Rise of Tech Conference
MLOps – AI-Anwendungen im Team erfolgreich operationalisieren

Kurt Fritz
Verbund Energy4Business
Team Data Intelligence & Pricing
17.06.2025

Verbund

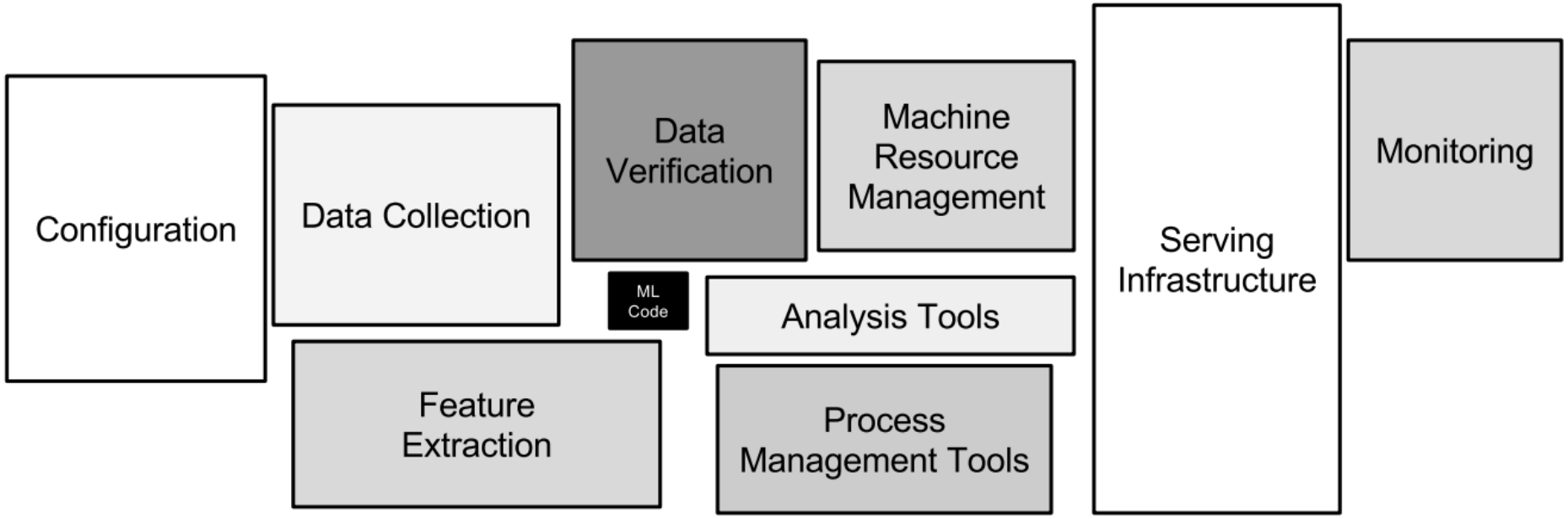
Intro und Zielsetzung des Workshops

Zielsetzung	Diskussion des Pfades von der ML-Idee zum operativ laufenden KI-Modell
Fokus	Konzeptionelle Ansätze und Prinzipien • Keine spezifischen Technologie- oder Anbieterdiskussionen • Beispielmodell für Veranschaulichungen und Erfahrungen
Interaktiver Austausch	Kurze thematische Impulse mit Folien, gefolgt von gemeinsamer Diskussion
Mein Hintergrund	Studium Mathematik und Physik 12 Jahre Erfahrung in der Energiewirtschaft
Team	Team Data Intelligence und Pricing Wir schaffen maßgeschneiderte quantitative Lösungen für Handel, Vertrieb und Asset-Vermarktung von Verbund Energy4Business

V

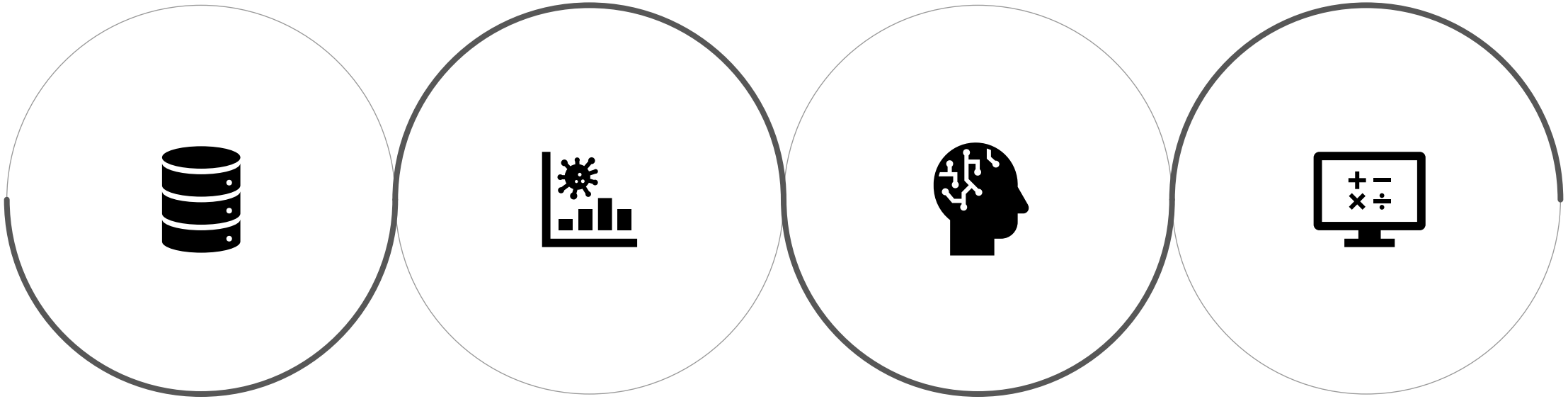
Hidden Technical Debt in Machine Learning Systems

Technical Debt (Technische Schuld) … zusätzlicher Aufwand, den man für Änderungen und Erweiterungen an schlecht geschriebener Software im Vergleich zu gut geschriebener Software einplanen muss.



Hidden Technical Debt in Machine Learning Systems, D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips (2015)
https://proceedings.neurips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

Skillsets Machine Learning Realisierung und Betrieb



Data Engineering

Aufbau und der Wartung von Infrastruktur und Prozessen, um Daten zuverlässig zu sammeln, zu transformieren, zu speichern und für Analyse und Training aufzubereiten

Data Analysis

Untersuchung, Bereinigung, Transformation und Modellierung von Daten, um nützliche Informationen, Zusammenhänge und Schlussfolgerungen aufzudecken

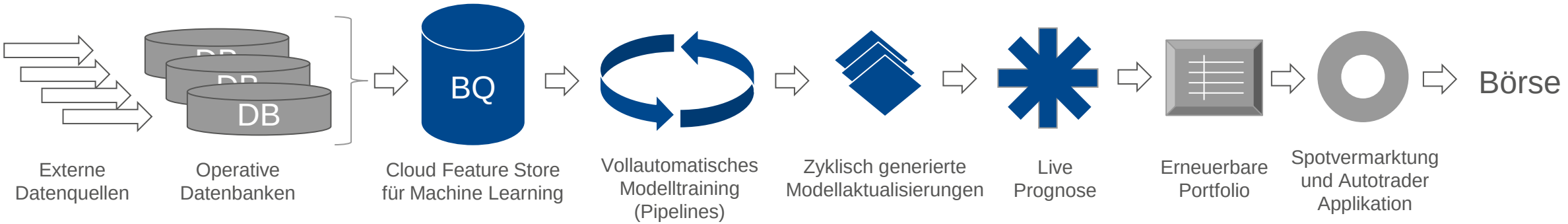
ML-Engineering

Aufbau, Bereitstellung und der Wartung von skalierbaren, zuverlässigen und automatisierten Pipelineprozessen für Machine Learning Modelle

Software Engineering

Entwicklung robuster, skalierbarer und wartbarer Softwareinfrastruktur und -prozesse, um ML-Modelle zuverlässig in Produktionsumgebungen bereitzustellen und zu verwalten

Illustrierender Beispiel-Case:
Integration ML-Modell in operativen Workflow Erneuerbare Energien



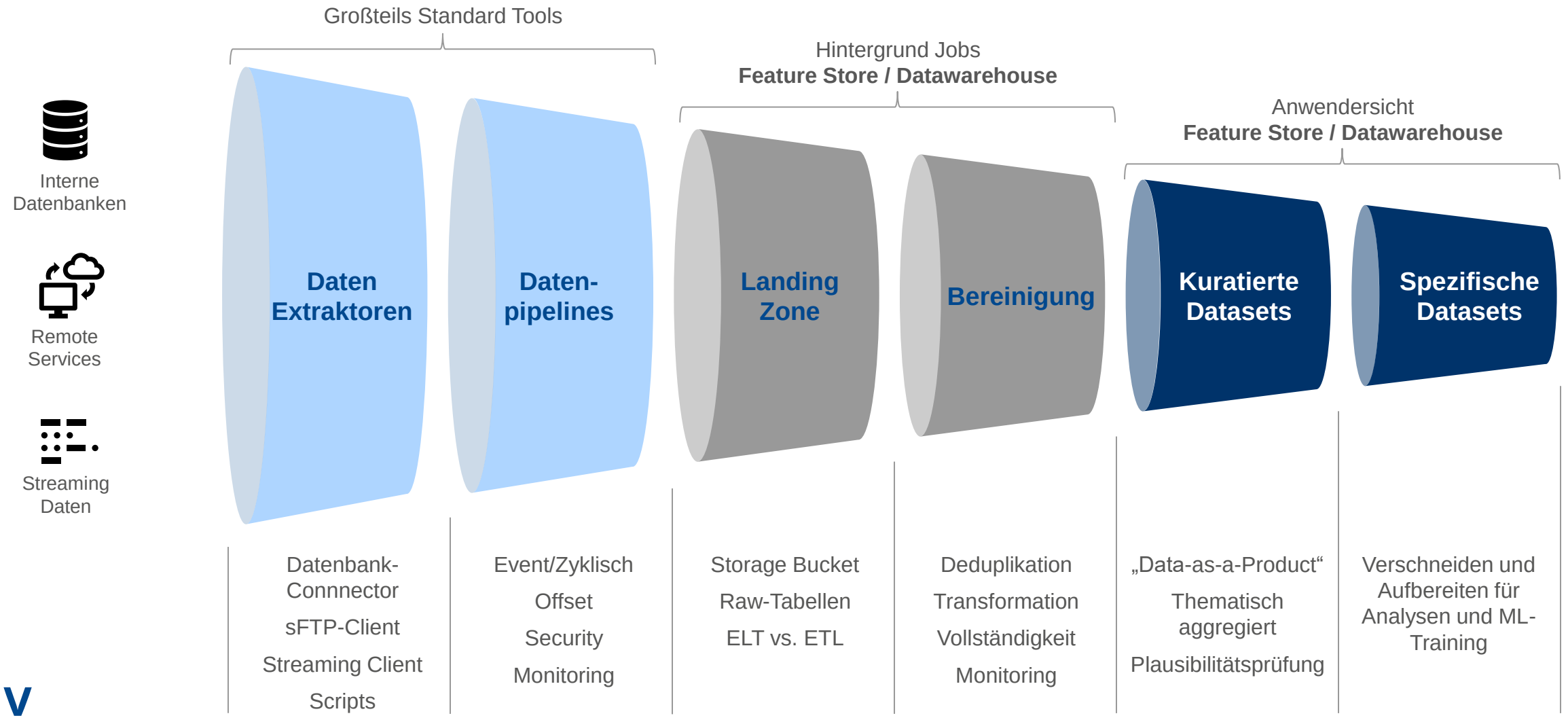
Eingangsdaten sind u.a.: Wind-, PV-Prognosen, Verbrauchsprognosen DE und Nachbarländer, akt. Regelzonensaldo, akt. ID-Preise, Spotpreise

Nutzung der Cloud als Machine Learning Infrastruktur

- Selbstverwaltet im Team
- Vorbildwirkung für Infrastruktur as Code und agiler Cloud-Nutzung (MLOps-Prinzip)
- Nutzung von State-of-the-Art AI-Technologien
- Serverless Architektur mit flexibler Ressourcenskalierung und hoher Verfügbarkeit
- Enge Anbindung an Verbund Infrastruktur bei gleichzeitiger Abschottung nach außen

Anmerkung: Erster Usecase in der neu aufgebauten Umgebung (ab 2020)

Solides Data Engineering als Fundament

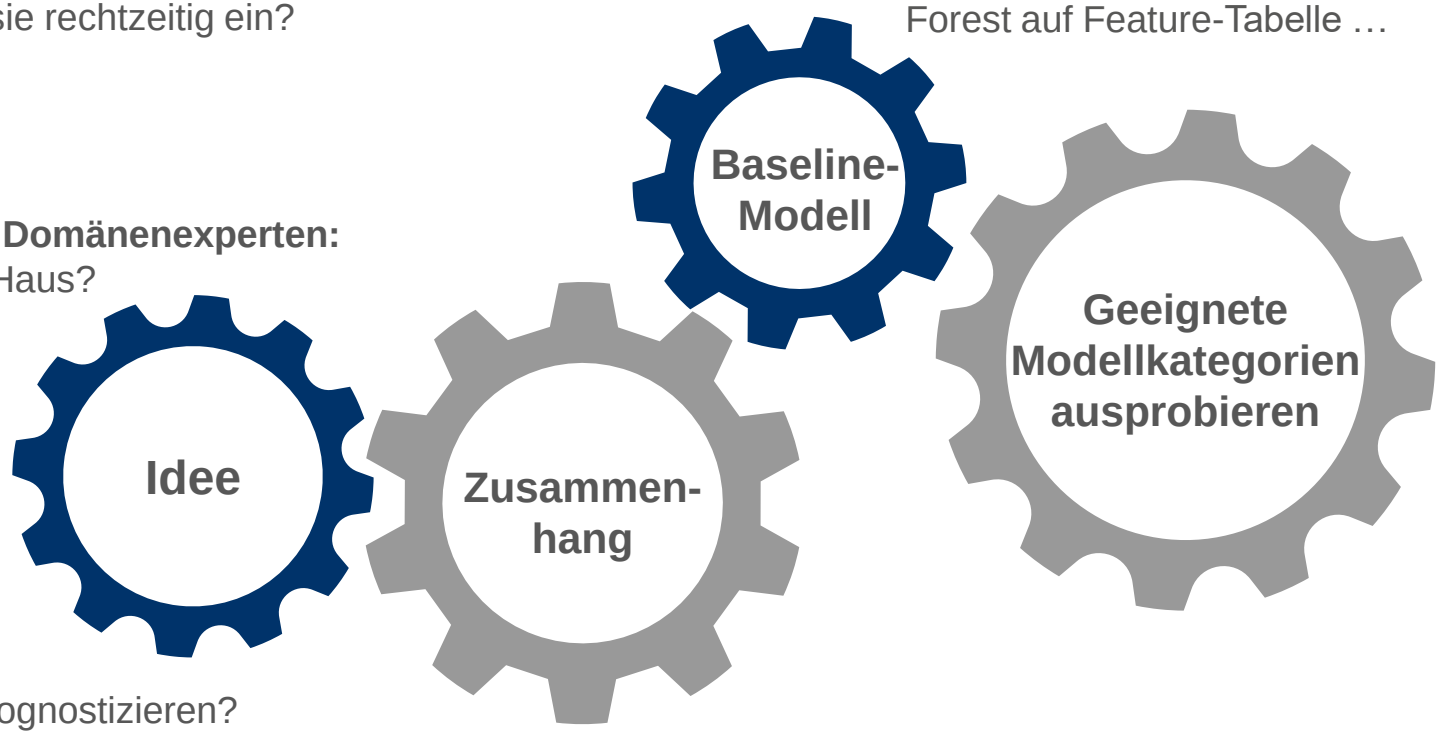


Data Analysis mit Domain Knowledge

Feature Store als Ausgangspunkt:
Haben wir die Daten?
Treffen sie rechtzeitig ein?

Einfach aber nicht zu einfach:
Was schafft ein einfach zu implementierendes Modell?
Z.B. letzten Wert fortschreiben, lineares Modell, Random-Forest auf Feature-Tabelle …

Kommunikation mit Domänenexperten:
Gibt es Expertise im Haus?
Bekannte Einflüsse?



Recherche
Regression vs. Klassifikation
Auto ML
Feature-Engineering
Zeitreihenmodelle, Bilder …

Eingrenzung:
Was genau will ich prognostizieren?
Wann genau möchte ich die Prognose erhalten?
Zeitintervall, Wert/Kategorie …
Definitionen schärfen

Kausalität vs. Korrelation:
z.B.: Eiscremeverkäufe und Ertrinkungsunfälle
Plots, KPI, Zeiträume, Ausnahmesituationen etc.

ML-Engineering – Realismus und Plausibilität

Experimentierumgebung

Vom Analyseergebnis zum Modell

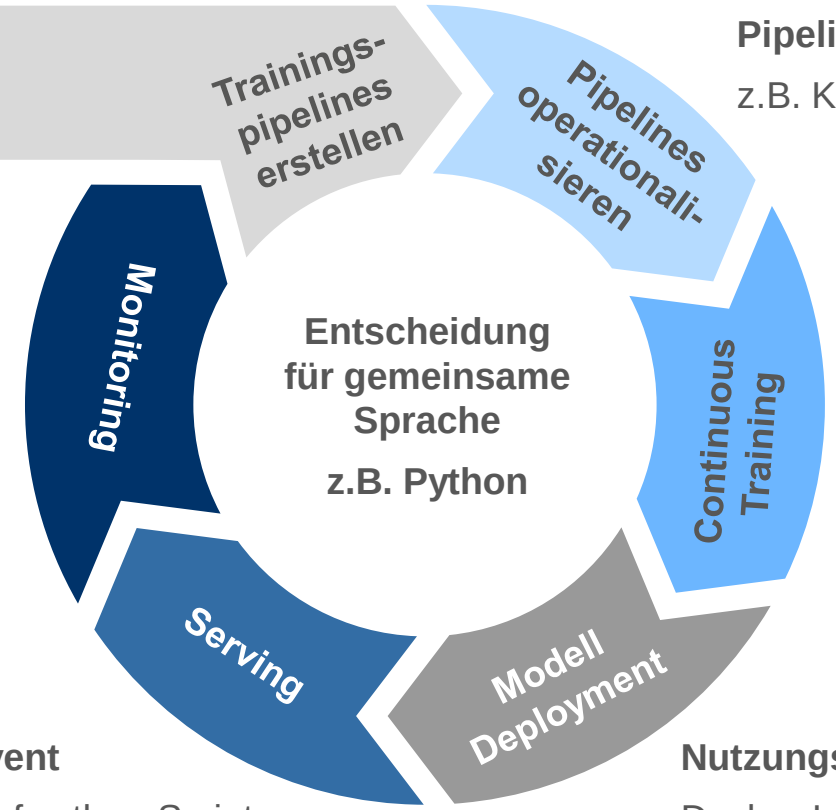
Wichtig: Freiheit aber Kompatibilität!

z.B. jupyter-Notebooks

Überwachung des Modellerfolgs

z.B. Handelserfolg festgehalten

Messbarkeit



Pipelines in Laufzeitumgebung ausführen

z.B. Kubeflow-Pipelines auf Kubernetes

Aktualisierung der Datenbasis

z.B. Wöchentliche Anpassung Trainingsfenster

Modelle versionieren

Scheduler oder Event

z.B. zyklischer Aufruf python-Script

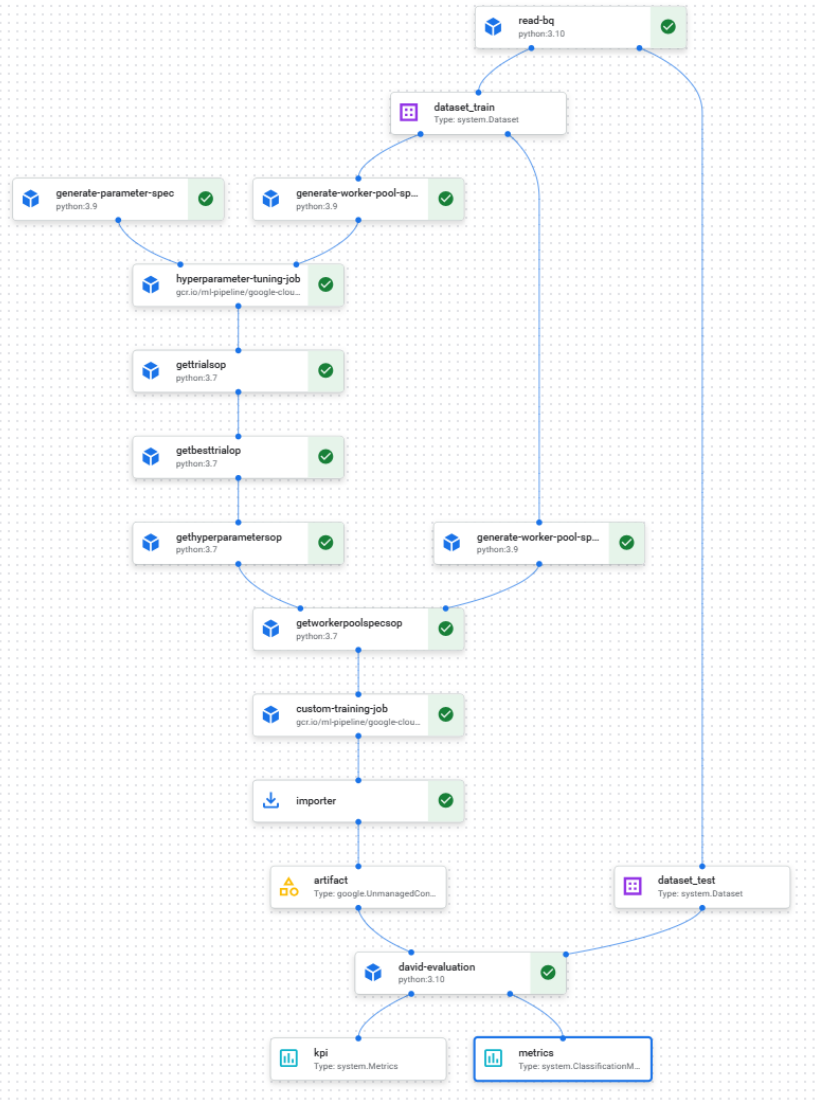
Reaktion auf Streaming Event

Nutzungshäufigkeit & Hosting-Kosten

Docker-Image & Serverless Cloud vs.

Modell bei Bedarf aus Speicher laden

ML-Engineering – Die konkrete ML-Pipeline
Überlegungen und Fragestellungen



Komplexes oder einfaches Modell?
- Bestmögliche Vorhersagequalität?
- Robust im operativen Betrieb?
- Feature-Pflegeaufwand?

Was ist das Maß für ein gutes Modell?
- Handelsergebnis insgesamt oder konstanter Zuwachs?
- Kontinuierlicher Ergebnisbeitrag über lange Zeiträume?

Was wird in der Trainingspipeline benötigt?
- Modelltyp-Selektion?
- Feature Engineering
- Hyperparametertuning?
- Modellevaluation?
- Qualitätscheck?

Ist das Gesamtsetup in sich schlüssig oder gibt es Problemstellen?

Wo läuft die Pipeline?
- Kubernetes/Cloud/Lokal?
- Abhängigkeiten von Infrastruktur/Package-Versionen?

Wo werden die trainierten Modelle verwaltet?

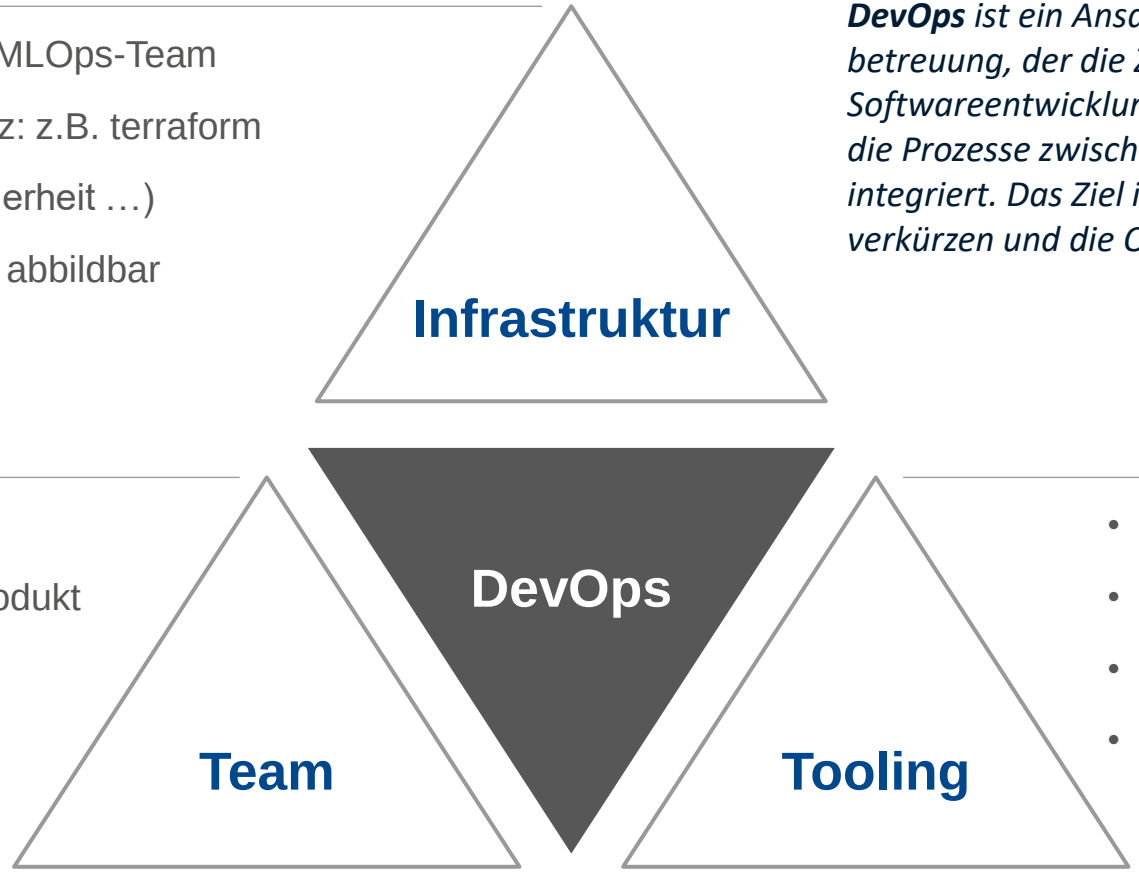
Wie erfolgt das Serving?
- App/Service/Individueller Aufruf?

Metadatenmanagement und Qualitätssicherung?
- Welche Prognose wurde mit welchem Modell gerechnet?

Software Engineering – Nutzen von Erfahrung aus DevOps-Ansatz

- Abgestimmter Rahmen („Bubble“) für MLOps-Team
- Infrastructure-as-Code für Transparenz: z.B. terraform
- IT-Security (Package-Scan, Datensicherheit …)
- Ziel: Idee bis Live-Betrieb durch Team abbildbar

***DevOps** ist ein Ansatz zur Softwareentwicklung und -betreuung, der die Zusammenarbeit zwischen Softwareentwicklungs- und IT-Betriebsteams fördert und die Prozesse zwischen diesen Teams automatisiert und integriert. Das Ziel ist, die Zeit vom Code zur Produktion zu verkürzen und die Qualität der Software zu verbessern.*



- Rollen/Skillsets vs. Personen
- Eigenverantwortlichkeit für das Endprodukt
- Spezialisten mit breiter Basis
- Arbeitsmodus (Sprints, Kanban…)
- Übergabepunkte
- Vertretungsregelungen

- Gemeinsame Umgebung
- Source Code Verwaltung (Git etc.)
- CI/CD
- Statische Code-Analyse

Literaturempfehlung DevOps Novelle: The Phoenix Project (Gene Kim, Kevin Behr, George Spafford)

Ausblick: MLOps für Generative AI und Agents (GenAIOps)

Modell-Ebene:

Provider — **Fine-Tuner** — **Applikationsentwickler**

Data Engineering | Data Analysis | ML-Engineering | Software Engineering

Agent-Ebene:

- Prompt-Engineering
- RAG (Retrieval-Augmented Generation)
- Function Calling
- Agent Testing
- MCP (Model Context Protocol) Server
- Agent Security

Vielen Dank!

Verbund

DE

Mag. Kurt Fritz , MSc

Teamleiter Data Intelligence

Portfolio Management and Energy Economics

VERBUND Energy4Business GmbH

https://vcard.verbund.com/kurt.fritz-0FIKE