MLOps in action for enterprise

Arnon Jirakittayakorn (CP40)
Lead Data Engineer & Data Architect - Sunday Technology

About Me

- Arnon Jirakittayakorn (Boat) - CP40
- Lead data engineer and data architecture at Sunday Technology
- Senior DE and MLE at Gojek
- Senior DE and DevOps at WNI, Japan
- More than 7 years experiences in the data profession

DATA is the new oil

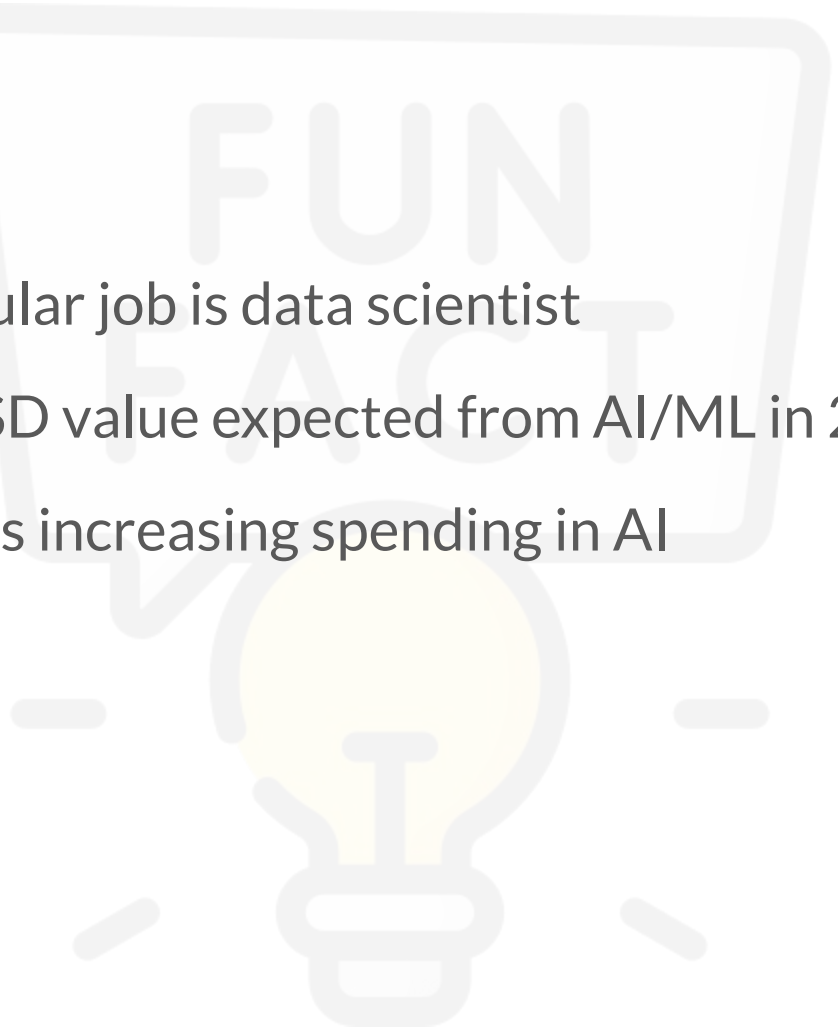
On average, every person created at least **1.7 MB of data per second** in 2020.

there are so many theories/ideas to explore, experiment with, and many discoveries to be made and models to be developed.

Real-Life business isn't a bed of roses…

- Acquiring & cleaning large amounts of data
- Tracking and versioning for experiments and model training runs
- Creating and managing ML pipelines
- Finding a way to scale ML operations
- Dealing with sensitive data at scale
- and about a million other problems…

#1	Most popular job is data scientist
3.9	Trillion USD value expected from AI/ML in 2022 (US data)
60 %	of business increasing spending in AI



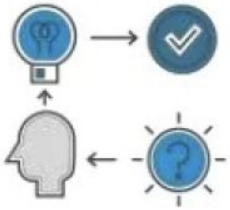
#1 Most popular job is data scientist

3.9 Trillion USD value expected from AI/ML in 2022 (US data)

60 % of business increasing spending in AI

85 % of AI and Big data analytic projects FAILED

Research to Value



Research

Academia

Presentations

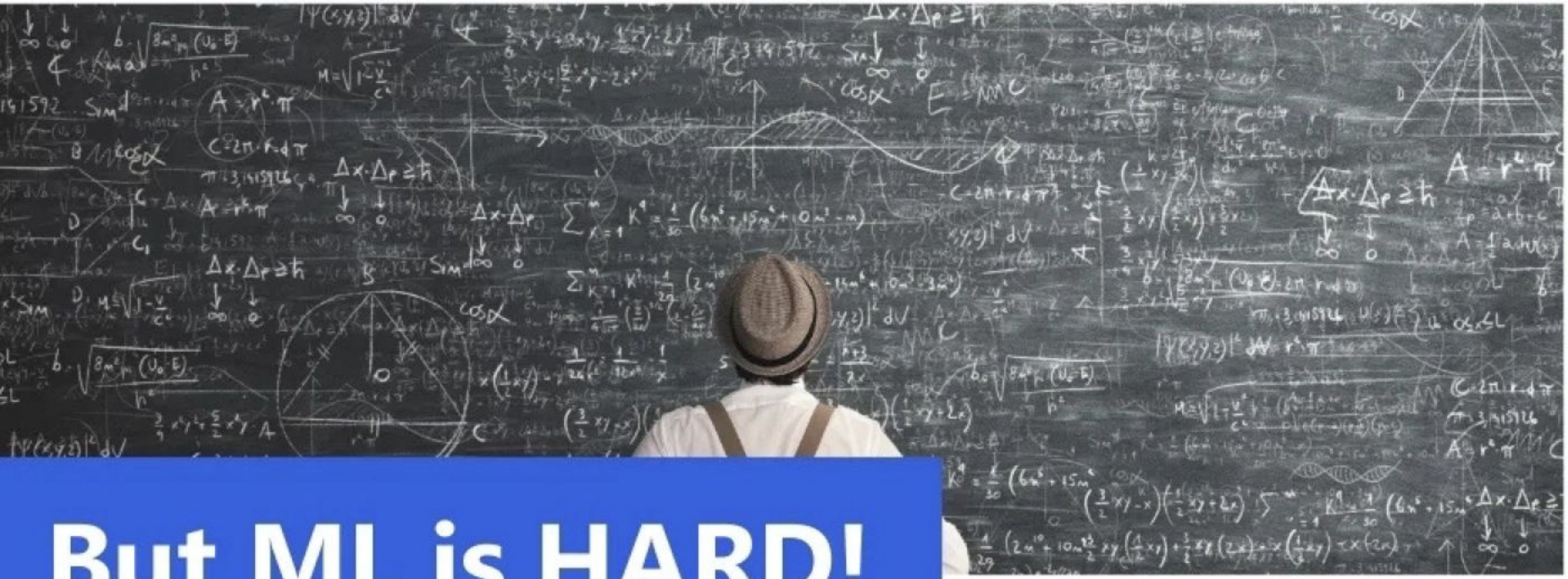
Corporate R&D

PoCs

Data Science

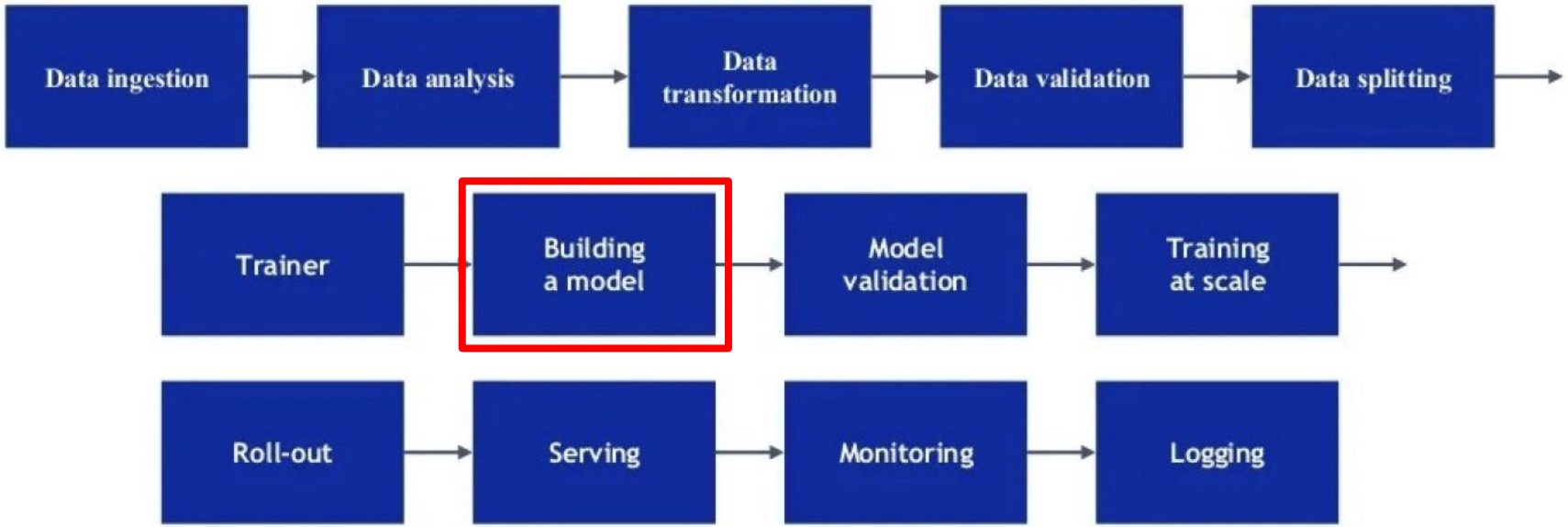
Industrialized ML

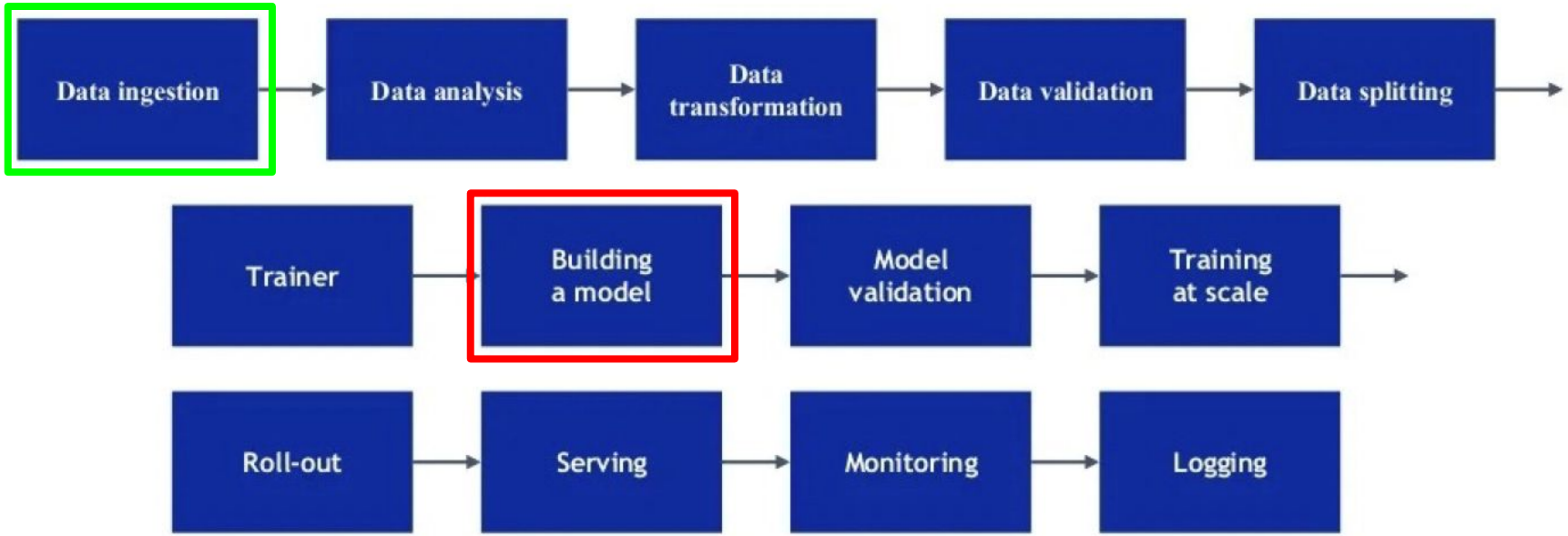
MLOps & ML Engineering

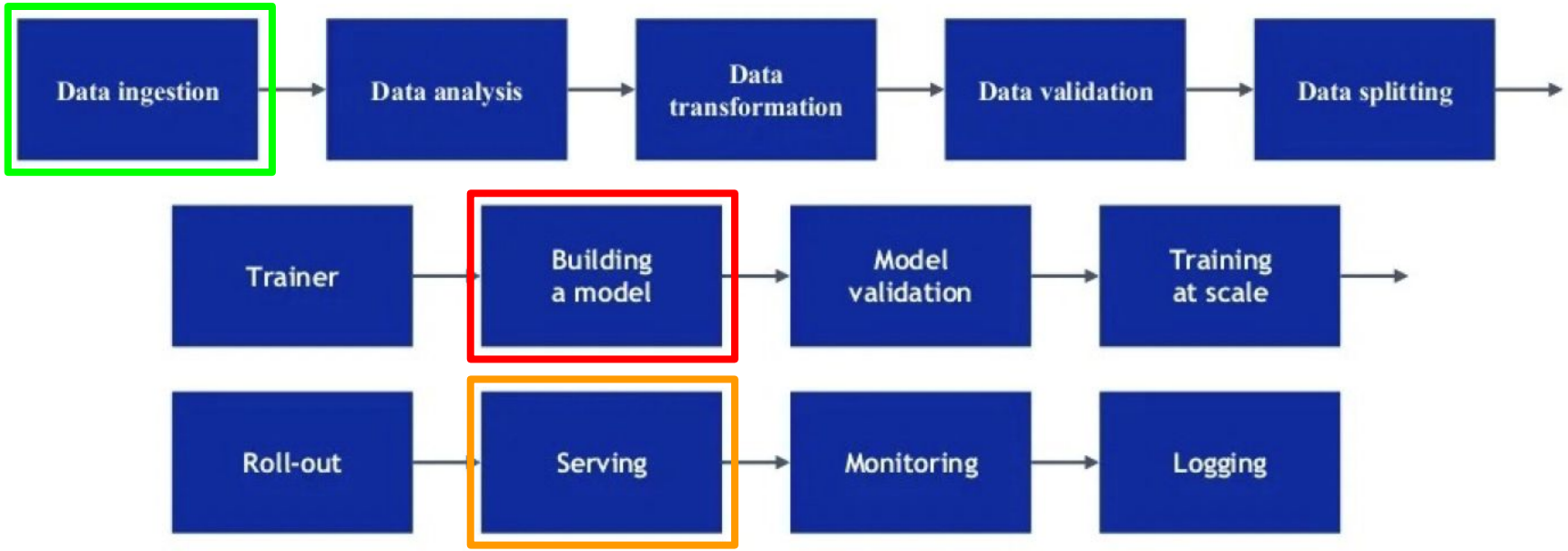


But ML is HARD!

Building a model

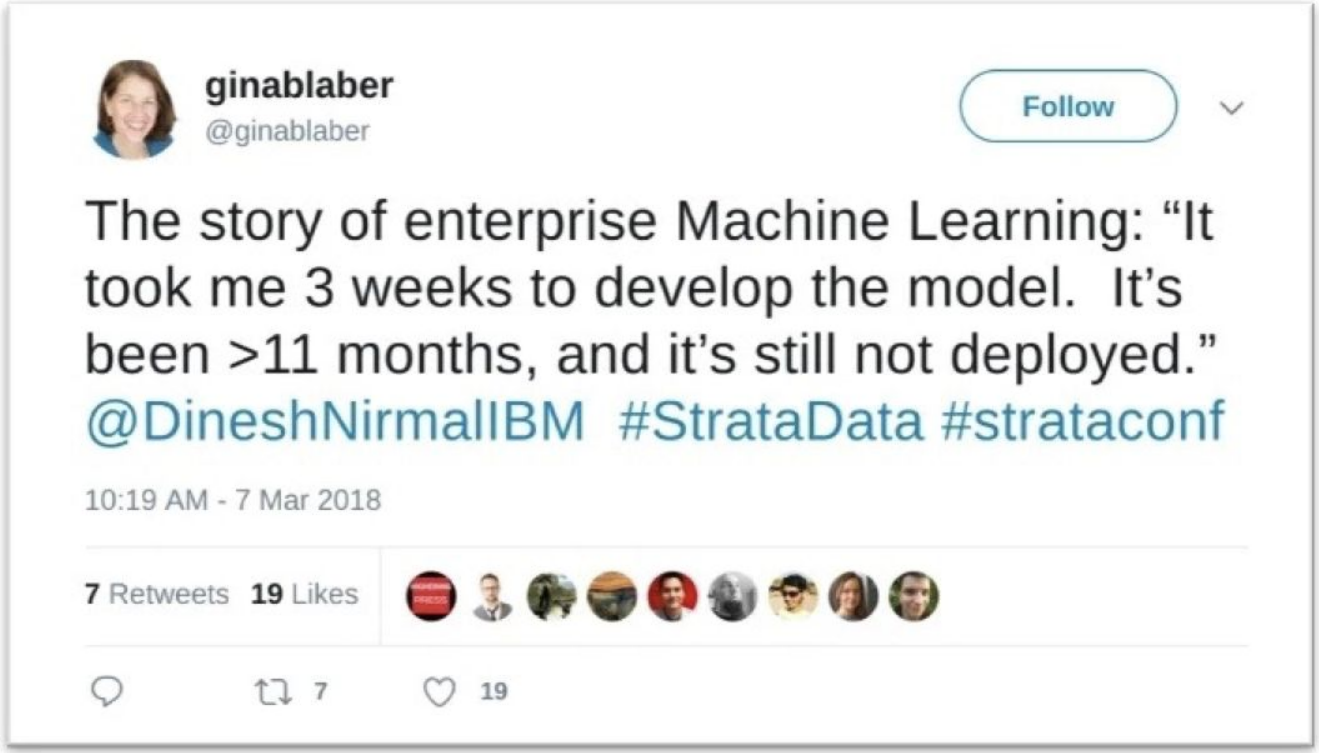

Data ingestion
Data analysis
Data transformation
Data validation
Data splitting
Trainer
Building a model
Model validation
Training at scale
Roll-out
Serving
Monitoring
Logging


Data ingestion
Data analysis
Data transformation
Data validation
Data splitting
Trainer
Building a model
Model validation
Training at scale
Roll-out
Serving
Monitoring
Logging

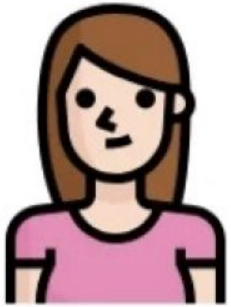

Data ingestion
Data analysis
Data transformation
Data validation
Data splitting
Trainer
Building a model
Model validation
Training at scale
Roll-out
Serving
Monitoring
Logging

Ok, but, like, I'm
a data scientist. ~~IDGAF~~
I don't care
about all that.

Yes You Do!

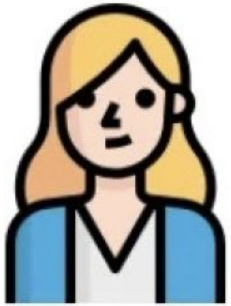

ginablaber
@ginablaber
Follow
The story of enterprise Machine Learning: “It took me 3 weeks to develop the model. It’s been >11 months, and it’s still not deployed.” @DineshNirmalIBM #StrataData #strataconf
10:19 AM - 7 Mar 2018
7 Retweets 19 Likes
7
19

Cowboys and Ranchers Can Be Friends!



Data Scientist

- Quick iteration
- Frameworks they understand
- Best of breed tools
- No management headaches
- Unlimited scale

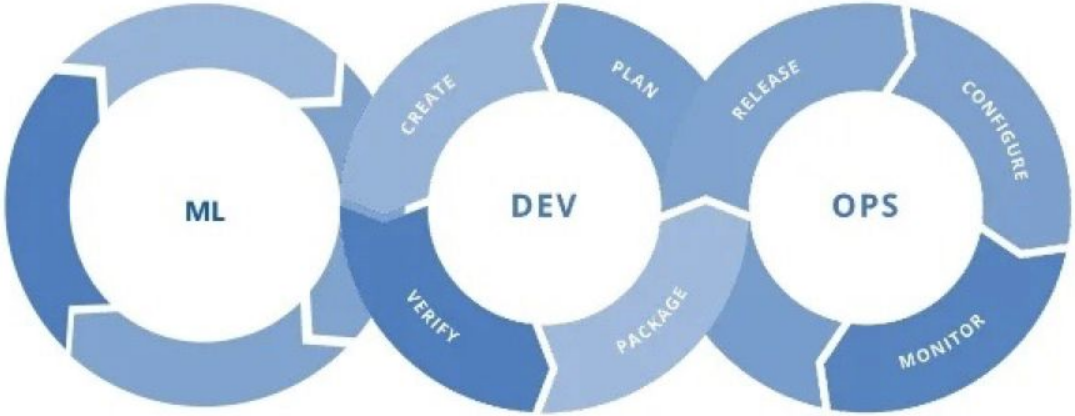


SRE/ML Engineers

- Reuse of tooling and platforms
- Corporate compliance
- Observability
- Uptime

MLOps!

MLOps = ML + DEV + OPS



Experiment
Data Acquisition
Business Understanding
Initial Modeling

Develop
Modeling + Testing
Continuous Integration
Continuous Deployment

Operate
Continuous Delivery
Data Feedback Loop
System + Model Monitoring

MLOps Benefits

Automation / Observability

- Code drives **generation** and **deployments**
- Pipelines are **reproducible** and **verifiable**
- All artifacts can be **tagged** and **audited**

Validation

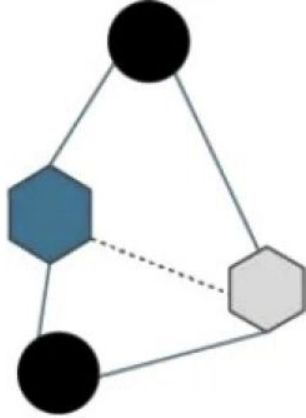
- SWE best practices for quality control
- Offline comparisons of model **quality**
- Minimize **bias** and enable **explainability**

Reproducibility /Auditability

- Controlled rollout capabilities
- Live comparison of predicted vs. expected performance
- Results fed back to watch for drift and improve model

== VELOCITY and SECURITY (For ML)

Why MLOps?

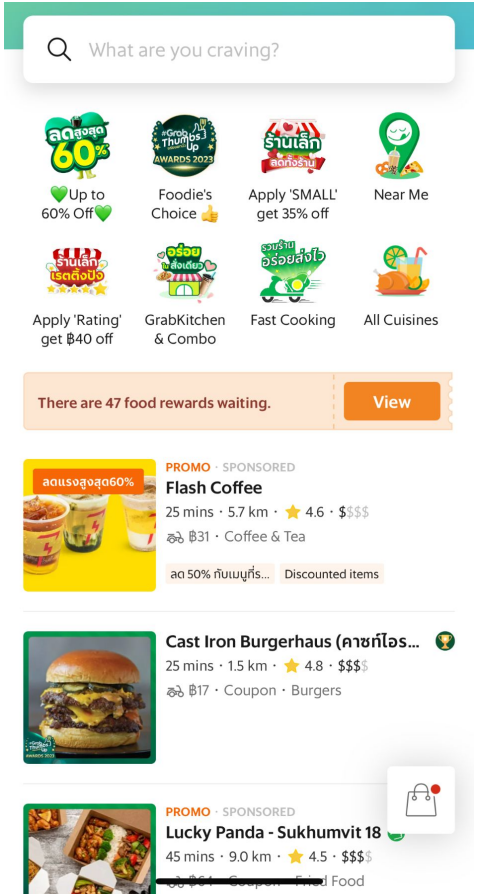


- Organizations need to deploy systems of models, not just one off solutions
- Models must be constantly monitored and retrained to maintain accuracy
- Structure is inconsistent from project to project
- Models and experiments are not properly tracked
- Code and dependencies aren't being properly managed

MLOps Engineer

- Responsible for ensuring that ML engineers can scale the machine learning models across the entire organization. They are responsible for building and maintaining the infrastructure that will allow this scaling to occur.
- Designs, builds, and runs machine learning systems at scale.
- Monitor the performance of your models, and they need to be able to troubleshoot any errors or bugs that may occur.
- In addition to these responsibilities, an MLOps Engineer might be tasked with improving your model's accuracy by tweaking its parameters or updating the data it uses for training.

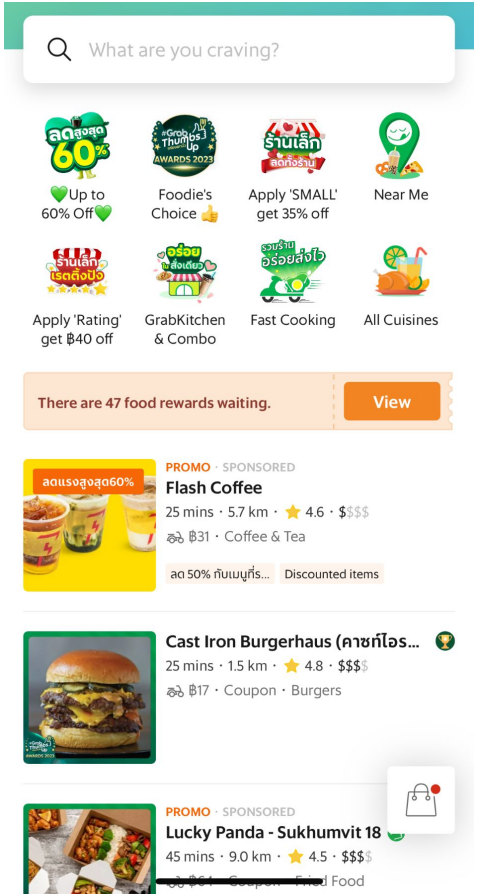
Use case : Recommendation Services



Food

- 200,000 Restaurant
- 20 menus per resto in average
- 4,000,000 SKUs

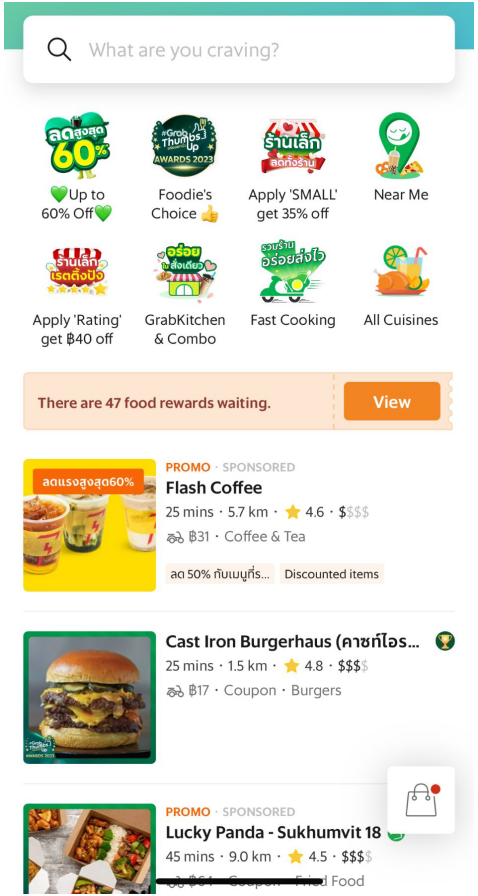
Use case : Recommendation Services



Food
- 200,000 Restaurant
- 20 menus per resto in average
- 4,000,000 SKUs

Users
- 10,000 users during 11.00 - 13.00

Use case : Recommendation Services

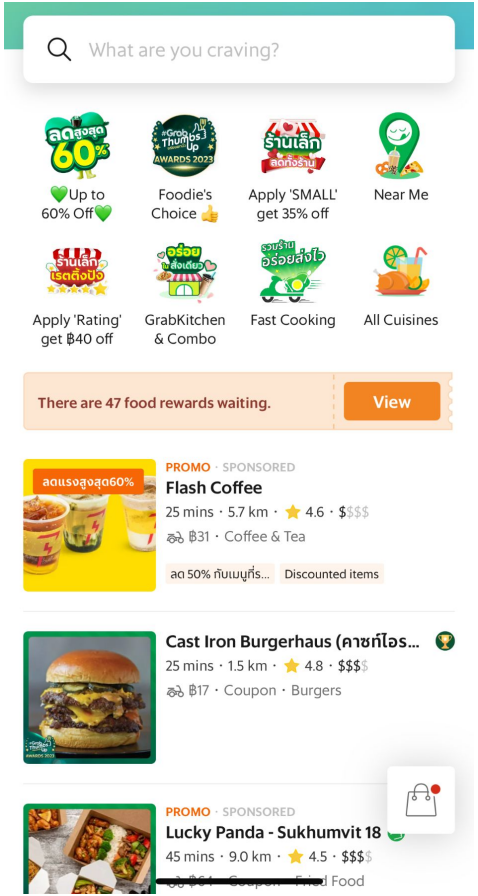


Food
- 200,000 Restaurant
- 20 menus per resto in average
- 4,000,000 SKUs

Users
- 10,000 users during 11.00 - 13.00

Transaction
- 5 million request per seconds

Use case : Recommendation Services



Food
- 200,000 Restaurant
- 20 menus per resto in average
- 4,000,000 SKUs

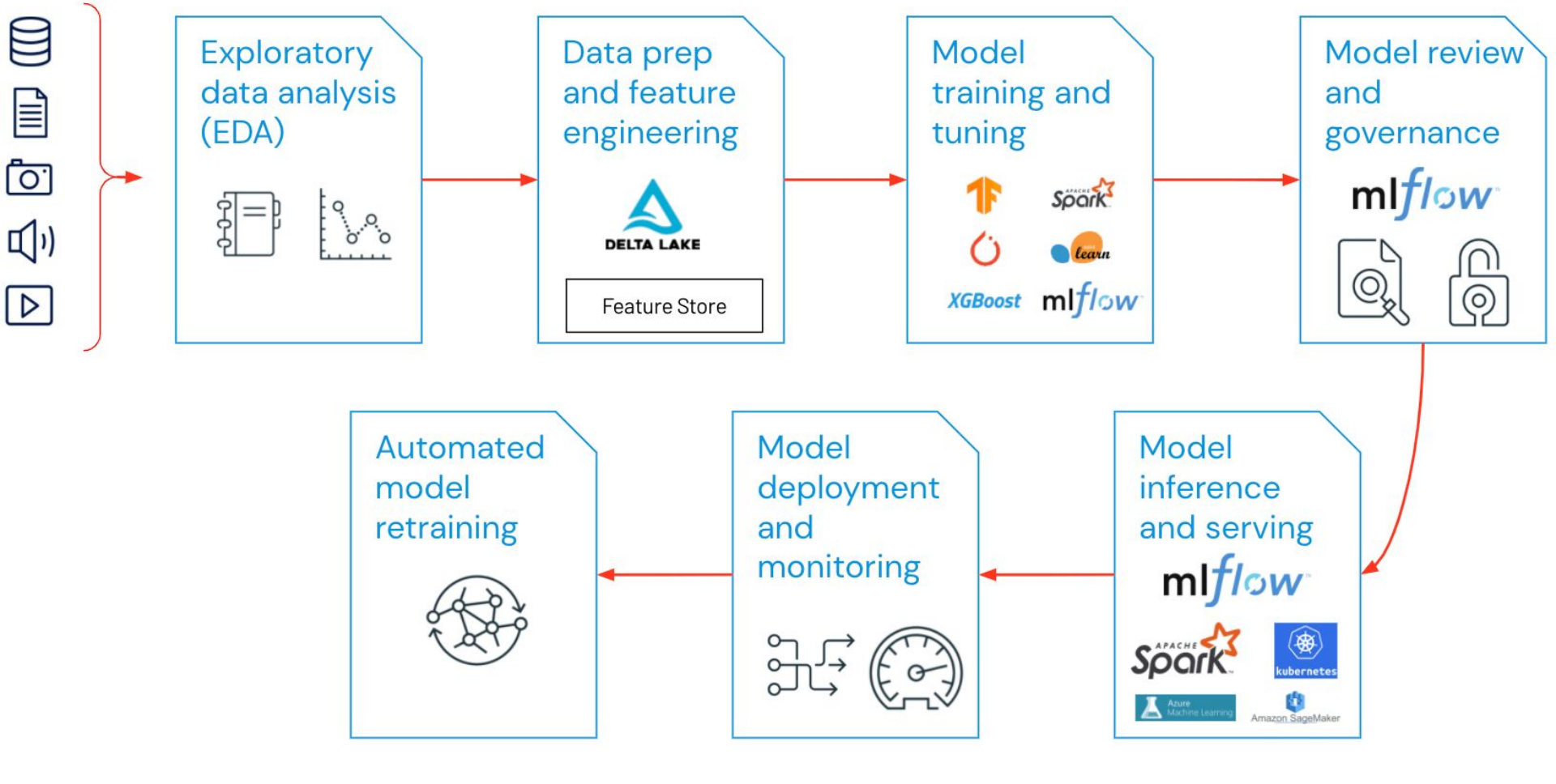
Users
- 10,000 users during 11.00 - 13.00

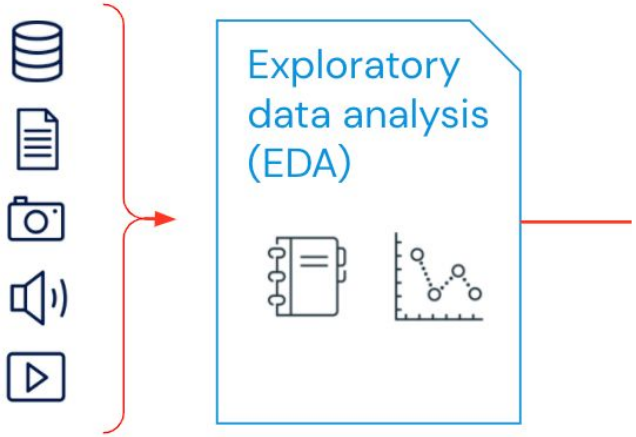
Transaction
- 5 million request per seconds

Models
- > 20 ML models serving

Experiments
- > 1000 A/B experiments in real time

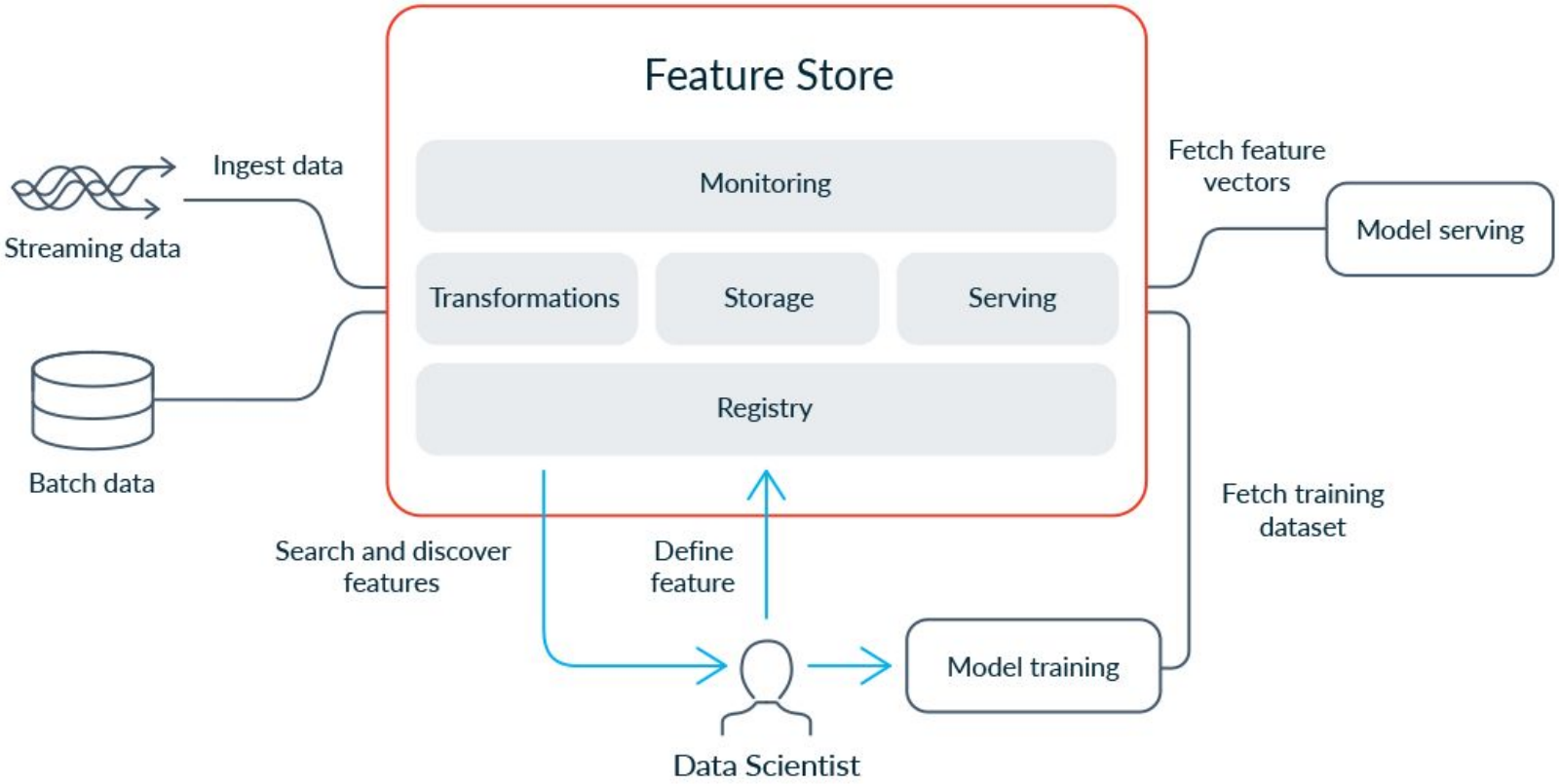
MLOps in ML development life cycle


Exploratory data analysis (EDA)
Data prep and feature engineering
DELTA LAKE
Feature Store
Model training and tuning
Spark
learn
XGBoost
mlflow
Model review and governance
mlflow
Model inference and serving
mlflow
Spark
kubernetes
Azure Machine Learning
Amazon SageMaker
Model deployment and monitoring
Automated model retraining



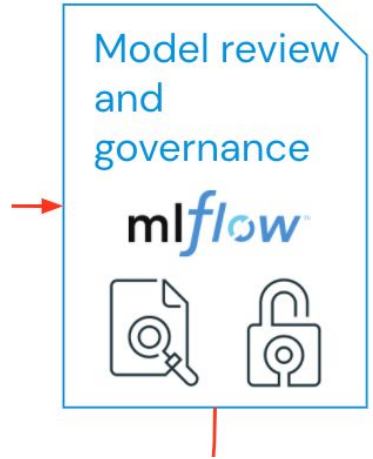
- **Exploratory data analysis (EDA)** - Iteratively explore, share, and prep data for the machine learning lifecycle by creating reproducible, editable, and shareable datasets, tables, and visualizations.

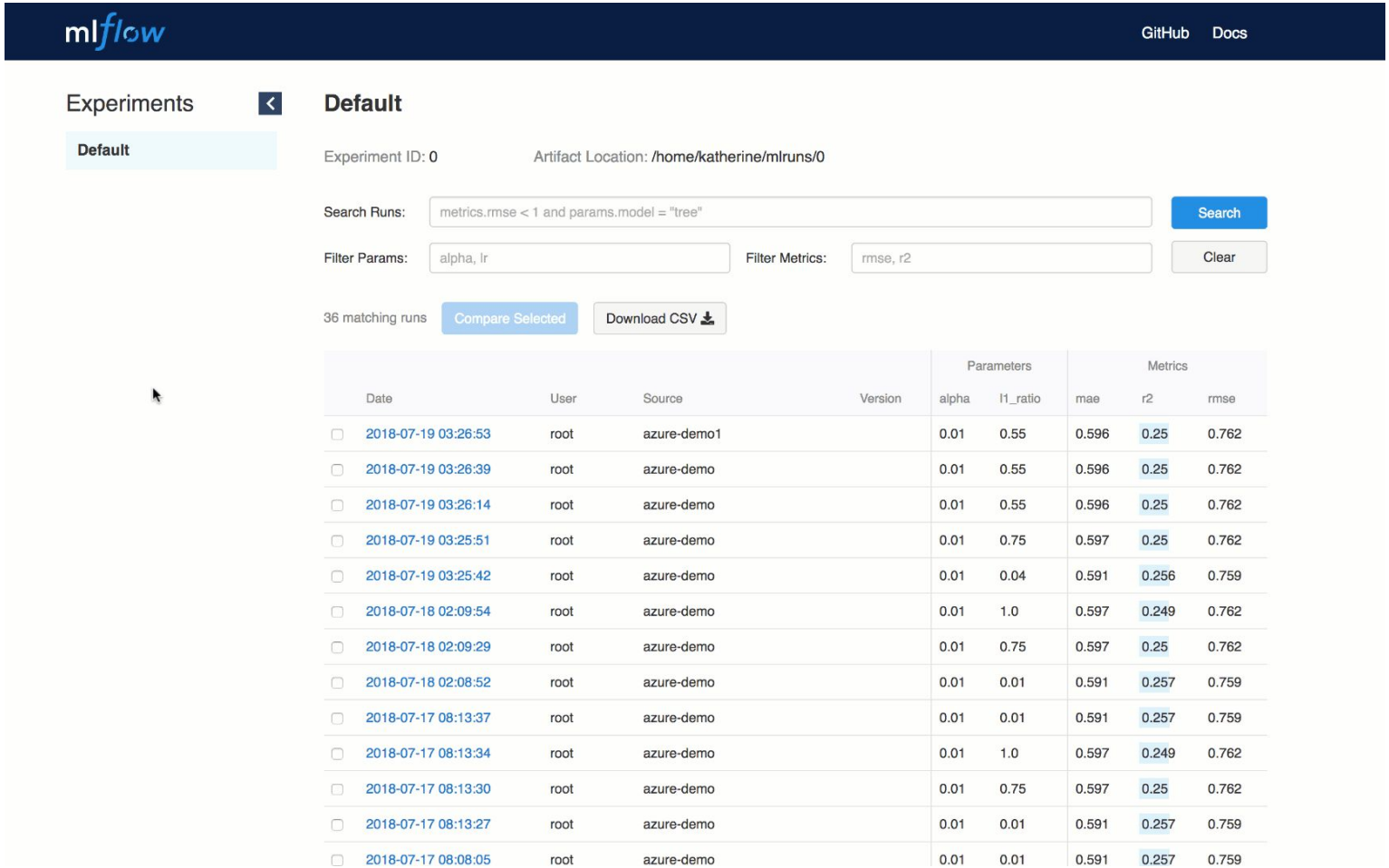
- **Data Prep and Feature Engineering** - Iteratively transform, aggregate, and de-duplicate data to create refined features. Most importantly, make the features visible and shareable across data teams, leveraging a feature store.


Feature Store
Monitoring
Transformations
Storage
Serving
Registry
Ingest data
Streaming data
Batch data
Fetch feature vectors
Model serving
Fetch training dataset
Search and discover features
Define feature
Model training
Data Scientist

- **Model training and tuning** - Use popular open source libraries such as scikit-learn and hyperopt to train and improve model performance. As a simpler alternative, use automated machine learning tools such as AutoML to automatically perform trial runs and create reviewable and deployable code.

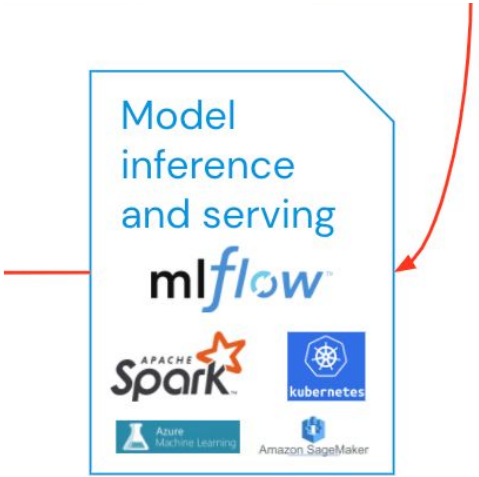
- **Model review and governance** - Track model lineage, model versions, and manage model artifacts and transitions through their lifecycle. Discover, share, and collaborate across ML models with the help of an open source MLOps platform such as MLflow.



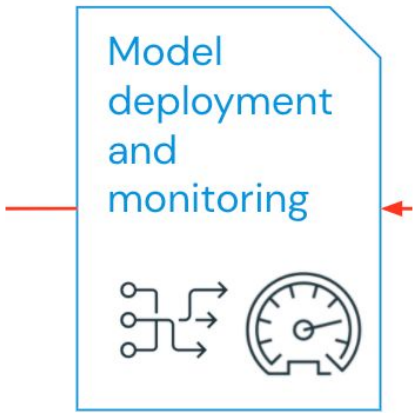


https://databricks-prod-cloudfront.cloud.databricks.com/public/793177bc53e528530b06c78a4fa0e086/0/5769800/100020/latest.html

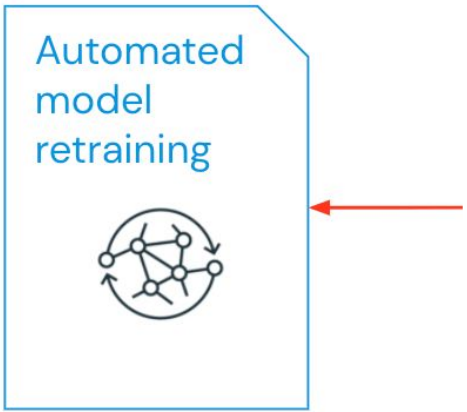
- **Model inference and serving** - Manage the frequency of model refresh, inference request times and similar production-specifics in testing and QA. Use CI/CD tools such as repos and orchestrators (borrowing devops principles) to automate the pre-production pipeline.



- **Model deployment and monitoring** - Automate permissions and cluster creation to productionize registered models. Enable REST API model endpoints.



- **Automated model retraining** - Create alerts and automation to take corrective action In case of model drift due to differences in training and inference data.



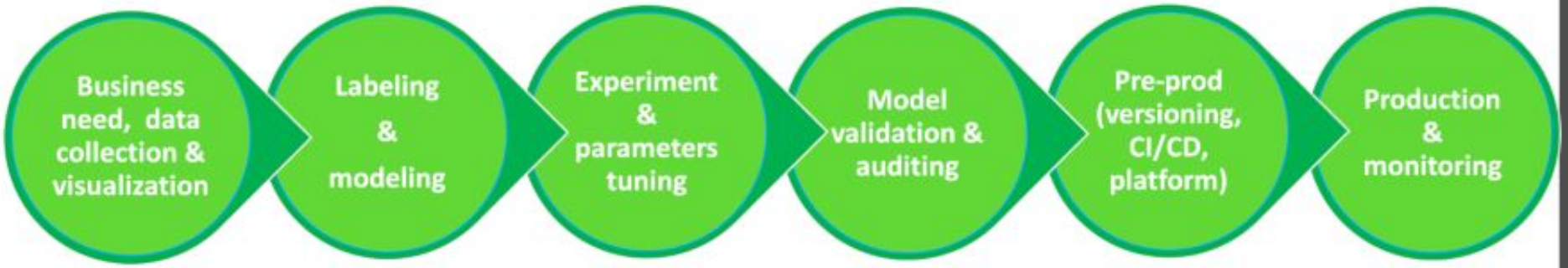
Practices for MLOps

- **Exploratory data analysis (EDA)** - Iteratively explore, share, and prep data for the machine learning lifecycle by creating reproducible, editable, and shareable datasets, tables, and visualizations.
- **Data Prep and Feature Engineering** - Iteratively transform, aggregate, and de-duplicate data to create refined features. Most importantly, make the features visible and shareable across data teams, leveraging a feature store.
- **Model training and tuning** - Use popular open source libraries such as scikit-learn and hyperopt to train and improve model performance. As a simpler alternative, use automated machine learning tools such as AutoML to automatically perform trial runs and create reviewable and deployable code.

Practices for MLOps

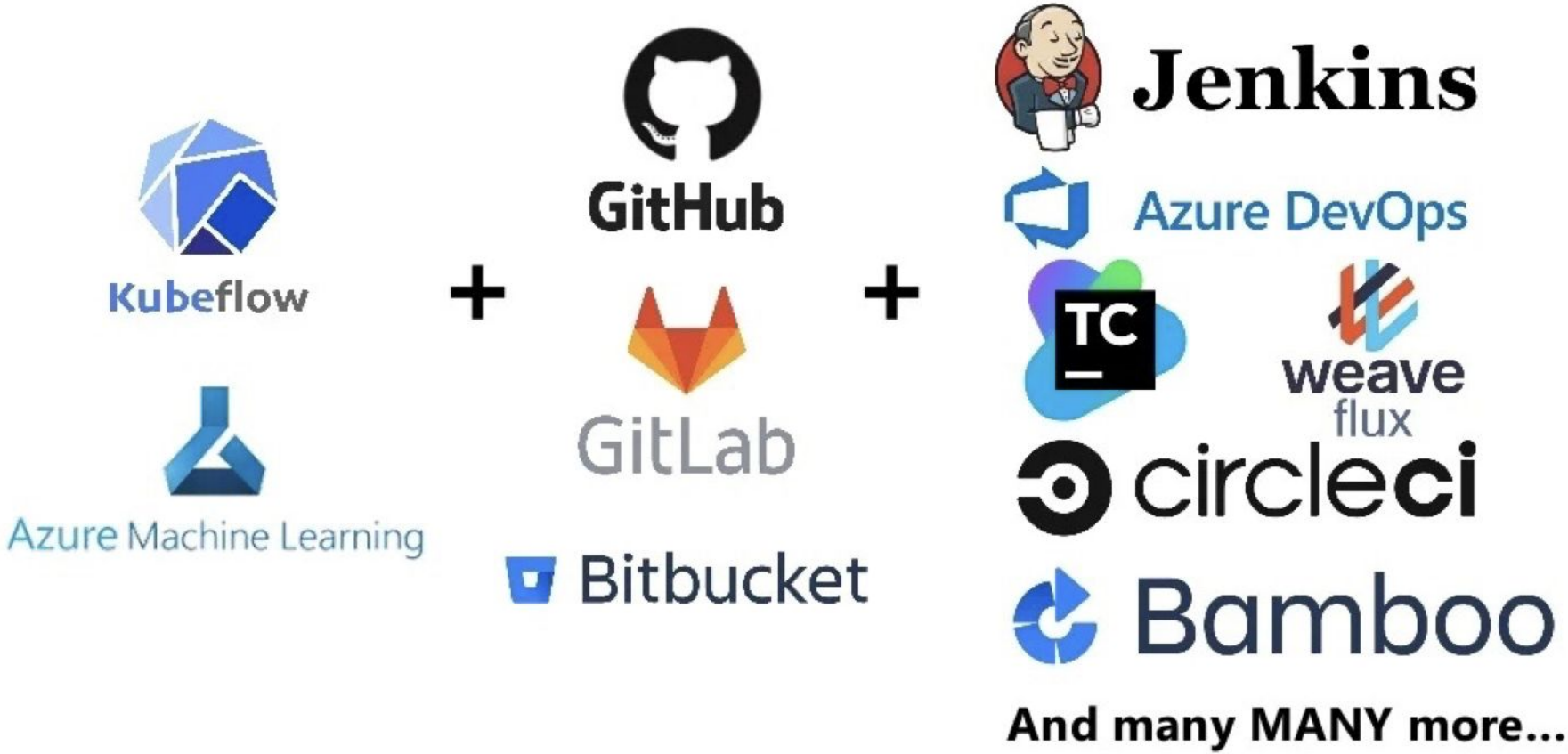
- **Model review and governance** - Track model lineage, model versions, and manage model artifacts and transitions through their lifecycle. Discover, share, and collaborate across ML models with the help of an open source MLOps platform such as MLflow.
- **Model inference and serving** - Manage the frequency of model refresh, inference request times and similar production-specifics in testing and QA. Use CI/CD tools such as repos and orchestrators (borrowing devops principles) to automate the pre-production pipeline.
- **Model deployment and monitoring** - Automate permissions and cluster creation to productionize registered models. Enable REST API model endpoints.
- **Automated model retraining** - Create alerts and automation to take corrective action In case of model drift due to differences in training and inference data.

MLOps Stages @ KBTG

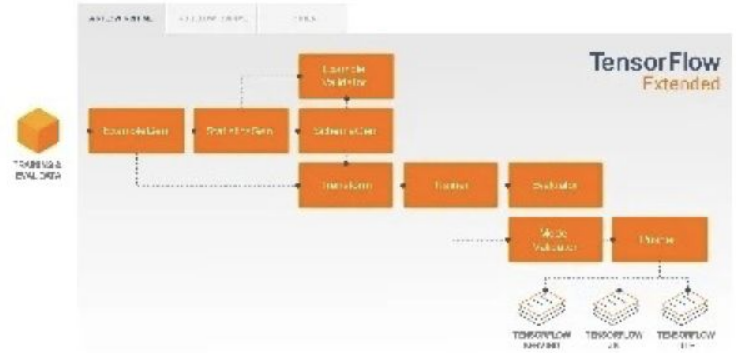


HIVE
hadoop
learn
TensorFlow
AllenNLP
mlflow
kubernetes
elastic
jupyterhub
GitLab
docker
kibana
+ableau
matplotlib
NVIDIA CUDA
H_2O.ai
A/B Testing
sonarqube
OPENSHIFT

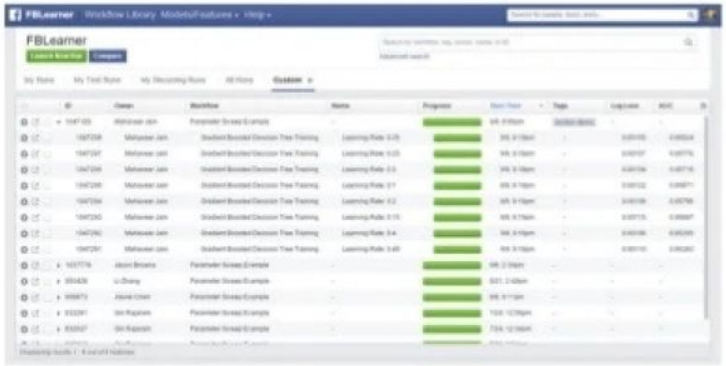
Build Your Own MLOps Platform



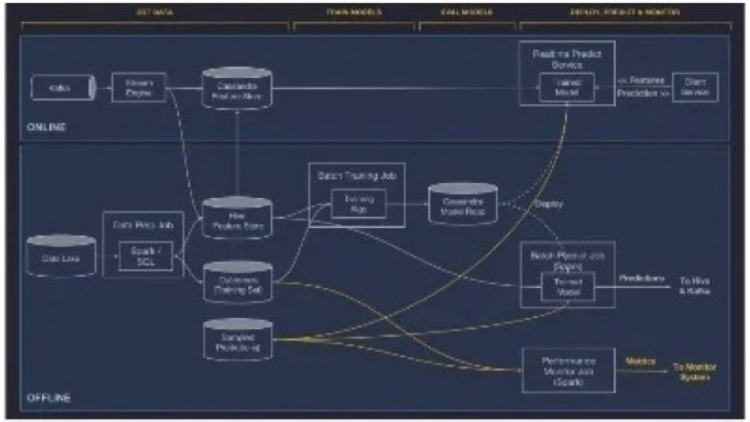
Internal MLOps Platforms



TensorFlow Extended



FBLearner Flow



Uber's Michelangelo

Microsoft Research

Microsoft Aether

When is the right time for MLOps ?

Ok... but WHY?

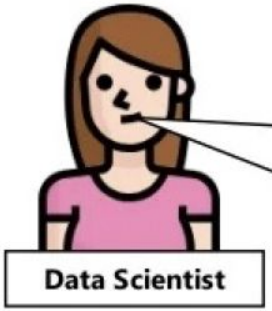
What Does All This Stuff Solve For?

1. Does My Model Actually Work?
2. What Did My Customers See?
3. Is My Model Still Good?

What Does All This Stuff Solve For?

1. Does My Model Actually Work?
2. What Did My Customers See?
3. Is My Model Still Good?

Does My Model Actually Work?



Data Scientist

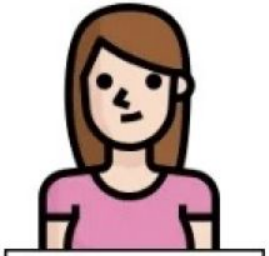
Time to test out my model...

Laptop

SRE/ML Engineers

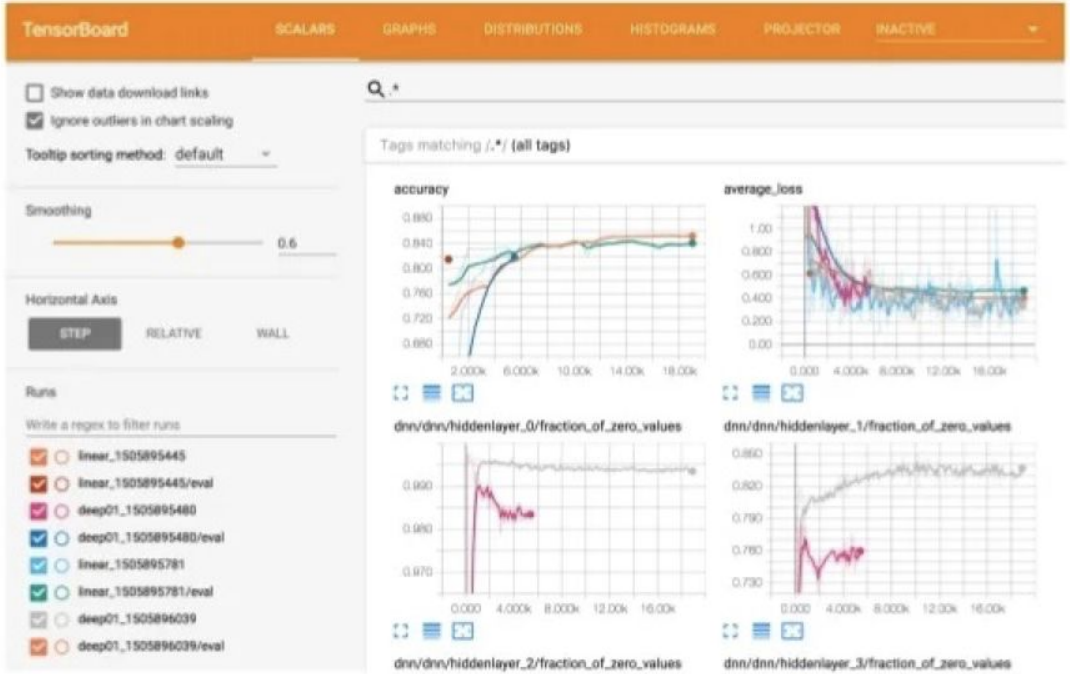
The Cloud

Does My Model Actually Work?



Data Scientist

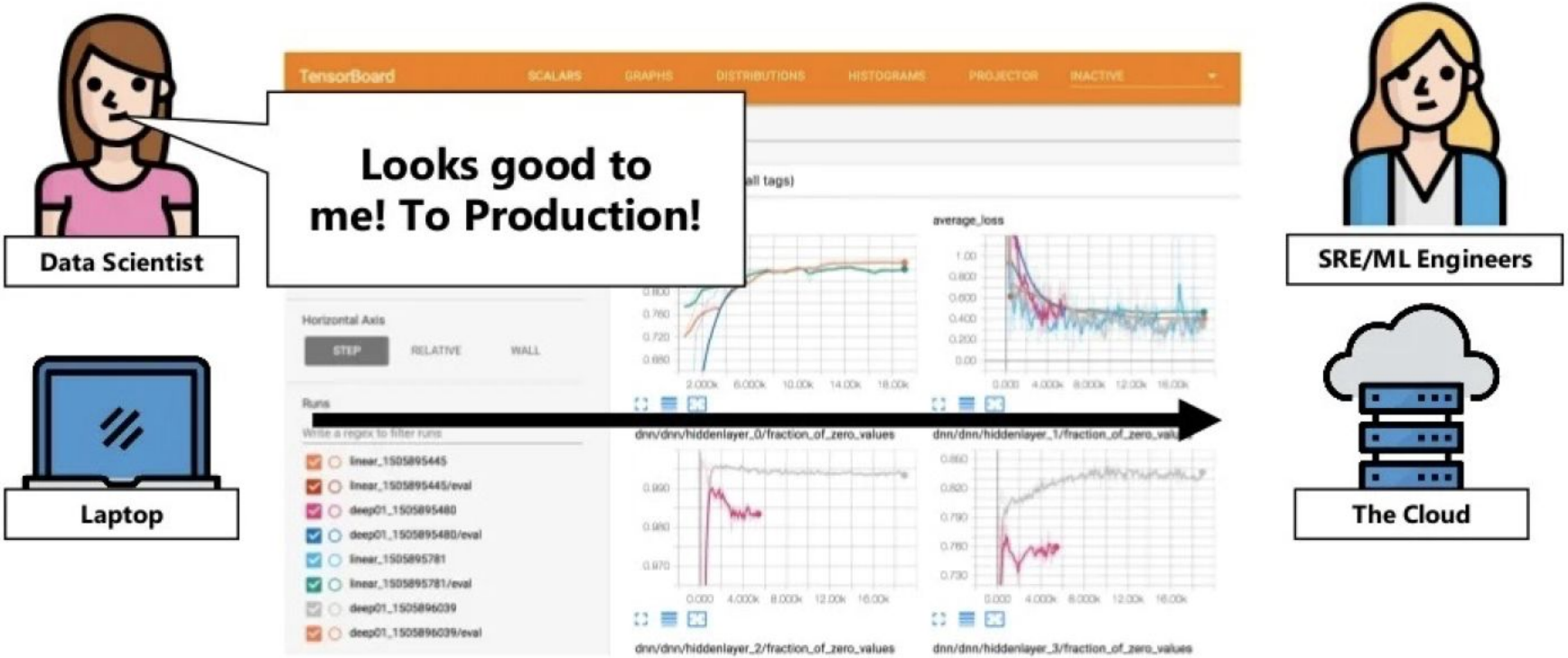
Laptop



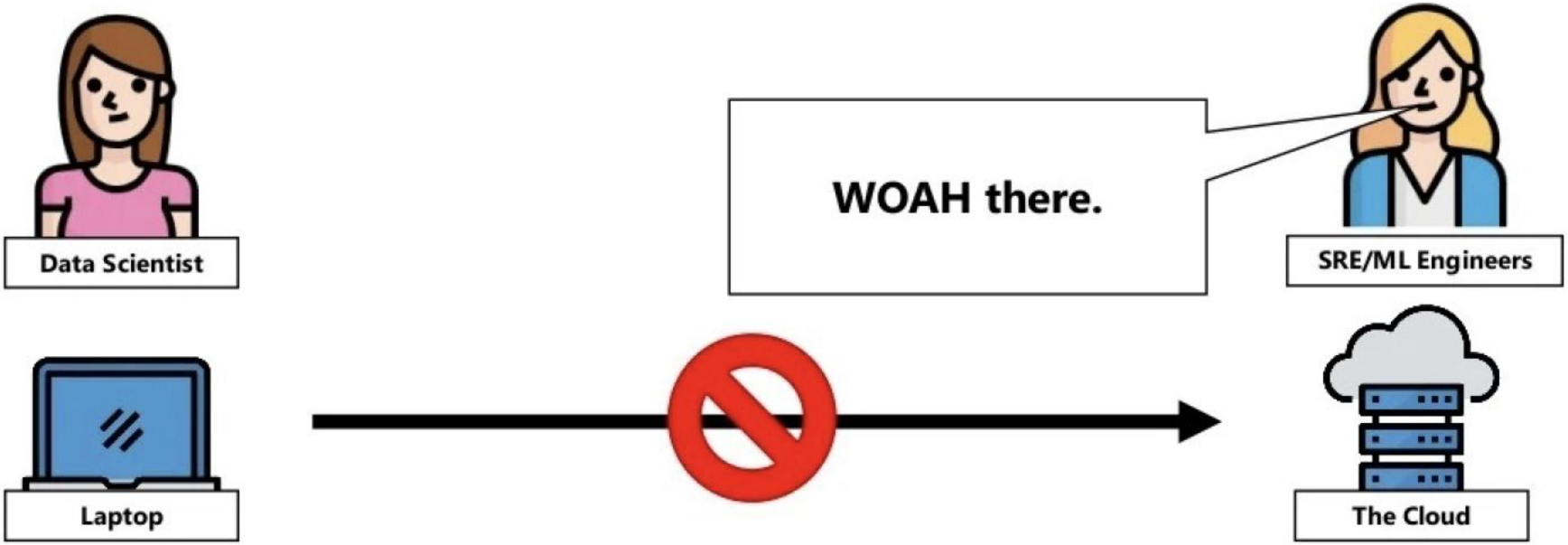
SRE/ML Engineers

The Cloud

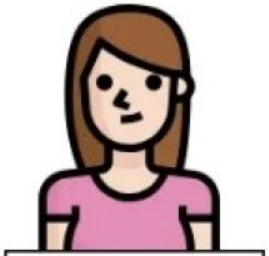
Does My Model Actually Work?



Does My Model Actually Work?



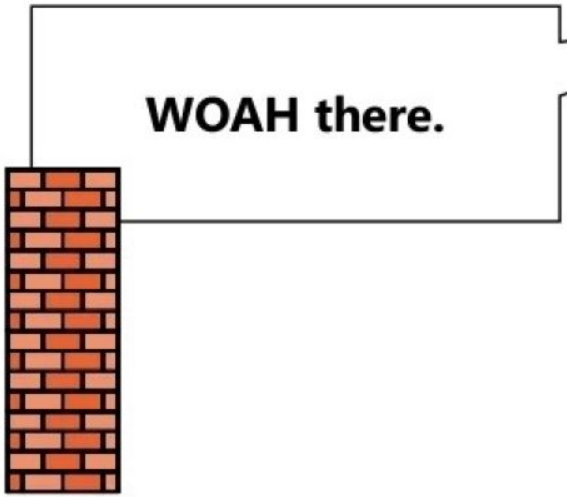
Does My Model Actually Work?

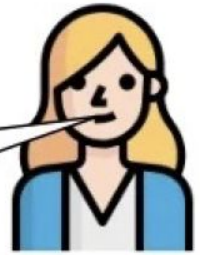


Data Scientist

Laptop

Source Control

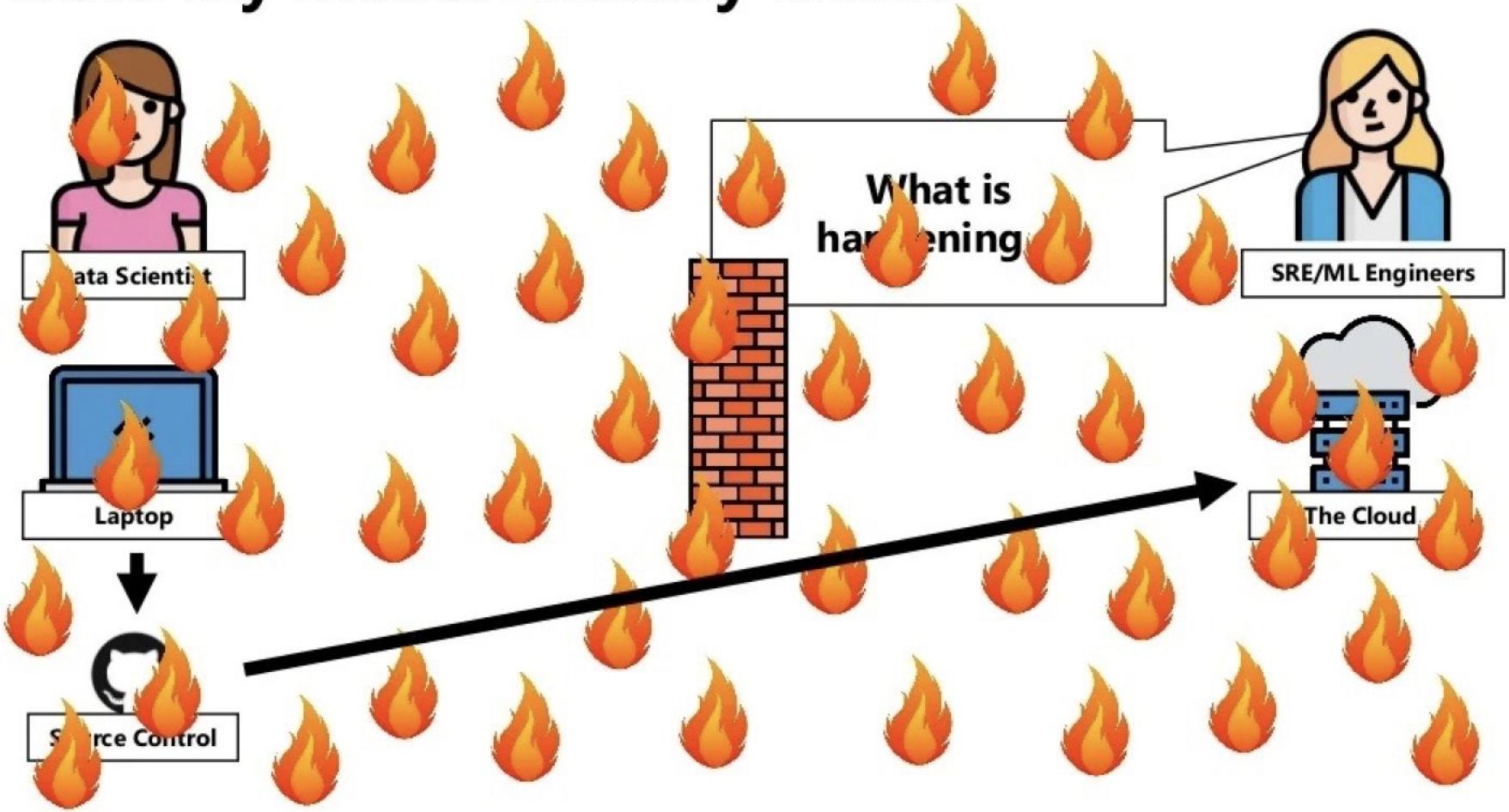




SRE/ML Engineers

The Cloud

Does My Model Actually Work?

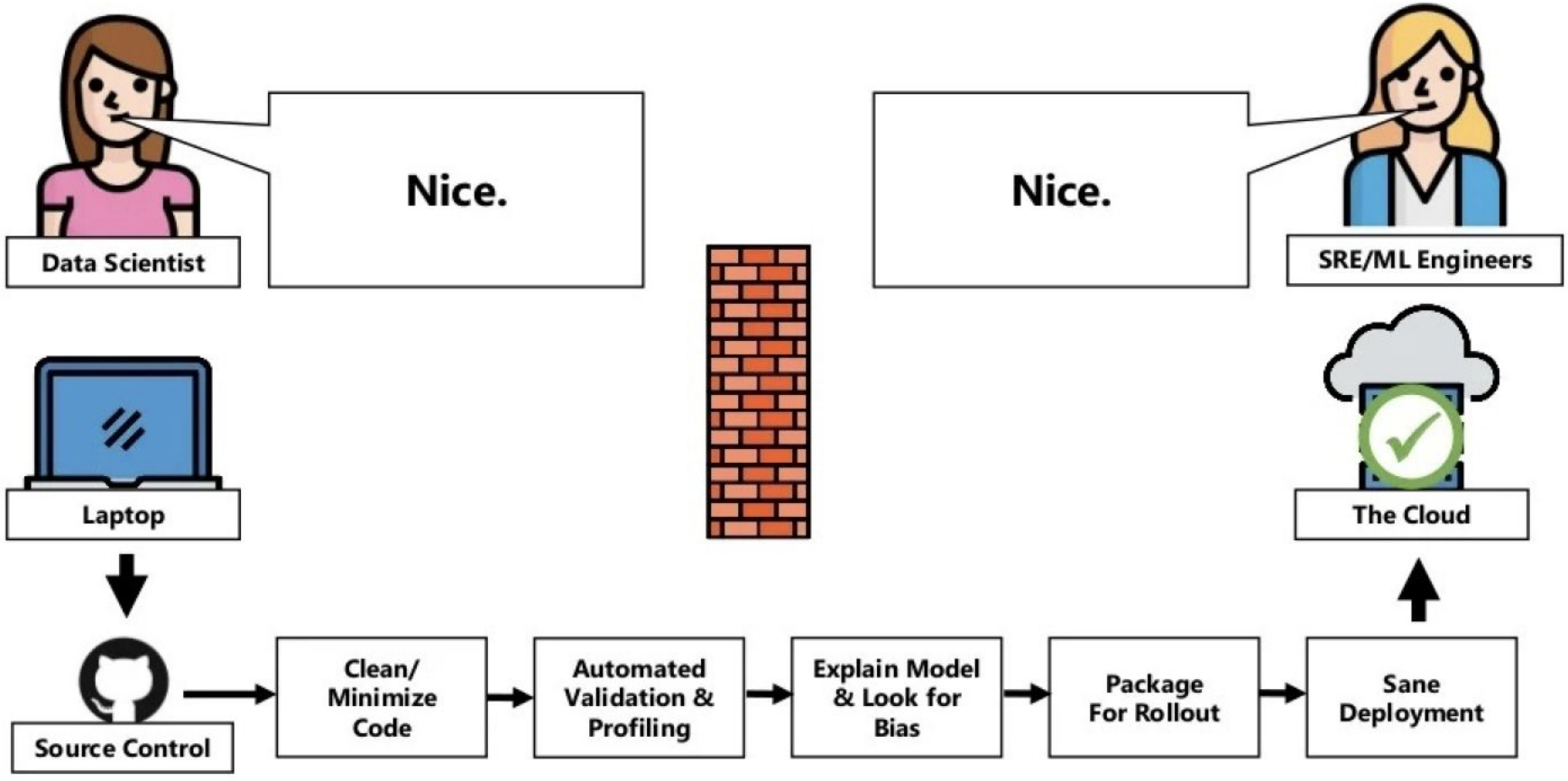


A Small Example of Issues You Can Have…

- Inappropriate HW/SW stack
- Mismatched driver versions
- Crash looping deployment
- Data/model versioning [Nick Walsh]
- Non-standard images/OS version
- Pre-processing code doesn't match production pre-processing
- Production data doesn't match training/test data
- Output of the model doesn't match application expectations
- Hand-coded heuristics better than model [Adam Laiacano]
- Model freshness (train on out-of-date data/input shape changed)
- Test/production statistics/population shape skew
- Overfitting on training/test data
- Bias introduction (or not tested)
- Over/under HW provisioning
- Latency issues
- Permissions/certs
- Failure to obey health checks
- Killed production model before roll out of new/in wrong order
- Thundering herd for new model
- Logging to the wrong location
- Storage for model not allocated properly/accessible by deployment tooling
- Route to artifacts not available for download
- API signature changes not propagated/expected
- Cross-data center latency
- Expected benefit doesn't materialize (e.g. multiple components in the app change simultaneously)
- Get wrong/no traffic because A/B config didn't roll out
- Get too much traffic too soon (expected to canary/exponential roll out)
- Lack of visibility into real-time model behavior (detecting data drift, live data distribution vs train data, etc) [Nick Walsh]
- Outliers not predicted [MikeBSilverman]
- Change was a good change, but didn't communicate with the rest of the team (so you must roll back)
- No dates! (date to measure impact/improvement against a pre-agreed measure; date scheduled to assess data changes) [Mary Branscombe]
- No CI/CD; manual changes untracked [Jon Peck]
- LACK OF DOCUMENTATION!! (the problem, the testing, the solution, lots more) [Terry Christiani]
- Successful model causes pain elsewhere in the organization (e.g. detecting faults previously missed) [Mark Round]

Or It Just Doesn't Work! At All!

Does My Model Actually Work?



But I Can Do All These Manually...

No.

MLOps is a Platform and a Philosophy

Even if:

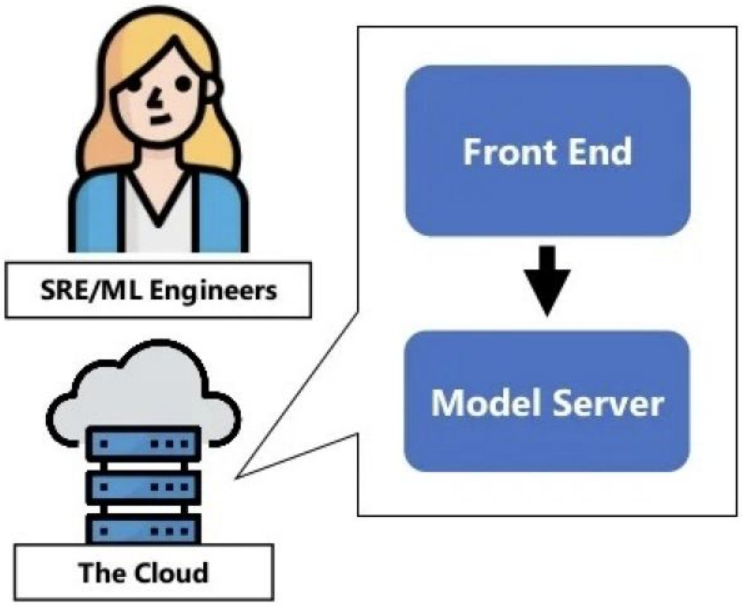
- Every data scientist trained…
- And you had all the tools necessary…
- And they all worked together…
- And your SREs understood ML modeling…
- And and and and …

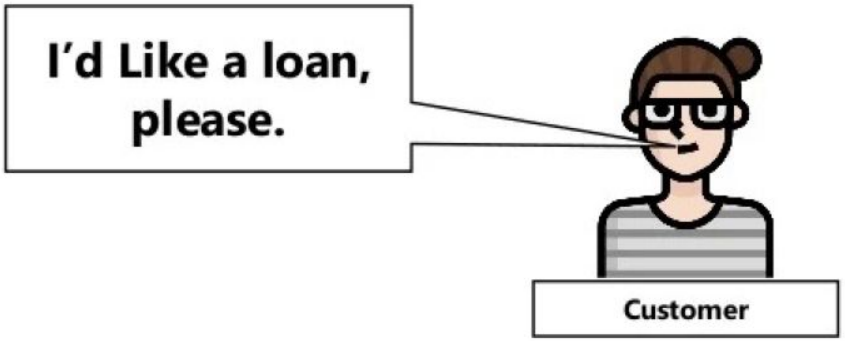
You'd still need a permenant, repeatble record of what you did

What Does All This Stuff Solve For?

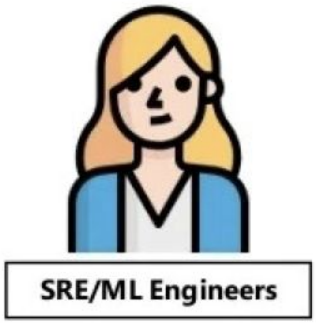
1. Does My Model Actually Work?
2. What Did My Customers See?
3. Is My Model Still Good?

What Did My Customers See?

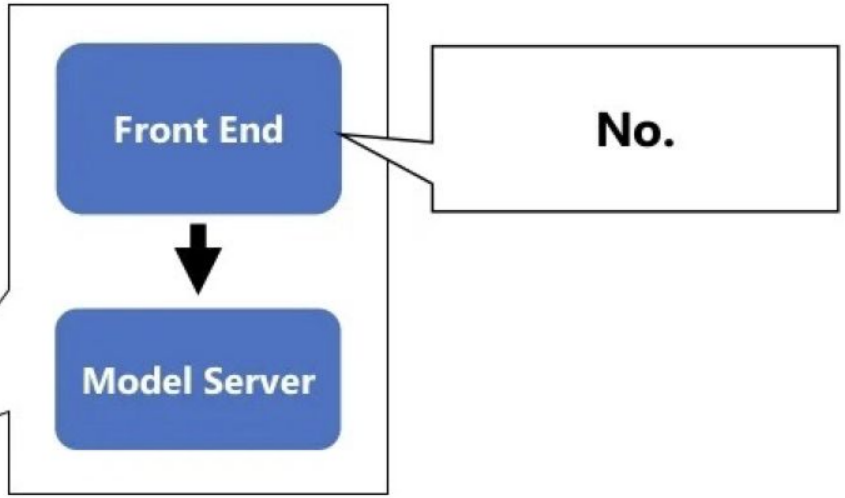




What Did My Customers See?



SRE/ML Engineers

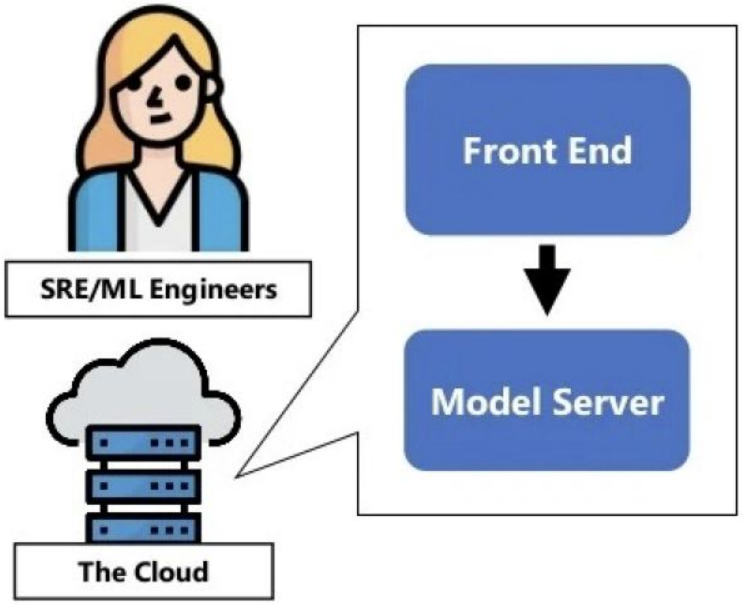


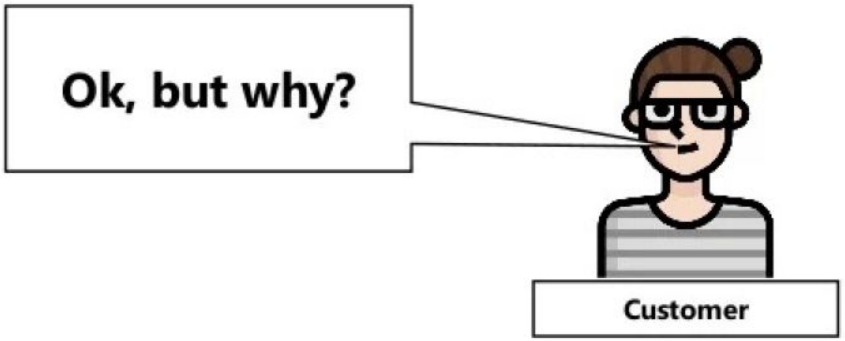
The Cloud

Customer

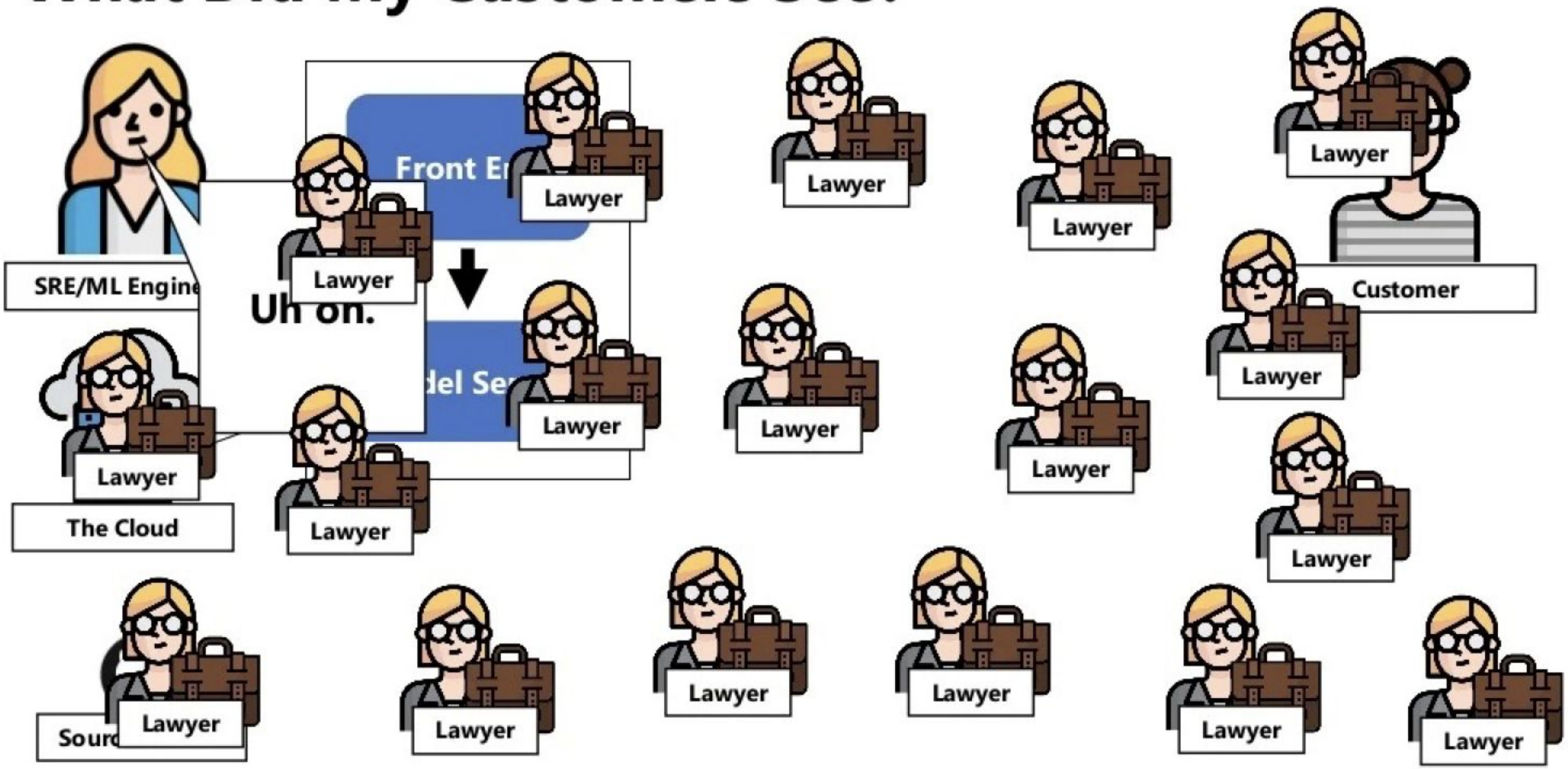
Source Control

What Did My Customers See?





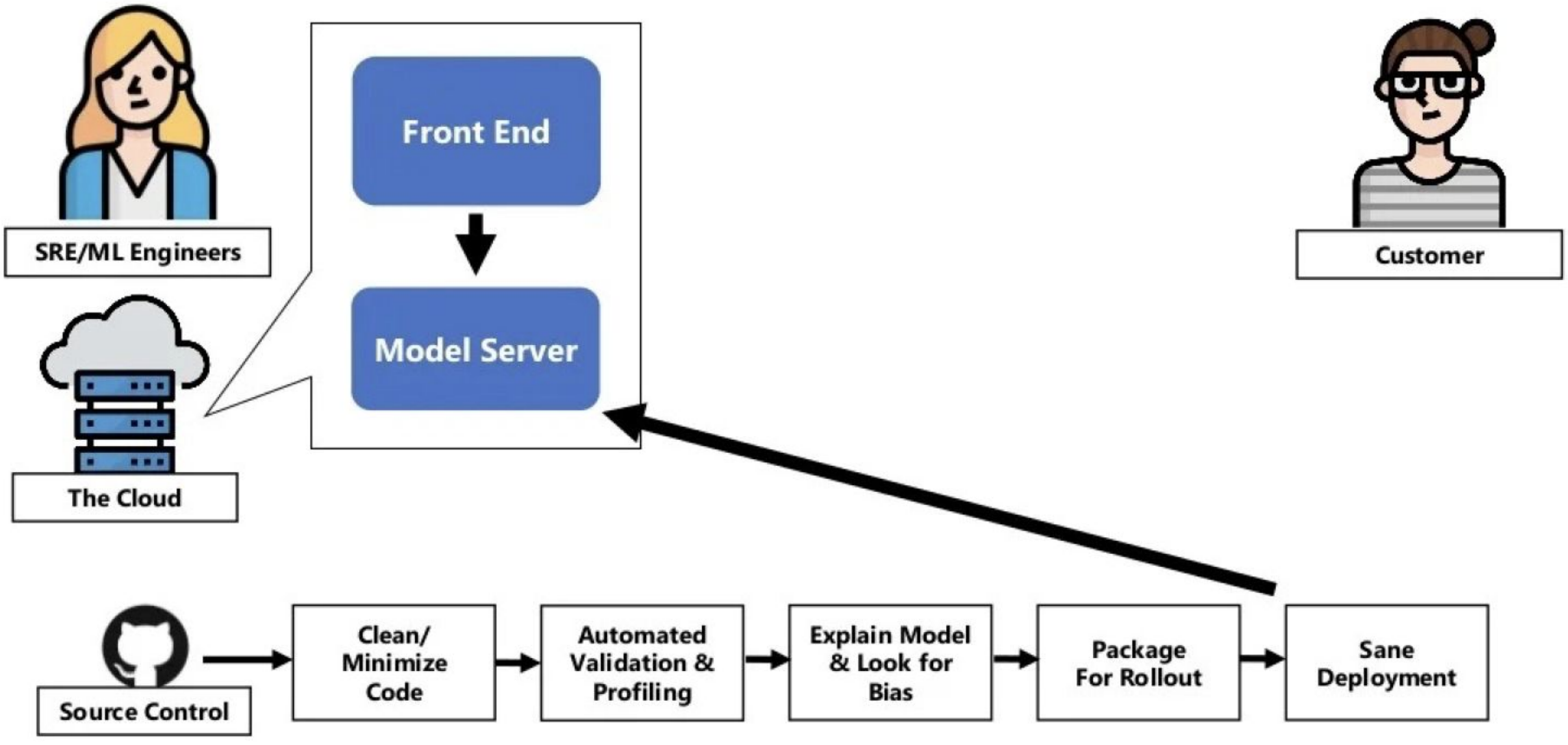
What Did My Customers See?



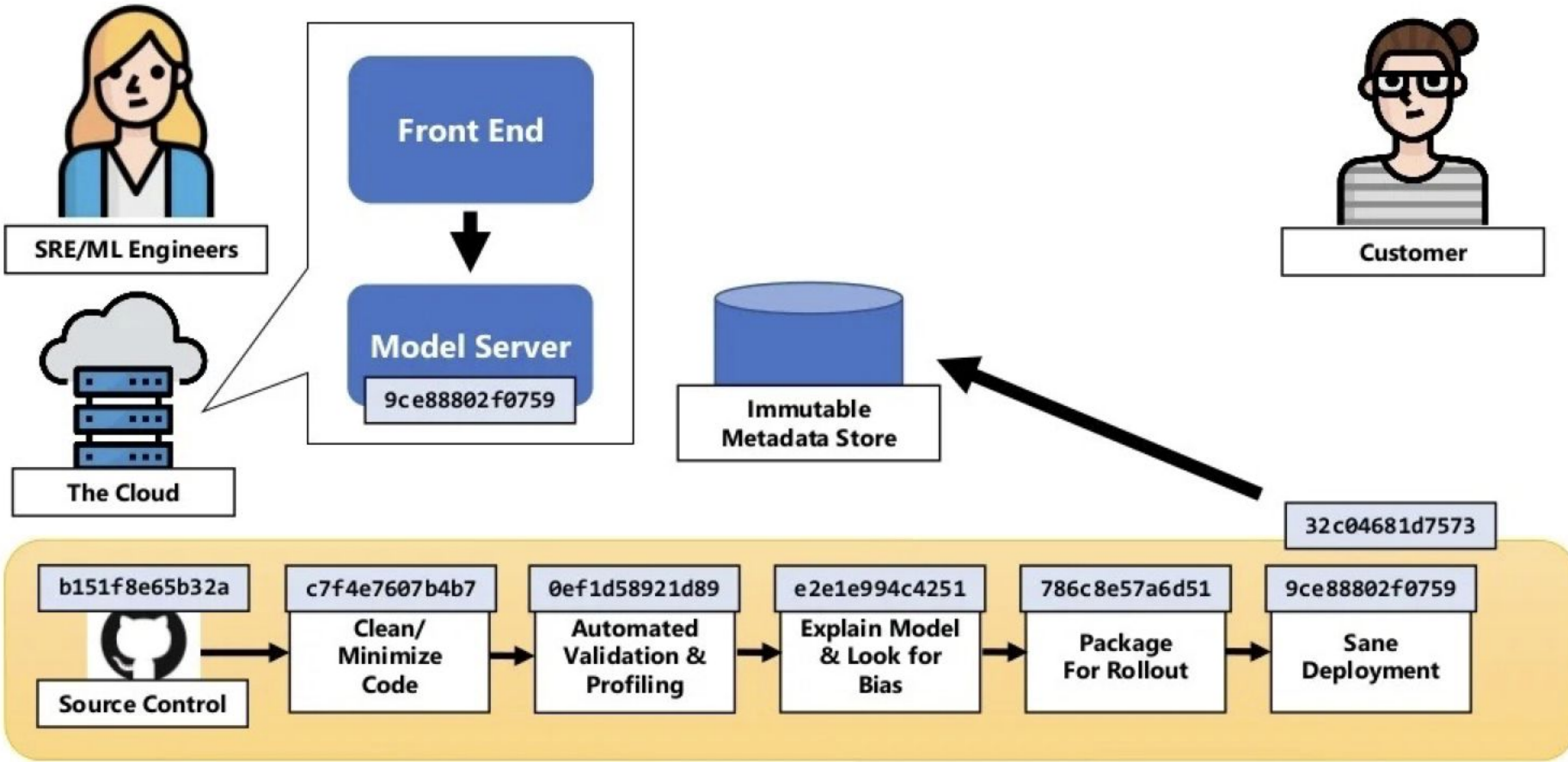
It's Not Just About Explainability!

- Yes, models are complicated
- But, that's not enough:
  - What data did you train on?
  - How did you transform/exclude outliers?
  - What are the data statistics?
  - Did anything change between code and production?
  - What model did you actually serve (to this person)?
- **MLOps can help!**

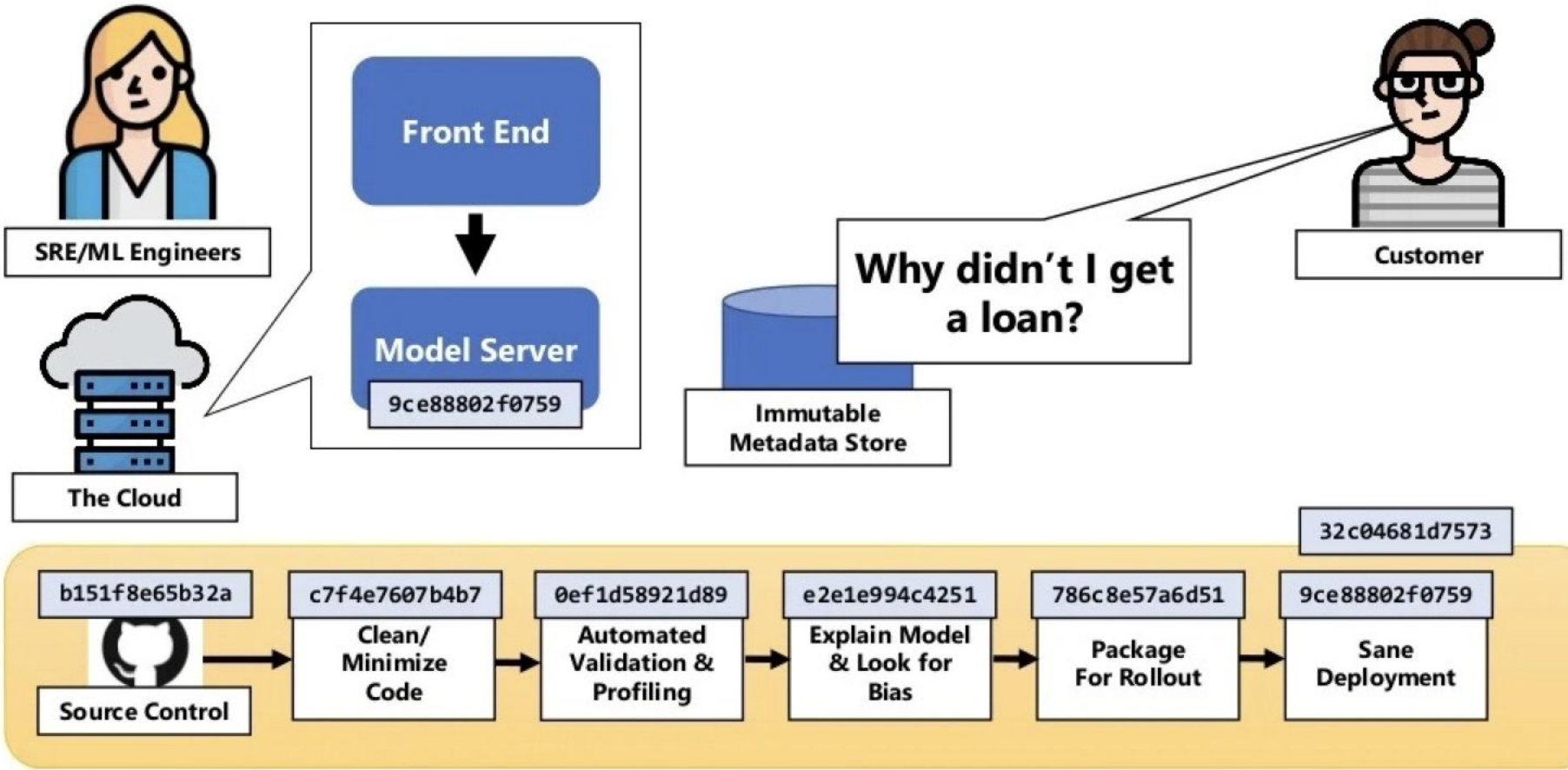
What Did My Customers See?



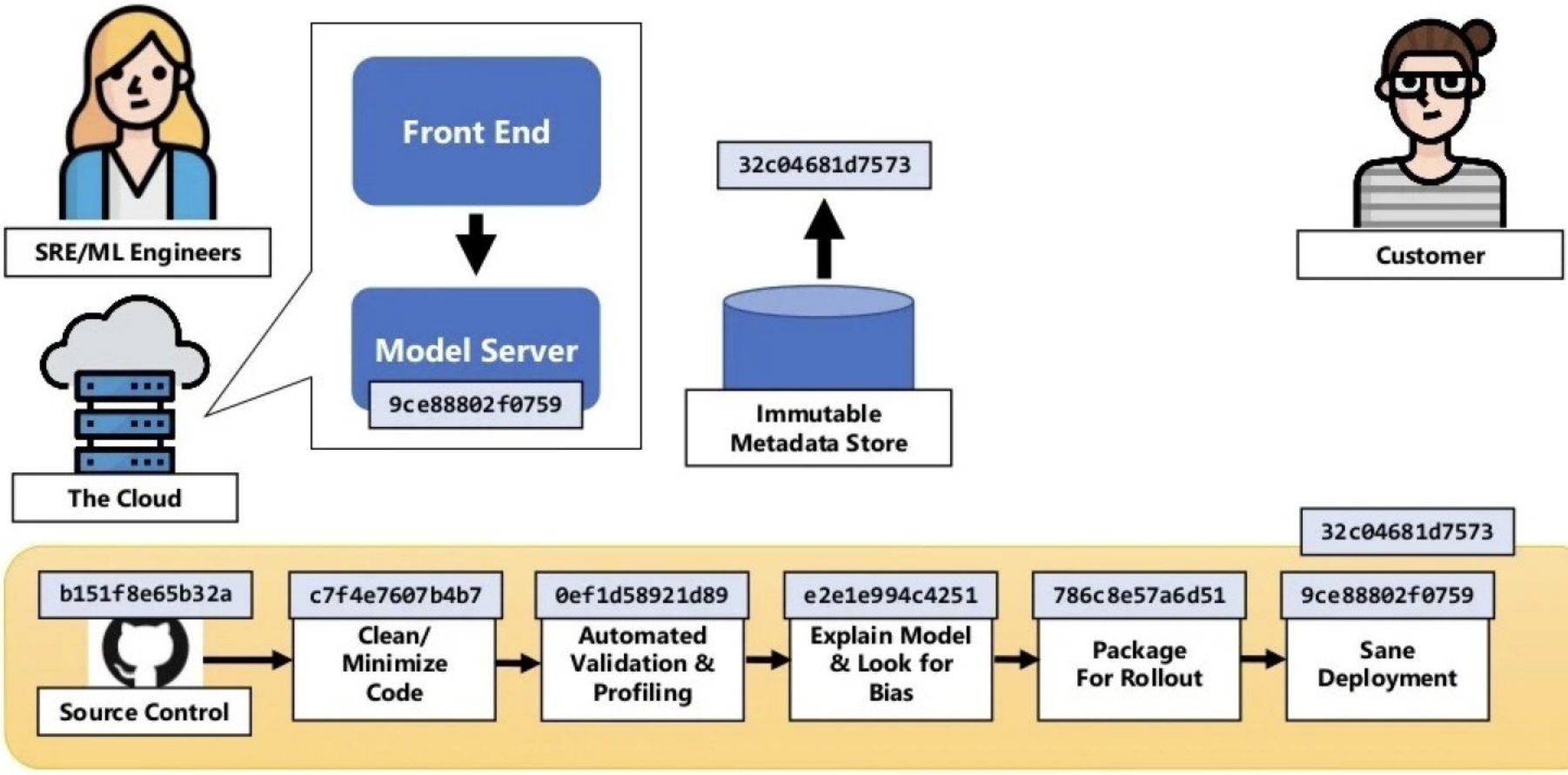
What Did My Customers See?



What Did My Customers See?



What Did My Customers See?

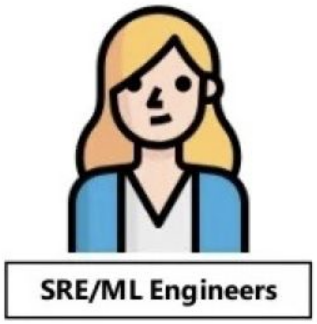


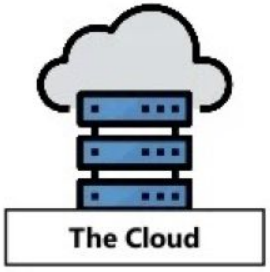
What Does All This Stuff Solve For?

1. Does My Model Actually Work?
2. What Did My Customers See?
3. Is My Model Still Good?

Is My Model *Still* Good?

Is My Model *Still* Good?



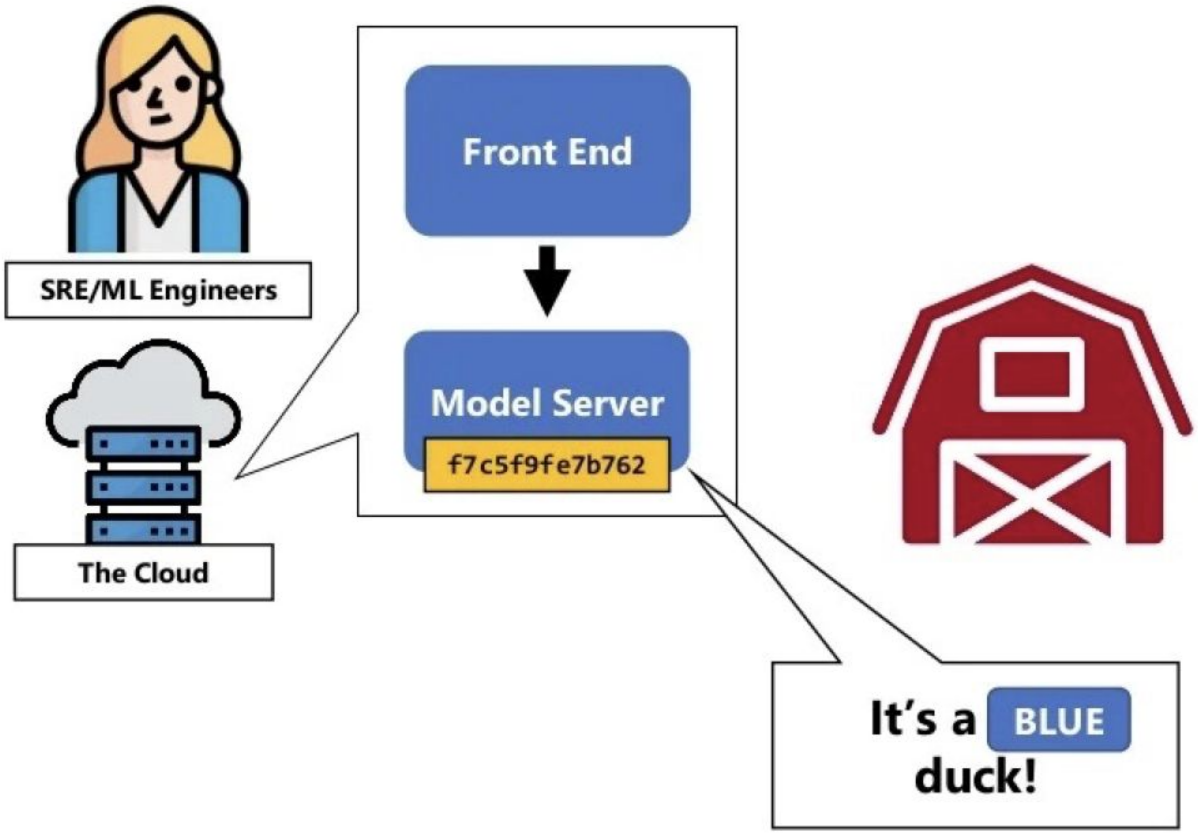


There is a blue or orange DUCK inside this barn.

What color is the duck?

Let’s Use Machine Learning!!

Is My Model *Still* Good?

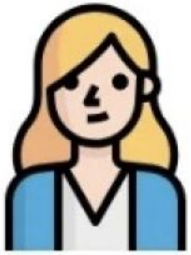


There is a blue or orange DUCK inside this barn.

What color is the duck?

But wait...

Is My Model *Still* Good?



SRE/ML Engineers

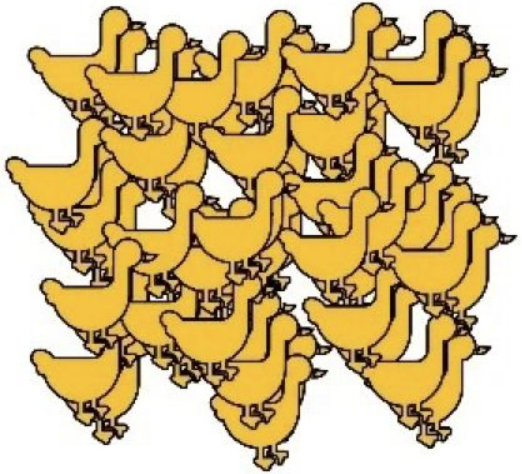
The Cloud

Front End

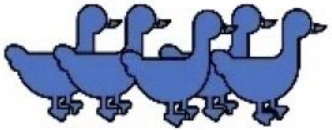
Model Server

`f7c5f9fe7b762`

It's a BLUE duck!



995 Yellow Ducks



5 Blue Ducks

Accuracy = 99%
False Positive = 1%

??????????????????

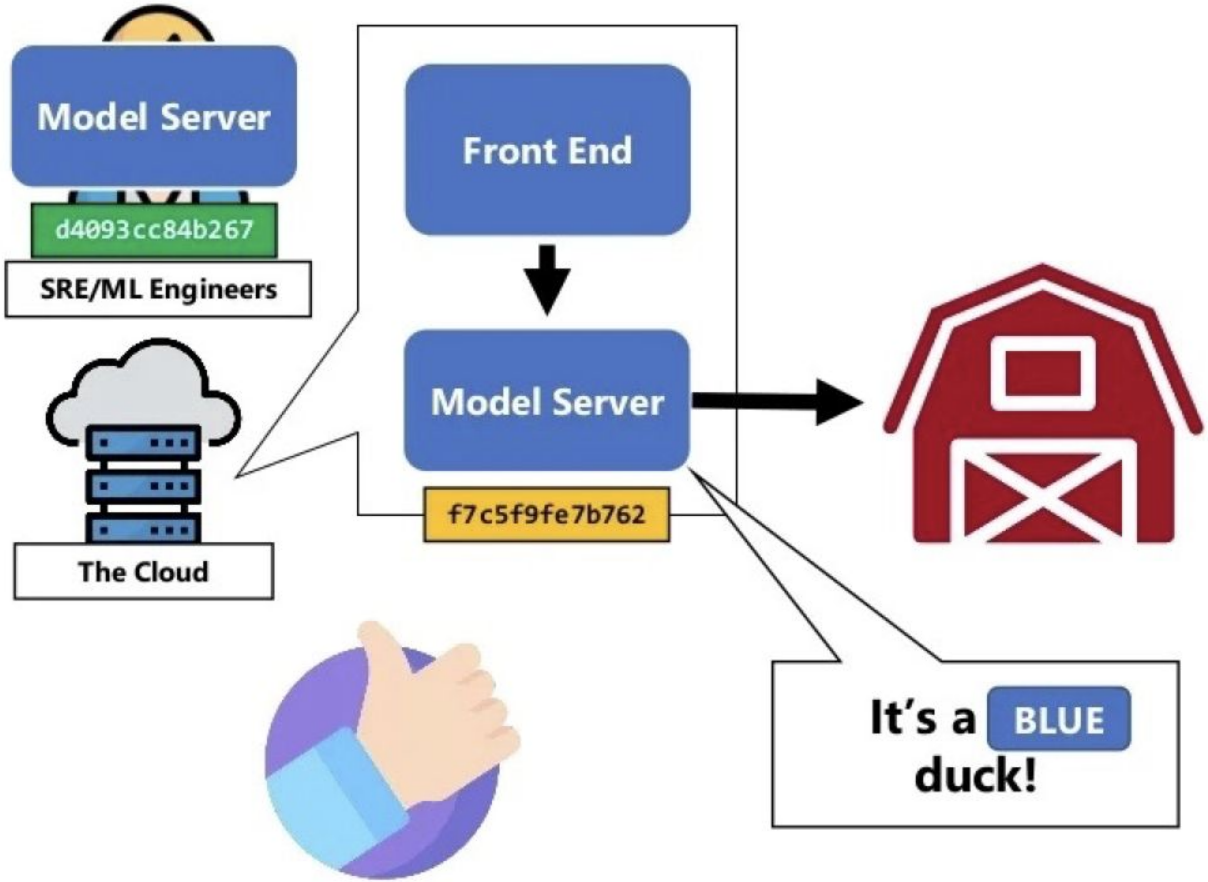
Thomas Bayes

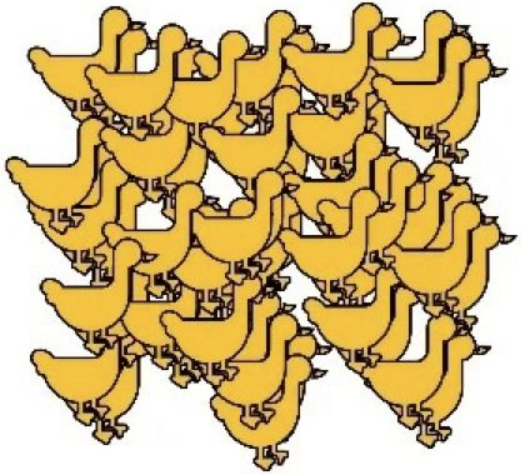
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Bayes' Theorem

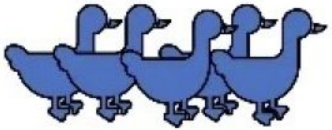
Accuracy depends on the population distribution!

Is My Model *Still* Good?





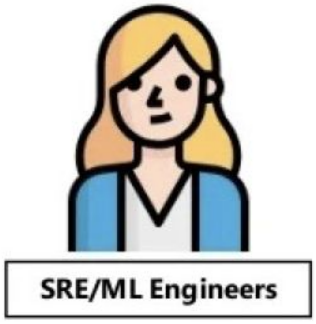
995 Yellow Ducks



5 Blue Ducks

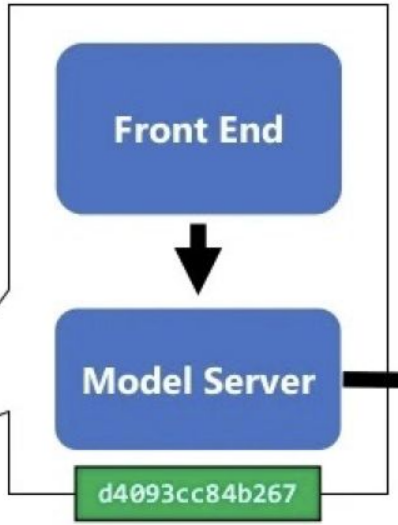
But...

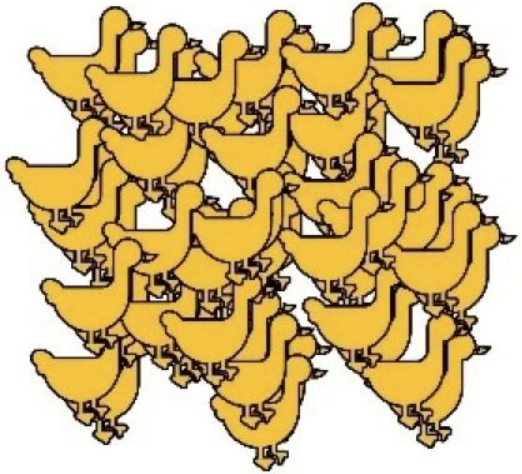
Is My Model *Still* Good?



SRE/ML Engineers

The Cloud

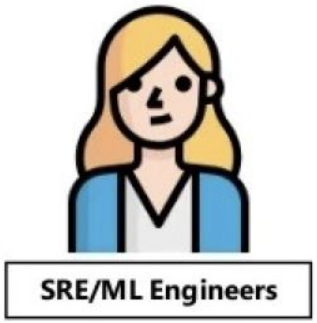




995 Yellow Ducks

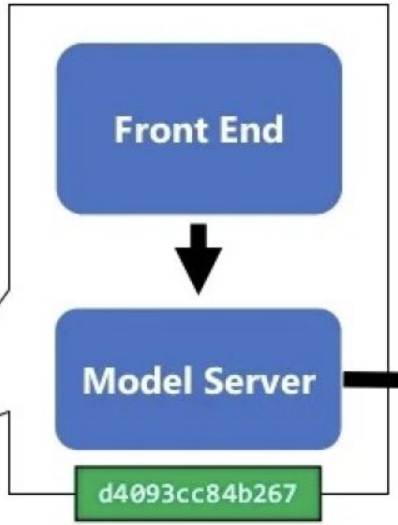
5 Blue Ducks

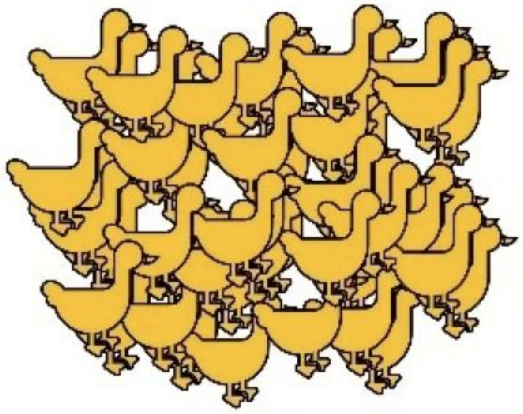
Is My Model *Still* Good?



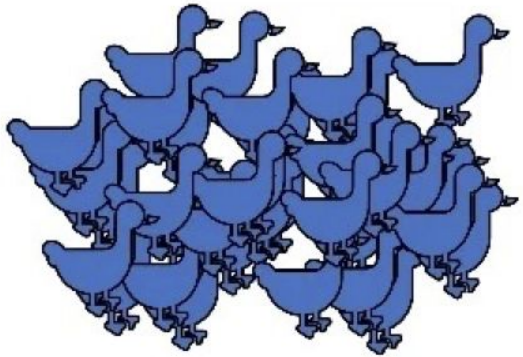
SRE/ML Engineers

The Cloud





500 Yellow Ducks



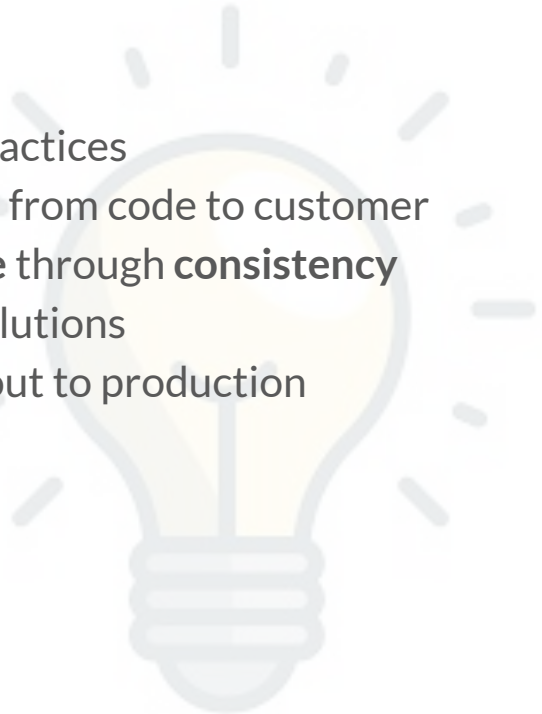
500 Blue Ducks

Is My Model *Still* Good?

- Models != Code – they can go stale… **QUICKLY**.
- IMPORTANT:
  - Watch your model & data for drift from training
  - Regularly (if not continuously) retrain, even before performance begins to fail
  - Multiple versions rollbacks are not uncommon!
- Without an e2e MLOps pipeline, many of the above are O(really really hard)!

MLOps Gives You...

- **End-to-end ownership** by data teams using SWE best practices
- **Continuously deliver of value** to end users, acceleration from code to customer
- Enables **lineage, auditability and regulatory compliance** through **consistency**
- Software best practices for building machine learning solutions
- Repeatable workflow for training a model and rolling it out to production
- An immutable record of what's actually running
- Lineage of model creation including data source.



My Lesson Learned & Ground Rules

- Always understand your data
- Right tools for the right time
- Collaboration is a key success
- Distributed computing, FTW!

THANK YOU

Me: Arnon Jirakittayakorn (arnon.jir@gmail.com)

Facebook: Arnon Jirakittayakorn