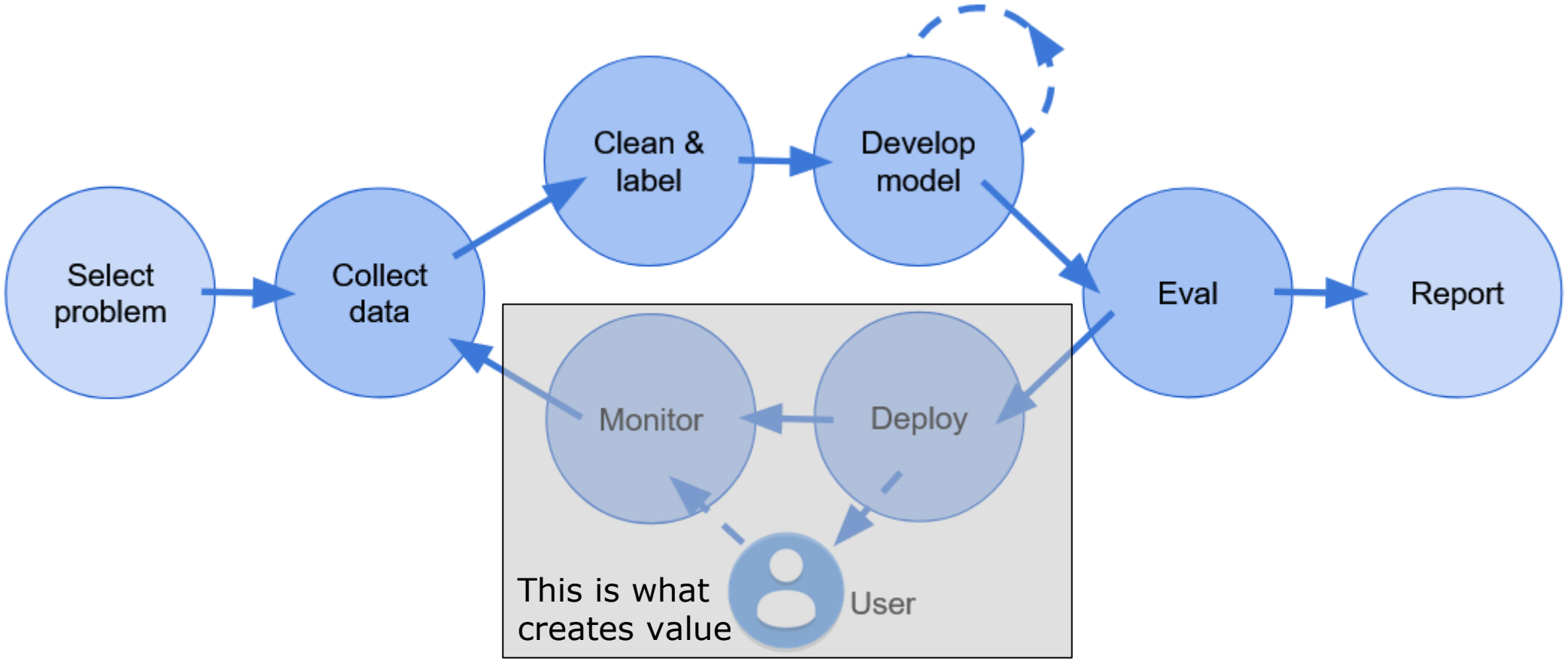
02476 Machine Learning Operations

Nicki Skafte Detlefsen

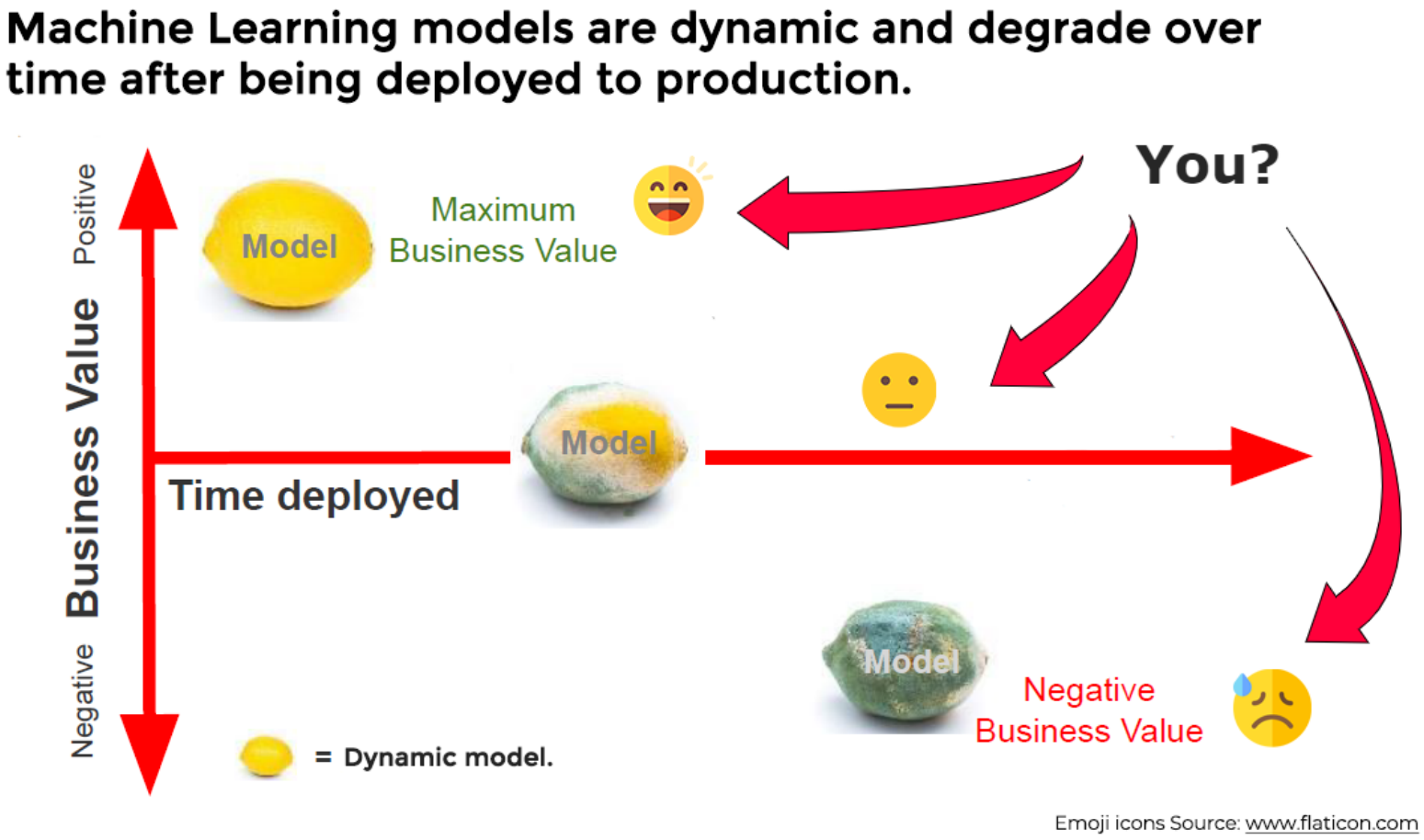
Monitoring

Based on slides by Duarte O. Carmo

Remember this figure



We are in the endgame now

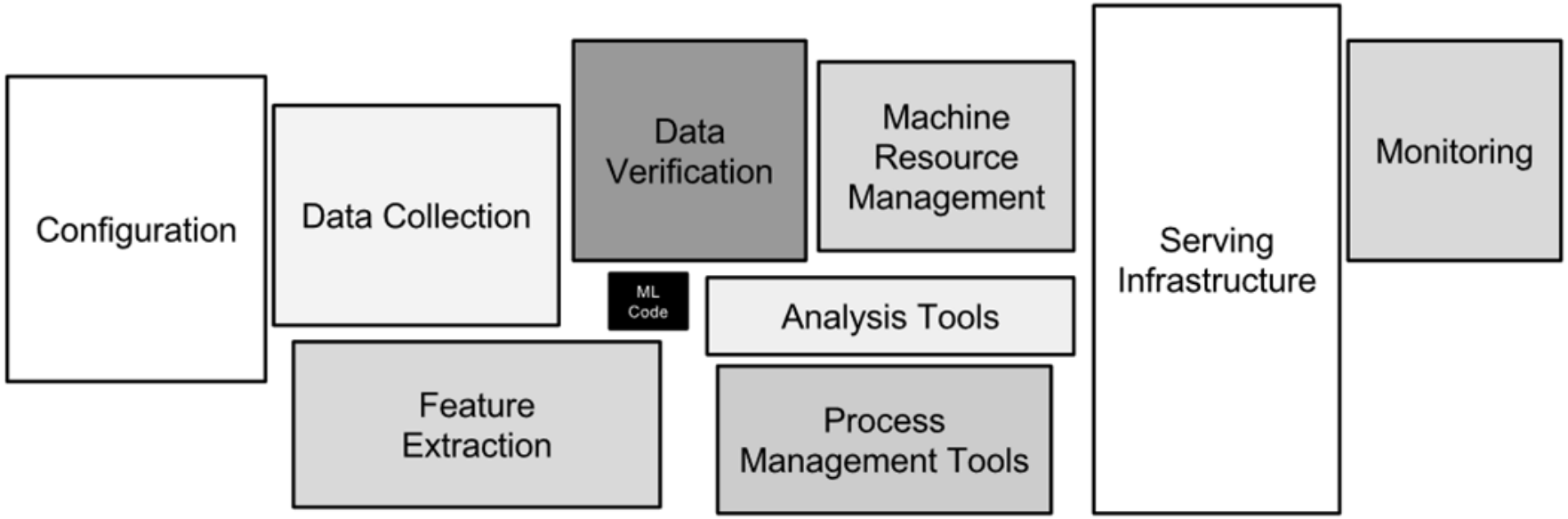


Emoji icons Source: www.flaticon.com

Operations is hard

“All models are wrong, but some are useful” - George Box

“87% of data science projects never make it into production”

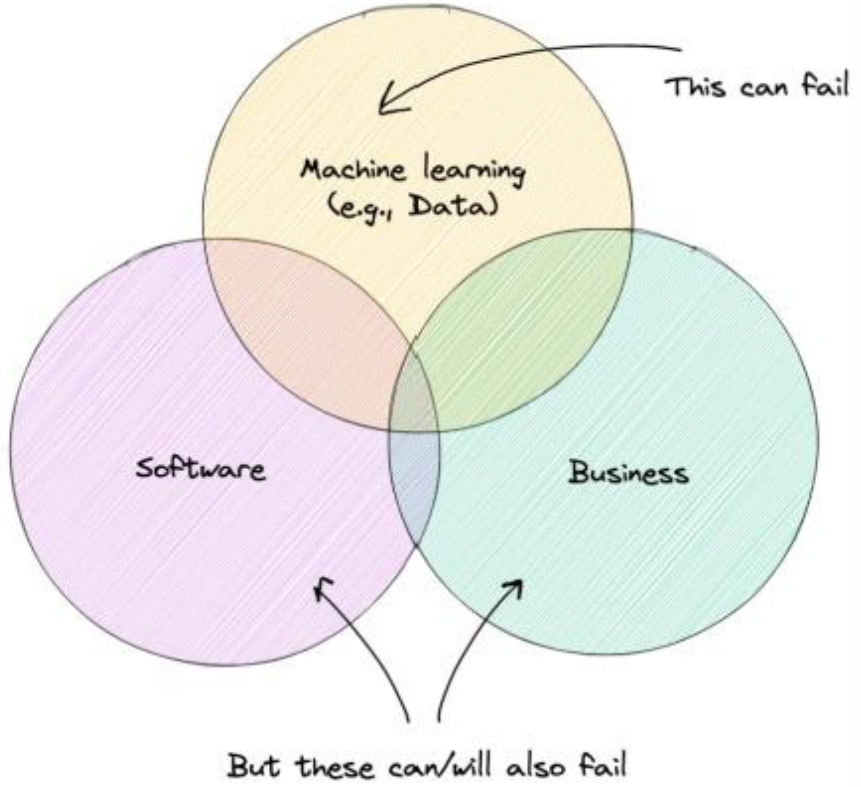


What can fail?

🚩 Monitoring deals with things ~~if~~ when they break

Applications will fail for many reasons, but we can group into three

- ML failures
- Software failures
- Business failures



Software failures

“Software is never done (only abandoned)”

All the reasons a non-ML application already can fail

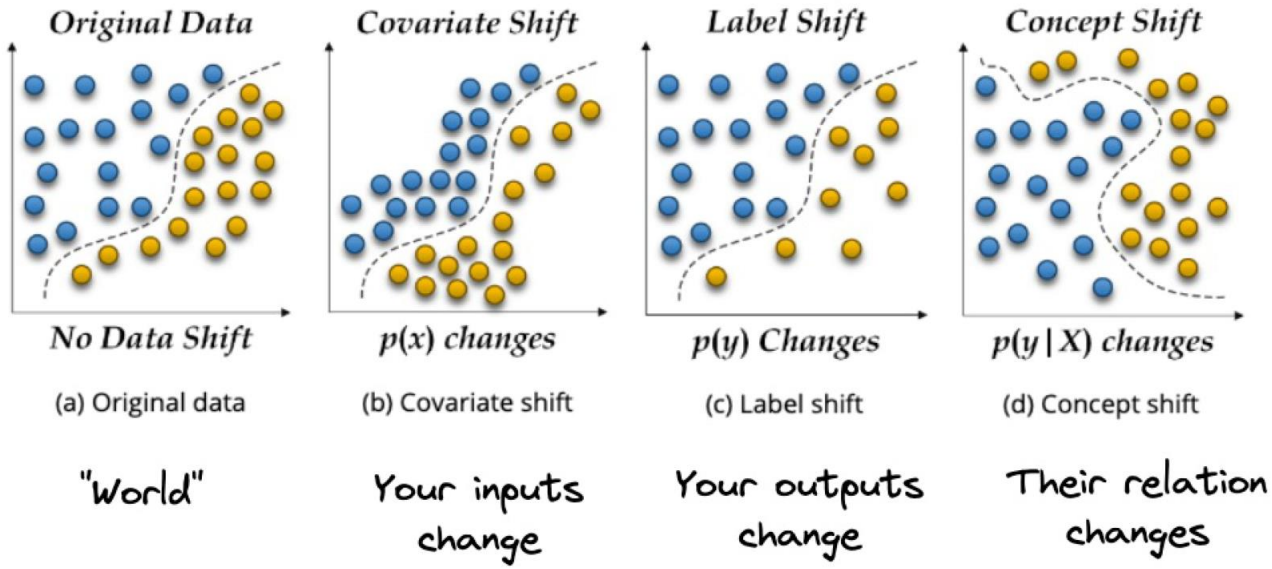
- Dependencies
- Deployments
- Hardware
- Downtime/crashing

ML failures

ML specific applications can fail for even more reasons

- Edge cases
- Feedback loops
- Training != production

x : your inputs
y: your outputs



ML failures

⚠️ Data drift

Model perform worse on unseen data

⚠️ Target drift

The world have changed, you need to wake up

Deal with by

✅ Train model on massive dataset initially

✅ Domain adaptation of large models

✅ Retrain from scratch or finetune

ML failures

To know how you are doing, you most likely is going to need labels for incoming data.

Three ways to get them:

- 🔥 Hand labels
  - Annotate by hand, expensive as hell
- 🔥 Natural labels
  - You get correct label in the future, leverage it
- 🔥 Programmatic labels
  - Any feedback is better than non, get creative

Business failures

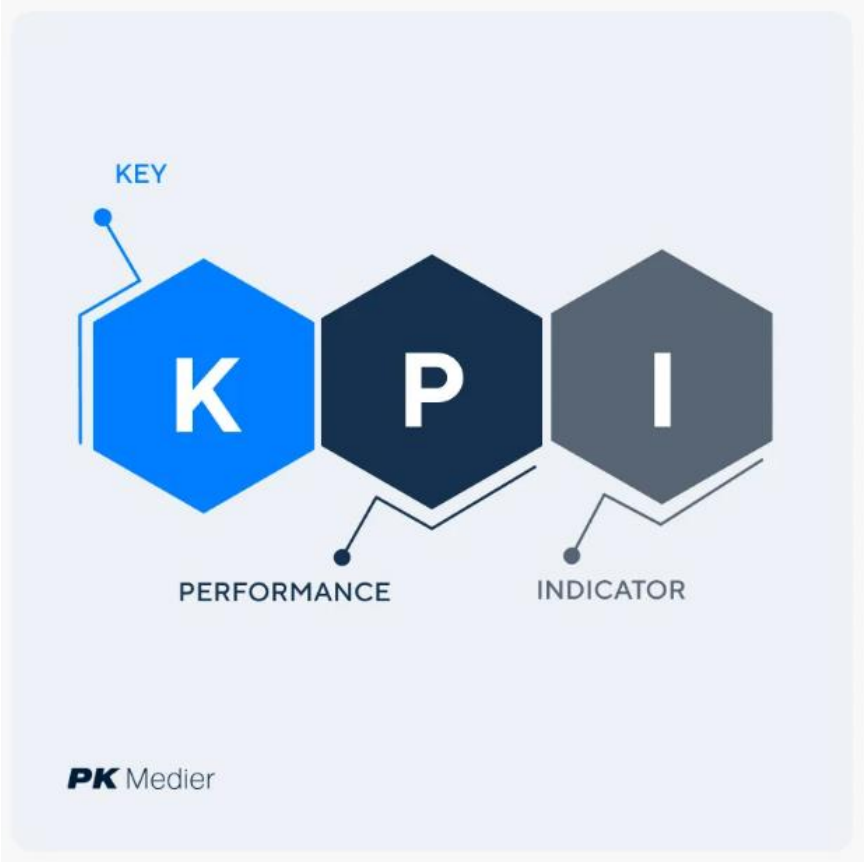
When ML deployments are not synchronized with business, then it fails

💡 We are not doing enough predictions

💡 Model predictions are not benefitting KPI

Deal with by

✅ Better business alignment



Logs

- Logs are textual or structured records generated by applications
- They provide a detailed account of events, errors, warnings, and informational messages that occur during the operation of the system
- Logs are essential for diagnosing issues, debugging, and auditing.

Traces

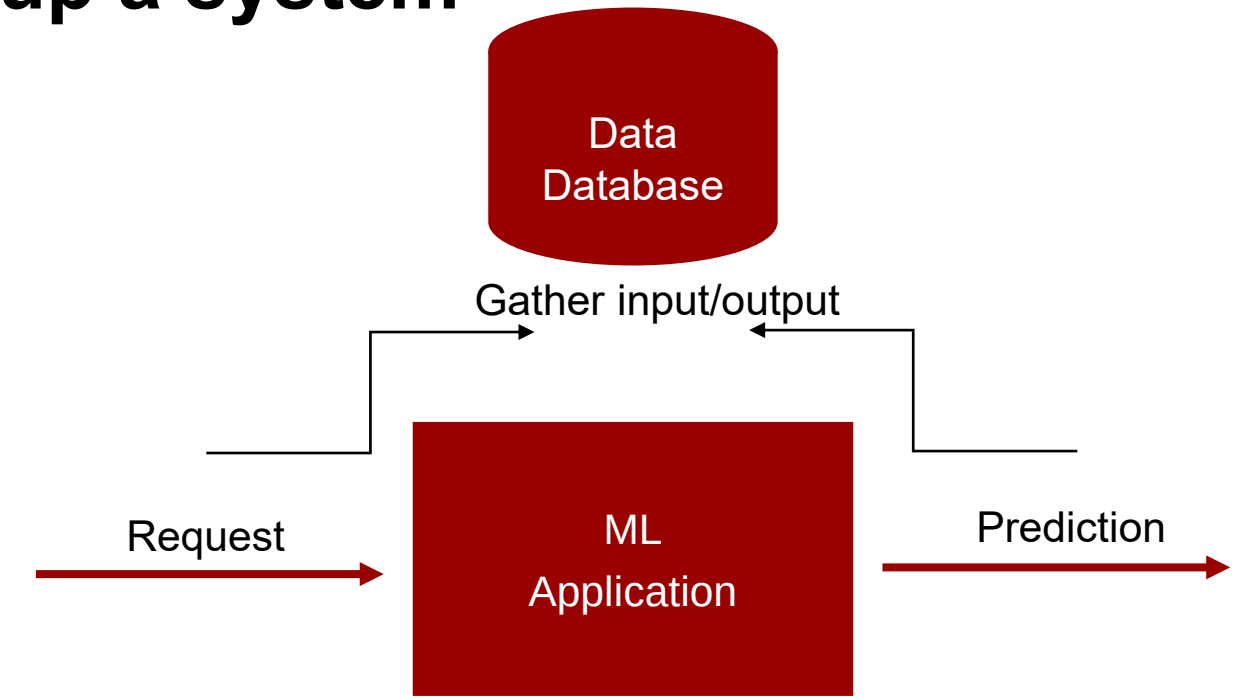
- Traces are detailed records of specific transactions or events as they move through a system.
- A trace typically includes information about the sequence of operations, timing, and dependencies between different components.
- Traces help in understanding the flow of a request or a transaction across different components.

Metrics

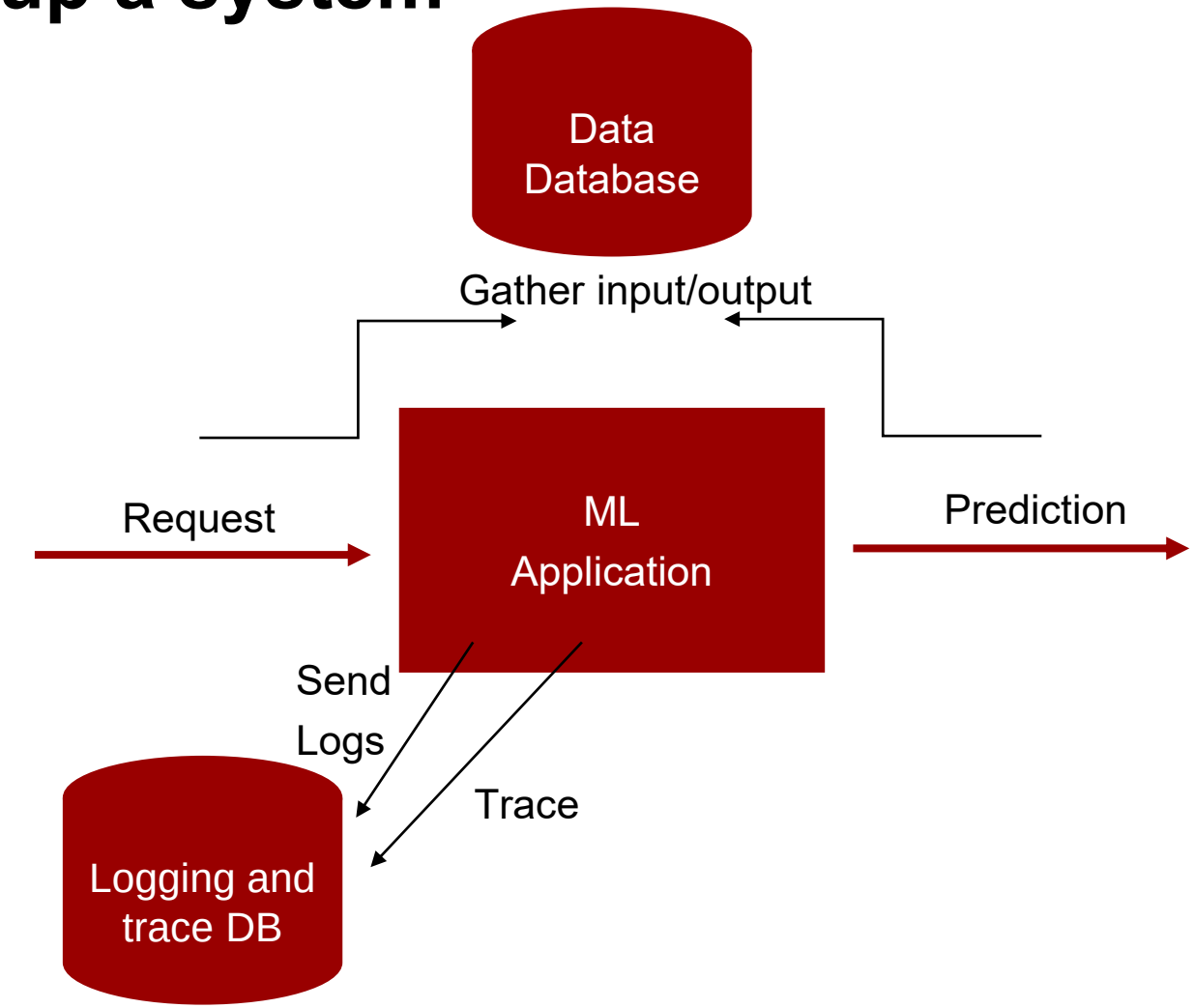
- Quantitative measurements of the system.
- They are usually numbers that are aggregated over a period of time. E.g. the number of requests per minute.
- Metrics are used to get an overview of the system

Let’s setup a system

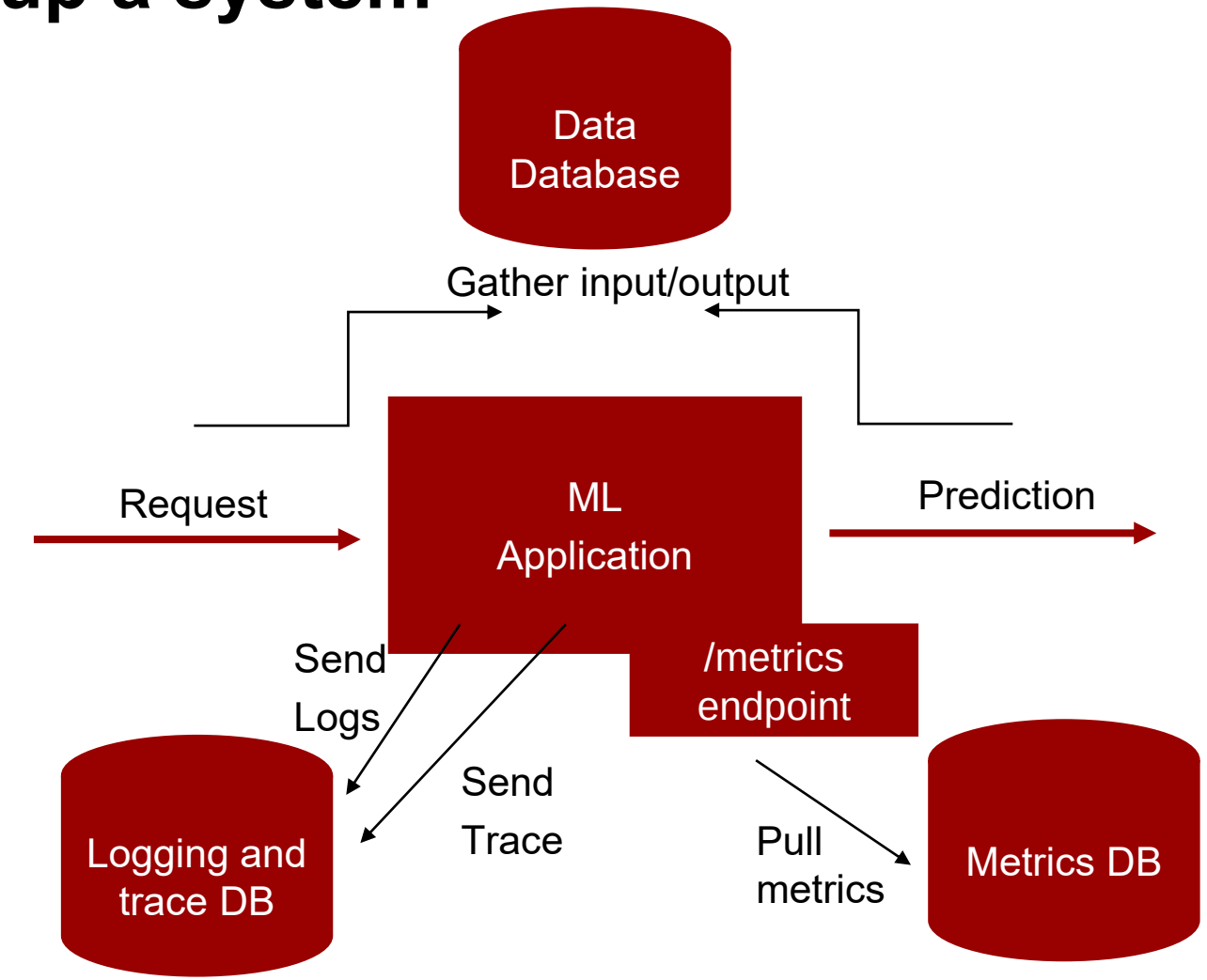
Let’s setup a system



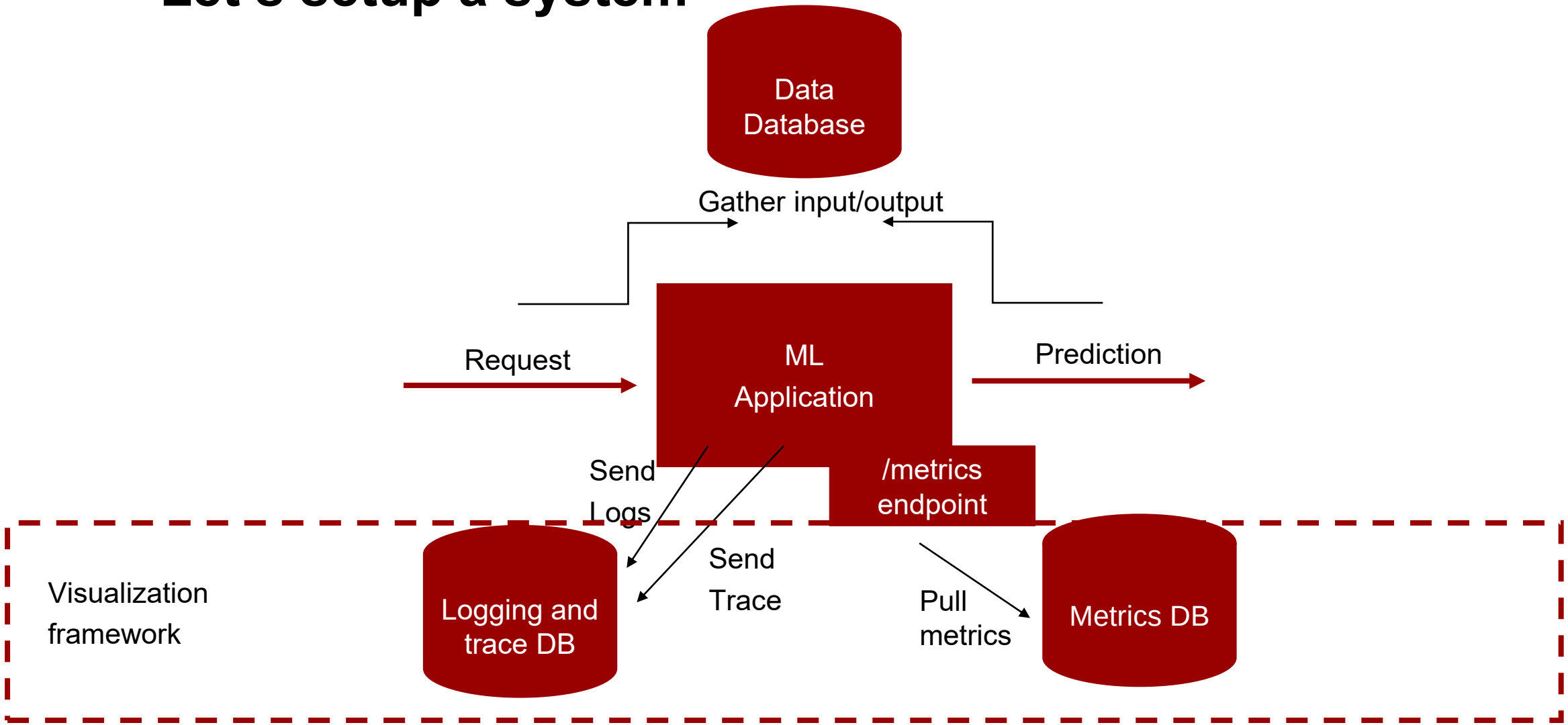
Let’s setup a system



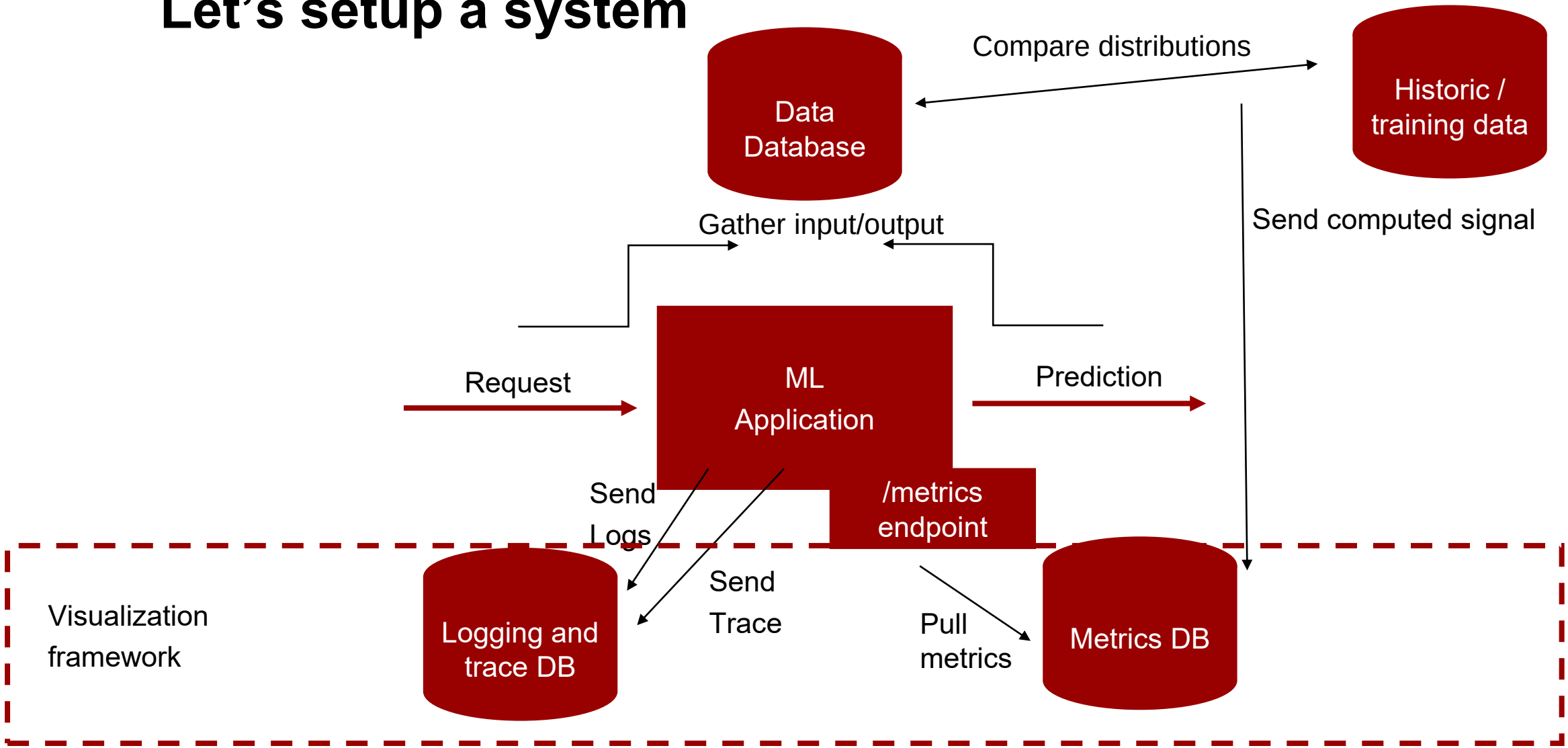
Let’s setup a system



Let’s setup a system



Let's setup a system



Let's setup a system

Data Database

Compare distributions

Historic / training data

Gather input/output

Send computed signal

Request

ML Application

Prediction

Send alerts

/metrics endpoint

Send Logs

Send Trace

Pull metrics

Visualization framework

Logging and trace DB

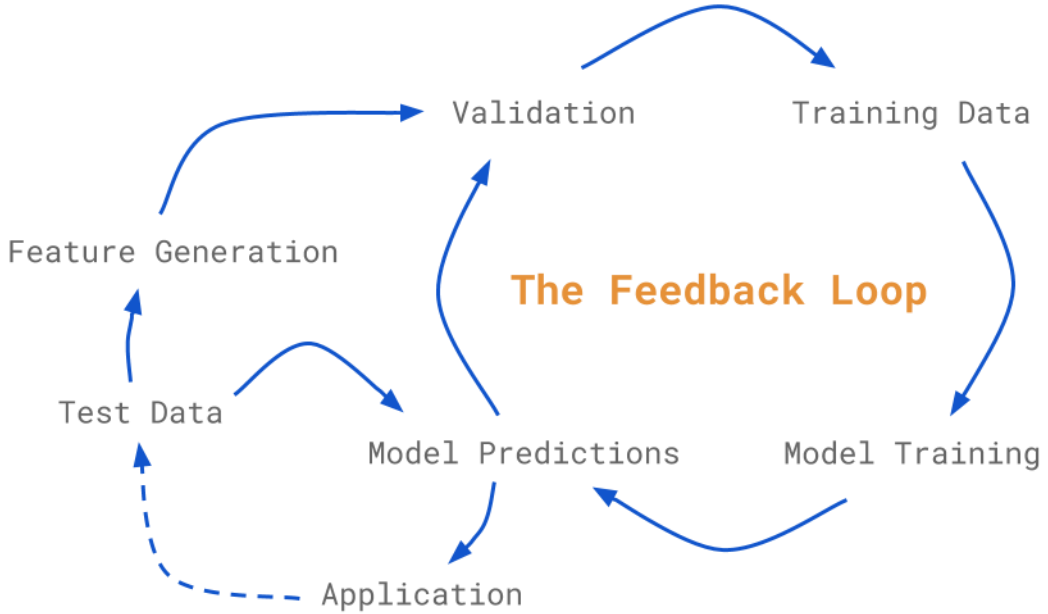
Metrics DB

Summary

✅ Things will break, so it is better to know when it happens

✅ You are trying to deliver value, make sure you know when

✅ Get feedback, save what you can



Meme of the day