

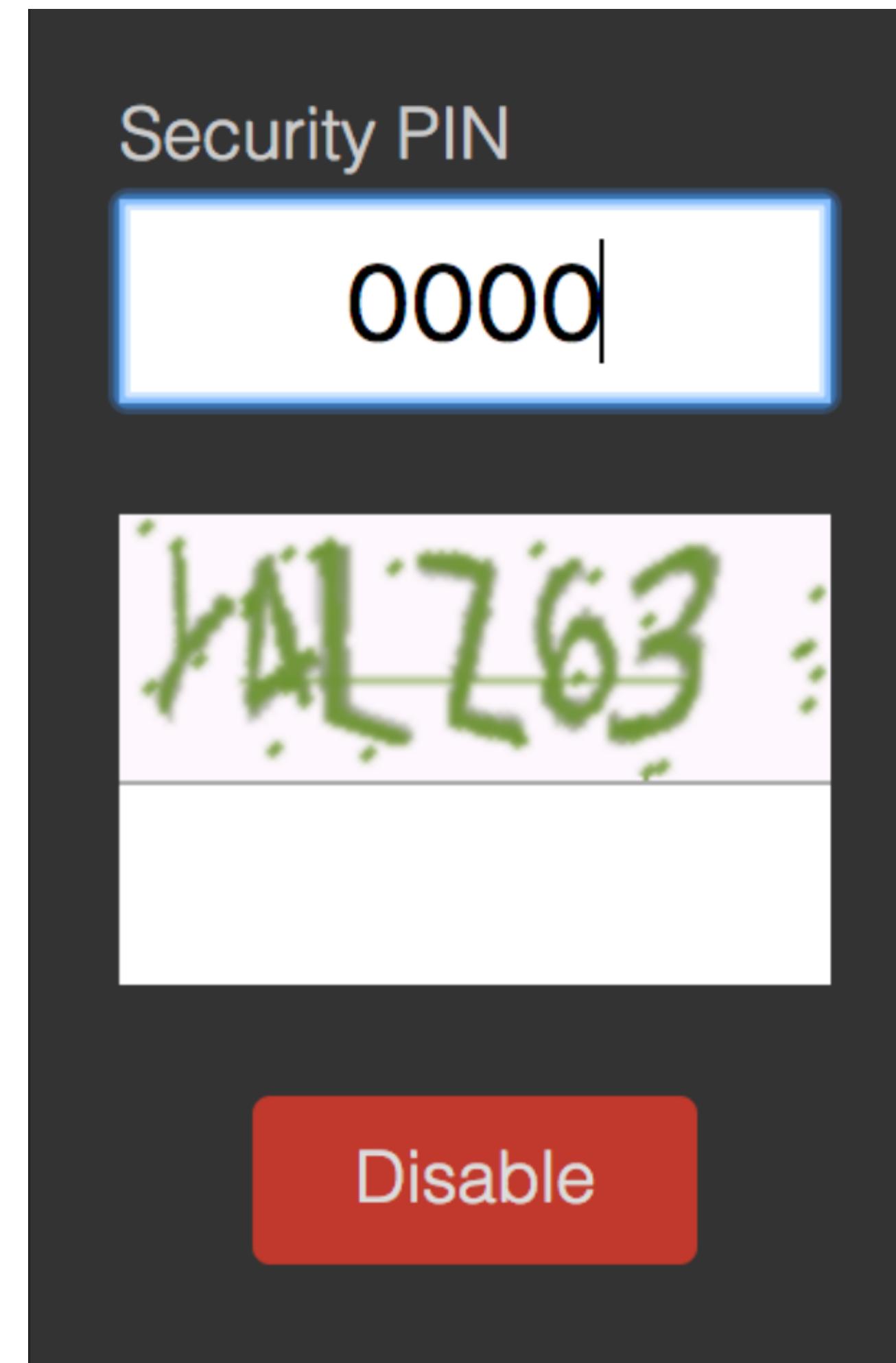
Infrastructure for Machine Learning Applications

Natalie Pistunovich
@NataliePis

@tweakers #TweakersDevSummit

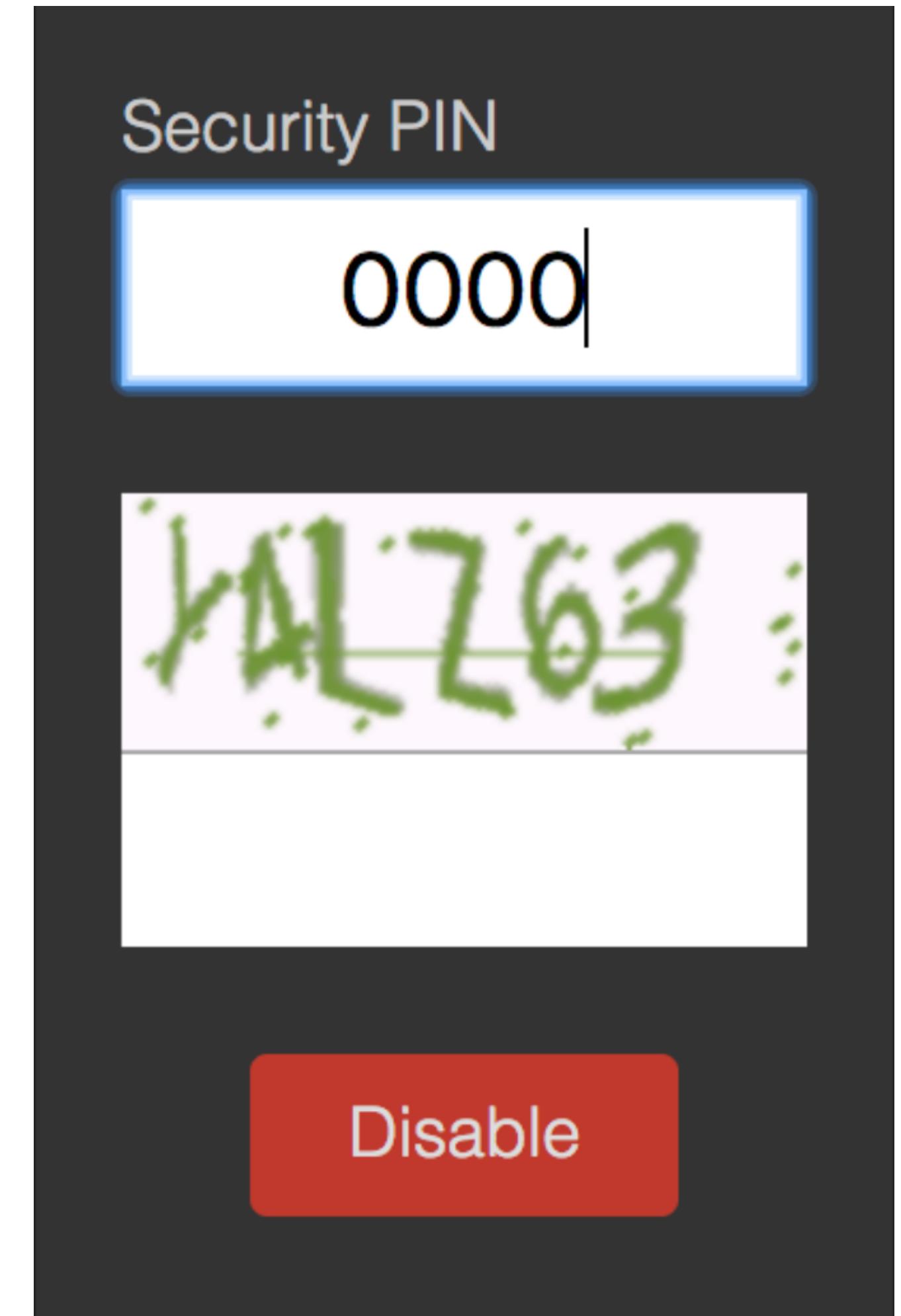


Captcha Challenge



Captcha Challenge

1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:
 - i. Open a cookie jar
 - ii. Get the CAPTCHA image
 - iii. Predict CAPTCHA using ML
 - iv. Guess the PIN + CAPTCHA
 - a. if false CAPTCHA,
fall back to (ii)



Captcha Challenge

Read all about it at the December 28 2017
Gophers Academy Advents Blog post

<https://github.com/Pisush/break-captcha-tensorflow>

Using Machine Learning: Go + TensorFlow











@tweakers
#TweakersDevSummit
@NataliePis

What is ML?



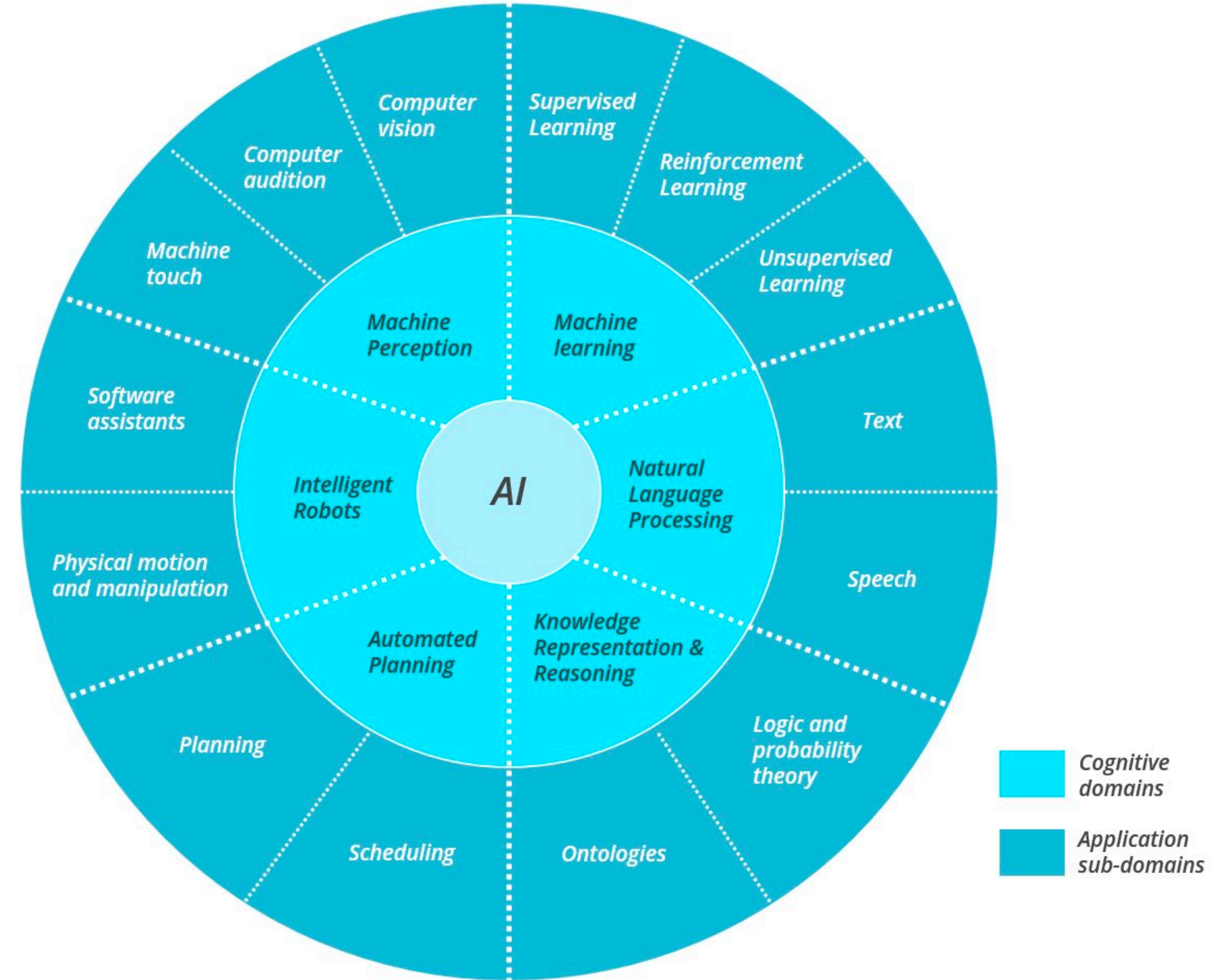
Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

It is seen as a subset of artificial intelligence.

- Wikipedia

What is AI?





In computer science, **artificial intelligence (AI)**
... is intelligence demonstrated by machines, in
contrast to the **natural intelligence** displayed by
humans.

- Wikipedia

The term **artificial intelligence** is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

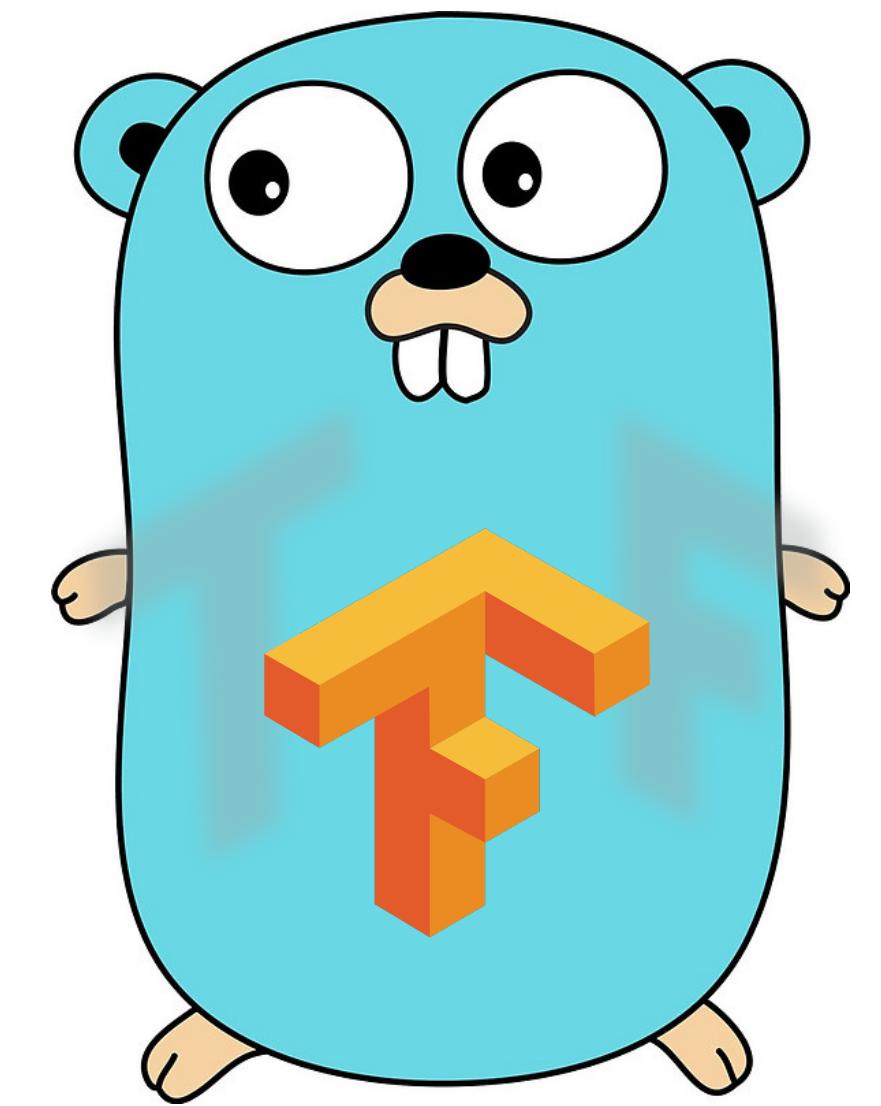
- Wikipedia

How to ML?



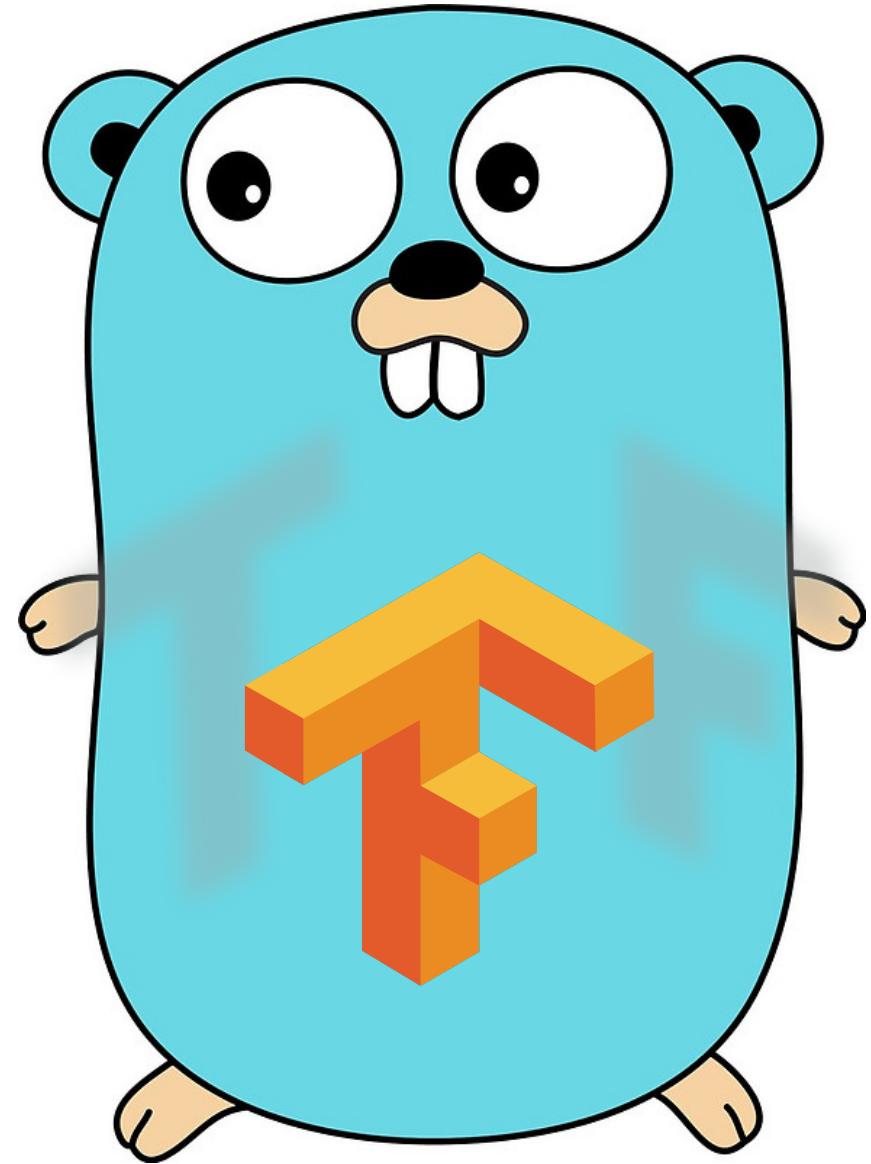
How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



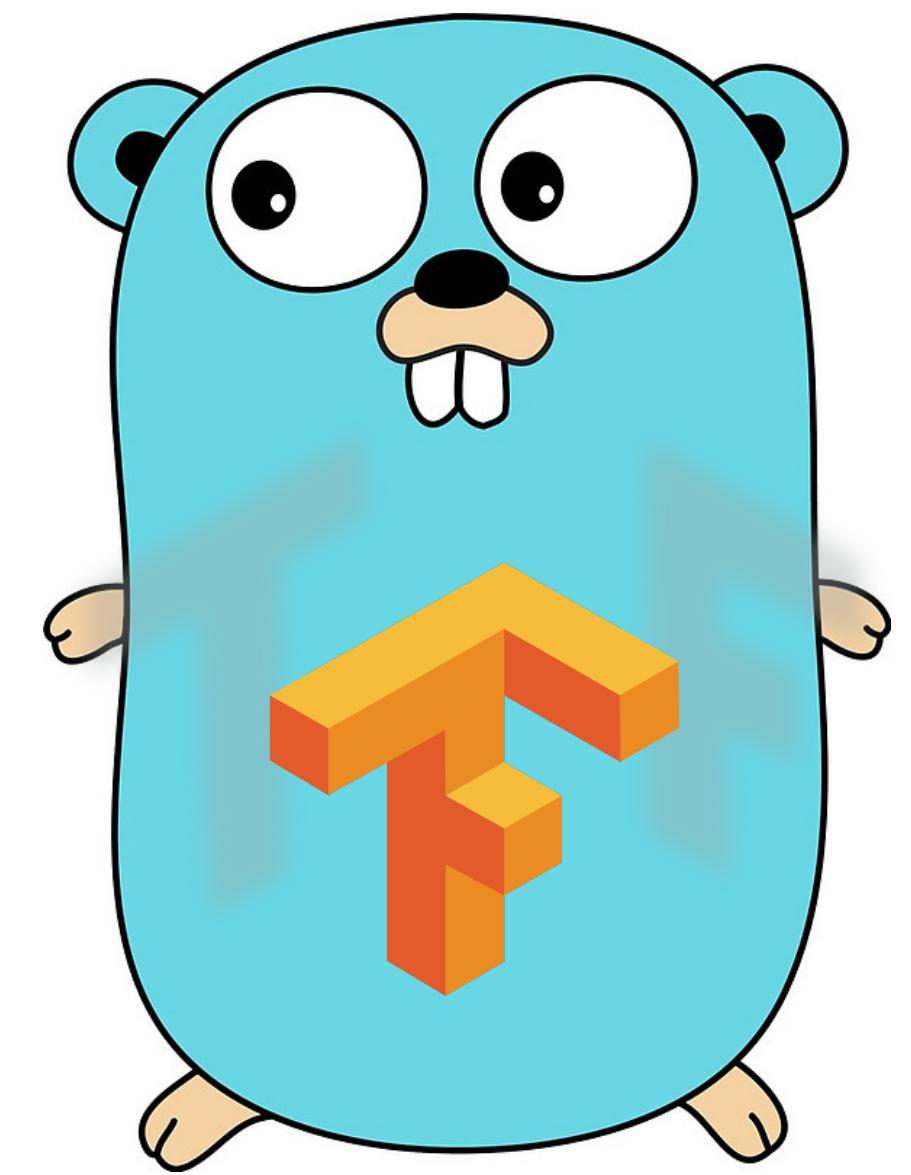
How to ML

- 1. Define the problem**
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



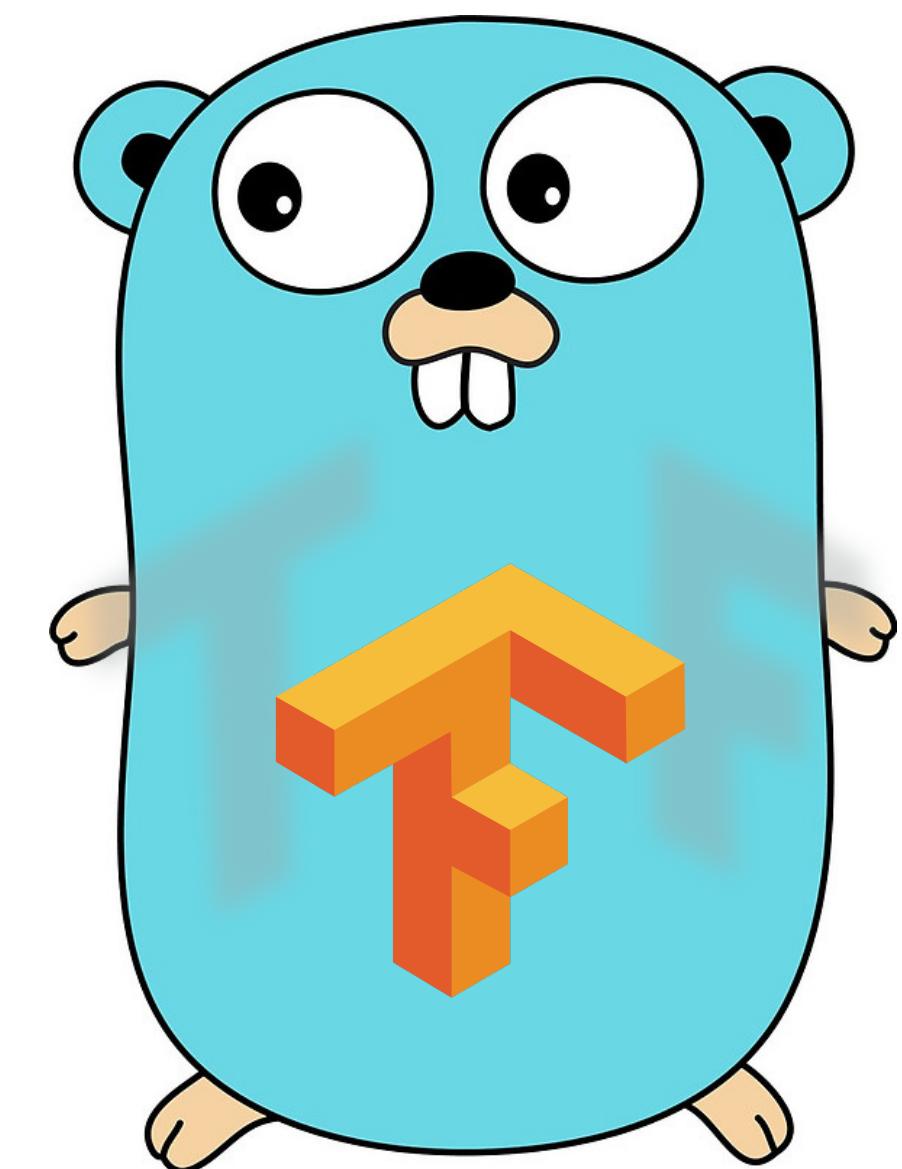
How to ML

1. Define the problem
- 2. Gather data**
 - relevant to the task
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



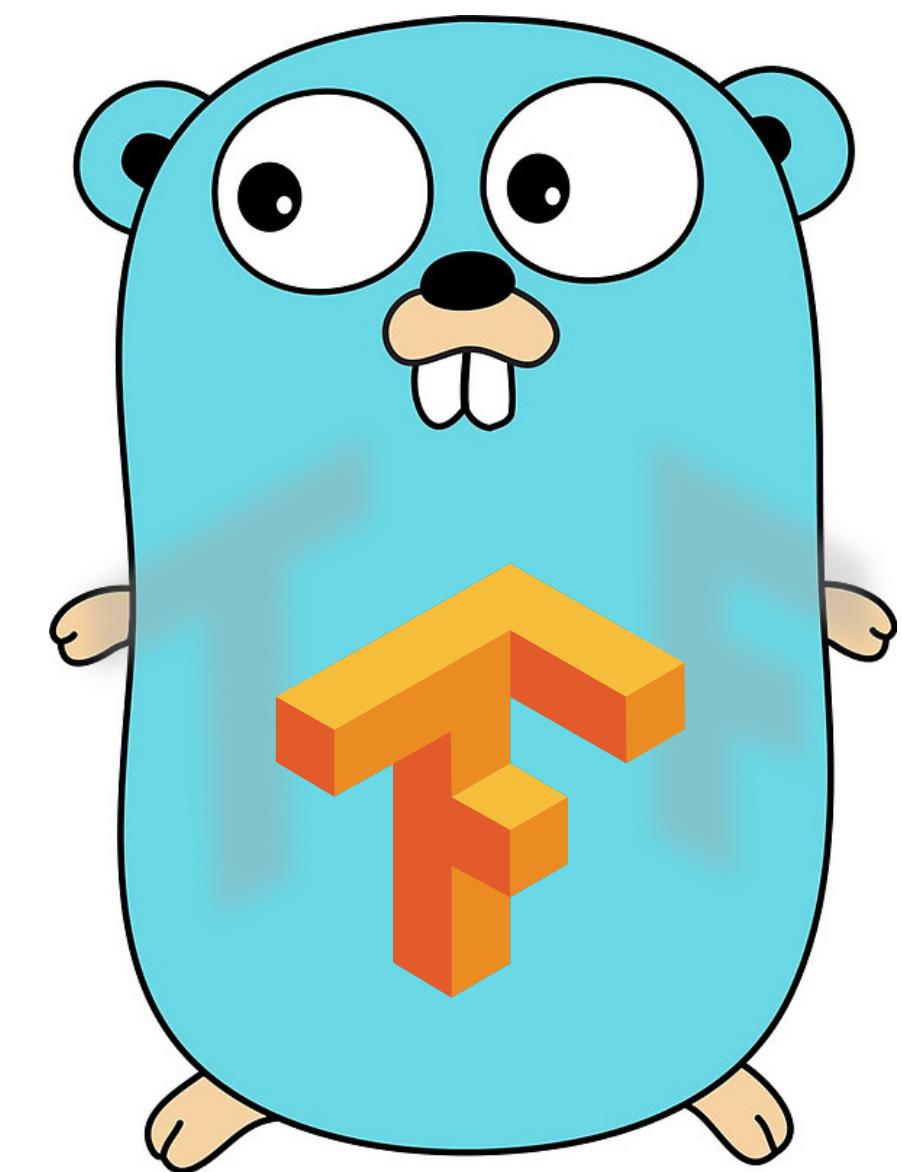
How to ML

1. Define the problem
 2. Gather data
 3. **Prepare data**
 4. Choose a model
 5. Train the model
 6. Evaluate the model
 7. Tune the hyperparameters
 8. Predict
- [] clean and pre-process
randomize
split: train/test



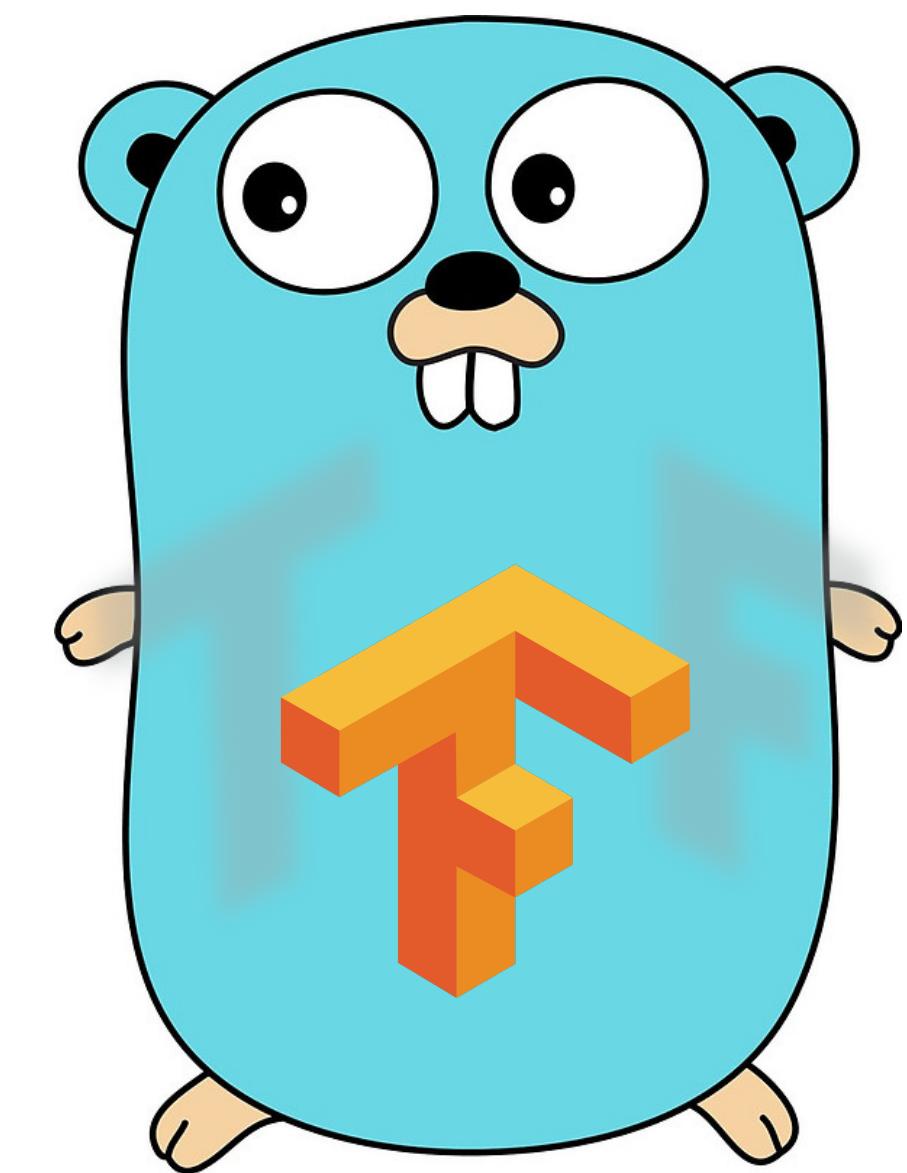
How to ML

1. Define the problem
 2. Gather data
 3. **Prepare data**
 4. Choose a model
 5. Train the model
 6. Evaluate the model
 7. Tune the hyperparameters
 8. Predict
- [] clean and pre-process
randomize
split: train/test
75/25



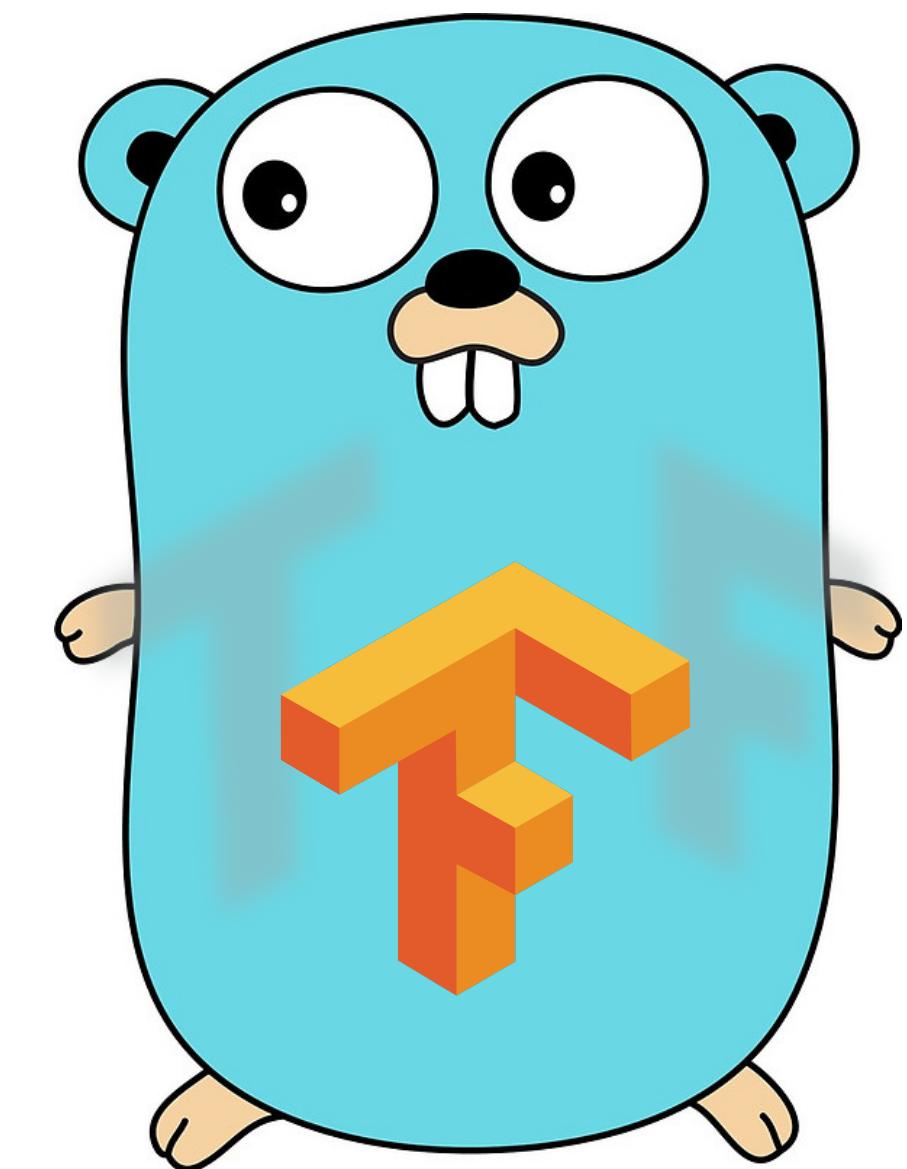
How to ML

1. Define the problem
 2. Gather data
 3. Prepare data
 - 4. Choose a model**
 5. Train the model
 6. Evaluate the model
 7. Tune the hyperparameters
 8. Predict
- learning task
input type
possible number
of categories



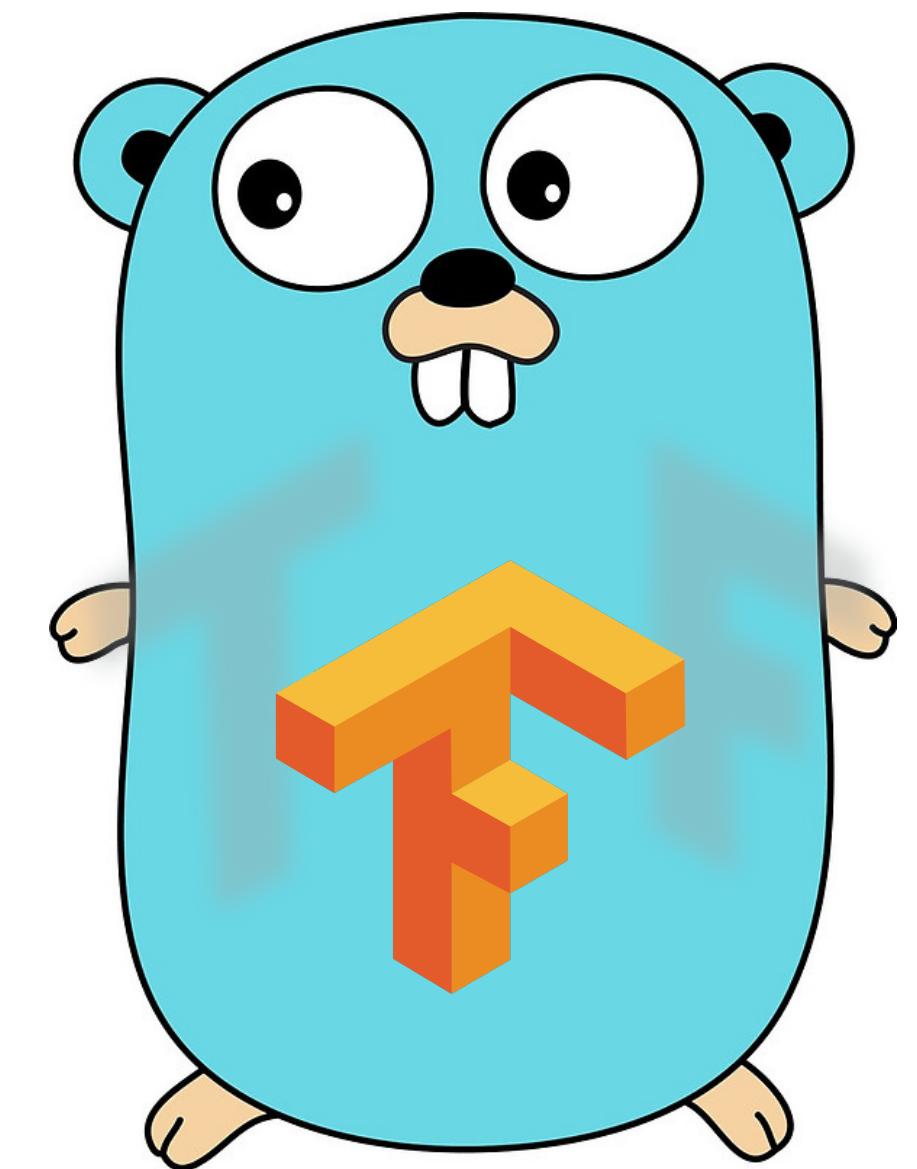
How to ML

1. Define the problem
 2. Gather data
 3. Prepare data
 4. Choose a model
 - 5. Train the model**
 6. Evaluate the model
 7. Tune the hyperparameters
 8. Predict
- assign random values
predict the train data
adjust weights



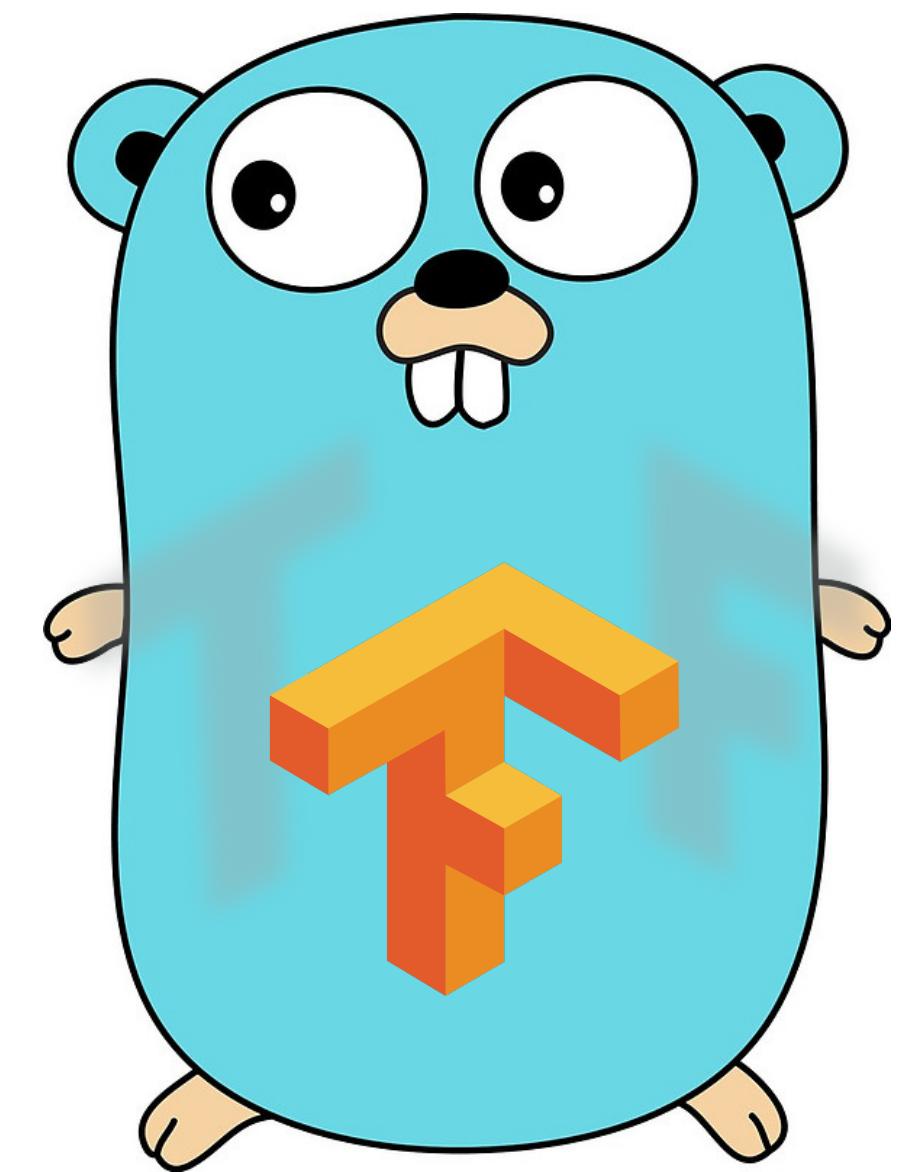
How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
- 6. Evaluate the model**
 - check test data metrics
7. Tune the hyperparameters
8. Predict



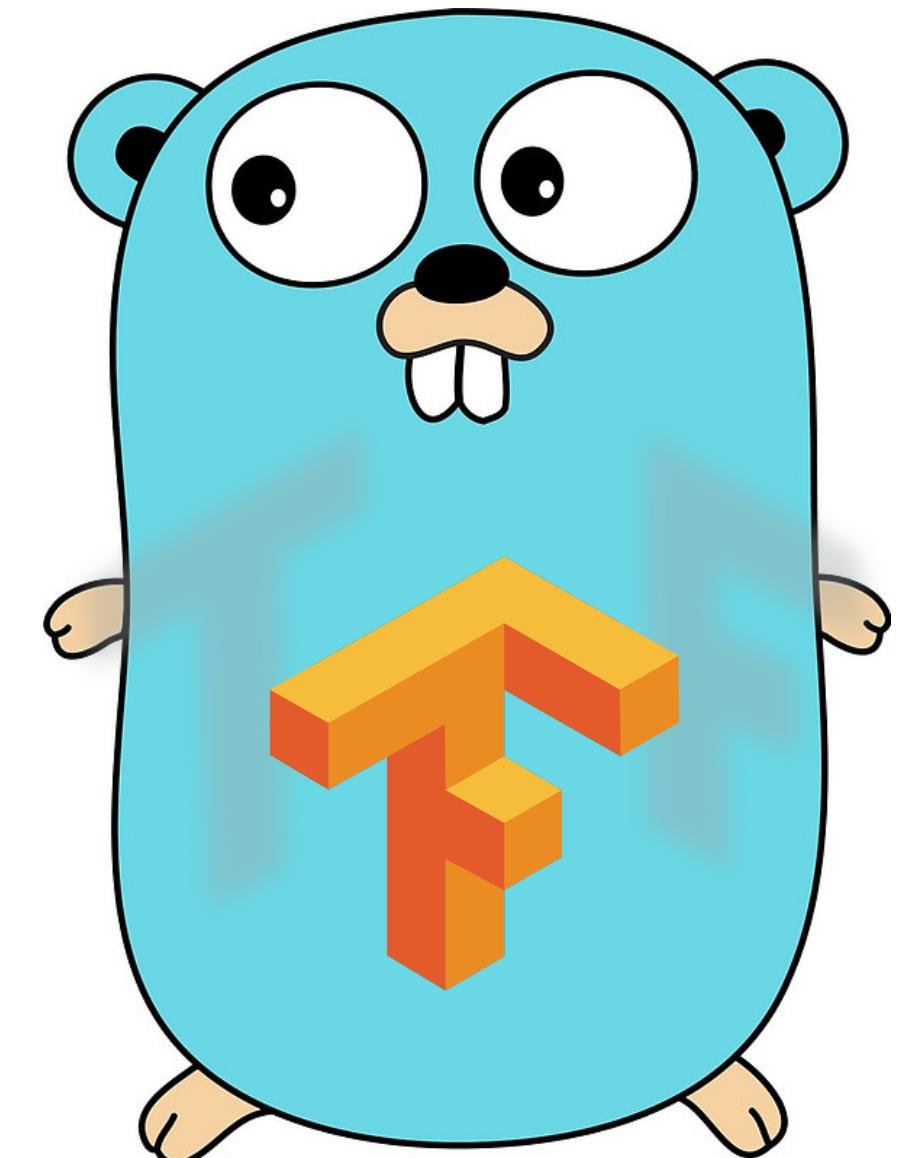
How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
- 7. Tune the hyperparameters**
 - or, fine tune
8. Predict



How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
- 8. Predict**



TensorFlow



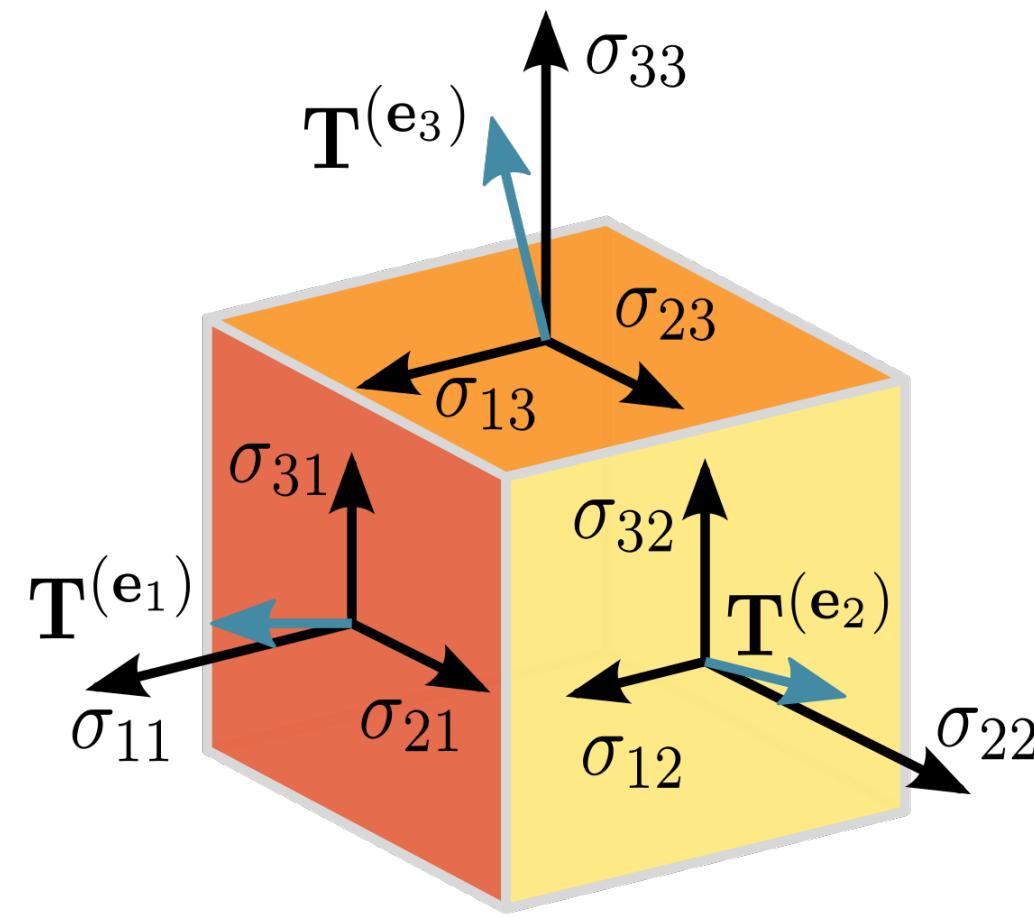


TensorFlow

TensorFlow is an open-source software
for Machine Intelligence,
used mainly for
Machine Learning applications
such as neural networks.



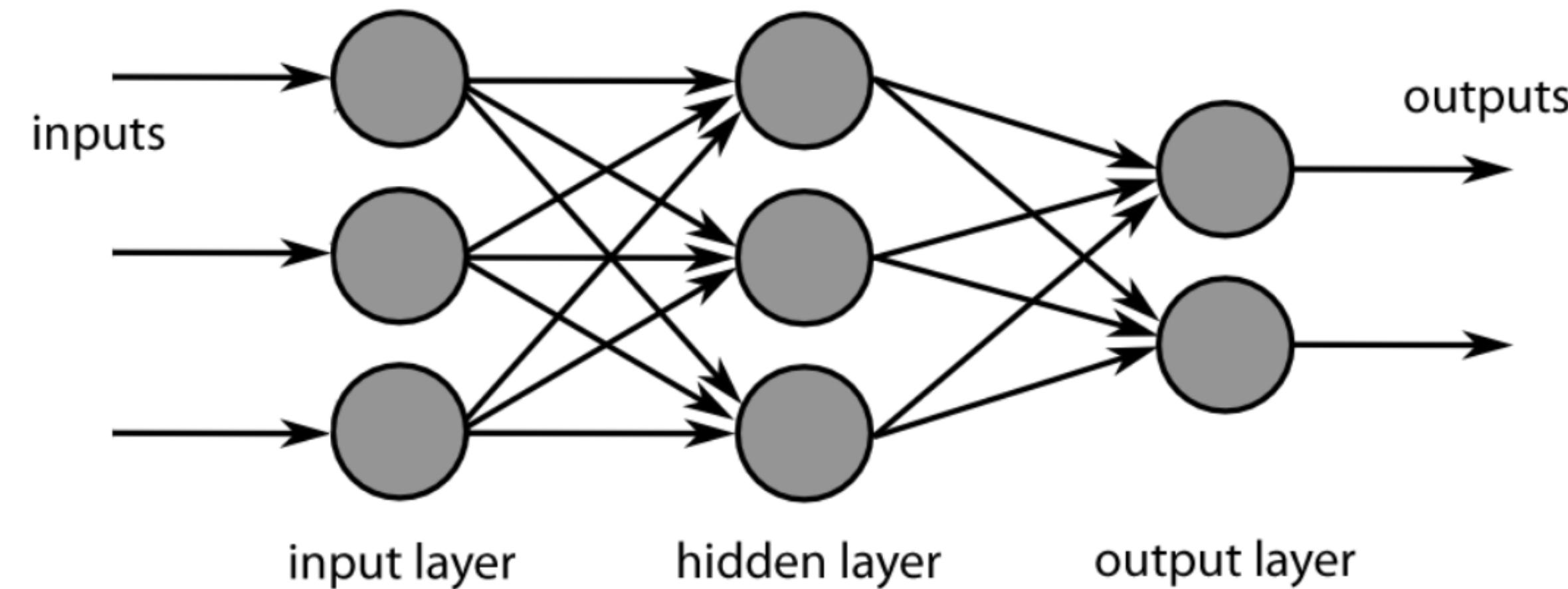
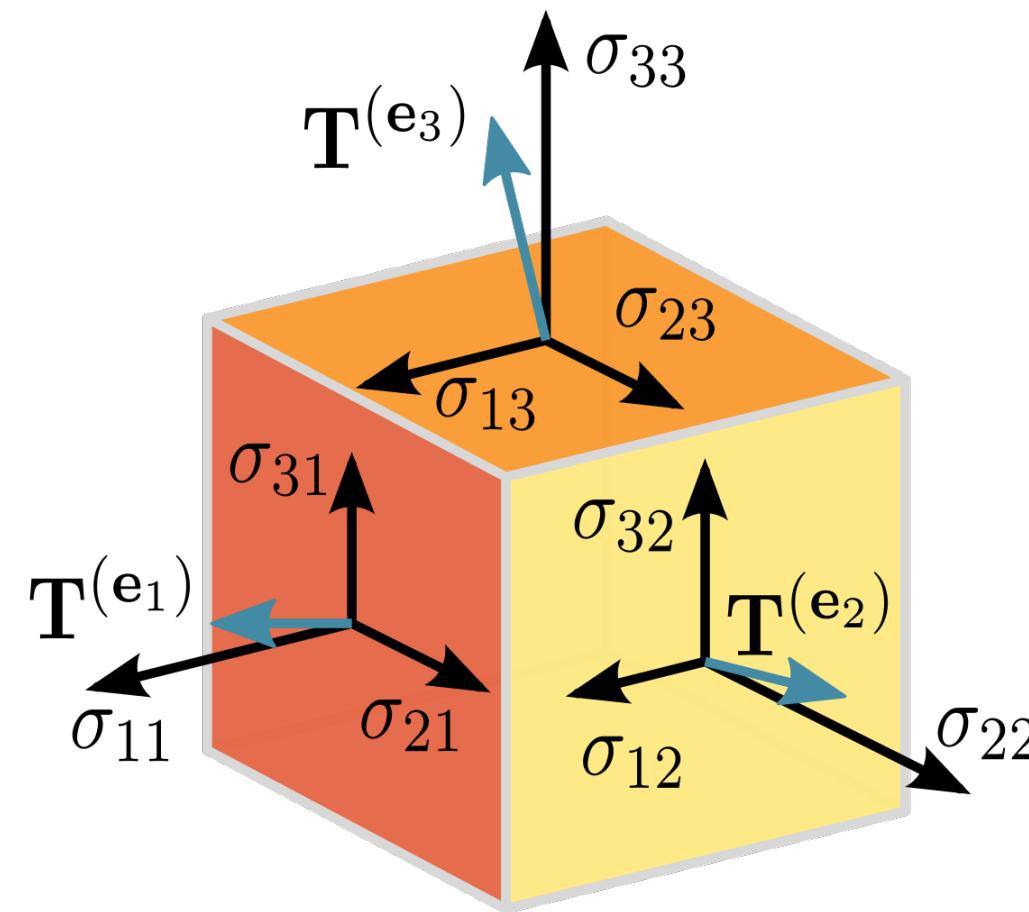
TensorFlow is an open-source software for Machine Intelligence, used mainly for machine learning applications such as neural networks.



A tensor is a generalization
of vectors and matrices to
potentially higher dimensions

1. data type
2. shape
 - number of dimensions
 - number of values / dimension

TensorFlow is an open-source software for Machine Intelligence, used mainly for machine learning applications such as neural networks.



A tensor is a generalization of vectors and matrices to potentially higher dimensions

1. data type
2. shape
 - number of dimensions
 - number of values / dimension

The flow part comes to describe:
- the graph (model) is a set of nodes (operations)
- the data (tensors) "flows" through those nodes, undergoing mathematical manipulation

You can look at, and evaluate, any node of the graph

TensorFlow

- Community driven
- Becoming friendly for developers
 - AutoML: automates ML models design
 - TF Hub: repo for modules
 - Black-box tools built on top of TF

TensorFlow

- Community driven
- Becoming friendly for developers
 - **AutoML: automates ML models design**
 - TF Hub: repo for modules
 - Black-box tools built on top of TF

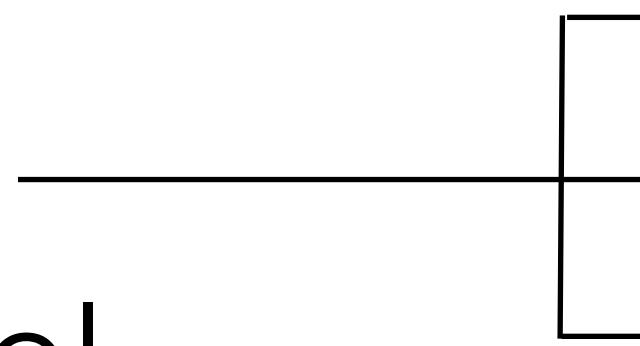
How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
- 7. Tune the hyperparameters**
 - or, fine tune
8. Predict

TensorFlow

- Community driven
- Becoming friendly for developers
 - AutoML: automates ML models design
 - **TF Hub: repo for modules**
 - Black-box tools built on top of TF

How to ML

1. Define the problem
 2. Gather data
 3. Prepare data
 4. Choose a model
 5. **Train the model**
 6. Evaluate the model
 7. Tune the hyperparameters
 8. Predict
- 
- assign random values
 - predict the train data
 - adjust weights

TensorFlow

- Community driven
- Becoming friendly for developers
 - AutoML: automates ML models design
 - TF Hub: repo for modules
 - **Black-box tools built on top of TF**

So, Infrastructure?



Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley, gholt, dg, edavydov, toddphillips}@google.com
Google, Inc.

Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison
{ebner, vchaudhary, mwyong, jfcrespo, dennison}@google.com
Google, Inc.

Abstract

Machine learning offers a fantastically powerful toolkit for building useful complex prediction systems quickly. This paper argues it is dangerous to think of these quick wins as coming for free. Using the software engineering framework of *technical debt*, we find it is common to incur massive ongoing maintenance costs in real-world ML systems. We explore several ML-specific risk factors to account for in system design. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns.

1 Introduction

As the machine learning (ML) community continues to accumulate years of experience with live systems, a wide-spread and uncomfortable trend has emerged: developing and deploying ML systems is relatively fast and cheap, but maintaining them over time is difficult and expensive.

Infrastructure

There's a lot more to machine learning than just implementing an ML algorithm.

Infrastructure

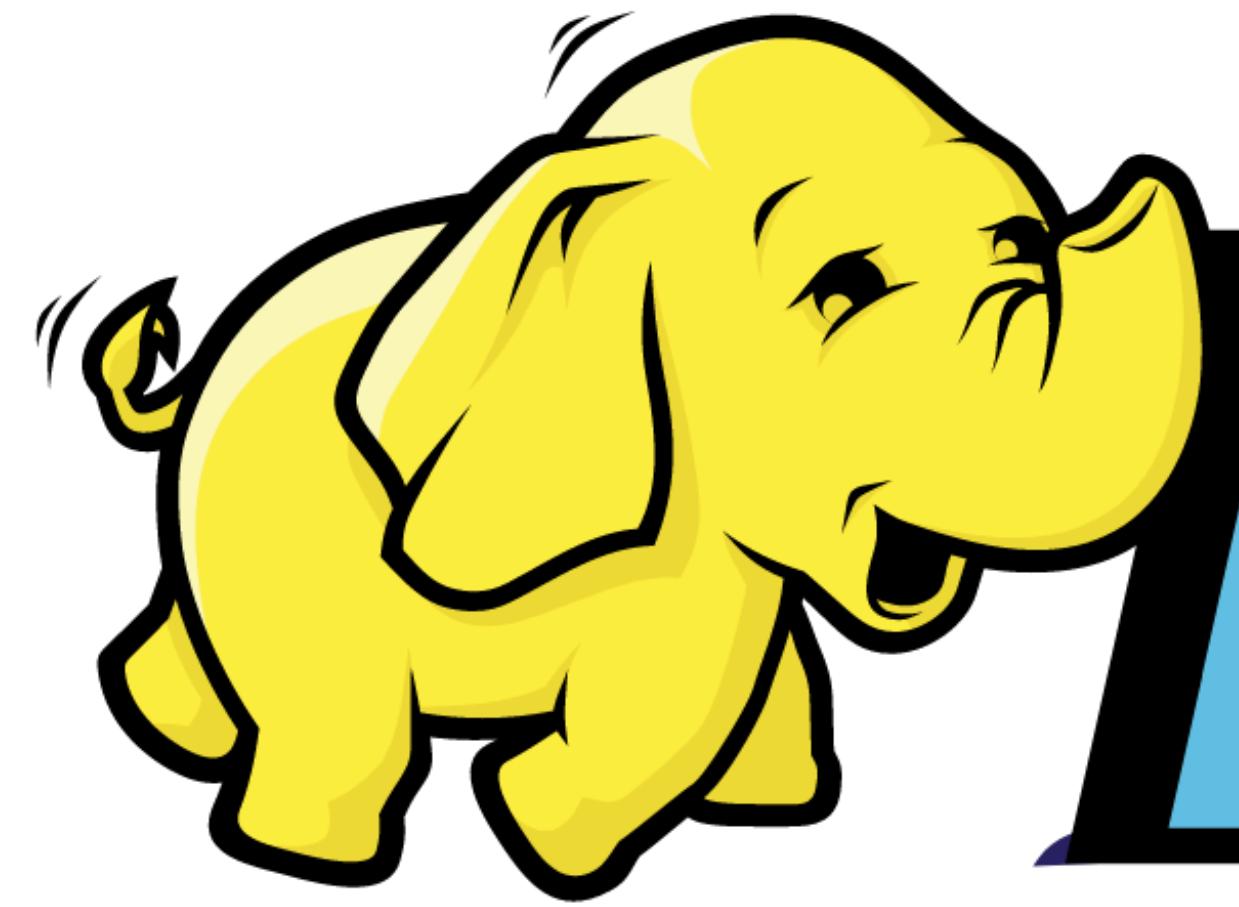
The ML code is at the heart of a real-world production system, but it accounts for **5% or less** of the overall code of that system.

Infrastructure

The ML code is at the heart of a real-world production system, but it accounts for **5% or less** of the overall code of that system.





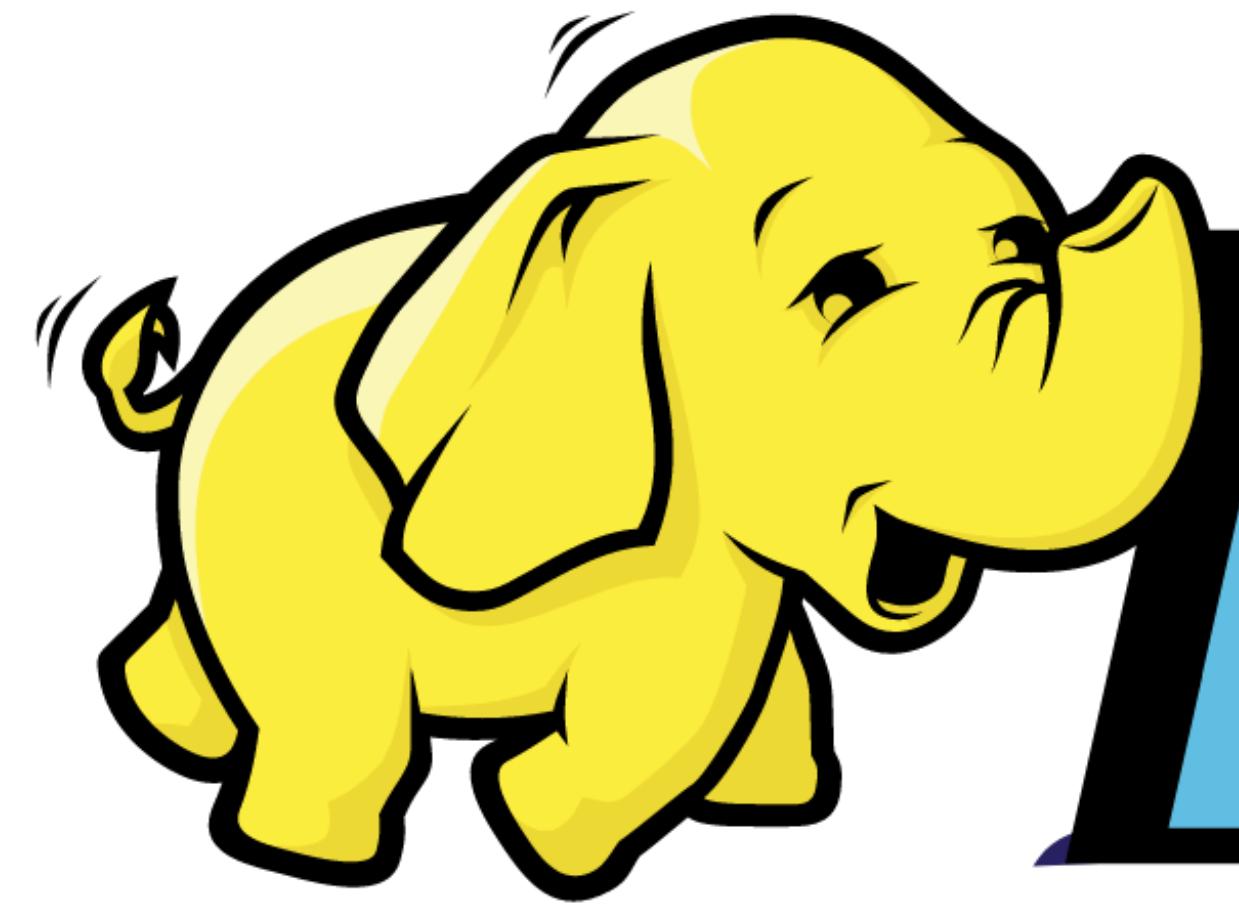


APACHE

hadoop

TM



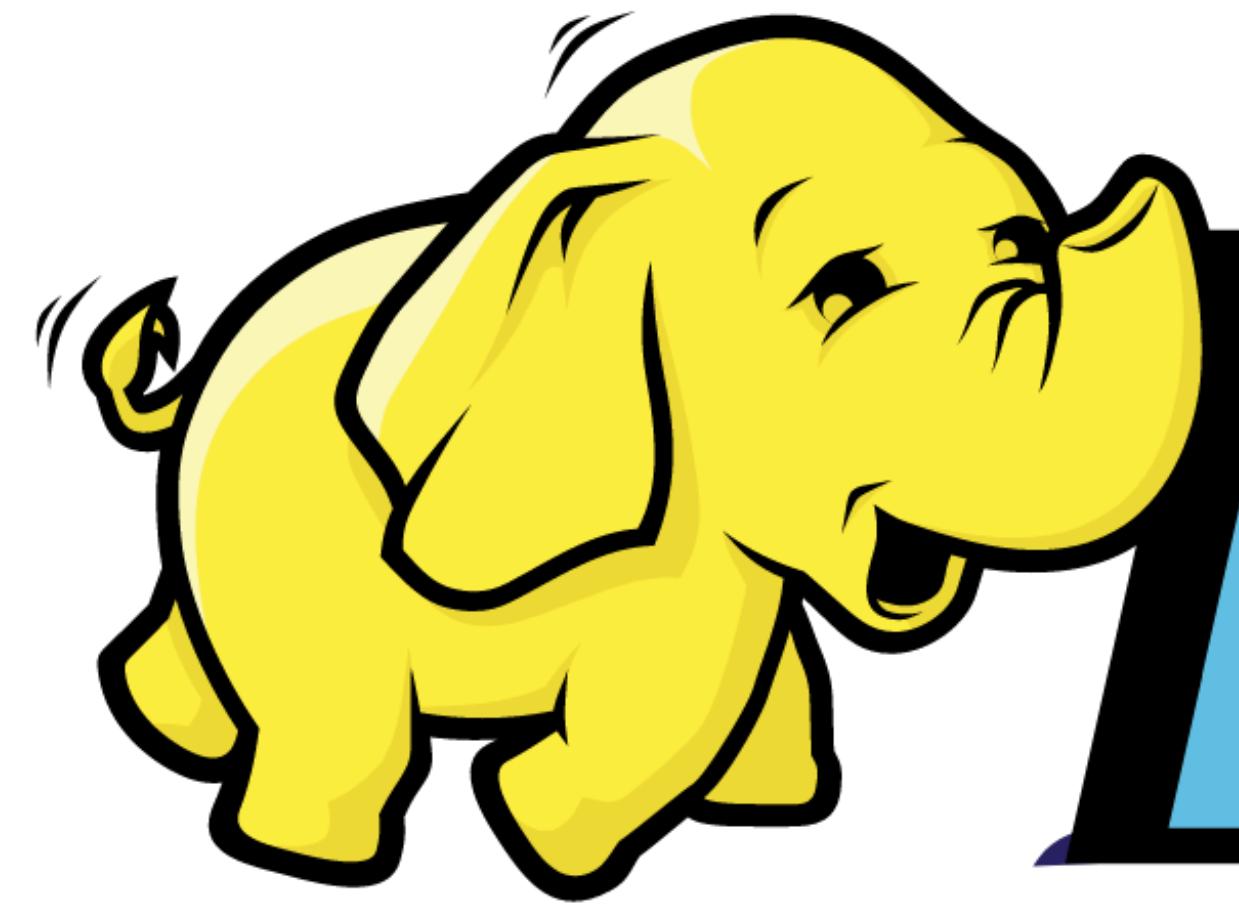


APACHE

hadoop

TM



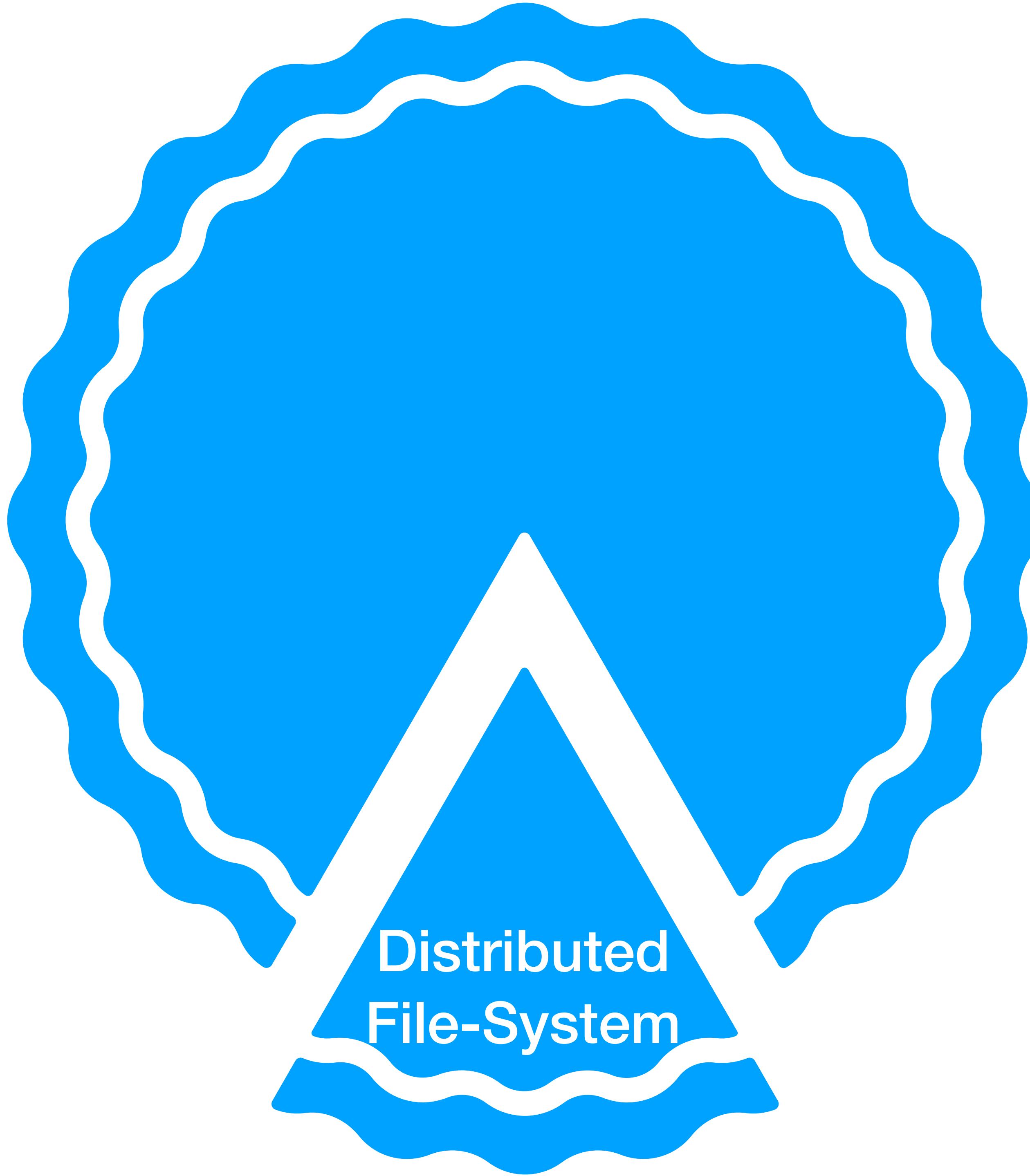


APACHE

hadoop

TM

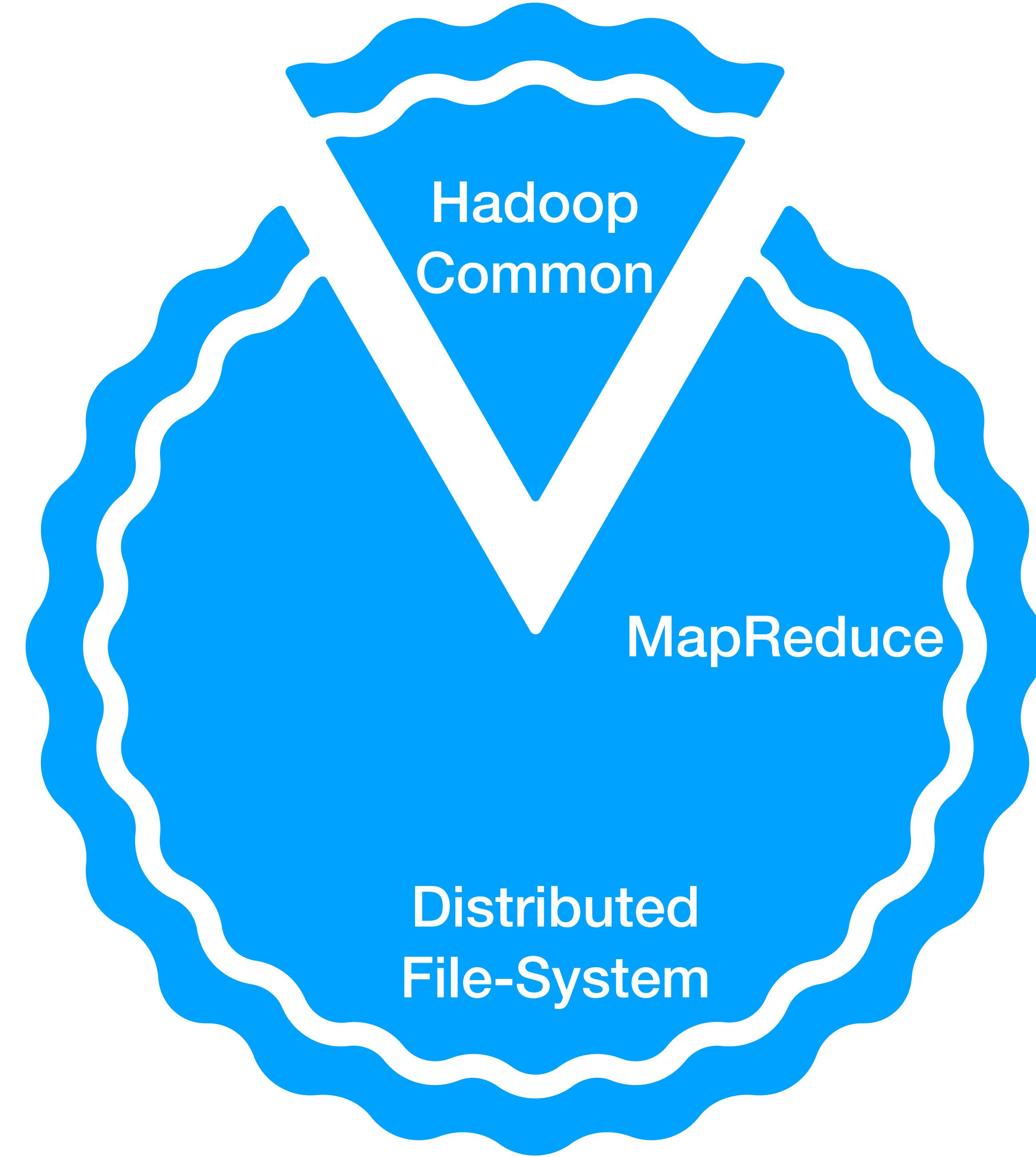


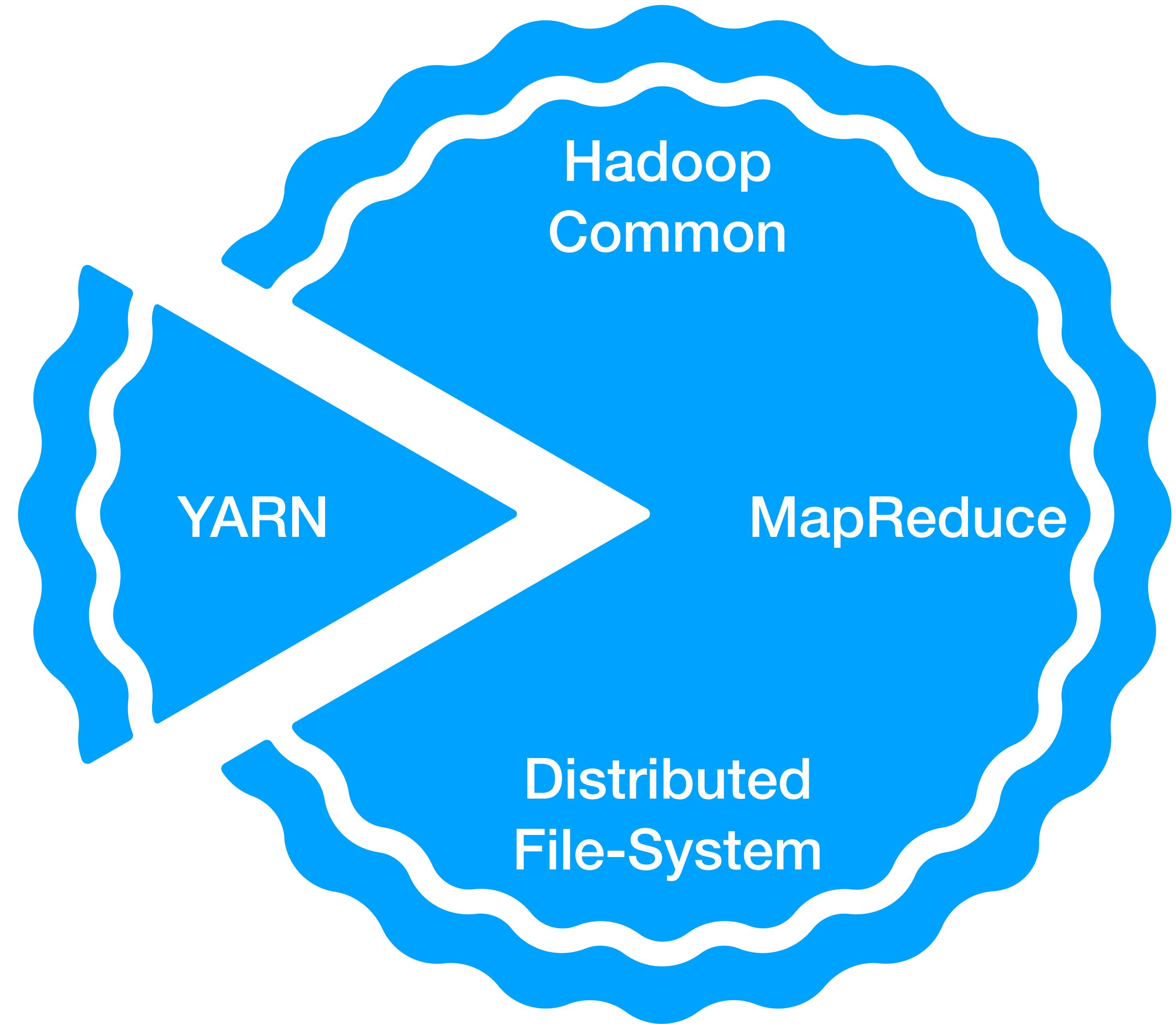




Distributed
File-System

MapReduce







<https://github.com/colinmarc/hdfs>

Data Lake Platform



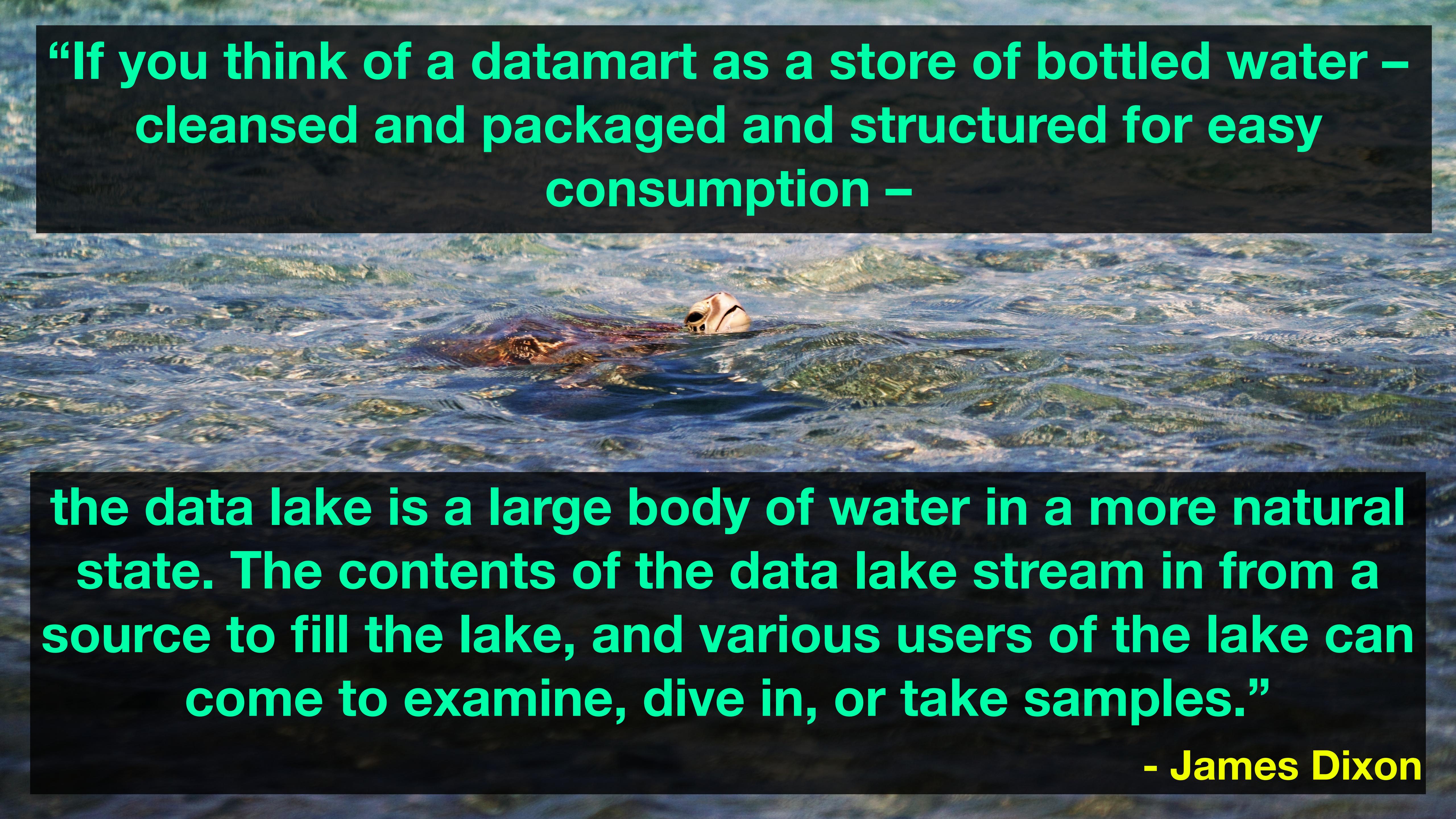




Data Lake Platform



**“If you think of a datamart as a store of bottled water –
cleansed and packaged and structured for easy
consumption –**

A close-up photograph of a sea turtle's head and upper body as it swims through dark, choppy ocean water. The turtle's head is above the surface, showing its eye and nostrils. Its skin is a mottled brown and tan color.

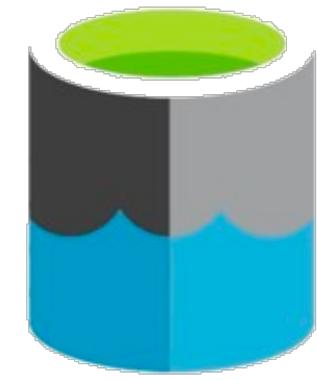
the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples.”

- James Dixon

Data Lake Platform



Google BigQuery



Azure Data Lake

Database



mongoDB®



Cockroach DB



Couchbase

Functions as a Service (aka Serverless)



AWS
Lambda



Azure Functions



Google Cloud
Functions



OPENFAAS

Data Governance



Monitoring

pagerduty



splunk®>

Putting it All Together: Bare Bones Infrastructure





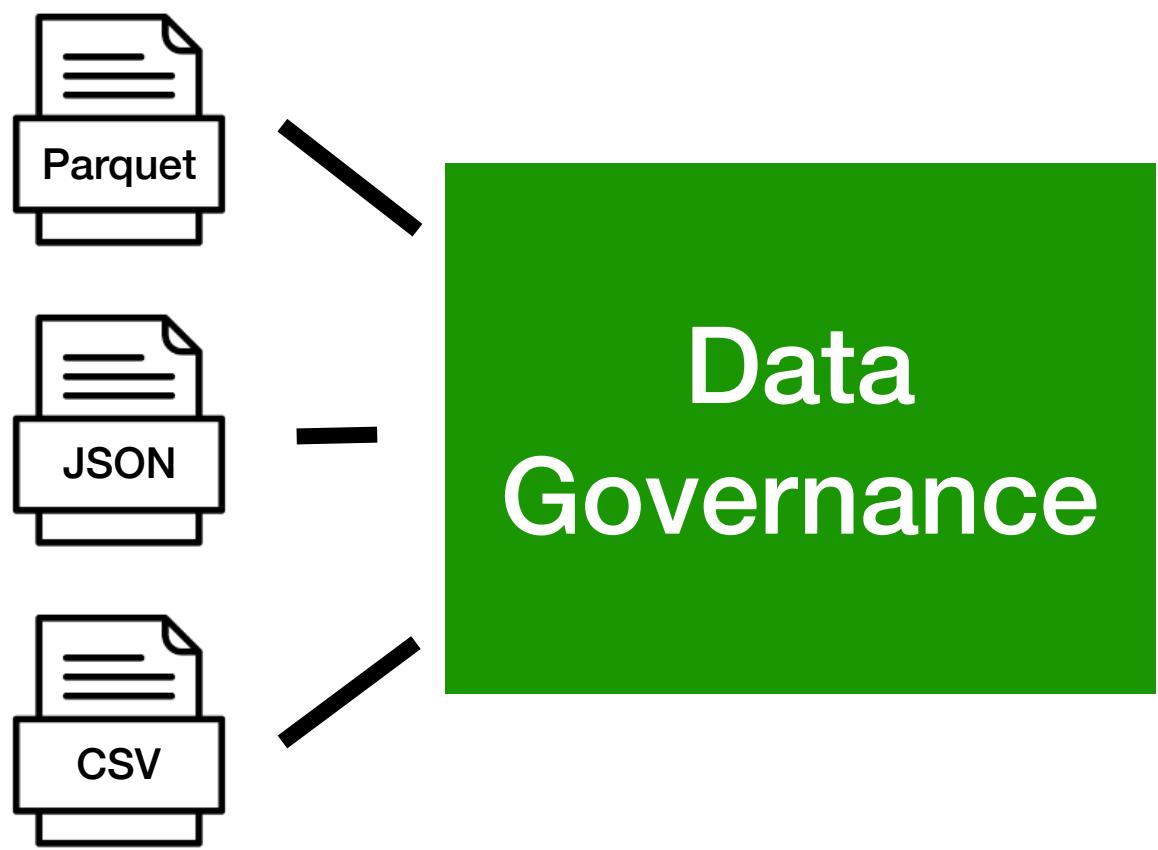
Parquet

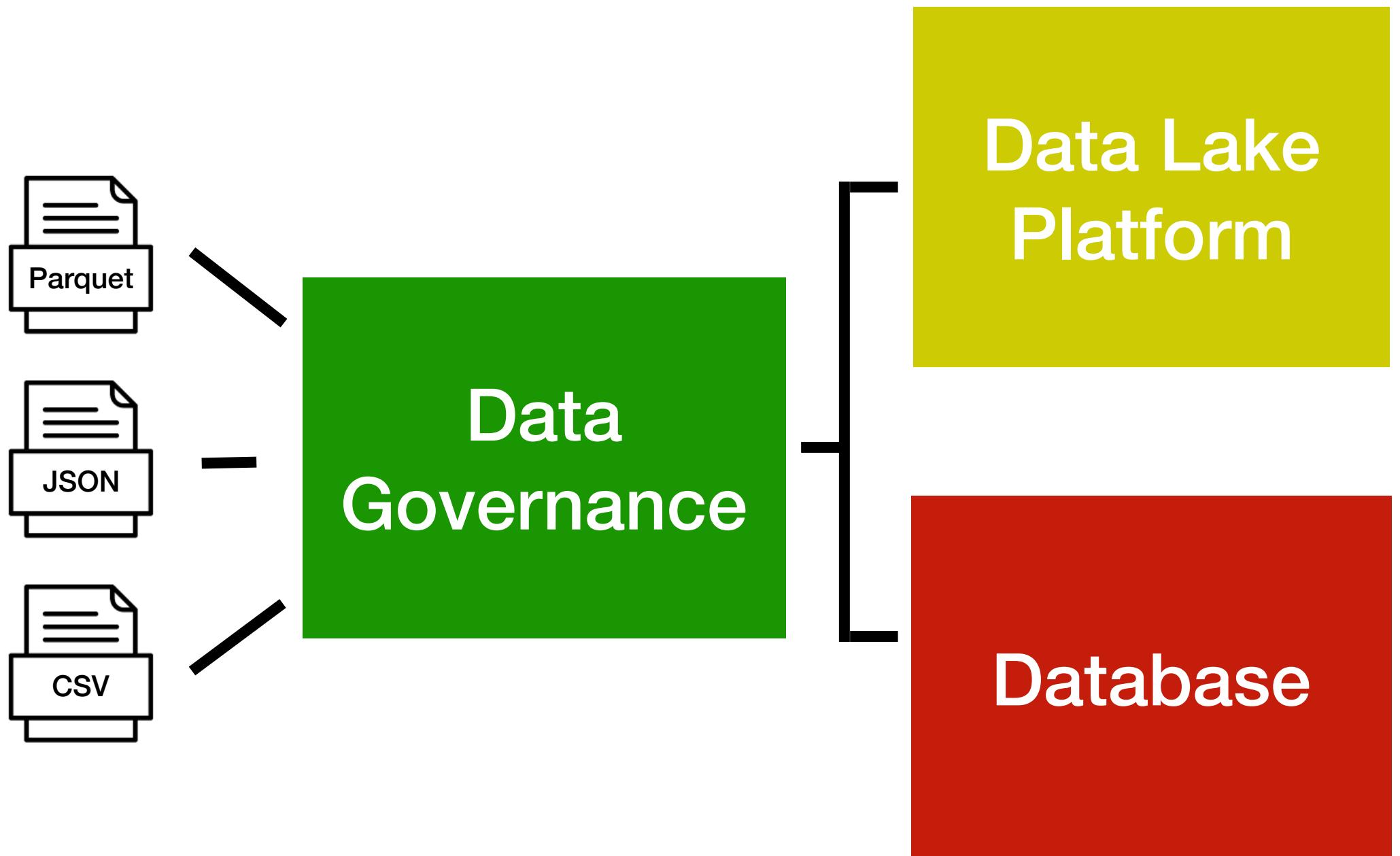


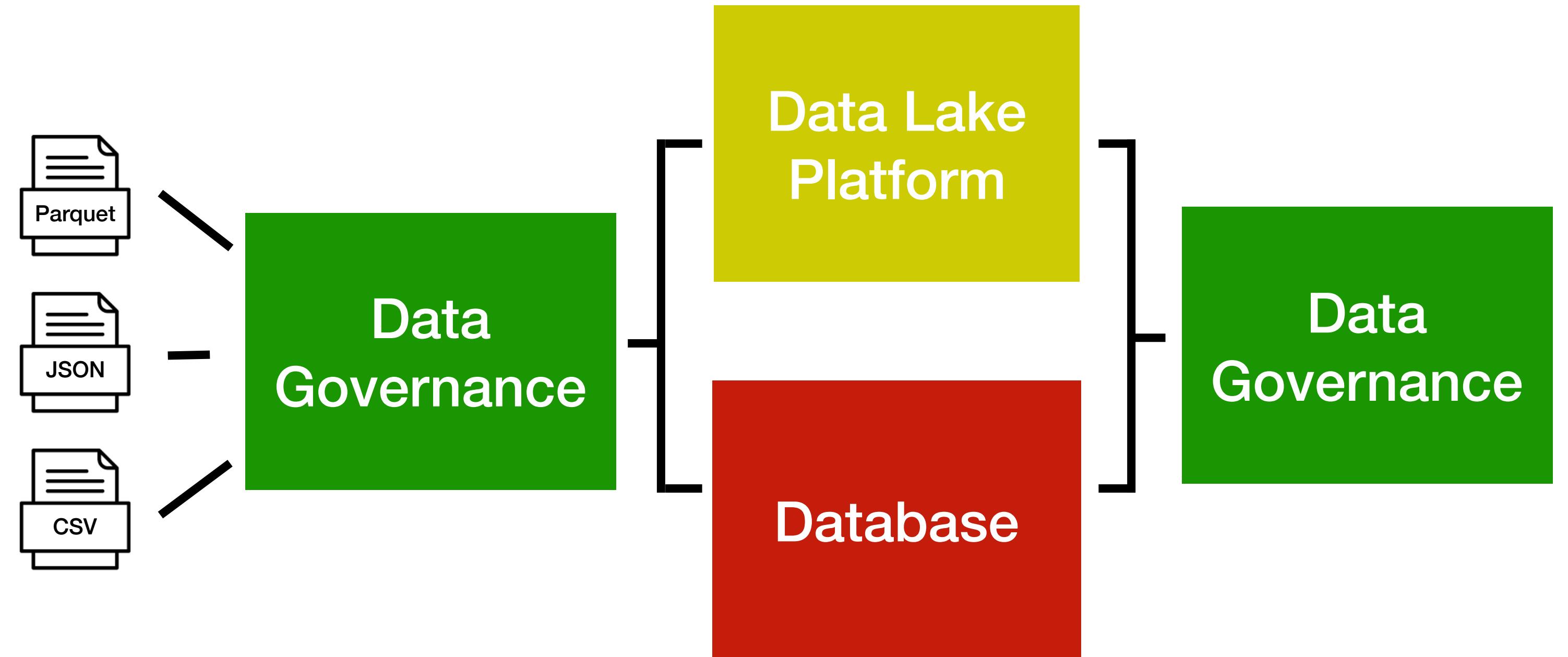
JSON

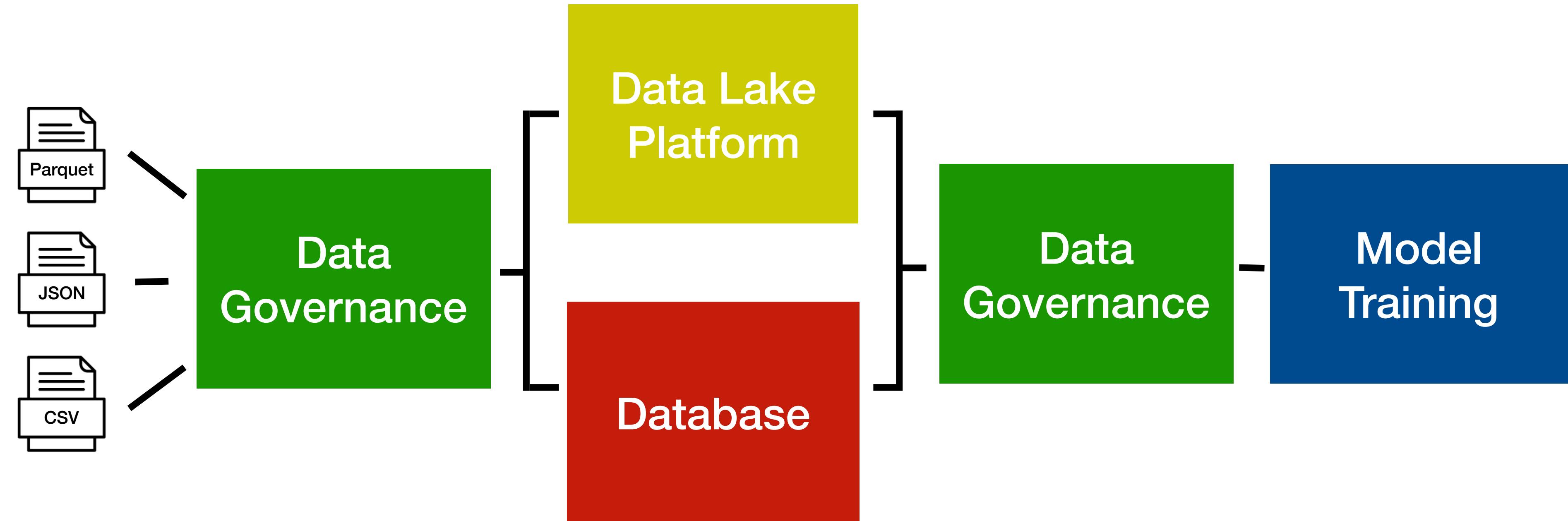


CSV







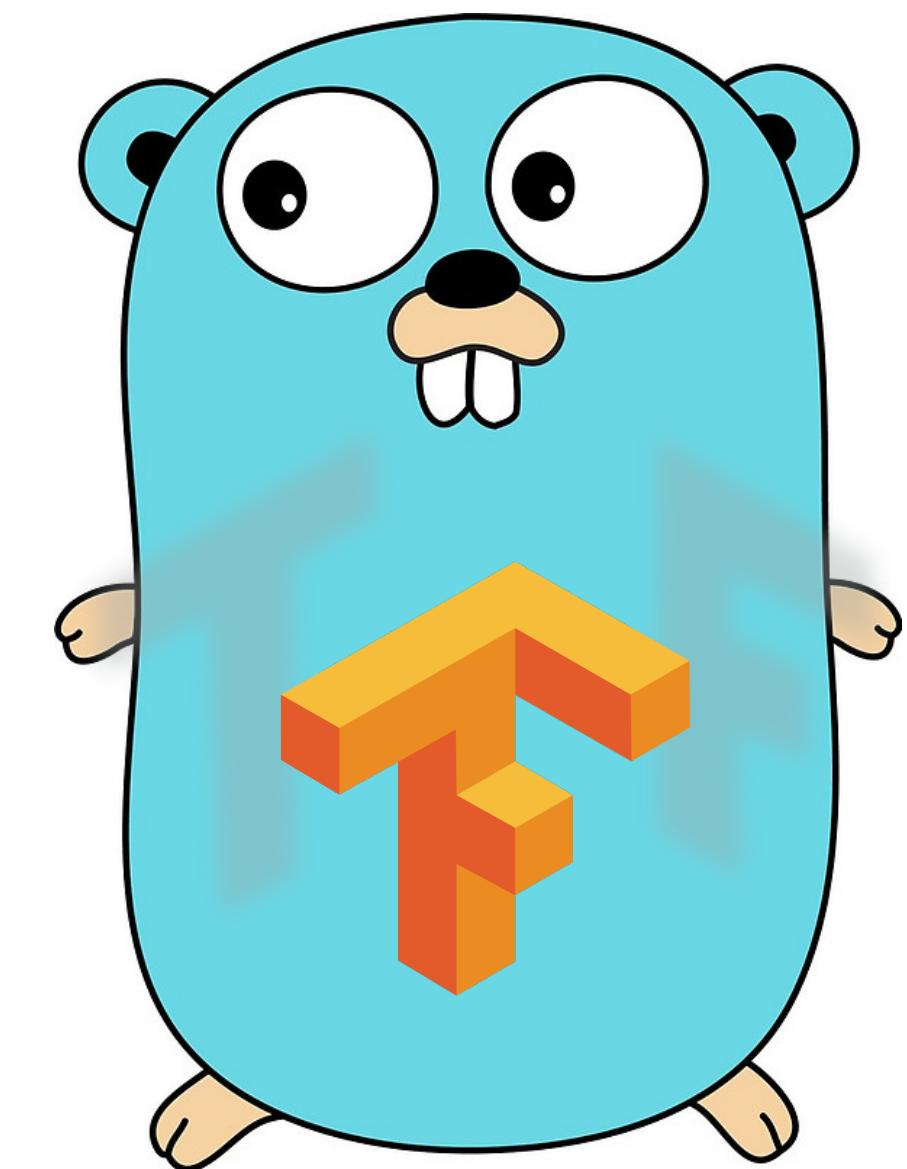


Zooming in on Training



How to ML

1. Define the problem
 2. Gather data
 3. Prepare data
 4. Choose a model
 - 5. Train the model**
 6. Evaluate the model
 7. Tune the hyperparameters
 8. Predict
- assign random values
predict the train data
adjust weights



Types of Training

Static Model



Easier to build and test.



Can only predict things we know about.

Update latency likely measured in hours or days.

Dynamic Model

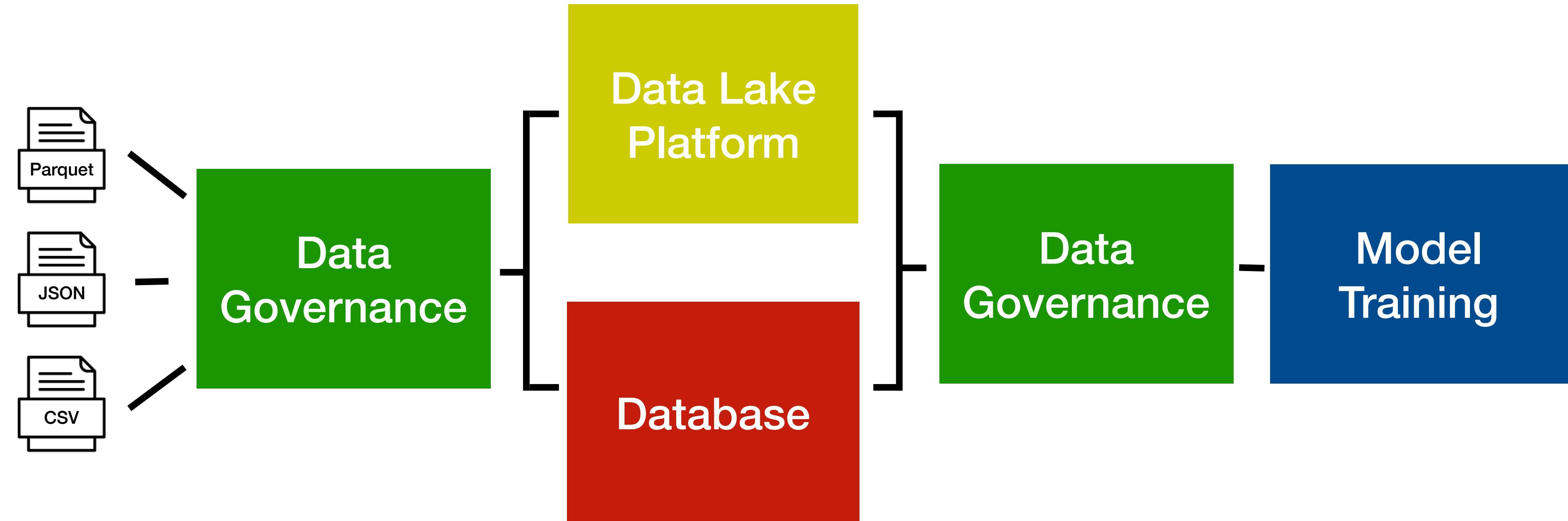
+

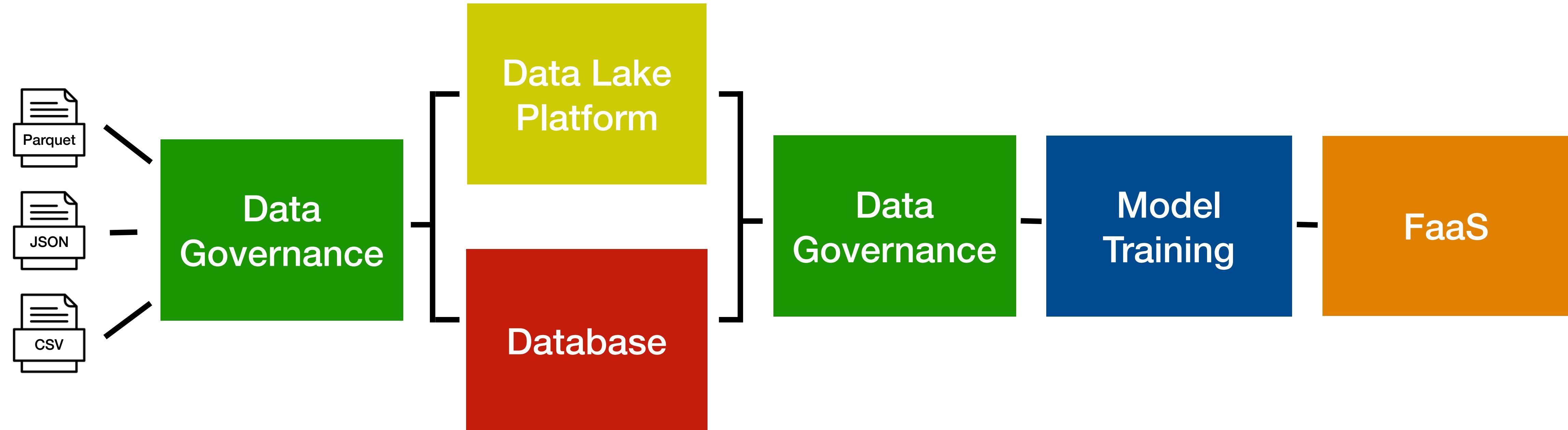
Adapted to changing data,
hence more likely to make
better predictions.

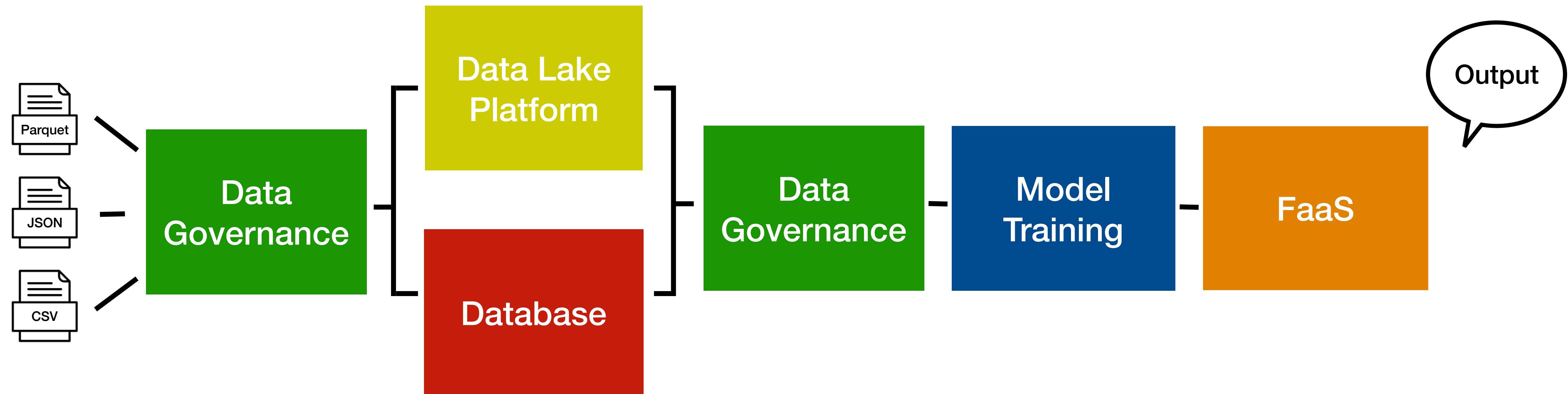
-

Compute intensive, latency
sensitive, may limit model
complexity.

Monitoring needs are more
intensive: outputs and
performance.





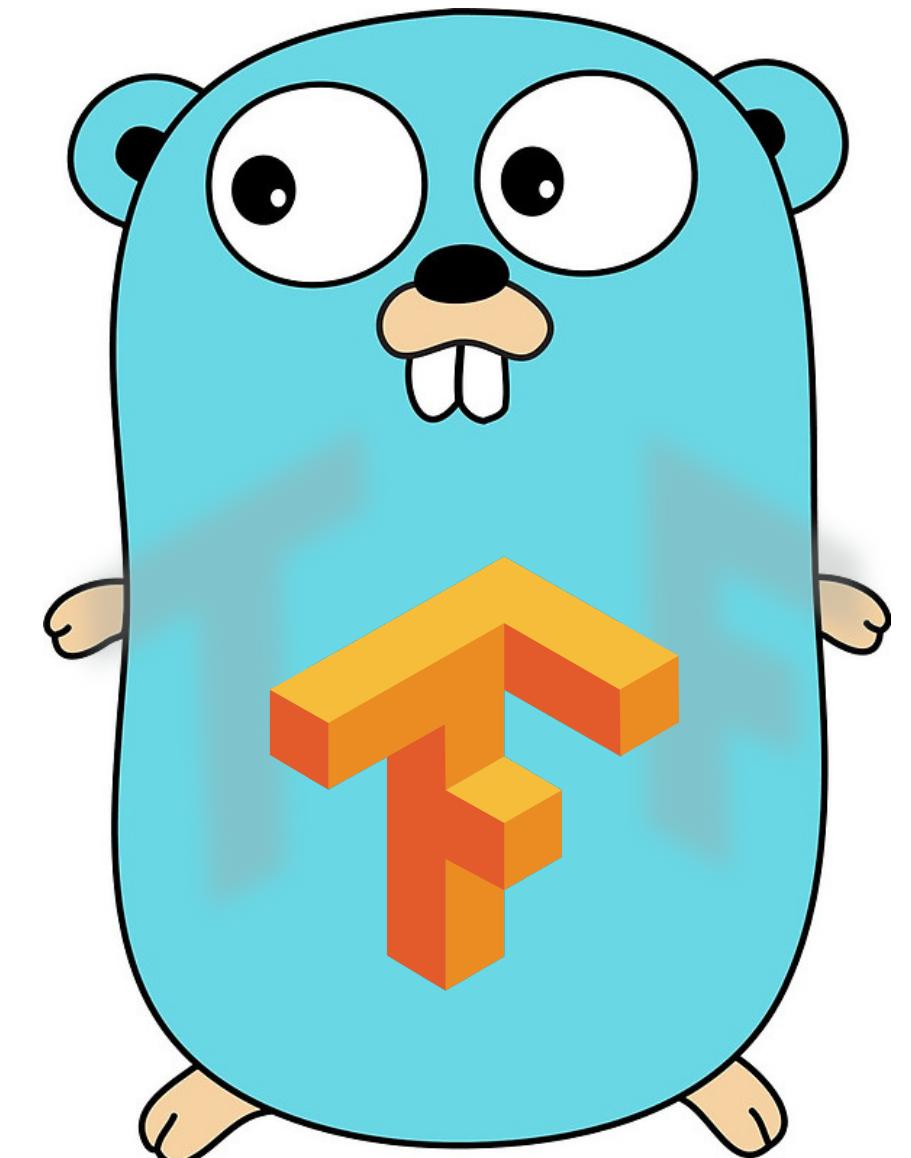


Zooming in on Inference



How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
- 8. Predict**



Types of Inference

Online Inference vs. Offline Inference

Online

+

Can make a prediction on any new item as it comes in — great for long tail.

-

Compute intensive, latency sensitive — may limit model complexity.

Monitoring needs are more intensive.

Offline

+

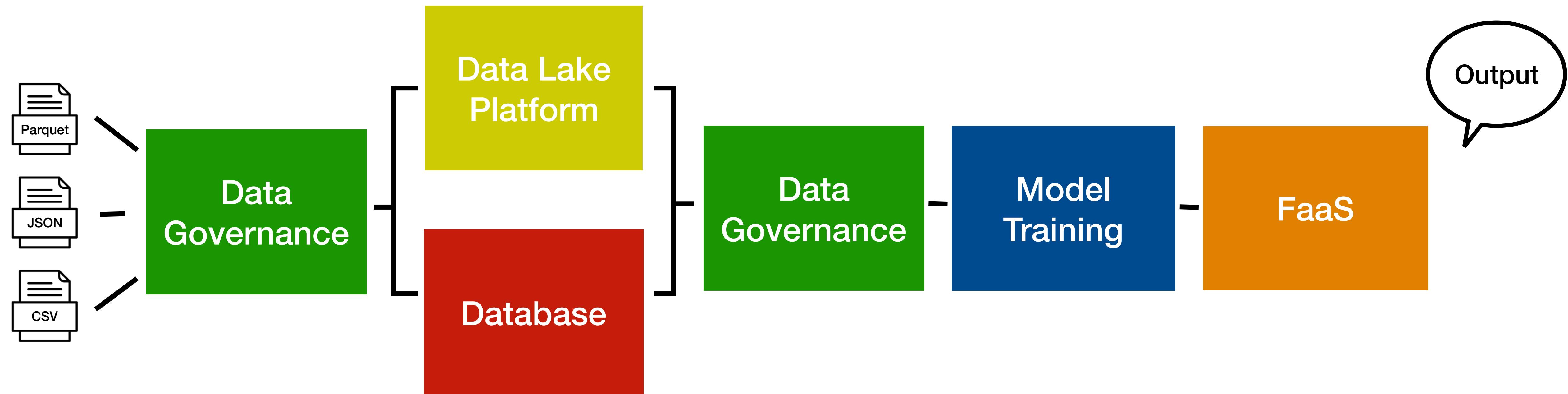
Don't worry much about cost of inference.

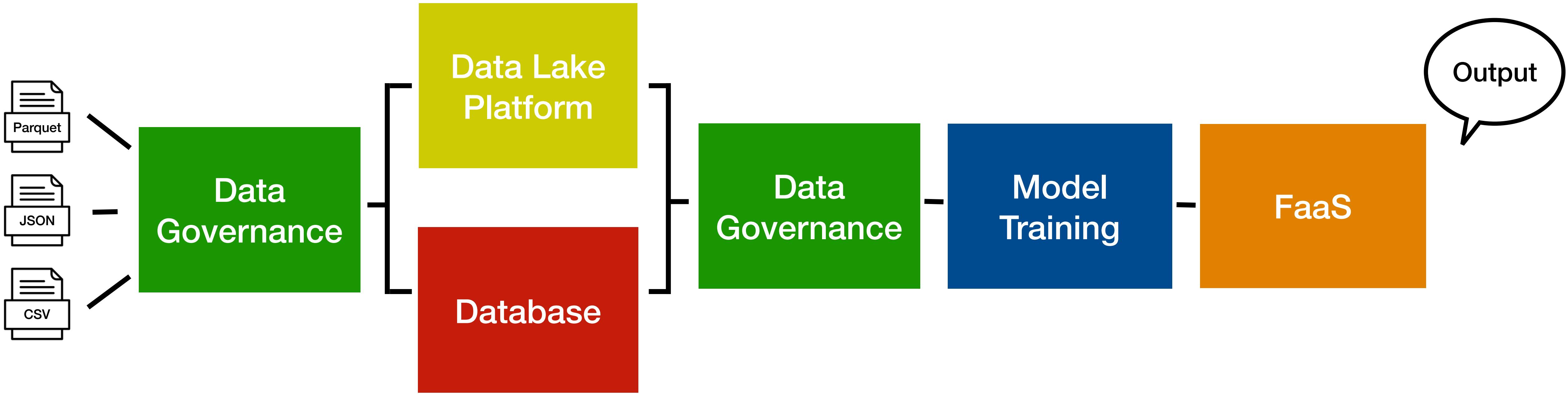
Likely to use batch quota.

Can do post-verification of predictions on data before pushing.

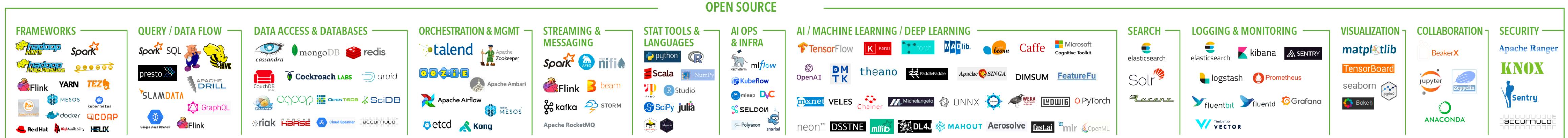
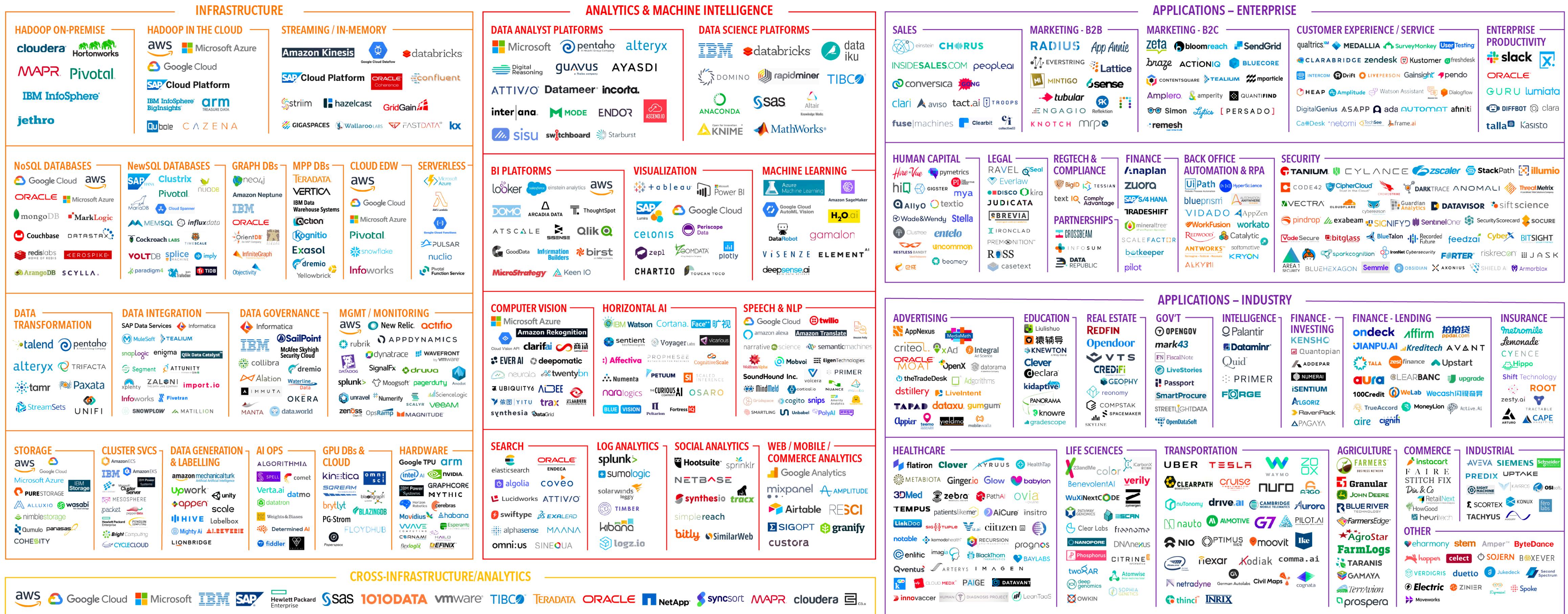
-

Can only predict things we know about.





Monitoring



HADOOP ON-PREMISE



HADOOP IN THE CLOUD



STREAMING / IN-MEMORY



NoSQL DATABASES



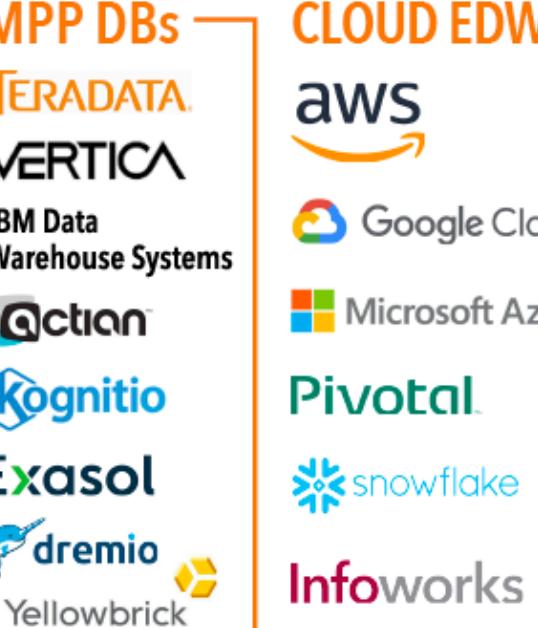
NewSQL DATABASES



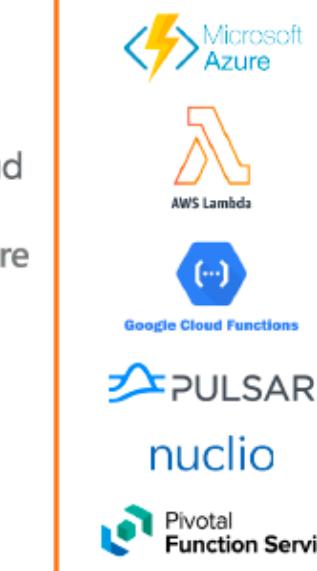
GRAPH DBs



MPP DBs



CLOUD EDW



SERVERLESS



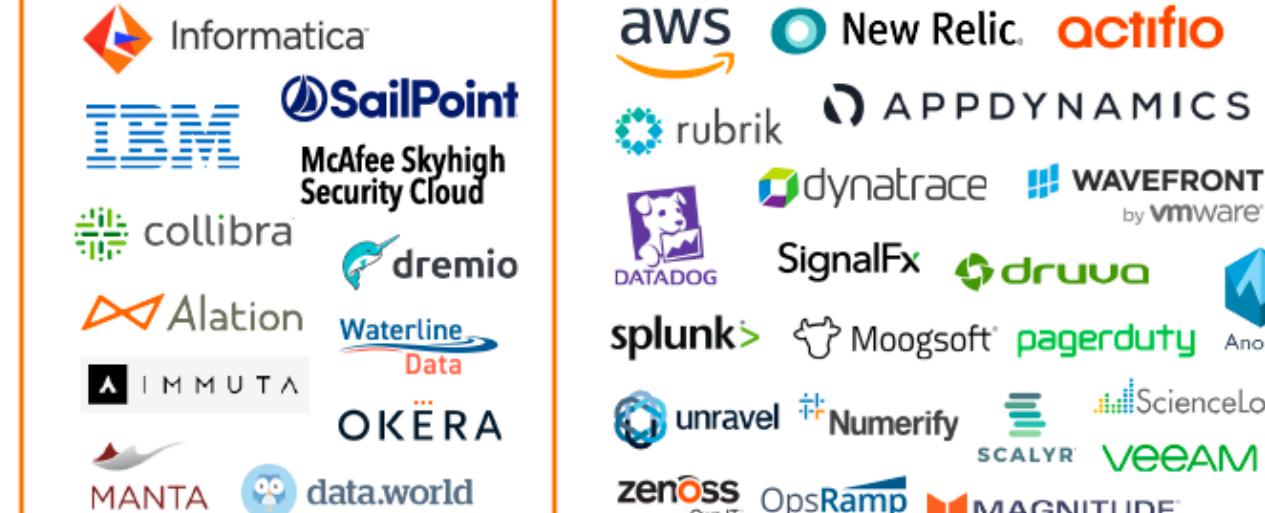
DATA TRANSFORMATION



DATA INTEGRATION



DATA GOVERNANCE



MGMT / MONITORING



STORAGE



CLUSTER SVCS



DATA GENERATION & LABELLING



AI OPS



GPU DBs & CLOUD



HARDWARE



CROSS-INFRASTRUCTURE/ANALYTICS



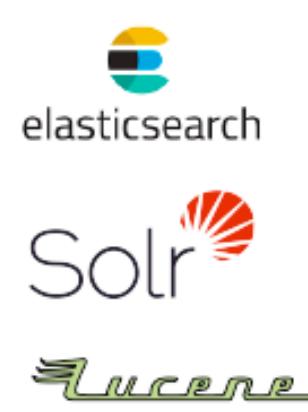
OPEN SOURCE



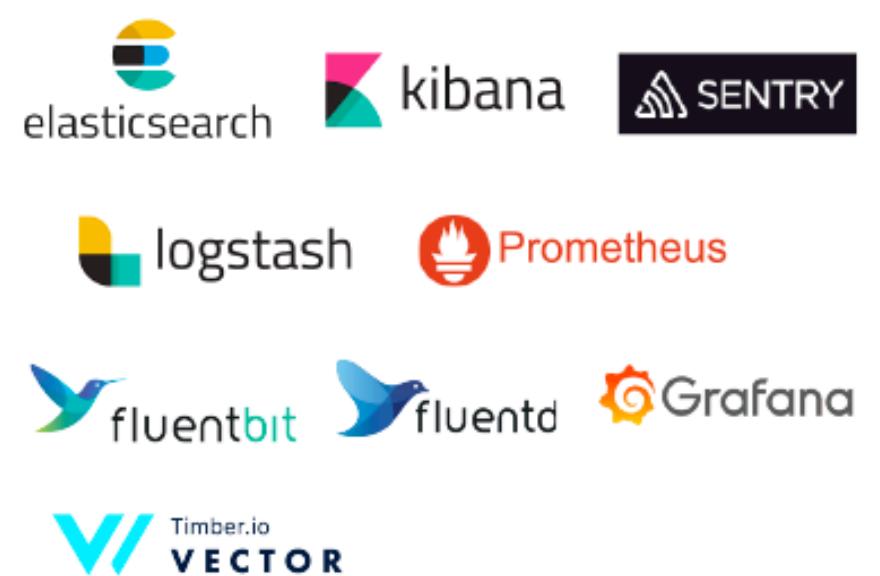
AI / MACHINE LEARNING / DEEP LEARNING



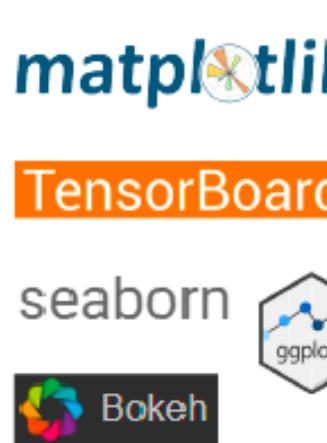
SEARCH



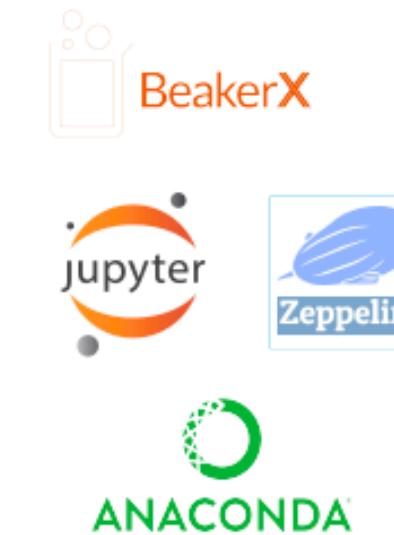
LOGGING & MONITORING –



VISUALIZATION



COLLABORATION



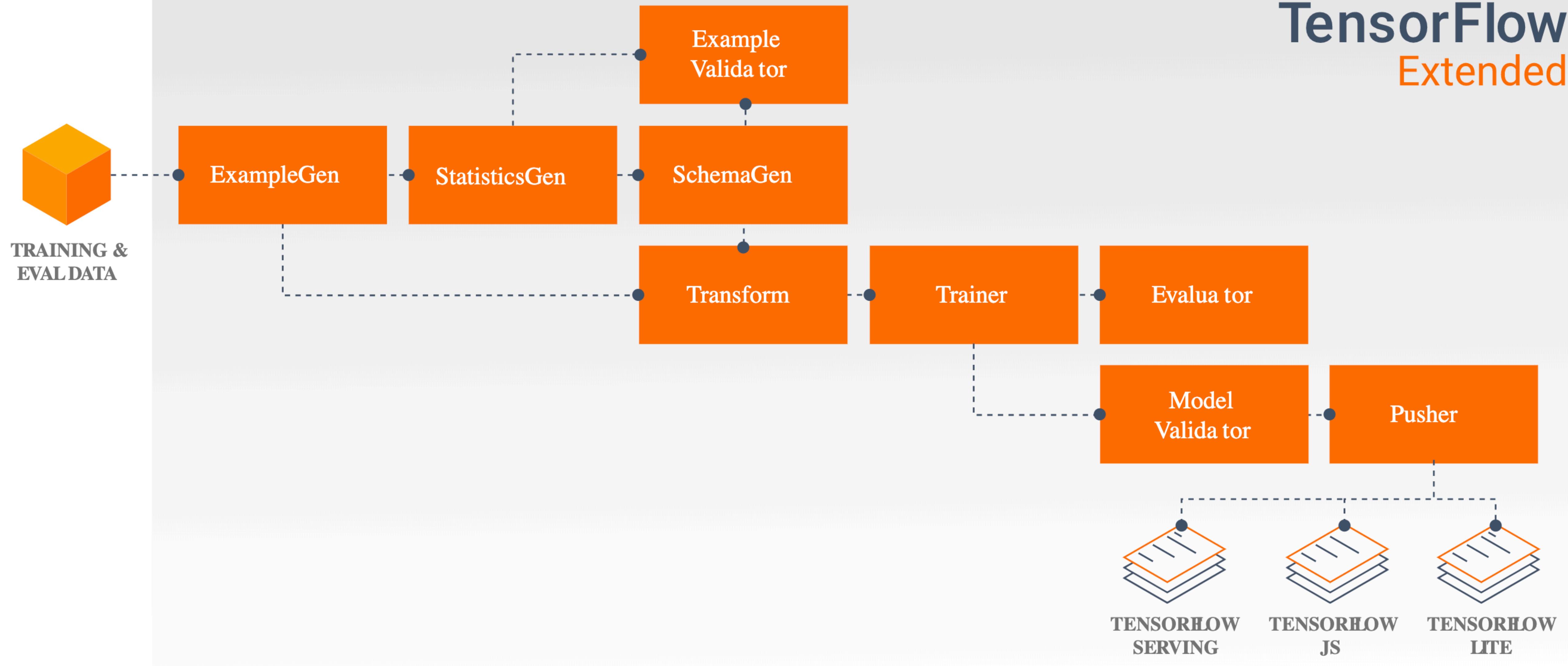
SECURITY —



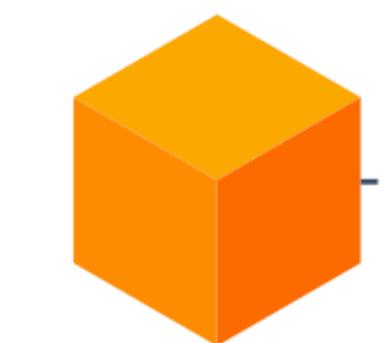
End to End Off the Shelf Solution



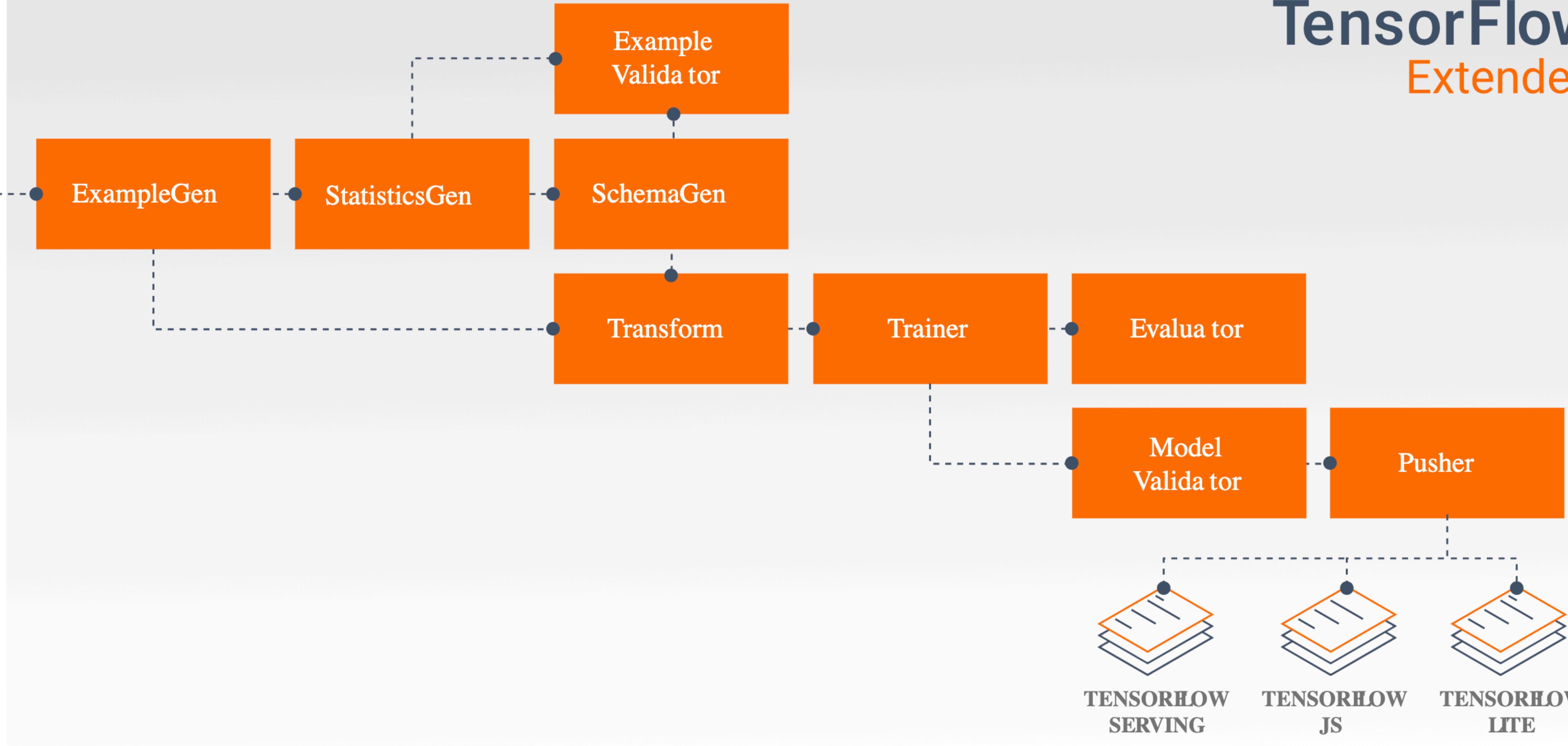
TensorFlow Extended



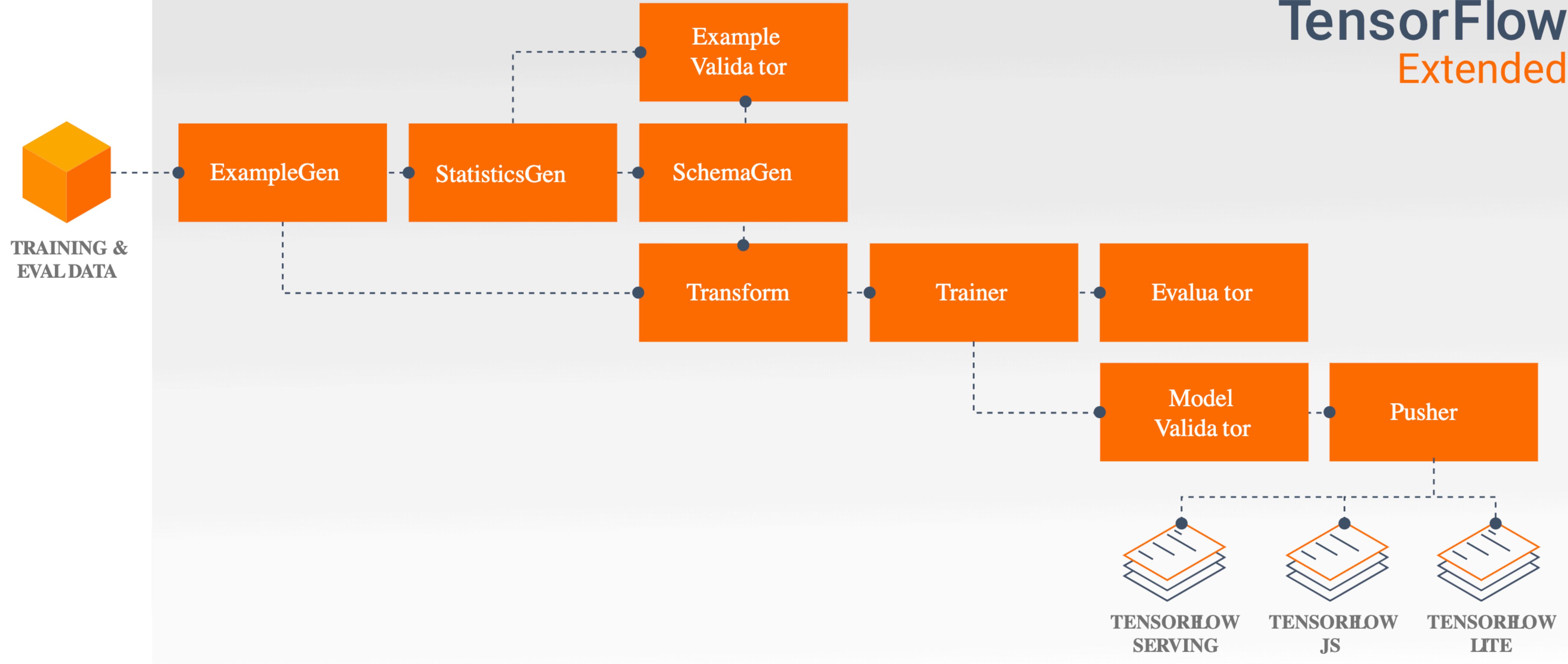
TensorFlow Extended



TRAINING &
EVAL DATA



TensorFlow Extended

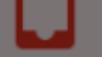


A Concrete Examples





New Message

 C Pa In St Sr Se Dr Ae Bl Co Nat

Recipients

Subject

 A font toolbar with icons for bold, italic, underline, superscript, subscript, and other text styles. Send A U E S A M T

⋮

 X

New Message

Recipients

Subject

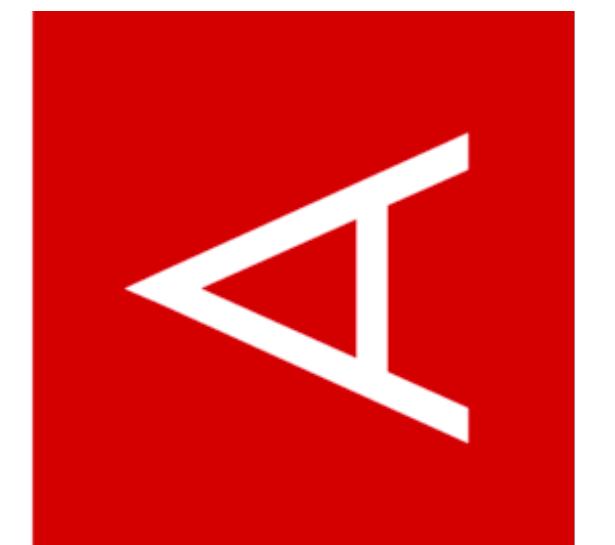
- Sender email
- Receiver email
- Email title
- Email content
- Header
- Footer



Send

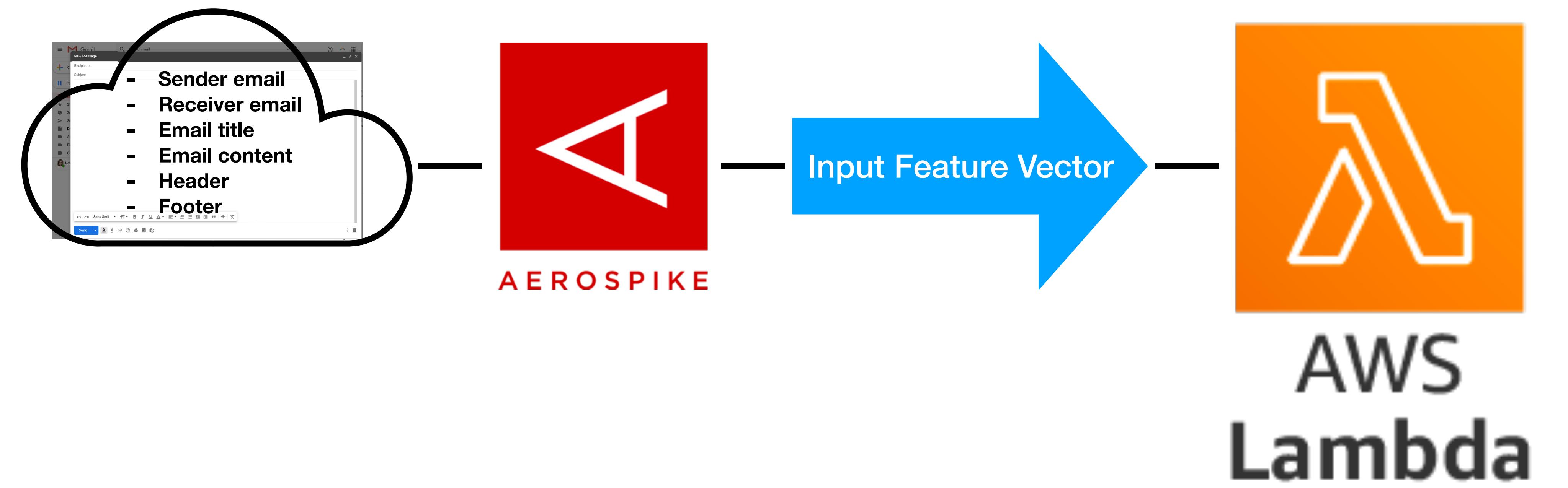


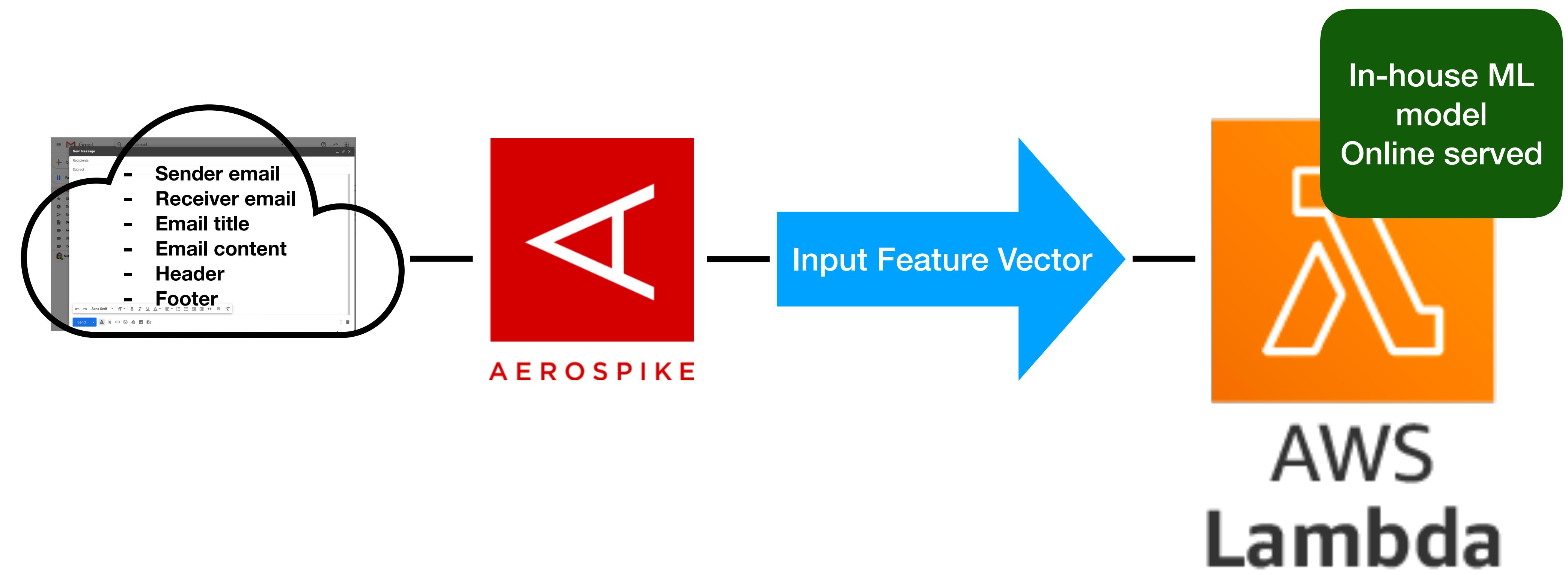


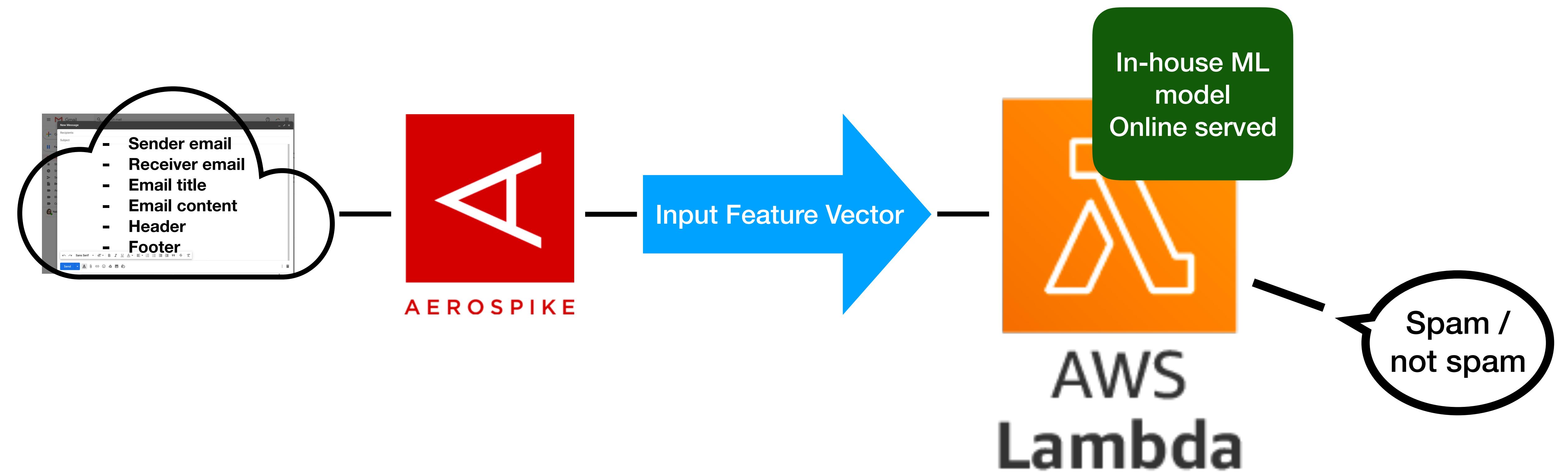


Presence of a recognised header	1
Presence of an official signature	0
Email structure	0.7
Language	0.78
Frequency of “special price”	0
Frequency of “prince”	0.9
Grammatical correctness	0.61











Recap

- What's AI? What's ML? How to ML?
- TensorFlow
- Training and Inference: online vs. Offline
- Bare bones architecture components, the shelf full platforms
- A concrete architecture example

To Summarize

- ML code is about **5% or less** of the overall code of a system
- Input data is a big and tricky part of an ML system
- Bare bones flow:
process data → store data → train model →
→ serve predictions → monitor

@NataliePis

@tweakers

#TweakersDevSummit



Thank You!

@NataliePis

@tweakers

#TweakersDevSummit

