Creating value with automation

3.1 Democratizing Machine Learning

3.2 Model Factories

3.3 Continous Learning

Democratizing Machine Learning

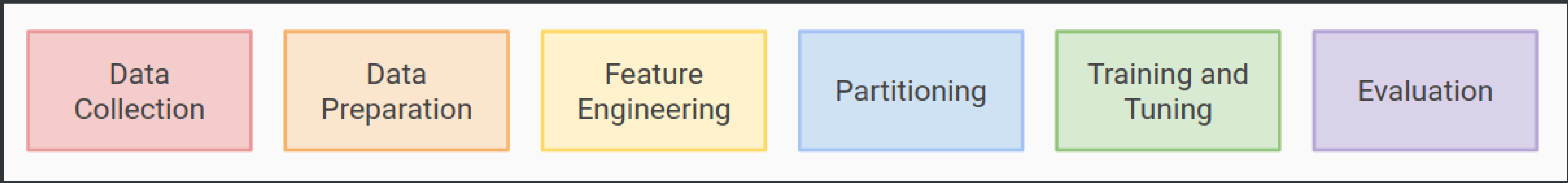
If AI is indeed *the new electricity* - how can we ensure that everyone gets access to it?

Barriers for democratization?

- Workforce shortage
  - Data Scientists cost top dollars and often lack the domain expertise required.
  - Empower Domain experts (e.g. Business Analysts, SWE) instead?
- Research -> Production Gap
  - Building models is easy - getting them into production and business processes is hard.
  - ~80% of ML dont get into production.
  - Unicorn: ML Engineer (strong Data Scientist and Software Engineer w/ DevOps skills; impossible to find).
  - [MLOps](https://en.wikipedia.org/wiki/MLOps) to the rescue \o/
- Access to data
  - Data sources
  - Data catalogues

True End-to-End Learning

- Assist/automate every step in the ML Process



- Open Problems
  - Problem formulation
    - Translate Business Problems into an objective that can be optimized.
    - Select metric and model selection scheme
- Data acquistion & cleansing
  - Automated Feature Engineering
- Operation & maintenance (more on that later).

How can AML help in Democratization?

Empower Business Analysts, Data Engineers, SWE, and Citizen Scientists to use ML to solve problems by providing capabilities to

- build accurate ML models reliabily with little to no human involvement (data acquisition, metric/partitioning selection, pipeline opt),
- with guardrails to avoid pitfalls and catastrophic failure (leakage),
- and guidance when the model should be used and when not!

Where does the AML community need to step up:

- Everywhere but pipeline optimization and hyper-parameter tuning

Model Factories

As organizations become more data driven, **reliable analytics and data science** will become an essential part of staying competitive and keeping costs under control.

Many Data Science teams, however, still **develop models in an ad-hoc fashion** on their workstations and hand over trained models to Data Engineers or SWEs for productionalization.

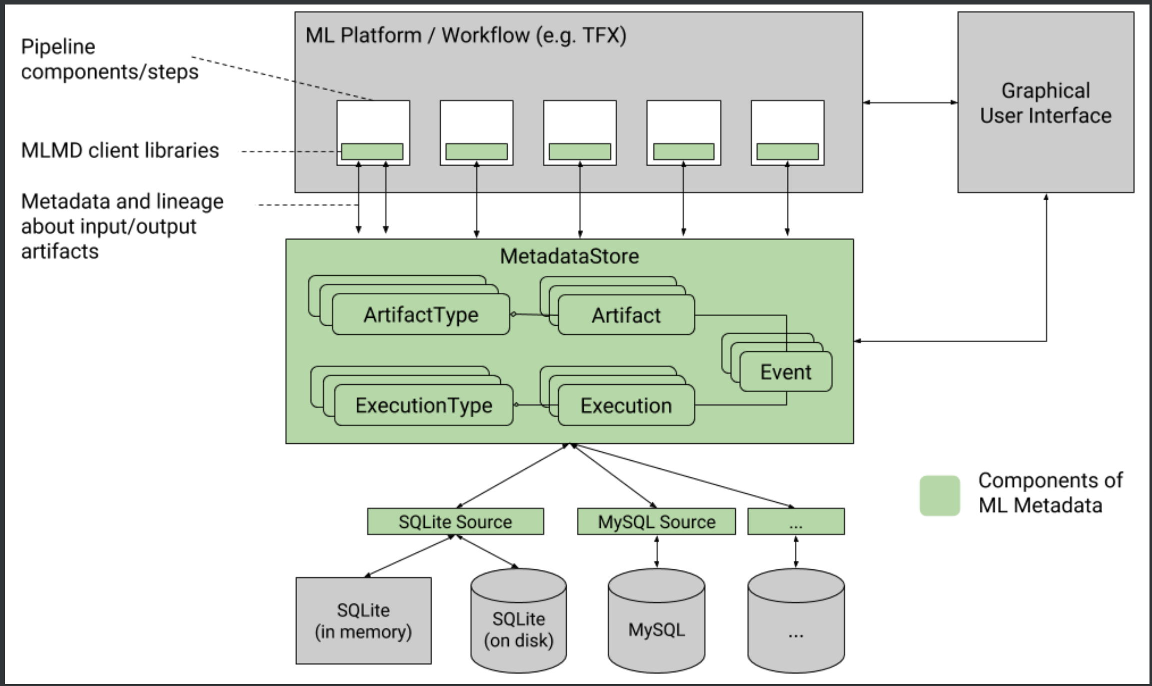
Requirements

- Version control of models & archiving
- Automated build & testing
- Input data checks
- Reproducibility & Lineage
- Governance & regulatory compliance
- Monitoring
- Scheduling

Model Metadata Stores

MLMD

- Provides Lineage
  - Reproducibility
  - Checkpointing (Pause/Resume)
- Versioning



How can AML help in Model Factories?

Consistent Quality

Having a solid model selection and assessment framework minimizes the surface for (human) error.

Governance

Platform ensures that lineage is tracked and metadata recorded (who built what model when and how).

Case-Study: Kubeflow

Open source ecosystem for Machine Learning (automation) for Kubernetes.

Not a *Model Factory* but contains relevant building blocks.

Ecosystem components:

- Pipelines: Workflow orchestration
- Argo CD: Continous Delivery, GitOps
- Fairing: package models trained in a Jupyter notebook
- KFServing/SeldonCore: Model Deployment/Serving
- Katib: Hyper-parameter tuning

Continous Learning

The world constantly changes... this begs the question:

- Are my model assessment results still valid?
- Am I doing worse... or can I do better?

ML Models make assumptions about the data generating process, we need to automatically recognize changes in data.

Data Drift

Sources of changes

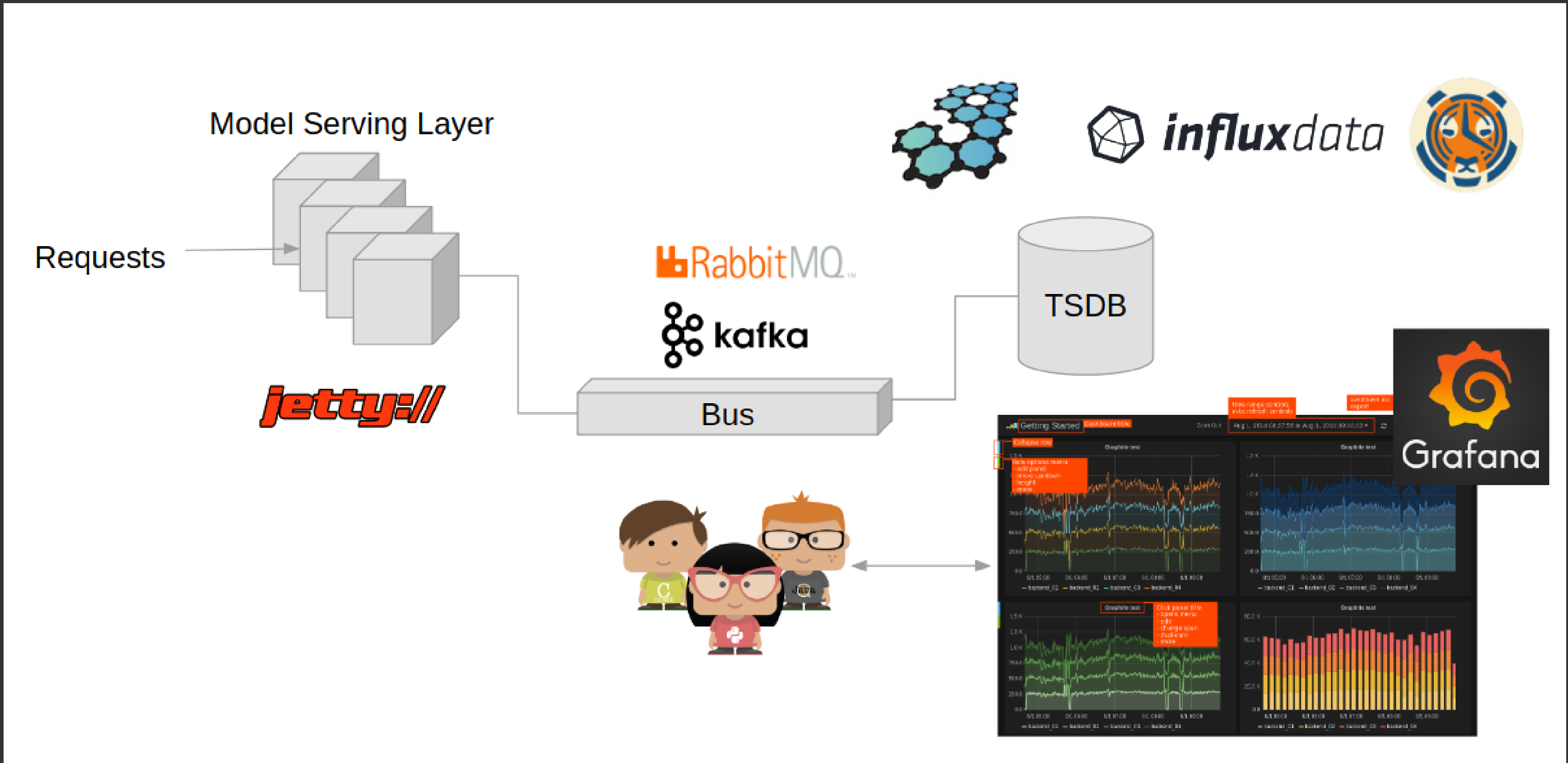
- Changes in DB design, broken sensors, new semantics, new user-interface alters interaction pattern, ...
- In real world, DBs and data sources are living, breathing, evolving entities

Automatically detect drift

- When labels are available: run model assessment again and compare to old values.
- When labels are not available
  - Look for changes in the distribution of the predictions (using histogram distance metrics like Population-Stability-Index)
  - Look for simple univariate statistic
    - Fraction of missing values
    - Fraction of new categorical levels
    - Fraction of values outside a certain range (e.g. 2 * std)

Automatically detecting drift is hard; regularily scheduled retrains are more common.

Model Monitoring Architecture



What to do when Data Drift is detected?

- Warning flags / alert
  - the model might still work (as measured by other KPIs)
- Default to a more robust model
  - This is where AML can help: quickly build a model without a drifting feature.
- Re-train model
  - This is usually a lengthy process..
- Adapt existing model
  - This is for a different time…

Case-study: TFX

TensorFlow Extended (TFX) is an end-to-end platform for deploying production ML pipelines

Key building blocks (not exhaustive):

- TF Data Validation: Understand, validate and monitor data.
- TF Transform: Feature preprocessing as a TF graph; limits what you can do but provides safety and interoperability.
- ML Metadata: Integral part of TFX, ensures compatibility of different artifacts (models)

TFX Training Orchestration