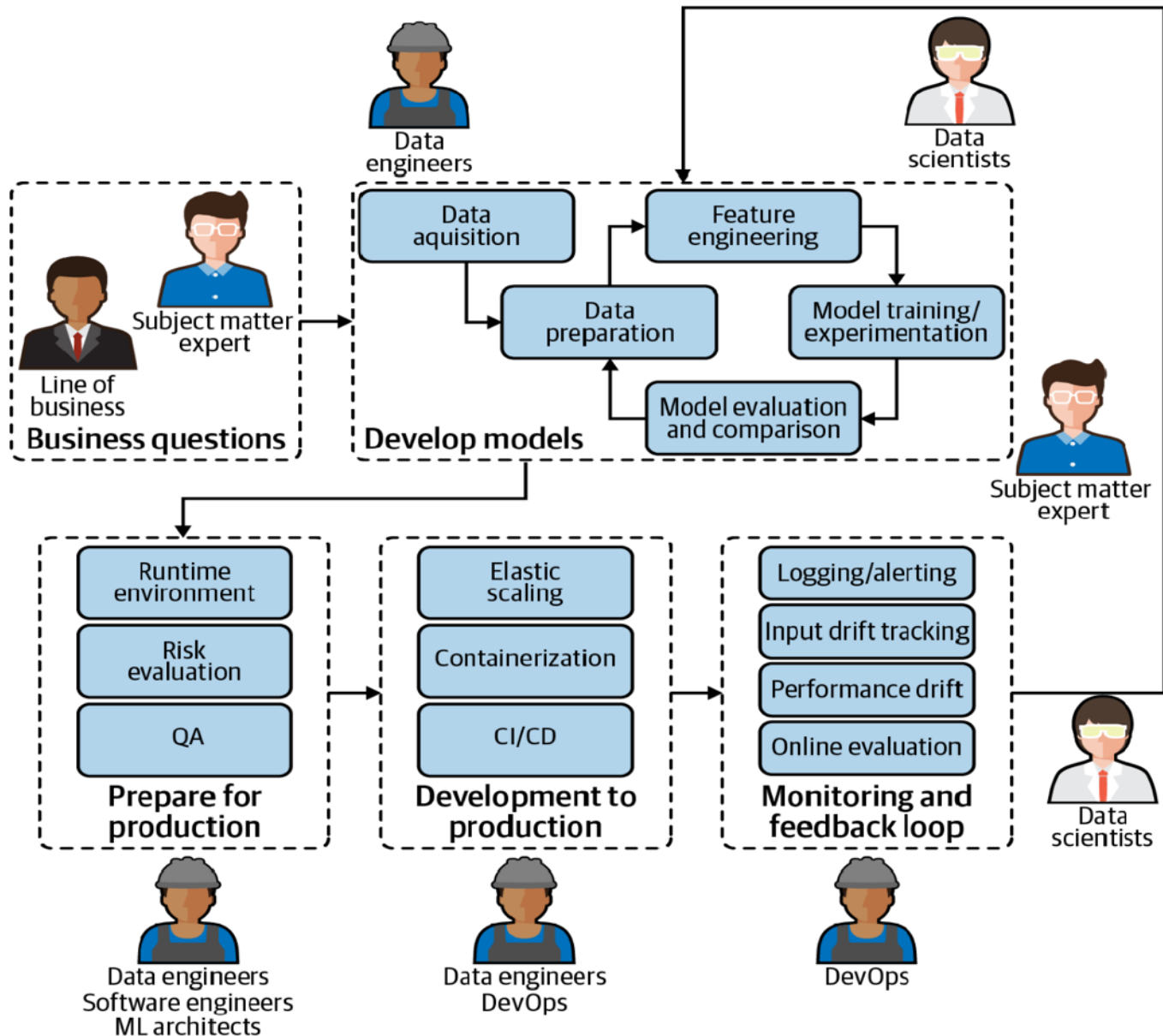
- Ensure a scalable and flexible environment for ML model pipelines, from design to development and monitoring.

Tieferer Einstieg in den Data Science-Teil



Ein Modell ist trainiert – und jetzt?

Rollen und ihr Zusammenspiel



Agenda

- Data Science Modelle als Softwaresystemen
- MLOps und DataScience
- MLOps und Data Engineering

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Was ist ein ML-Modell



ML-Modell besteht aus :

- ◆ Trainingsdaten
- ◆ Performancemetrik
- ◆ ML Algorithmus
- ◆ Hyperparameter
- ◆ Evaluation- oder Test-Datensatz

Vergleich von ML-Modellen

Versionierung und Reproduzierbarkeit

Qualitätssicherung

Governance und Risiko Management



Data Governance

- ◆ What is the data's provenance?
- ◆ How was the original data collected and under what terms of use?
- ◆ Is the data accurate and up to date?
- ◆ Is there PII or other forms of sensitive data that should not be used?



Risikomanagement

- ◆ Welche Risiken gibt es ?
 - Unklare Datenherkunft , DSGVO-Verstoß, politisch inkorrekt Bias



Process Governance

Was kann mit ML-Modellen in Produktion passieren?

- After deploying the ML model into a software system, we might recognize that as time goes by, the model starts to decay and to behave abnormally, so we would need new data to re-train our ML model.
- After examining the available data, we might recognize that it's difficult to get the data needed to solve the problem we previously defined, so we would need to re-formulate the problem.
- In the ML project at some stages, we might go back in the process and either collect more data, or collect different data and re-label training data. This should trigger the re-training of the ML Model.
- After serving the model to the end-users, we might recognize that the assumptions we made for training the model are wrong, so we have to change our model.
- Sometimes the business objective might change while project development and we decide to change the machine learning algorithm to train the model.

Agenda

- Data Science Modelle als Softwaresystemen
- MLOps und DataScience
- MLOps und Data Engineering

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

ML-Model Degradation

- Modellverschlechterung muss durch ständiges Beobachten überwacht werden
- Ist die Modellverschlechterung zu groß, so muss (je nach Problem):
 - ◆ Modell re-trainiert werden
 - ◆ Hyperparameter angepasst werden
 - ◆ Neues Modell entwickelt werden
- Die aktuellen Daten werden als Trainings- und Testdaten bereit gestellt

Monitor und Feedback-Loop

Deployment Pipeline

- **Deployment meint, dass**
 - ◆ ... ein ML-Modell verpackt wird
 - ◆ ... und in unterschiedlichen Umgebungen reaktiviert werden kann

- **ML Modell muss so ‚verpackt‘ werden, dass**
 - ◆ ... dem ML-Modell eine eindeutige und feste Version zugeordnet werden kann
 - ◆ ... jederzeit ein Test des Modells möglich ist
 - Metriken müssen geprüft werden
 - Trainings- und Testdaten müssen verfügbar sein
 - Software, in der das Modell geschrieben ist, ist verfügbar
 - Code zum Testen des Modells ist verfügbar
 - ◆ ... das ML-Modell in der Produktions-Infrastruktur in Betrieb genommen werden kann
 - ML-Modell muss eine Schnittstelle anbieten

Deployment Pipeline

> Anpassung der

Verpackungsmaterial

- Containerisierung
 - ◆ Docker ...
- Paketierungsformate

Model Degradation – Verschlechterung von ML-Modelle

➤ Ground Truth Evaluation

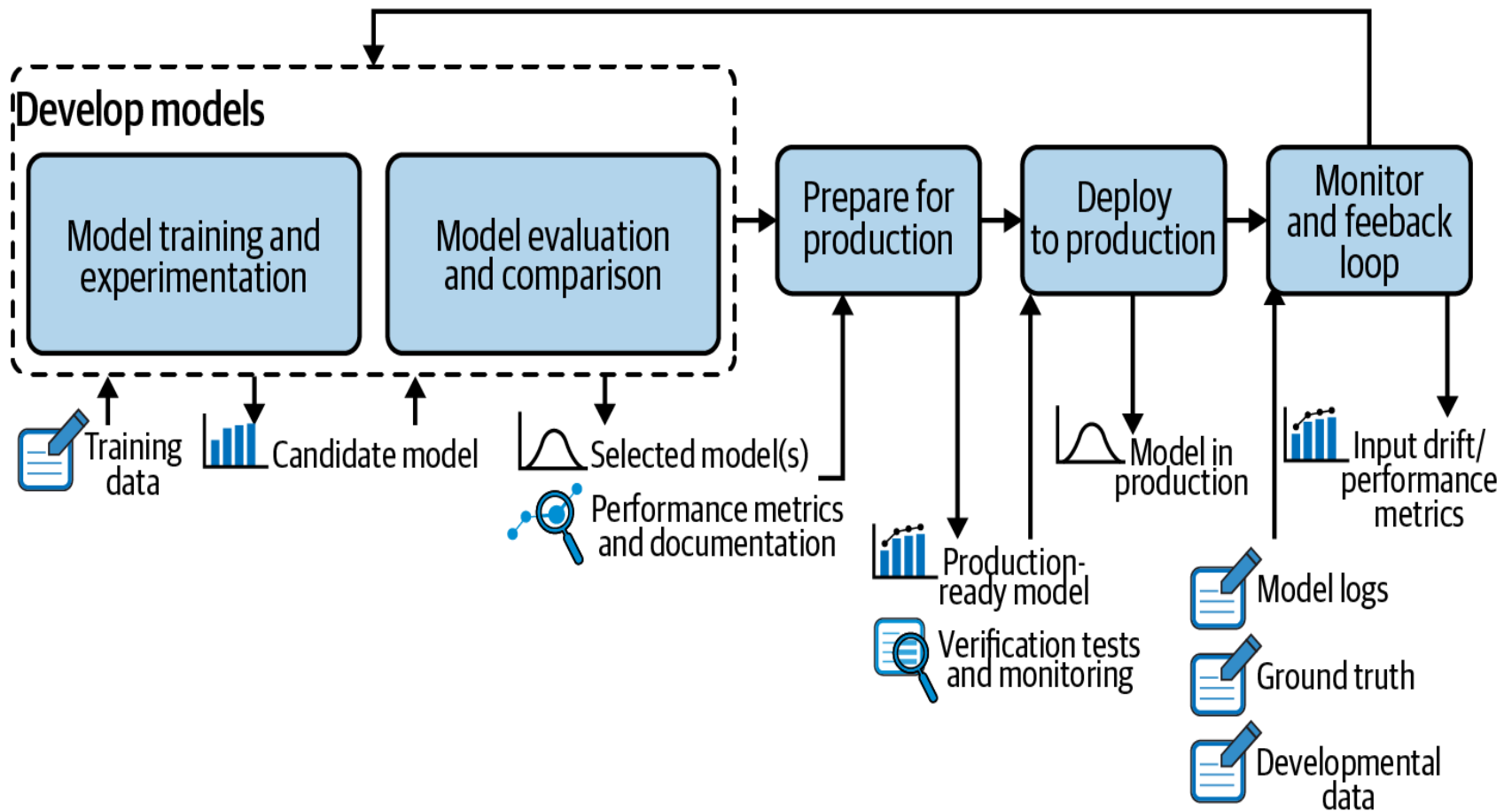
- ◆ ML-Modelle mit *labeling* (das Ergebnis/Performance der Vorhersage wird im Prozess ermittelt)
 - System stellt eigene Trainingsdaten her !!
 - ... damit kann untersucht werden, ob das Modell noch zu den aktuellen Daten passt
- ◆ Metriken, um die Qualität des ML-Modells zu messen, basieren auf neuen Daten
- ◆ Probleme:
 - Performance lässt sich erst nach langer Zeit ermitteln
 - Vorhersage und Performance lassen sich schwer oder gar nicht zuordnen
 - Technisch liegen die Daten in getrennten Systemen
 - Es ist sehr auswendig/teuer, die Performance zu ermitteln

Model Degradation – Verschlechterung von ML-Modelle

➤ Input Drift Detection

- ◆ ML-Modelle ohne *labeling* (das Ergebnis/Performance der Vorhersage steht nicht zur Verfügung)
- ◆ ... untersuche die statischen Eigenschaften der Trainings- und der aktuellen Daten
- ◆ ... oder ML-Modelle, um *data drift* zu erkennen

Monitor und Feedback-Loop



Ein Modell ist trainiert – und jetzt?

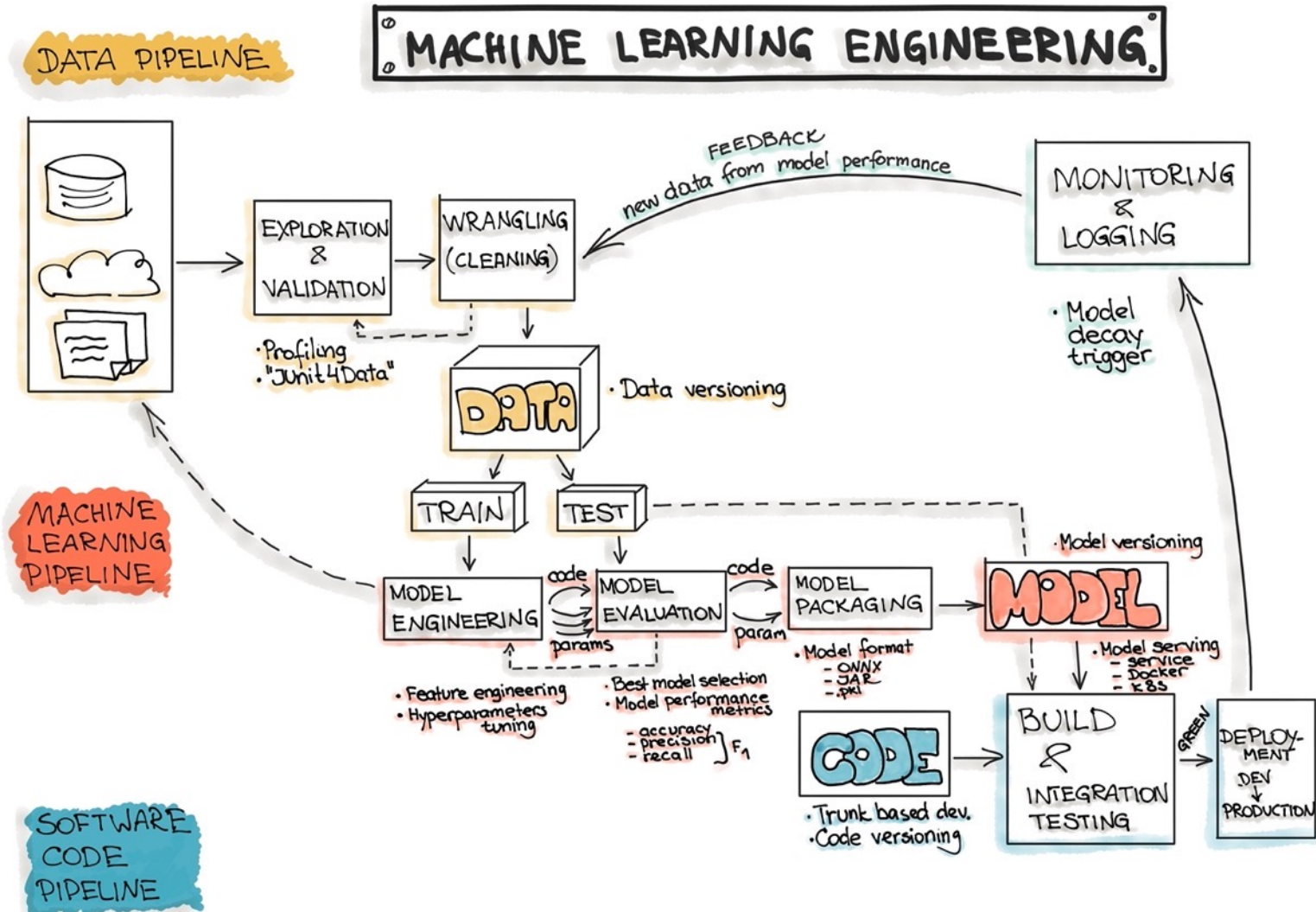
Agenda

- Data Science Modelle als Softwaresystemen
- MLOps und Data Science
- MLOps und Data Engineering
- Abschluss

Ein Modell ist trainiert – und jetzt?

Alles dreht sich um Daten | Data Science 'at Laptop' | Data Science 'at Scale'

Zusammenfassung - ML Engineering



WWW.IKS-GMBH.COM