MLOps for Machine Vision.

By CAPTIC

Our learnings distilled for the new generation of ML Engineers

Agenda.

- Introduction
    - MLOps for Machine Vision
    - About Captic
- MLOps: The necessary skills
- MLOps throughout the ML Lifecycle
- Q/A

PS: Break will be at the end since I have to leave at 15:15

Check list...

You know…

✅ what today's talk is about

Introduction.

Introduction.

MLOps for Machine Vision.

The cons of AI Machine Vision.

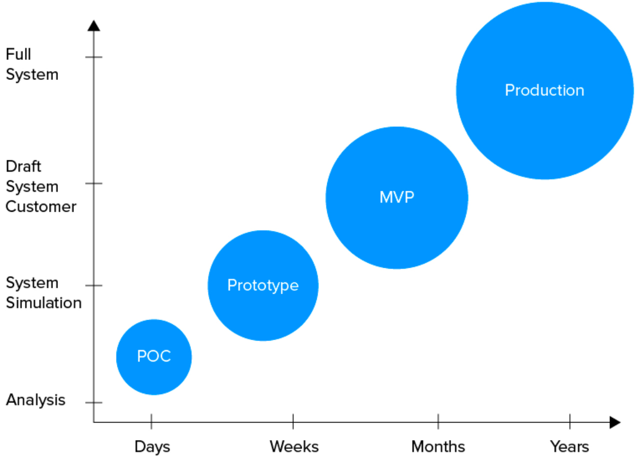
Don’t be fooled by quick notebook demo’s. Building ML solutions require a lot of:

- Knowledge
- Skill
- Ongoing effort

→ Driving up the cost and risk of failure

Many companies fail when deploying and monitoring ML models **in production**. **But only in production is the value created.**

Specific expertise is needed to successfully move beyond a simple PoC.

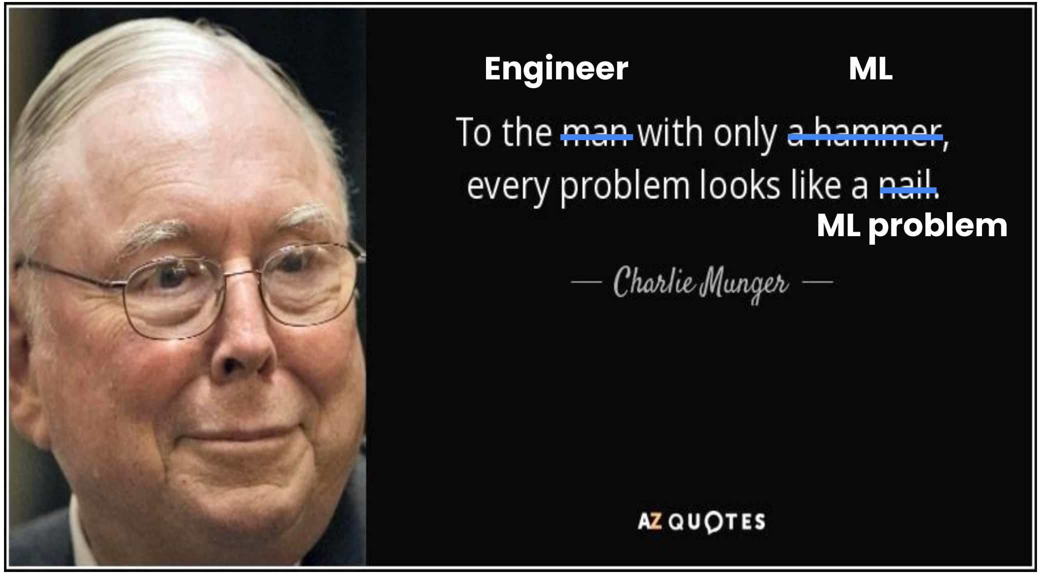


The pros of AI Machine Vision.

	Traditional Vision	AI Vision
Learns and improves over time	❌	✅
Doesn’t require config between known products	❌	✅
Doesn’t require config for changes in surroundings	❌	✅
Able to handle complex (sequences of) tasks	❌	✅

💡 These learnings and models **can be leveraged globally in a uniform way**

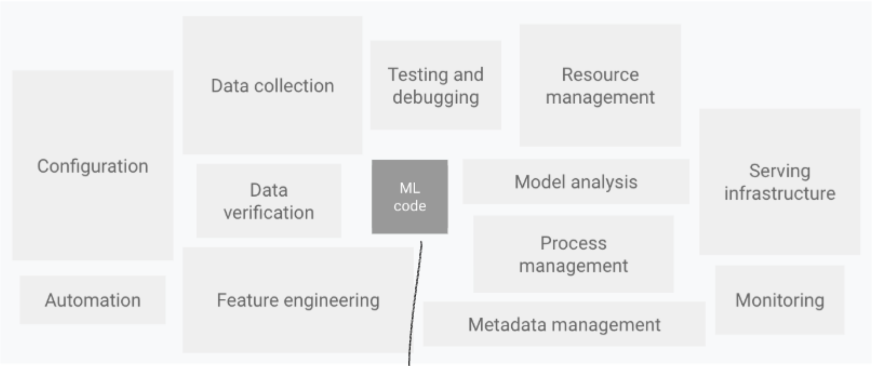
Only use AI when you have to.



ML by itself isn’t enough.

Many companies struggle with deploying and monitoring ML models in production.

This is why MLOps is becoming a critical component of successful ML projects.



Becoming easier everyday

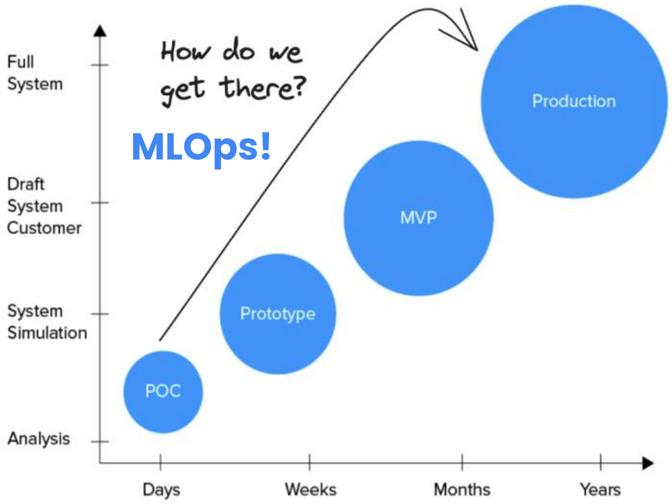
But this is just a Proof of Concept (PoC) by itself

MLOps.

Set of best practices that standardize and streamline ML Systems Lifecycle to make it reproducible, reliable and efficient:

- Modularity
- Containerization
- Versioning
- ...

→ allows you to **get to** - **and stay in** - **production**.



How, What and When?

“MLOps: The necessary skills”

“MLOps throughout the ML Lifecycle”

Check list...

You know…

✅ what today’s talk is about

Check list...

You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important

Introduction.

About Captic.

CAPTIC

The most flexible AI Vision System designed for industrial applications.

Part of SKYHAUS

High-end automation is set for
a transformative leap

Building on a decade of R&D, Captic now delivers the incredible power of AI to industry leaders.

ML6

Pfizer
P&G
BEKAERT
wienerberger
ASML
balta
gsk
HOLCIM

And more...

CAPTIC

Belgian Pork Group
WHAT'S COOKING?
GREENYARD
ANTARCTIC FOODS
WYZO
ELECTROTECHNIEK DECALF
Anna Faggio
BISCUITERIE JULES DESTROOPER BELGIUM SINCE 1886

And more...

Our partners:

Startup guidance

Microsoft

Mondelēz International

Industrial Hardware partner

ADVANTECH

Developer reinforcement

ML6

Captic's experience, flexibility and service are key to success.

Captic's framework

Your production environment.

We select the optimal camera module:

- RGB Area
- 3D
- Line Scan
- Other spectra

We leverage our unified AI core.

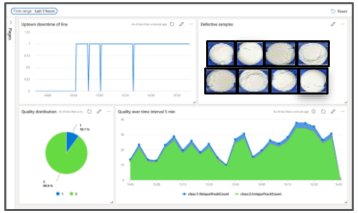
CAPTIC

We configure an automation module:

- PLCs
- Robotics
- Wyzo
- ...

We pick an insights module:

- Dashboards
- Datalake



Hardware monitoring	AI model performance	Remote support	Fleet management		
Device Security	Video Streaming	Data Collection	AI inference	Testing	...

Leverage the incredible power of Captic's AI technology.

Check list...

You know…

- ✅ what today's talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it's important

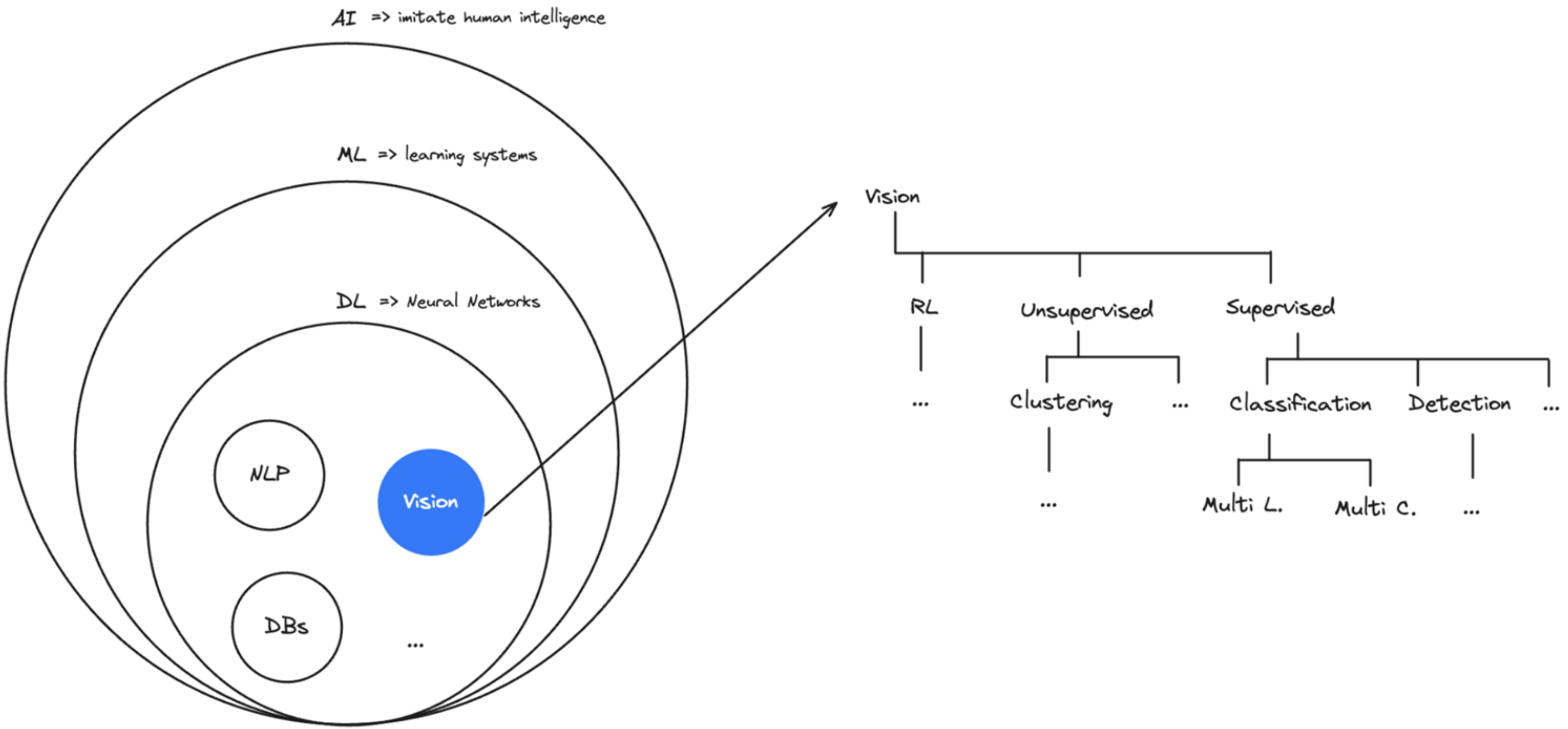
Check list...

You know…

- ✅ what today's talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it's important
- ✅ Captic and what we do

MLOps: The necessary skills.

ML Knowledge.



Python (for MLOps).

Python is the leading language for machine learning and MLOps due to its simplicity and the vast ecosystem of data science libraries. Used to implement ML models, to automate workflows, to script for deployment, …

Key Libraries:

- NumPy & Pandas & OpenCV	Data manipulation
- Matplotlib	isualization
- Tensorflow / Pytorch / Keras	Machine Learning

Best Practices:

- Write clean and readable code
- Use virtual environments (poetry)
- Leverage libraries extensively

Docker.

Open source virtualization platform to build, run and manage containers. Used to ensure reproducibility and consistency across platforms through isolation.

Key Concepts:

- Image	A read-only & self-sufficient package
- Dockerfile	Process of combining two branches
- Registry	Hosts the images
- Container	A running instance of an image
- Volume	A storage unit that allows for data persistence

Reproducible environments for:

- Model development, inference, …
- APIs
- …

Code Versioning (Git).

Version control is essential for managing changes to code, collaborating with others, and maintaining a history of project evolution.

Key Concepts:

- Repository — The directory containing code, files, …
- Branch — A copy (isolated environment) of the proj
- Commit — Adding updates, creating a new snapsho
- Merge — Process of combining two branches
- Pull Request — A proposal for a merge that needs to be reviewed

Best Practices:

- Commit regularly
- Write descriptive commit messages

CI/CD.

Continuous Integration (CI) and Continuous Deployment (CD) are practices that automate the integration and deployment of code changes. Facilitates frequent updates, minimizes integration issues, ensures deployment is systematic and predictable.

How?

YAML definition of steps that are executed during certain stages of the Git

Often used for:

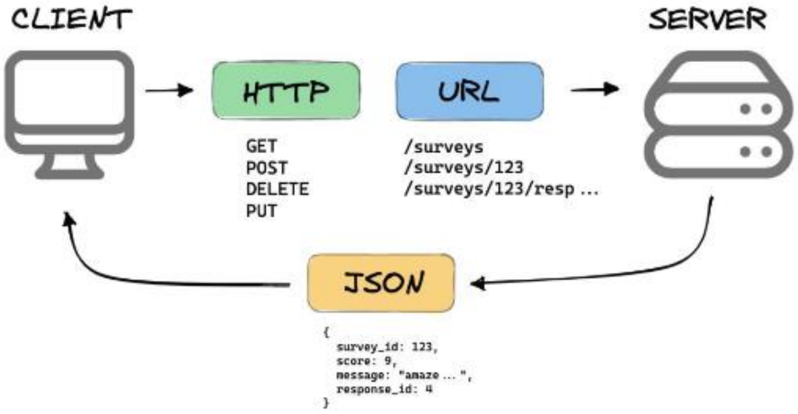
- Linting
- Building docker images
- Testing
- Deployment
- ...

APIs.

An Application Programming Interface is a set of protocols that enable interaction with software in a typical request/response manner.

There are many types of APIs (REST, Websocket, RPC, …) that are different from each other in structure and communication.

REST (Representational State Transfer):



Cloud infrastructure.

Cloud platforms provide scalable, flexible infrastructure for deploying and managing all types of systems (including ML systems). This allows you to scale as needed without limits (aside from budget).

ML = Big Data = Big Compute → Cloud offers specialized hardware at pay-as-you go pricing

Providers: AWS, Azure, GCP, …

Key concepts:

- VM	Virtual Machines
- Storage	Remote storage
- IAM	Identity and Access Management

Tip: Most providers have free courses and credit packs to get you started on their platforms. This allows you to get valuable hands on experience before graduating.

Check list...

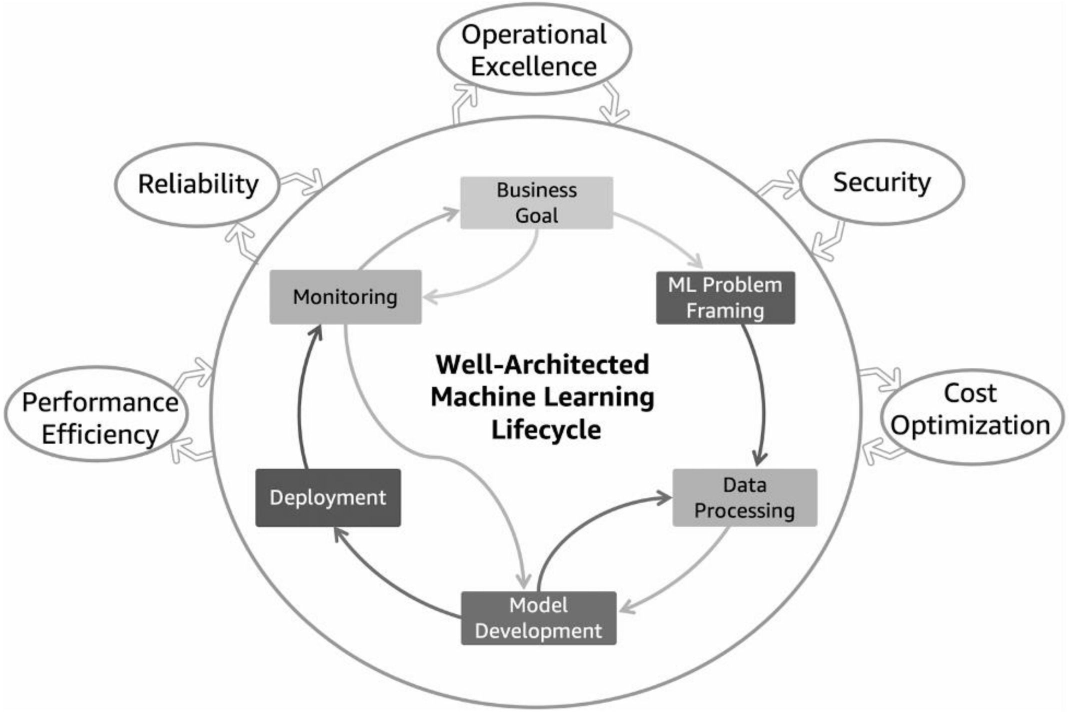
You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks

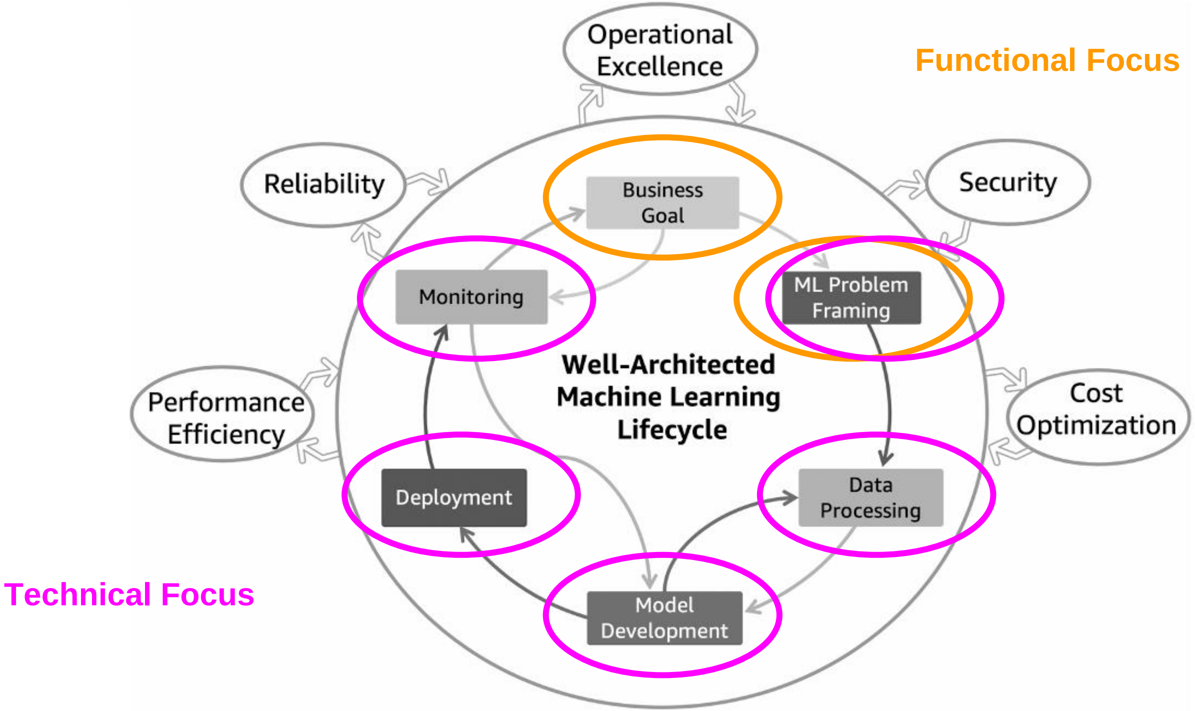
MLOps throughout the ML Lifecycle.

The ML Lifecycle.

The ML-Lifecycle (our day-to-day).



The ML-Lifecycle (our day-to-day).



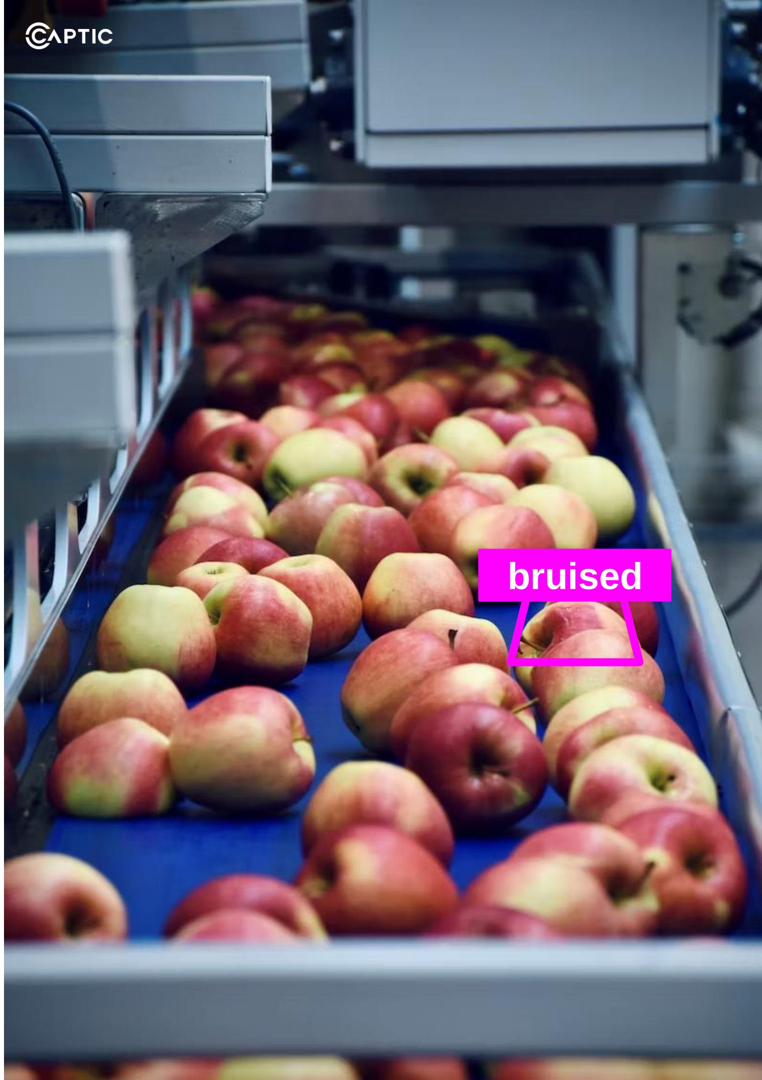
Our example.

Customer:

APPLE BV

Need:

Wants to “know” how much “bad” product are in their suppliers’ deliveries.



Check list...

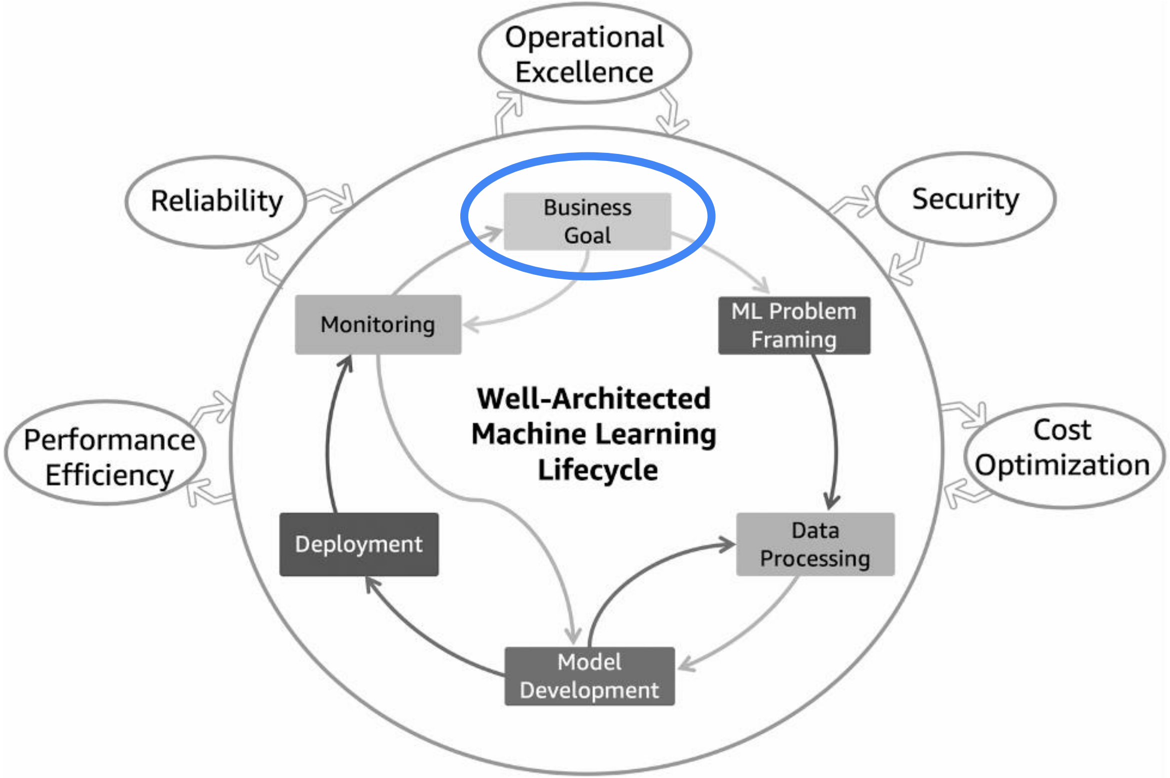
You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system

MLOps throughout the ML Lifecycle.

Business Goal.

The ML Lifecycle.



Business Goal.

"What are the best practices concerning business goals?"

Tip: Don't engineer for the sake of engineering

Questions you need to ask yourself:

1. What real problem are we trying to solve? (What is the goal?)
    - Staffing issues?
    - Safety issues?
    - Quality issues?
    - Throughput issues?
    - Waste issues?
    - ...

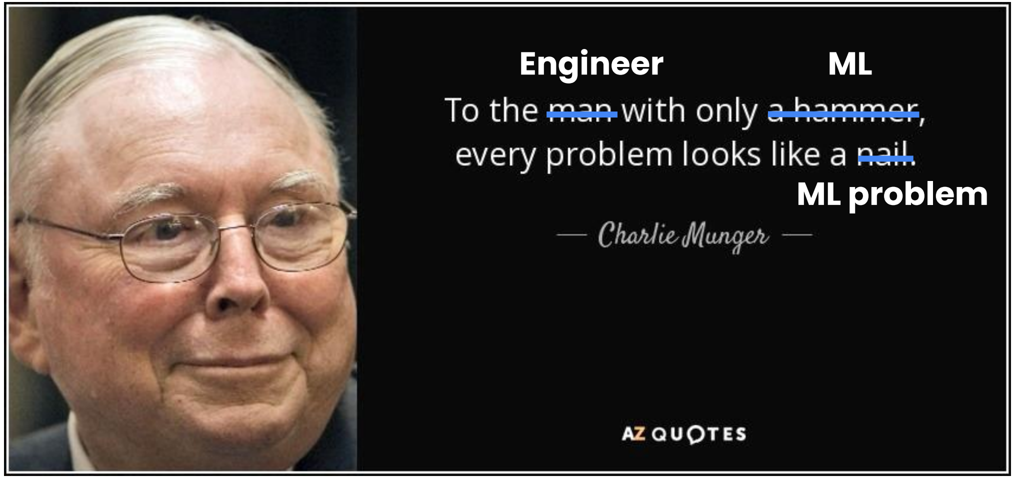
1. Is ML the best way to solve the problem?
    - No?
    - Yes?

Business Goal: A pitfall.

Business is always looking for the:
- ROI
- Lowest risk

ML solutions require a lot of:
- Knowledge
- Skill
- Ongoing effort

→ Driving up the cost and risk of failure



Tip 1: If the problem can be solved without ML, then don't use it just because it seems cool
Tip 2: always a good idea to make goals "smart" (specific, measurable, achievable, relevant, timely)

Example: Goal?

Company: APPLE BV (-> packages apples)

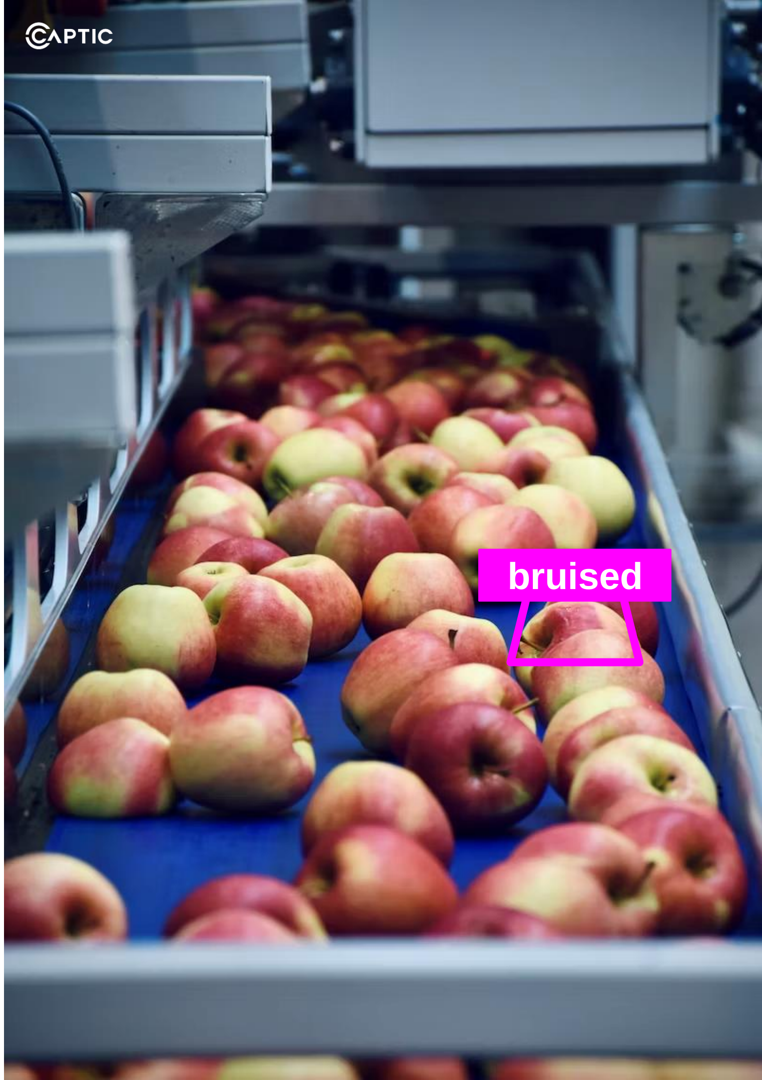
Problem:

- Finding operators is hard
- Labor cost keeps increasing
- Complaints about quality

Opportunity:

Automated Quality Analysis is a great first step.

✅ Question 1 (= goal) answered!



Example: ML?

Question 2 (= Use ML)?

Some kind of sensor is needed.

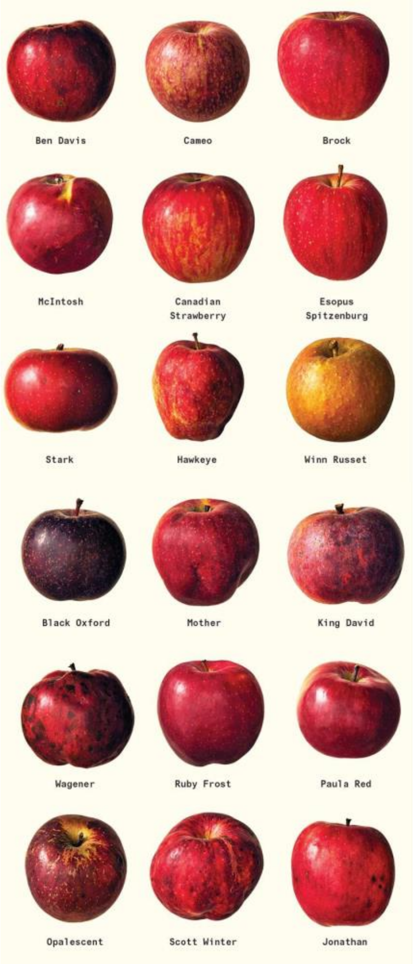
→ Weigher? Too similar
→ Camera? Makes sense

Can we use traditional methods?

- Color? Could work but won't handle variety well
- Shape? Too similar

We've exhausted all other options
→ We'll use ML to solve this problem

= Automated inspection through AI-Vision



Check list...

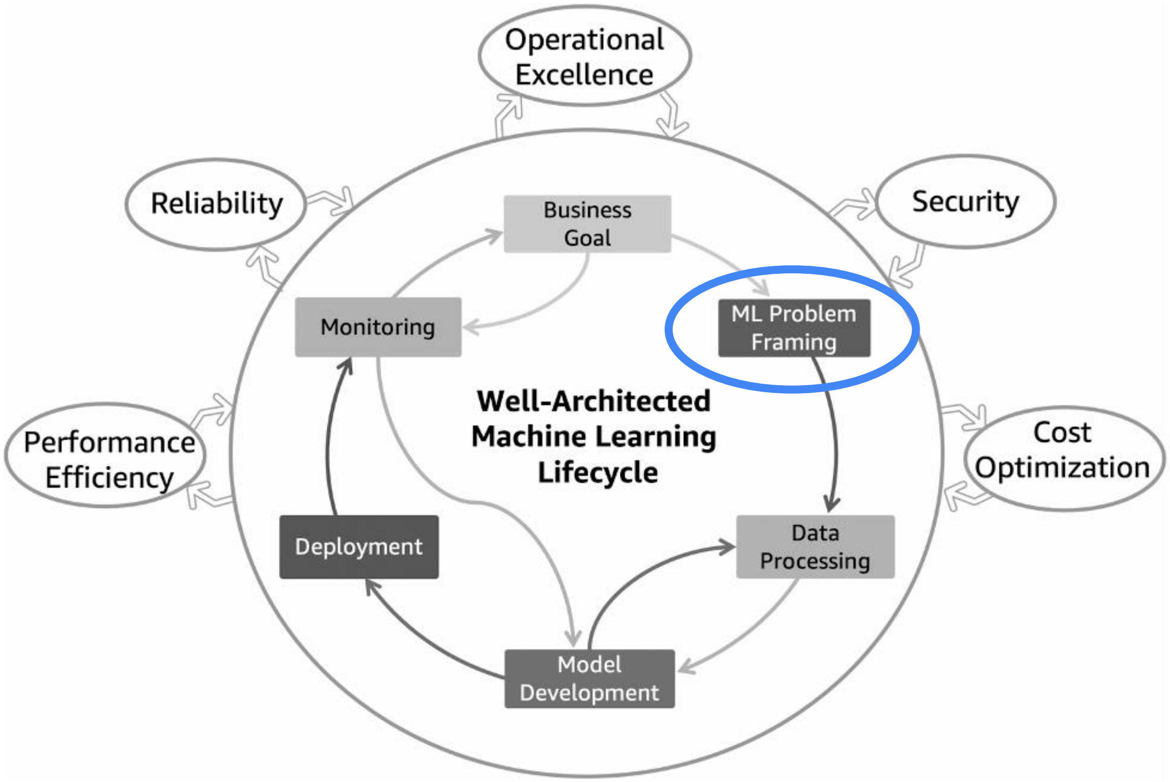
You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system
- ✅ why and how to define the business goal

MLOps throughout the ML Lifecycle.

ML Problem framing.

The ML Lifecycle.



ML Problem Framing.

Our case: Automated inspection through AI-Vision

"What are the best practices concerning ML problem framing?"

Questions to ask yourself:

1. What do we want to predict?
2. Do we have performance expectations?

Answers:

1. We need to be able to detect defects

1. Requirements
   a. Defect removal in the future so system needs to work real-time → Deployed on the edge
   b. We want to limit the amount of False Positives since this will negatively affect adoption

Check list...

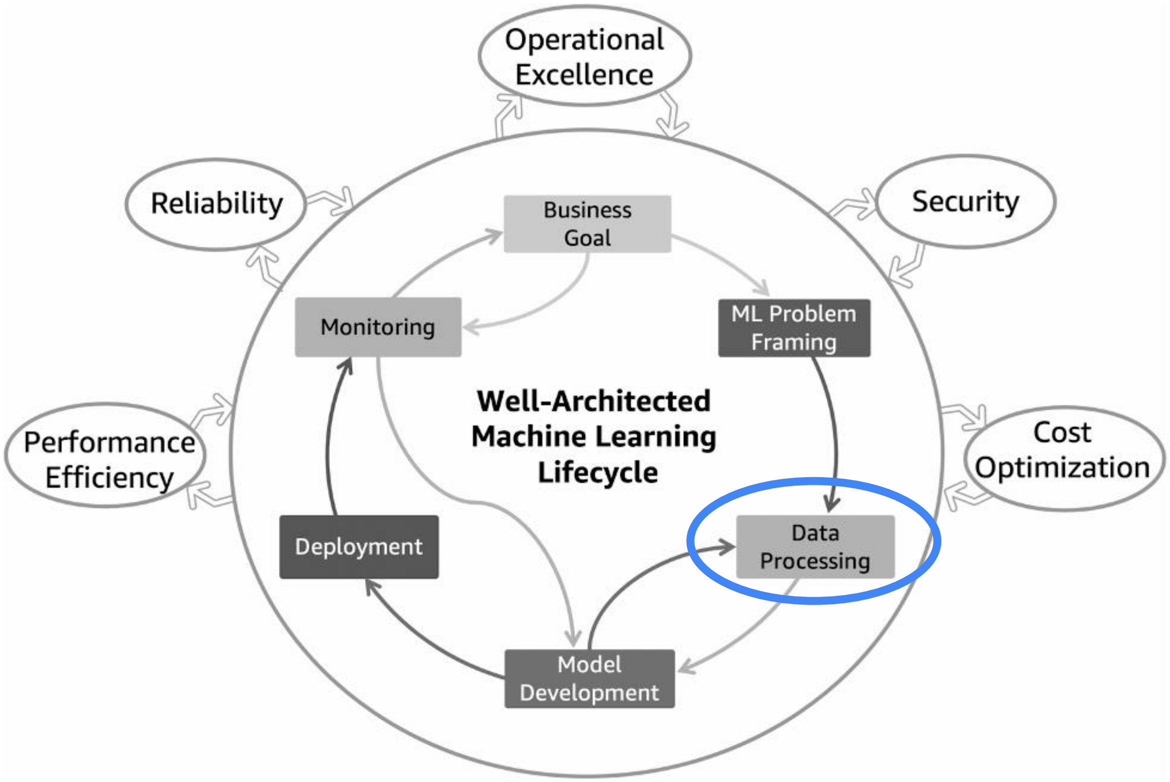
You know...

- ✅ what today's talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it's important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system
- ✅ why and how to define the business goal
- ✅ how to frame your problem in terms of ML

MLOps throughout the ML Lifecycle.

Data Processing.

The ML Lifecycle.



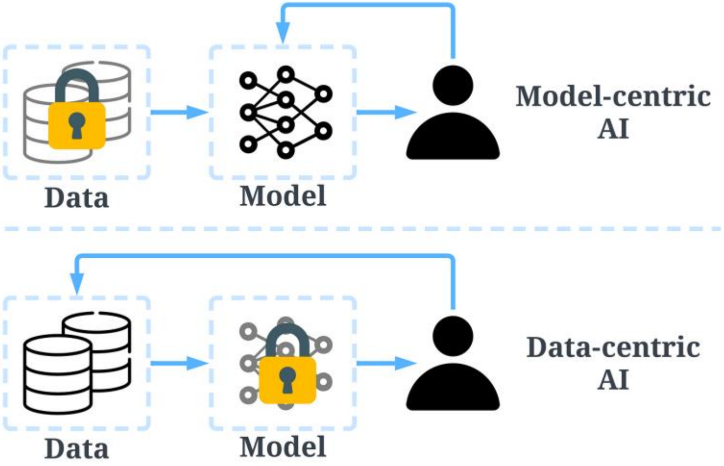
Data processing.

"What are the best practices concerning data processing?"

Questions to ask yourself:

- What data do we need?
- How much data do we need?
- Do we need particular examples that are hard to come by?
- How will we collect the data?
- Where do we store the data?
- How will we secure the data?
- How will we label it?
- In what format do we store it?
- Do we need to modify the data?
- How long will we keep the data?
- …

Data is hard work:

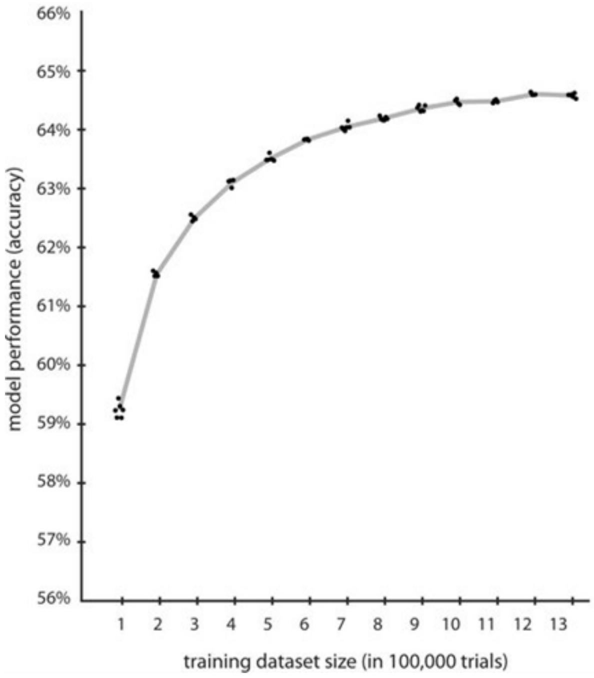


Data processing: Example.

What data do we need?
→ Images of apples with and without defects under different circumstances
→ It helps to have uniform distributions across varieties

How much data do we need?
→ Varies based on the complexity of the (vision) problem
→ No way to answer

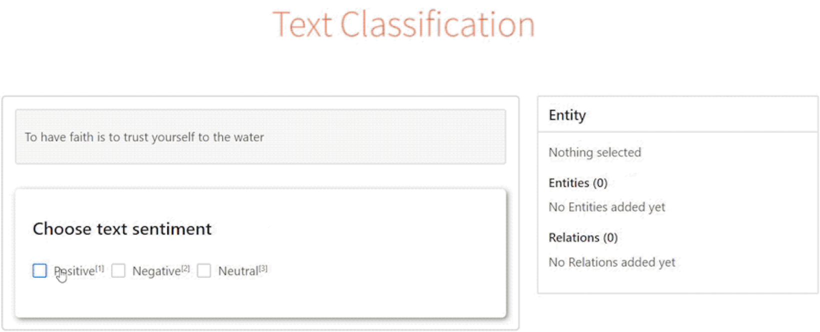
Do we need particular examples that are hard to come by?
→ Staging
→ Synthetic data
→ More data collection (camera install)



Data processing: Example.

Where do we store the data?
→ Cloud (blob) storage is a logical fit

- Large capacity
- Secure
- Lifecycle management
- Traceability



How will we label it?
→ There are many tools out there. [Label Studio](#) is a good starting point for beginners.

Tips:

- Version your datasets and splits
- track which versions you use for which models → MLFlow, W&B, …
- Analyze your data, understanding your dataset is key

Check list...

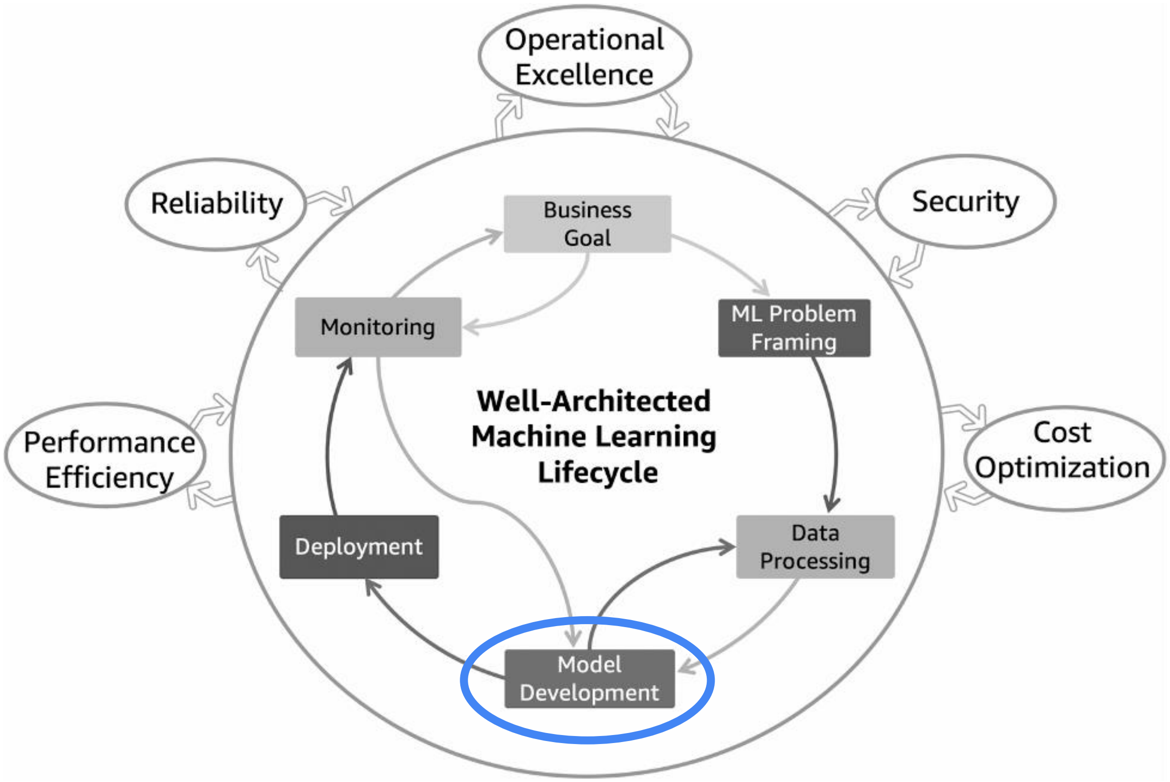
You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system
- ✅ why and how to define the business goal
- ✅ how to frame your problem in terms of ML
- ✅ how to properly process your data

MLOps throughout the ML Lifecycle.

Model Development.

The ML Lifecycle.



Model selection.

"What are the best practices concerning model development?"

There are many different approaches you can take:

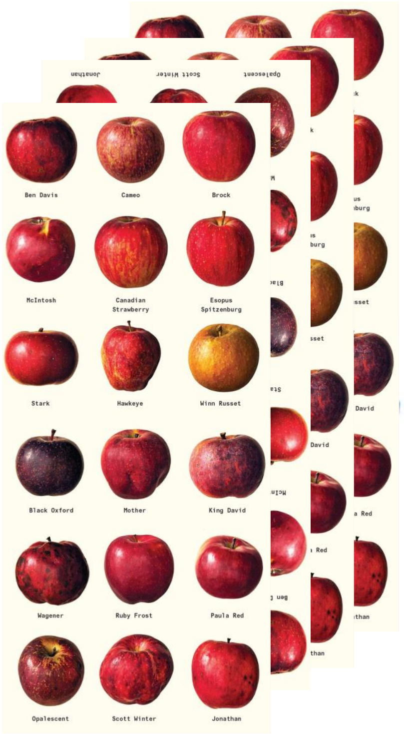
1. Build from scratch (allows for full control and optimization)

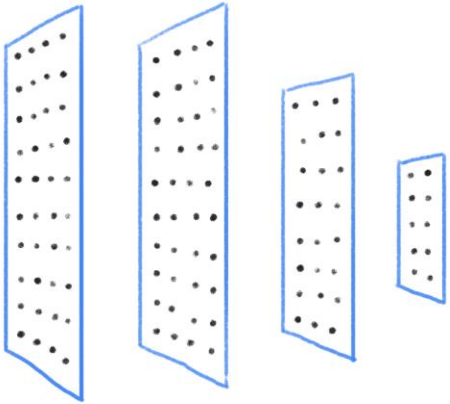
1. Give AutoML a go
    - Tends to be worse than SOTA-models, but perhaps good enough
    - Always good to have a baseline
    - Cloud costs

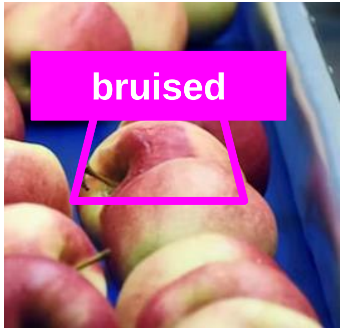
1. Use existing and proven architecture (Find them on Papers With Code)

Let’s build from scratch.

jupyter







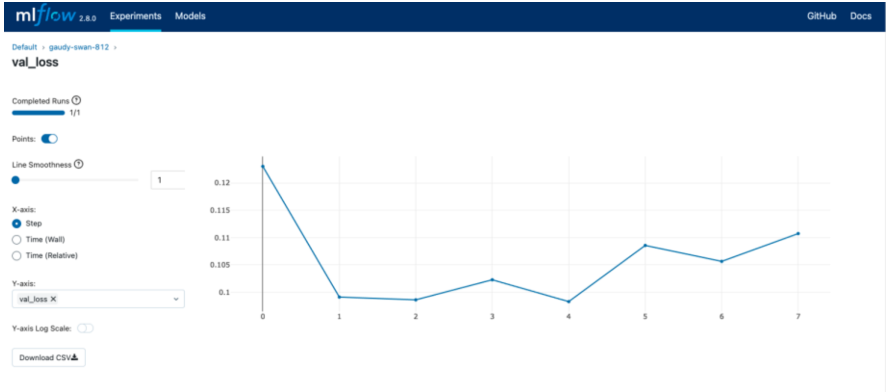
PyTorch Keras TensorFlow

Model training.

Tip: Training on specialized hardware makes it a lot faster. Try to avoid using your own laptop.
Google Colab gives you free GPU usage.

Track and register everything
- What dataset
- What augmentations
- What model
- What parameters
- ...

You can use a tool like MLflow for this.



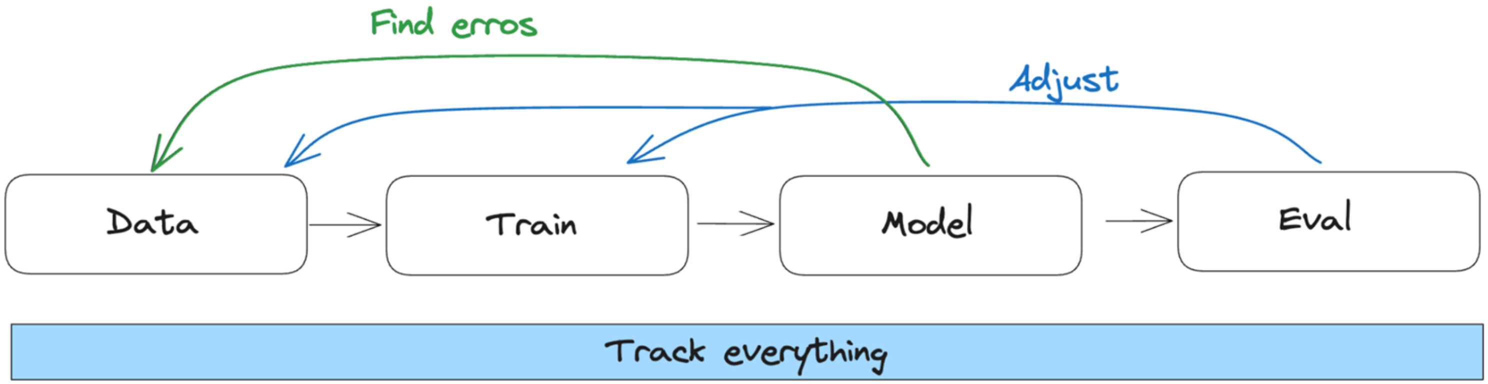
Additional “Hacks”:
- **Transfer learning =** start from model weights for similar task → Faster (and better)
- **Hyperparameter tuning** = try different parameters for different runs to see which are best
- **Data augmentation =** generate more data by changing the data itself

Model evaluation.

Evaluating the model properly is key. Make sure your test dataset is high-quality and balanced.

This step should also be tracked for reproducibility.

Model development is never done → Iterative process



ML Pipelines.

= Automated sequence of steps to build, train, evaluate and deploy machine learning models. Used to streamline the end-to-end process.

Why?

- Help organize and automate
- Keeps track of everything → reproducibility
- Can scale as needed in the Cloud
- Splitting into steps allows for collaboration
- Enforces systematic approach

Many flavors:

- Kubeflow
- TFX
- Azure
- …

The basics of ML Pipelines.

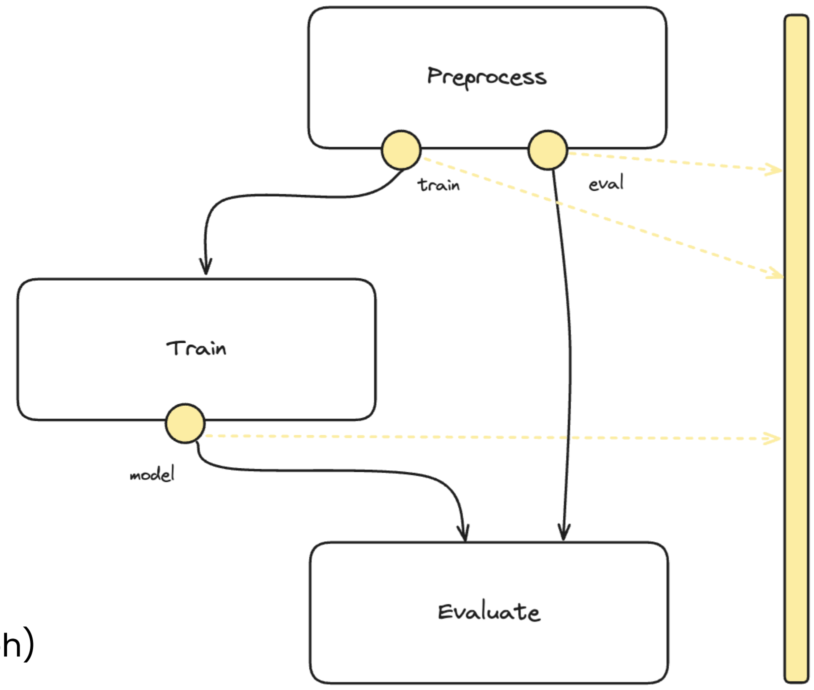
You define steps that need to run

→ *Often just Docker containers*

Steps can produce artifacts that are registered

→ *Often just a wrapper around storage*

Steps can require artifacts from previous steps (= graph)



Check list...

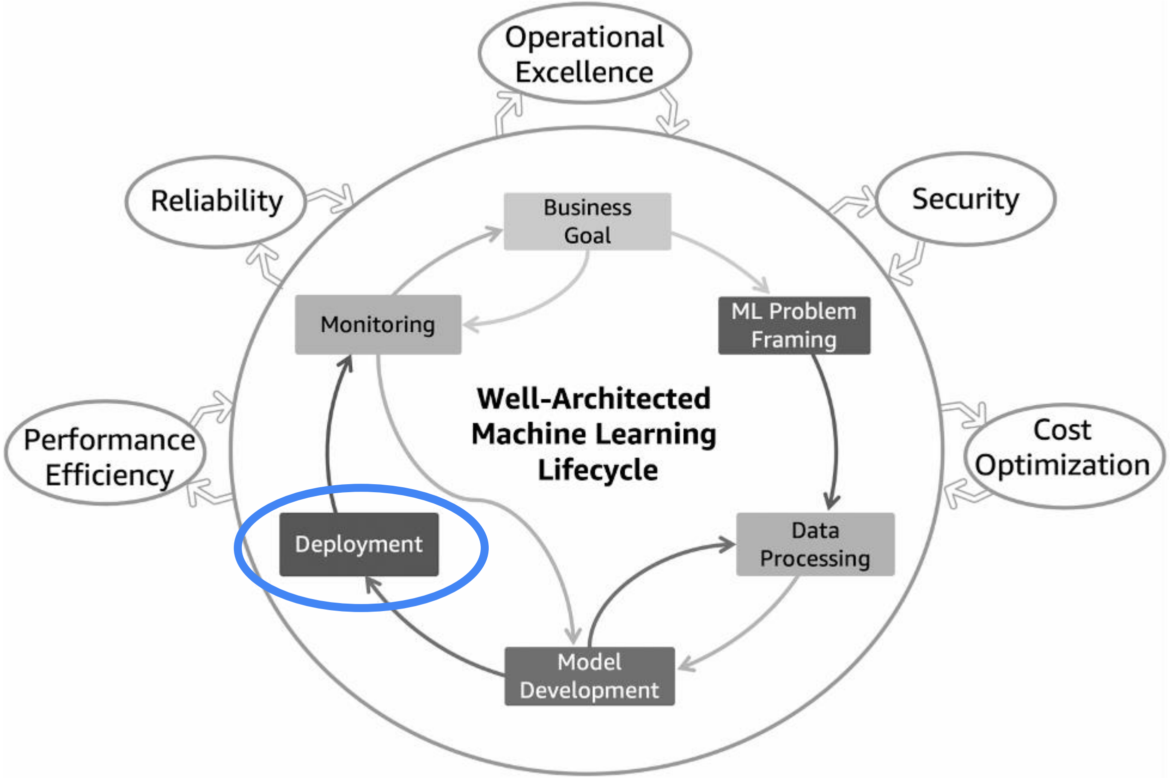
You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system
- ✅ why and how to define the business goal
- ✅ how to frame your problem in terms of ML
- ✅ how to properly process your data
- ✅ how to properly develop a model

MLOps throughout the ML Lifecycle.

Deployment.

The ML Lifecycle.

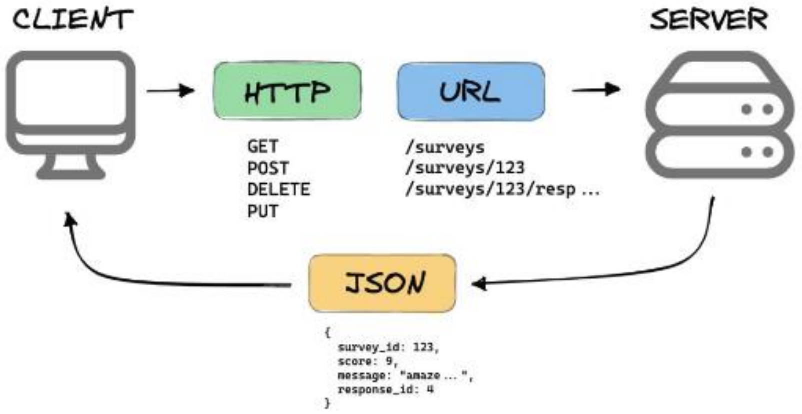


Deployment.

“What are the best practices concerning model deployment?”

There are many ways to deploy an ML model:
- Real-time serving
- Serverless
- Batch processing
- Edge deployments

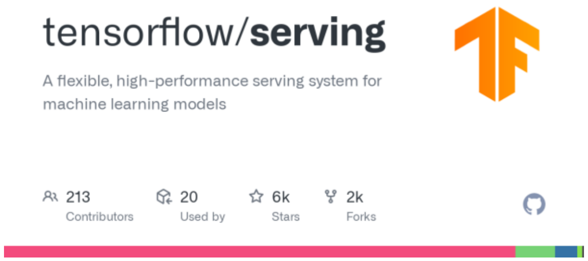
It’s a common pattern that an API is used:



Example: Tensorflow Serving.

Questions to ask yourself:

- How do I want to work with my model?
- How fast should I get an answer?
- What's my budget?
- How do we automate the release process? (CI/CD)
- What and how do we monitor?



TF Serving handles many things for us:

- serve multiple models (version) simultaneously
- Exposes both gRPC as well as HTTP inference endpoints
- Seamless canarying and A/B testing of experimental models
- Adds minimal latency to inference
- Automatic request batching
- …

```
# Start TensorFlow Serving container and open the REST API port
docker run -t --rm -p 8501:8501 \
    -v "$TESTDATA/saved_model_half_plus_two_cpu:/models/half_plus_two" \
    -e MODEL_NAME=half_plus_two \
    tensorflow/serving &

# Query the model using the predict API
curl -d '{"instances": [1.0, 2.0, 5.0]}' \
    -X POST http://localhost:8501/v1/models/half_plus_two:predict
```

Deployment: Example.

We need our Classifier to work real-time. Real-time serving?
→ No, we can’t let our production rely on whether we have internet connection.

We need an **Edge Deployment**

There are many options for the hardware:
- Coral
- Raspberry Pi
- Nvidia Jetson
- …

What do they have in common?
They’re tiny, so our model will have to be as well

Model optimization.

Since we want our model to run on Edge, we need to make it tiny and fast

→ Best to start with a model that is already pretty small

Many techniques for model compression:

- Pruning = removing unimportant weights
- Quantization = reducing precision of weights
- Knowledge distillation = training a smaller student model that learns from the bigger teacher

There are tools that do this for you:

- In TensorFlow
- TensorFlow Lite
- In ONNX
- ...

Check list...

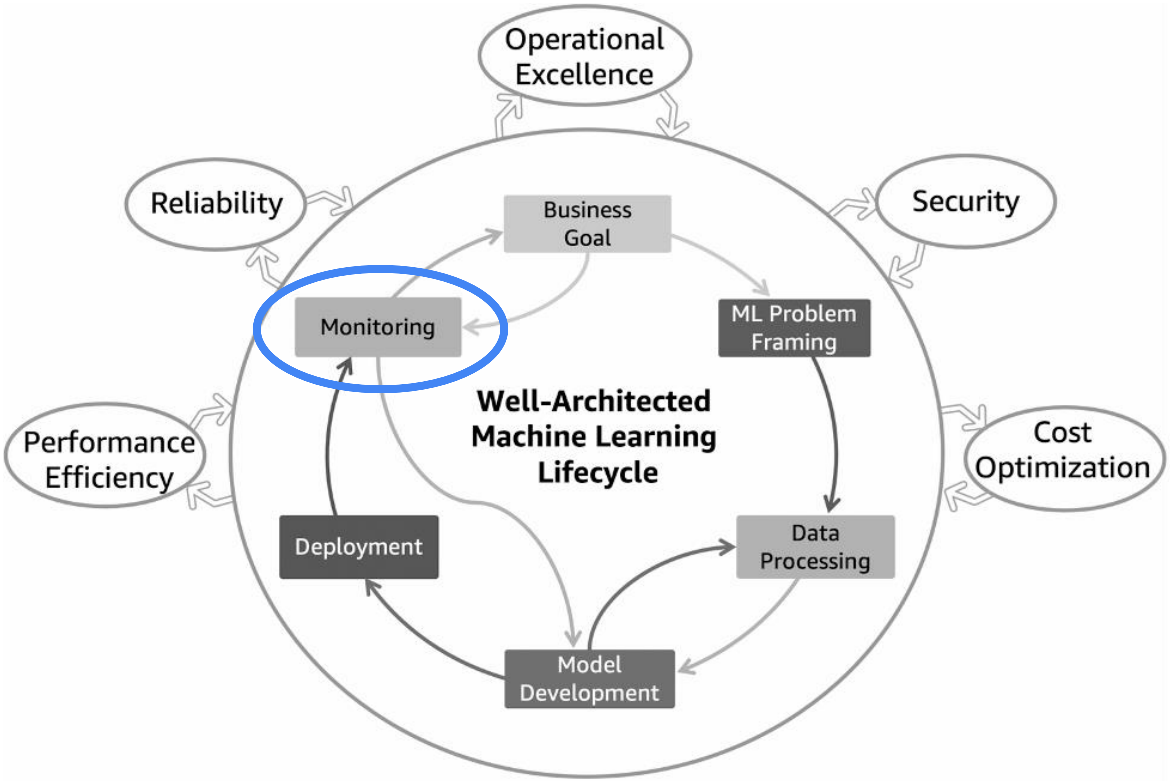
You know…

- ✅ what today's talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it's important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system
- ✅ why and how to define the business goal
- ✅ how to frame your problem in terms of ML
- ✅ how to properly process your data
- ✅ how to properly develop a model
- ✅ how to properly deploy a model

MLOps throughout the ML Lifecycle.

Monitoring.

The ML Lifecycle.

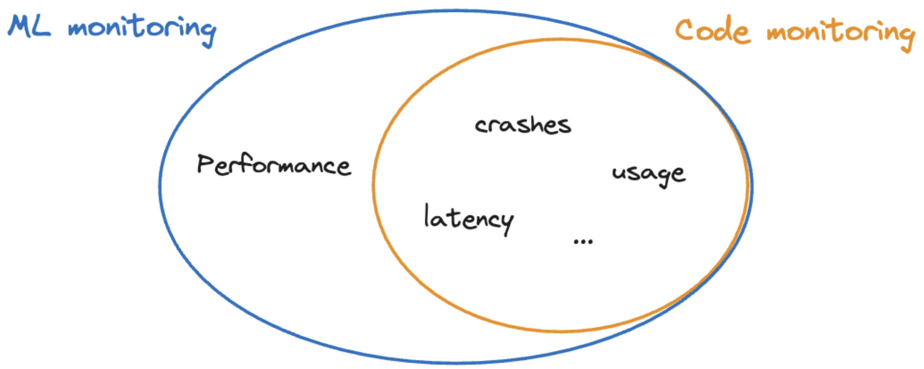


Monitoring.

Why?

- We want to know how our model is doing
- Every model will get bad over time. How fast depends on the use case

ML monitoring is different from normal code monitoring
→ Bad doesn't mean crash (latency, up, …)



Model Decay.

No model lives forever, but the speed of decay varies.

Usual culprits:

- Data drift = changing of input data
- Concept drift = relationship between input and output has changed

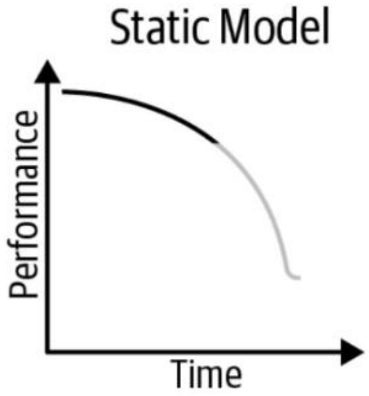
Example of Data drift:
Our lens is dirty so our images look different

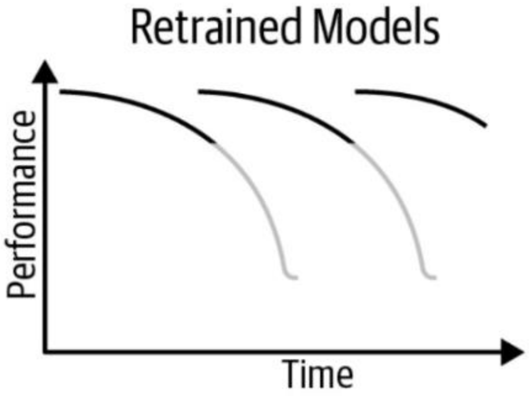
Example of Concept drift:
The Quality manager’s interpretation of a certain class changes over time

→ Monitoring allows us to spot model decay and to retrain (back to model development)

Acting on Model Decay.

Monitoring is needed to avoid negative impact of stale models





One of the reasons to have an automatic (scheduled) retraining pipeline!

Monitoring.

What can be monitored:

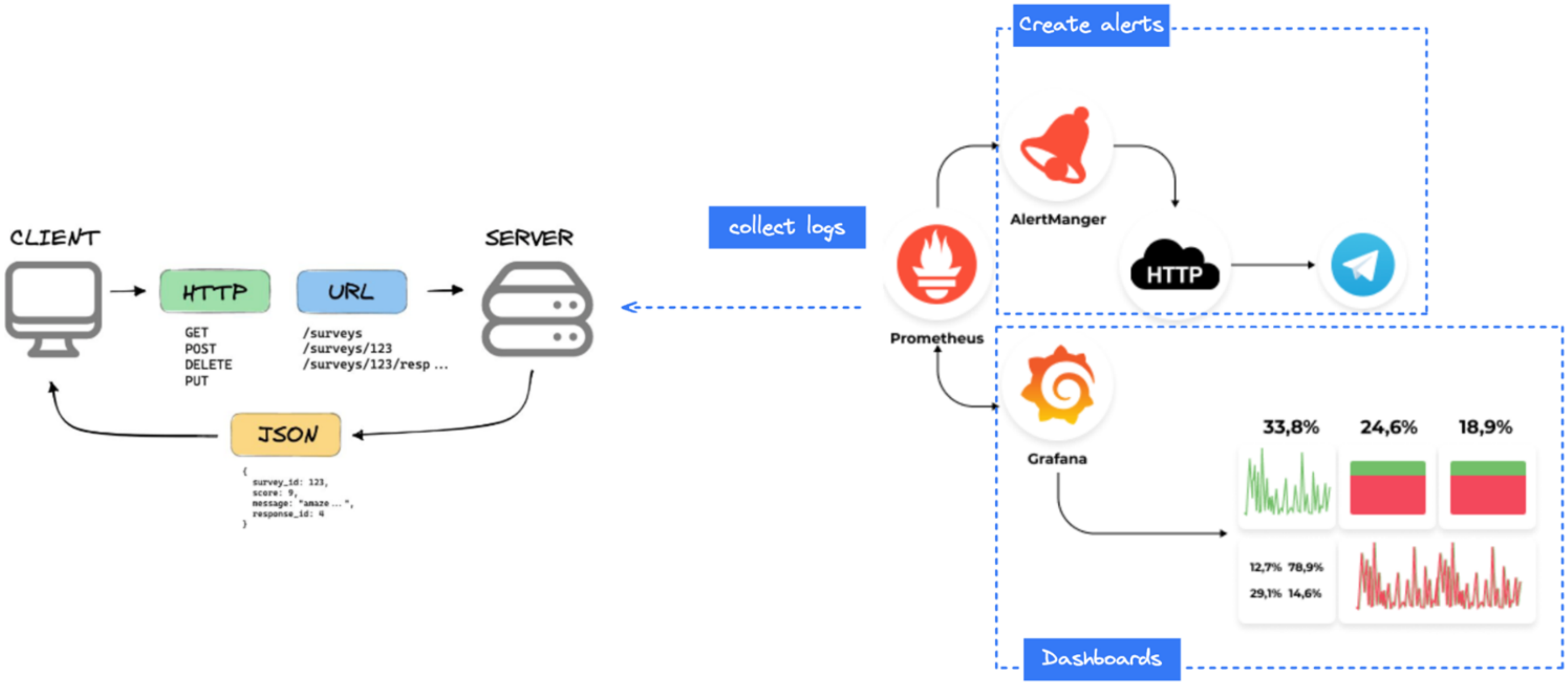
- User feedback collection
    - Instances themselves
    - How many over time

- Output
    - Confidence scores
    - Distributions as expected?
        - Dataset
        - Previous models

- Annotated data

You'll also want to monitor whether you're actually solving the business problem.

Typical monitoring setup.



Check list...

You know…

- ✅ what today’s talk is about
- ✅ to only use ML when necessary
- ✅ what MLOps is and why it’s important
- ✅ Captic and what we do
- ✅ the necessary skills to perform MLOps tasks
- ✅ lifecycle of an ML system
- ✅ why and how to define the business goal
- ✅ how to frame your problem in terms of ML
- ✅ how to properly process your data
- ✅ how to properly develop a model
- ✅ how to properly monitor an ML system

Q&A.

Want to learn more?

- Follow us on LinkedIn
- Apply for an internship