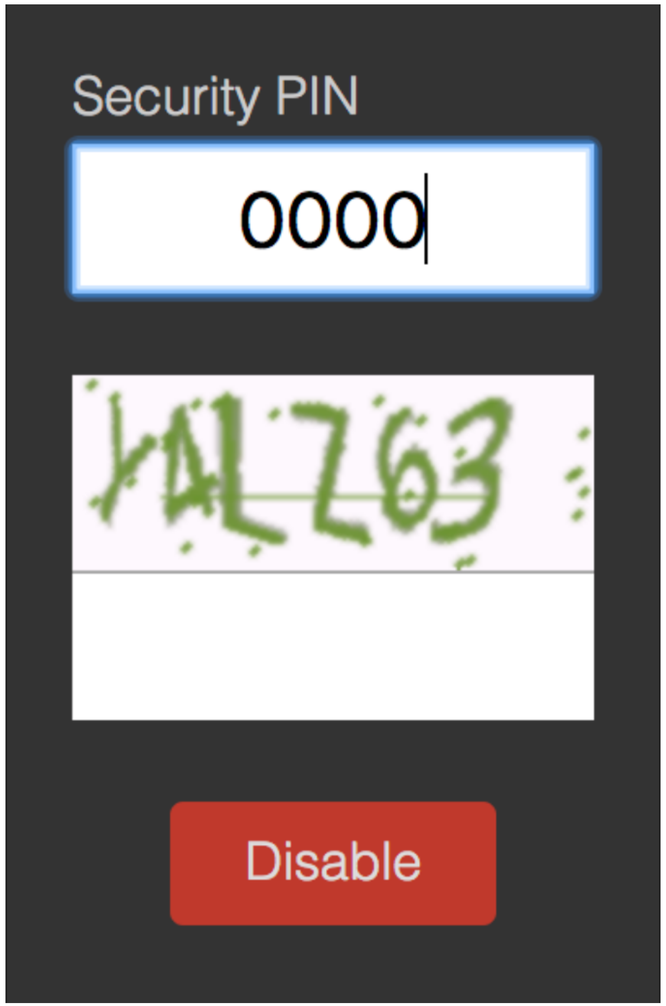
Infrastructure for Machine Learning Applications

Natalie Pistunovich
@NataliePis

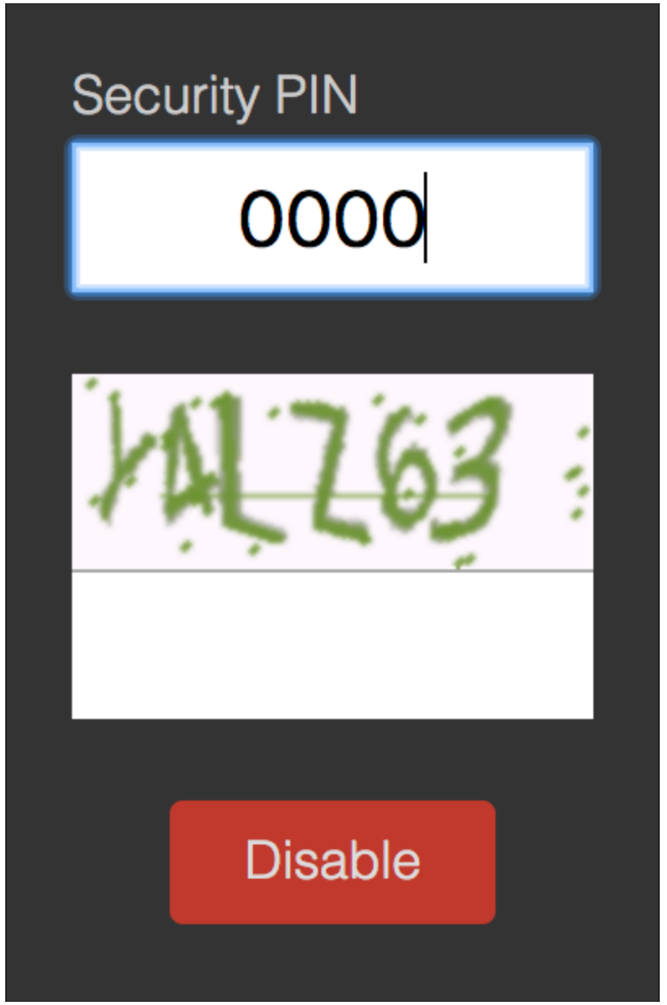
@golangwaw
#golang

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

@golangwaw
#golang
@NataliePis

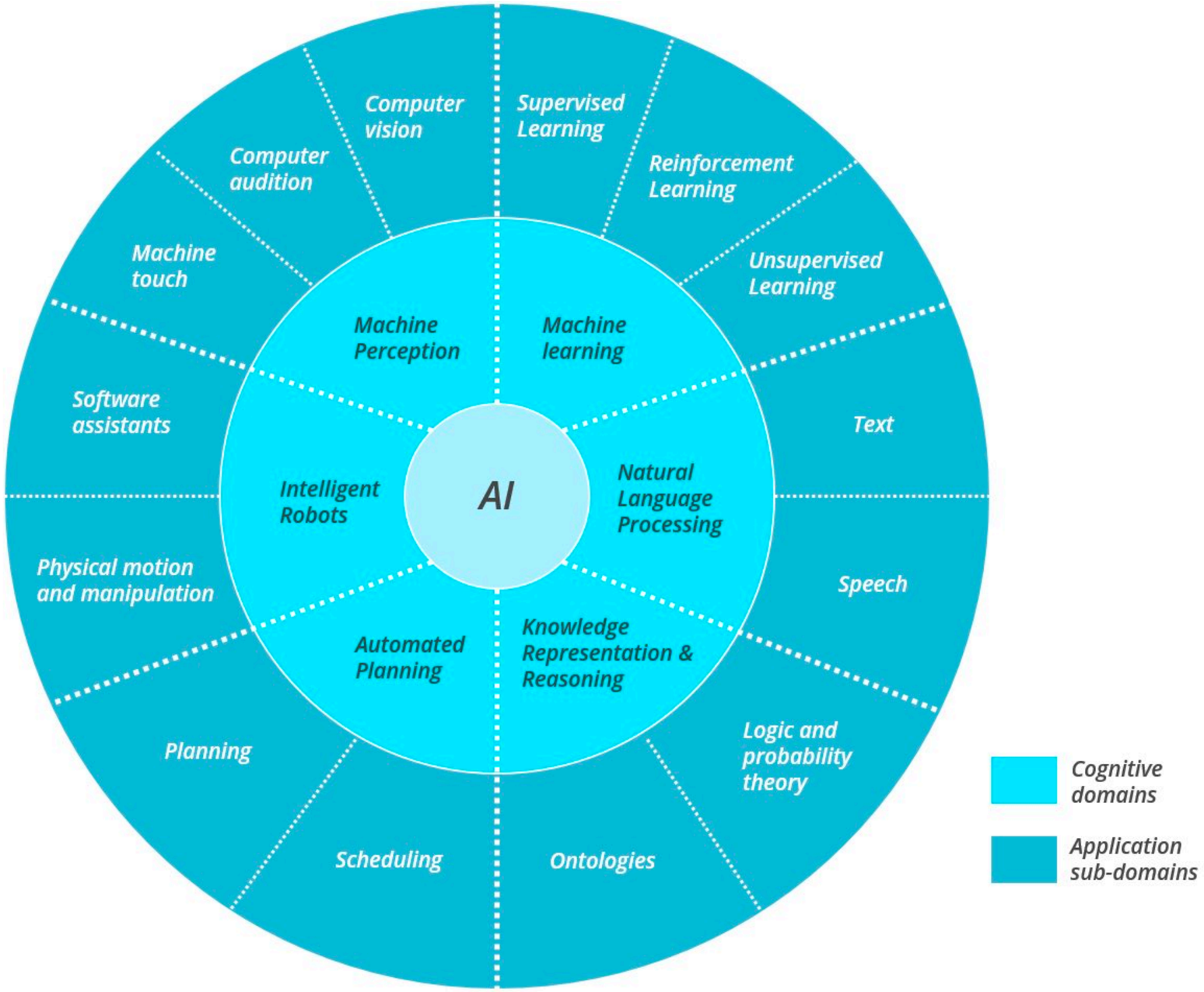
What is ML?

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?



@theAlexFreeman

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

- Wikipedia

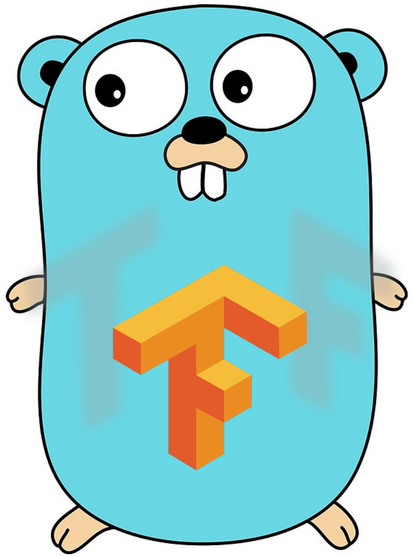
The term **artificial intelligence** is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

- Wikipedia

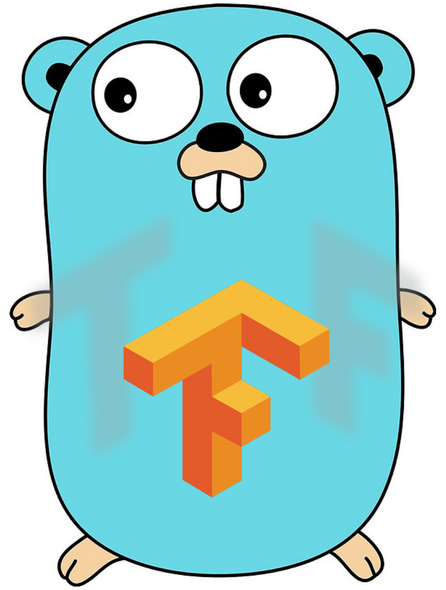
How to ML?

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

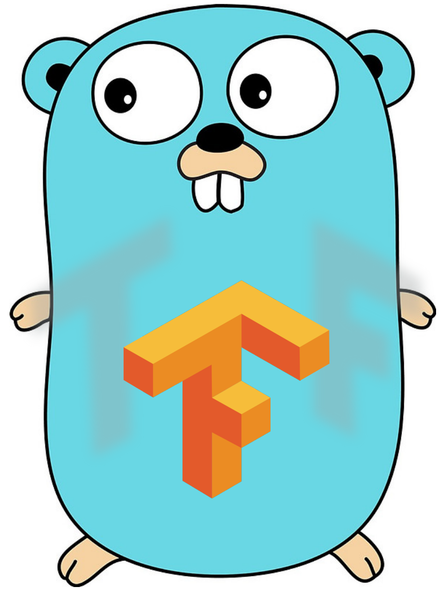


How to ML



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

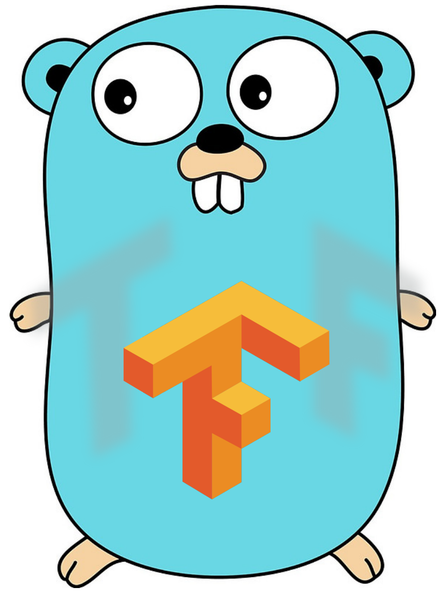


1. Define the problem
2. **Gather data**
   - relevant to the task
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict

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

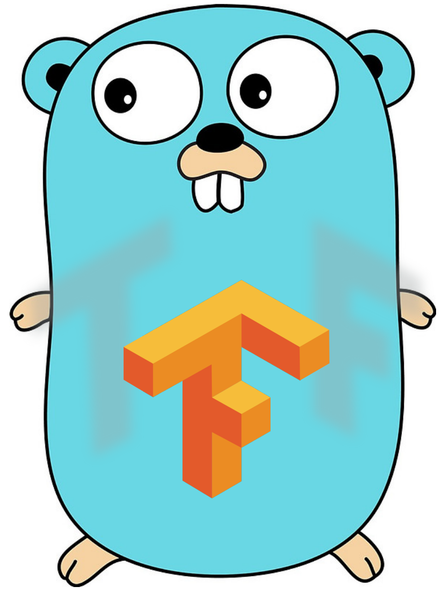
- clean and pre-process
- randomize
- split: train/test



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

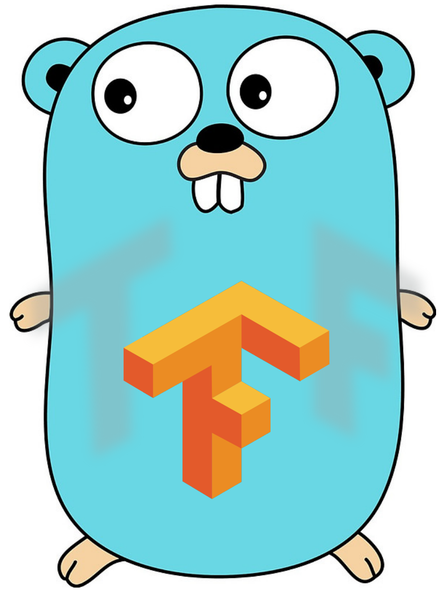
- clean and pre-process
- randomize
- split: train/test
  75/25



How to ML

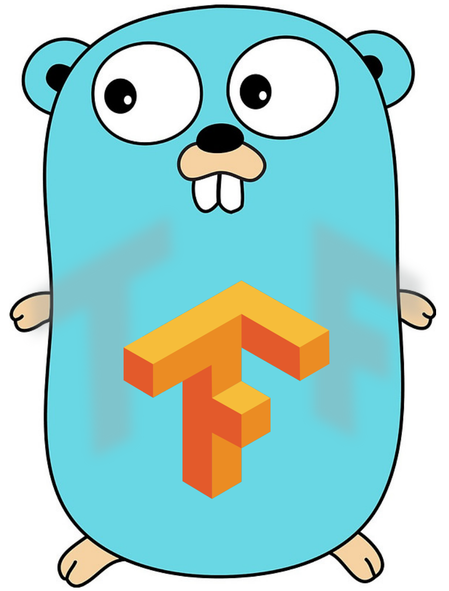
1. Define the problem
2. Gather data
3. Prepare data
4. **Choose a model**
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict

- learning task
- input type
- possible number of categories



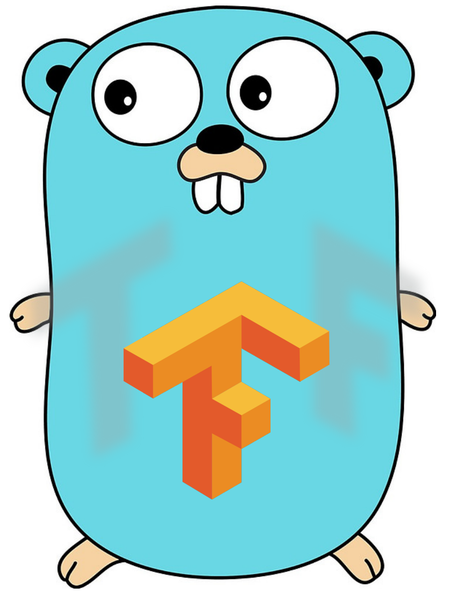
How to ML

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



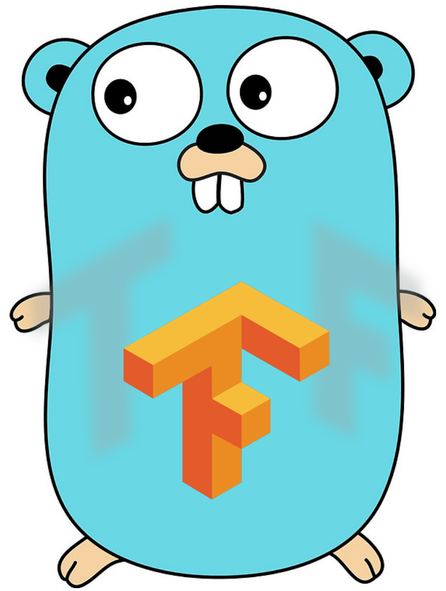
How to ML

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



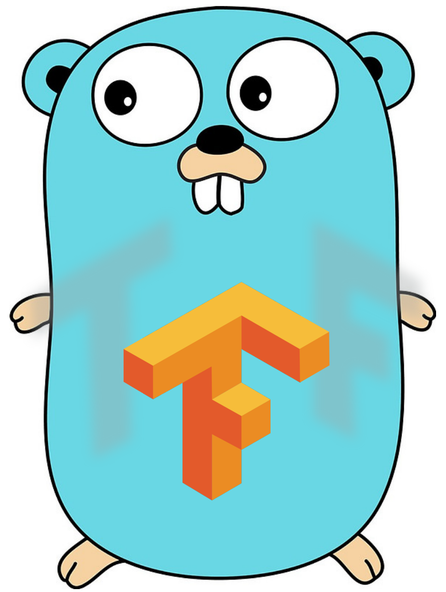
How to ML

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**



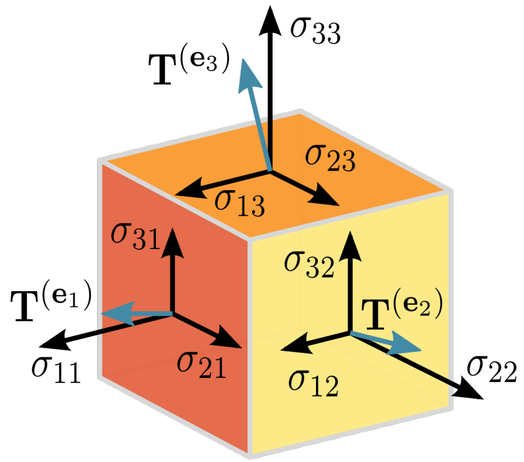
TensorFlow

TensorFlow

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

TensorFlow

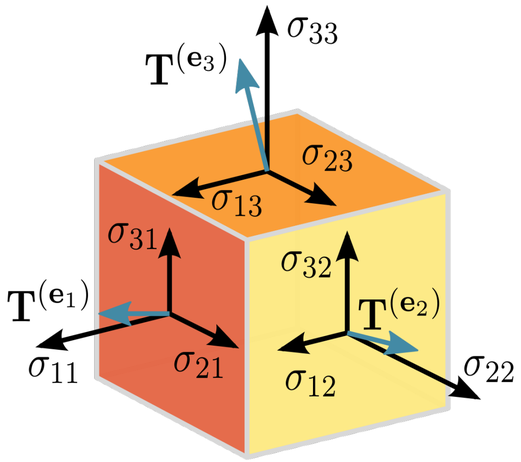
TensorFlow is an open-source software for Machine Intelligence,
used mainly for machine learning applications such as neural networks.

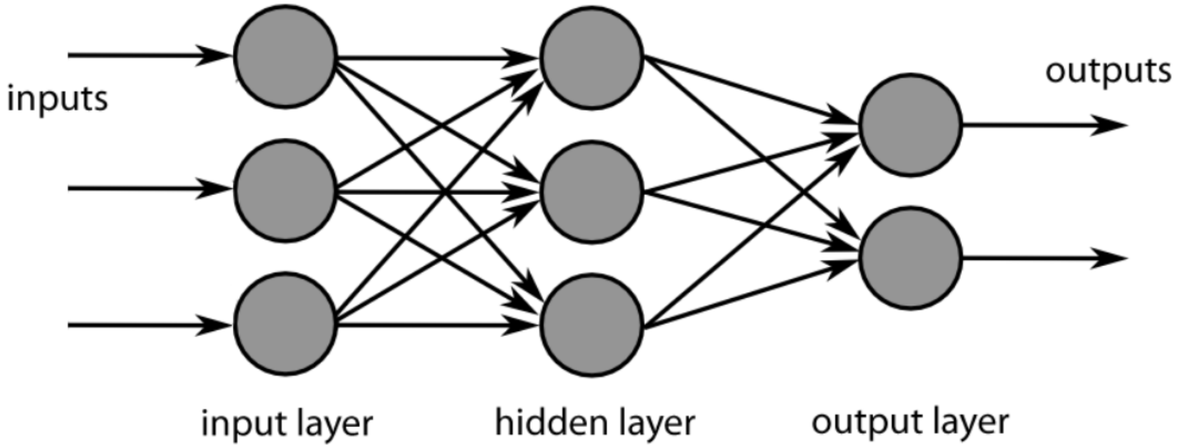


A tensor is a generalization
of vectors and matrices to
potentially higher dimensions

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**
   - assign random values
   - predict the train data
   - adjust weights
6. Evaluate the model
7. Tune the hyperparameters
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**

So, Infrastructure?

Hidden Technical Debt in Machine Learning Systems

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

Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison
{ebner,vchaudhary,mwyoung,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

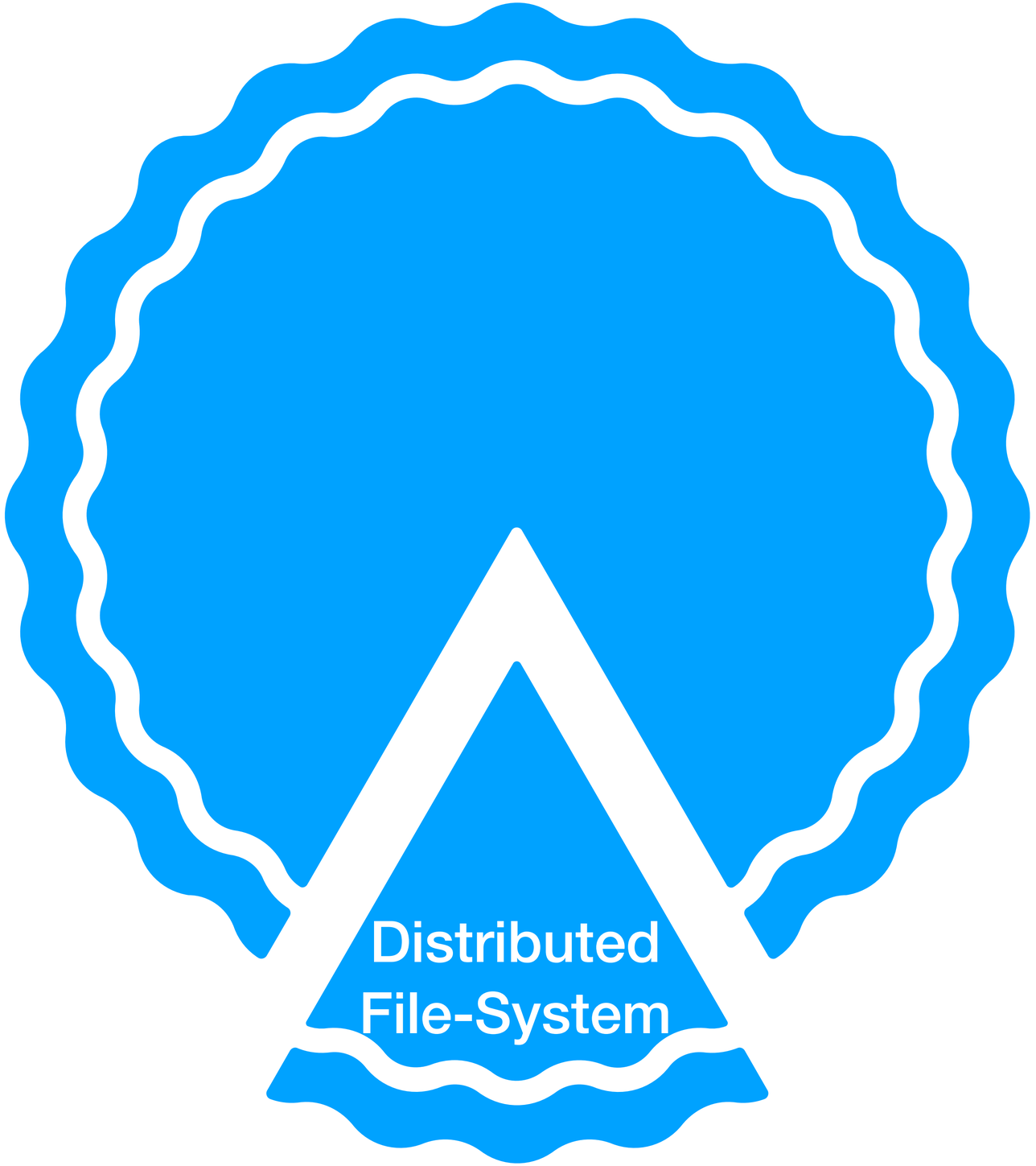
The ML code is at the heart of a real-world production system, but it accounts for **5% or less** of the overall code of that system.

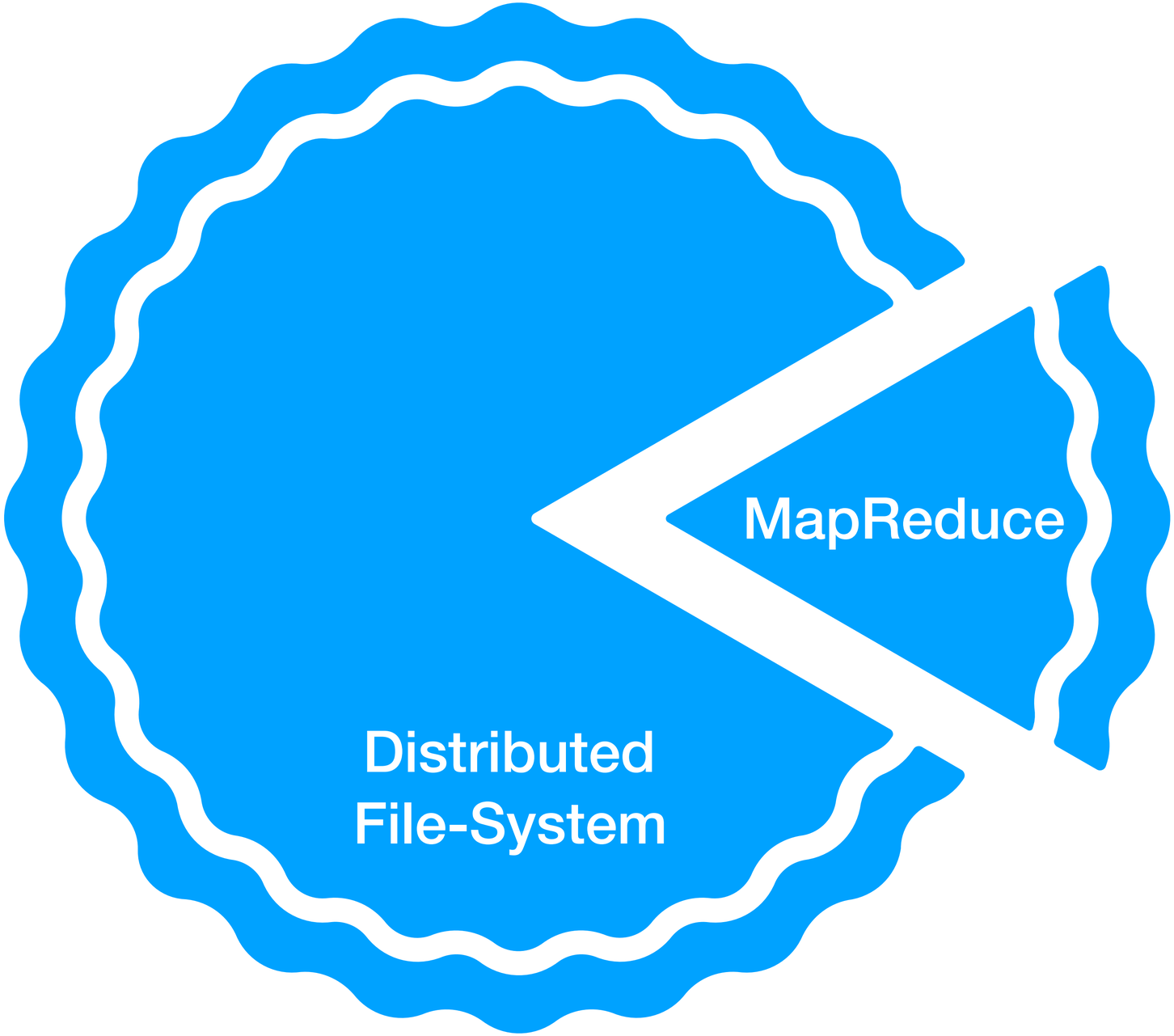

hadoop

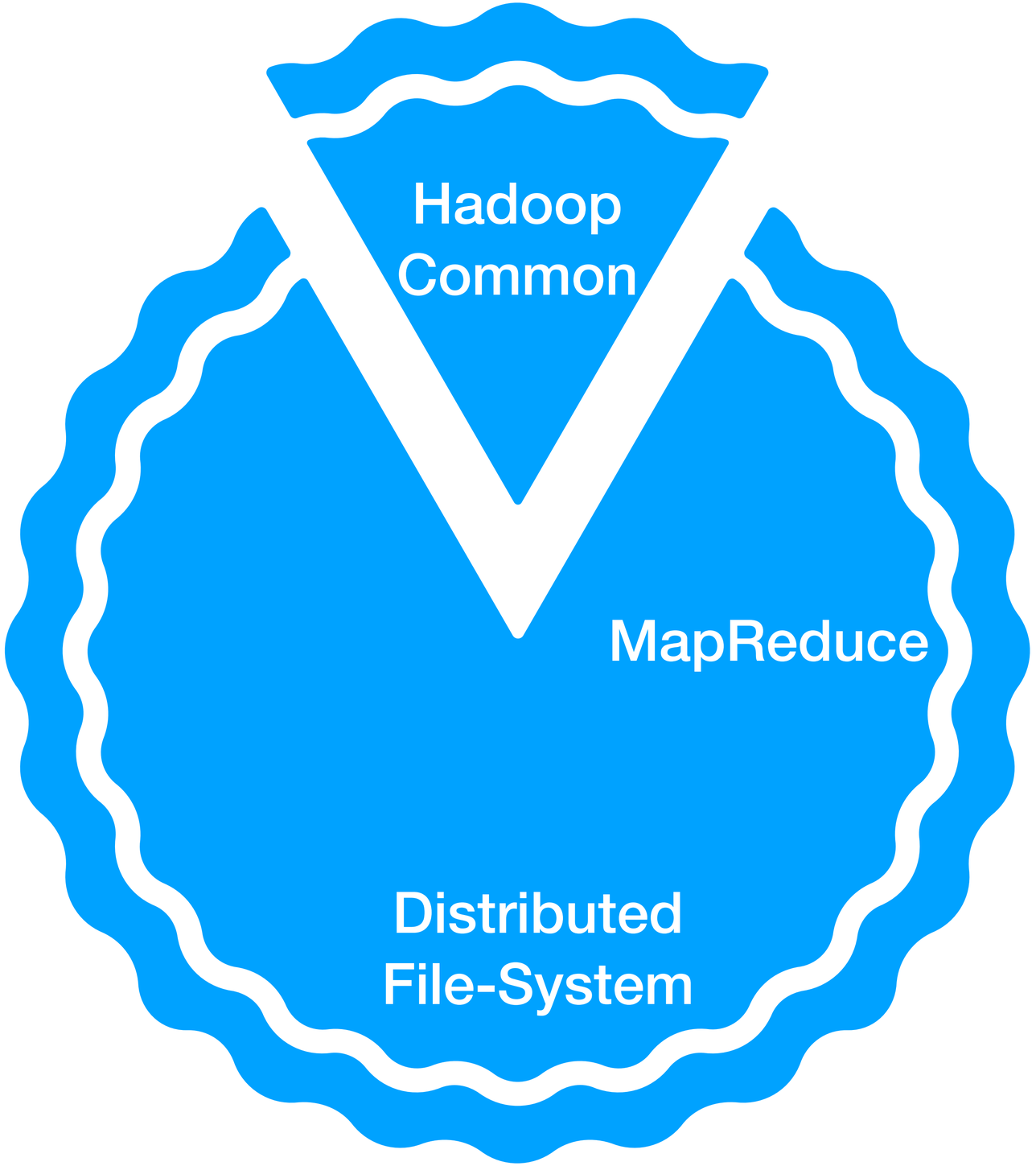

APACHE
hadoop
TM

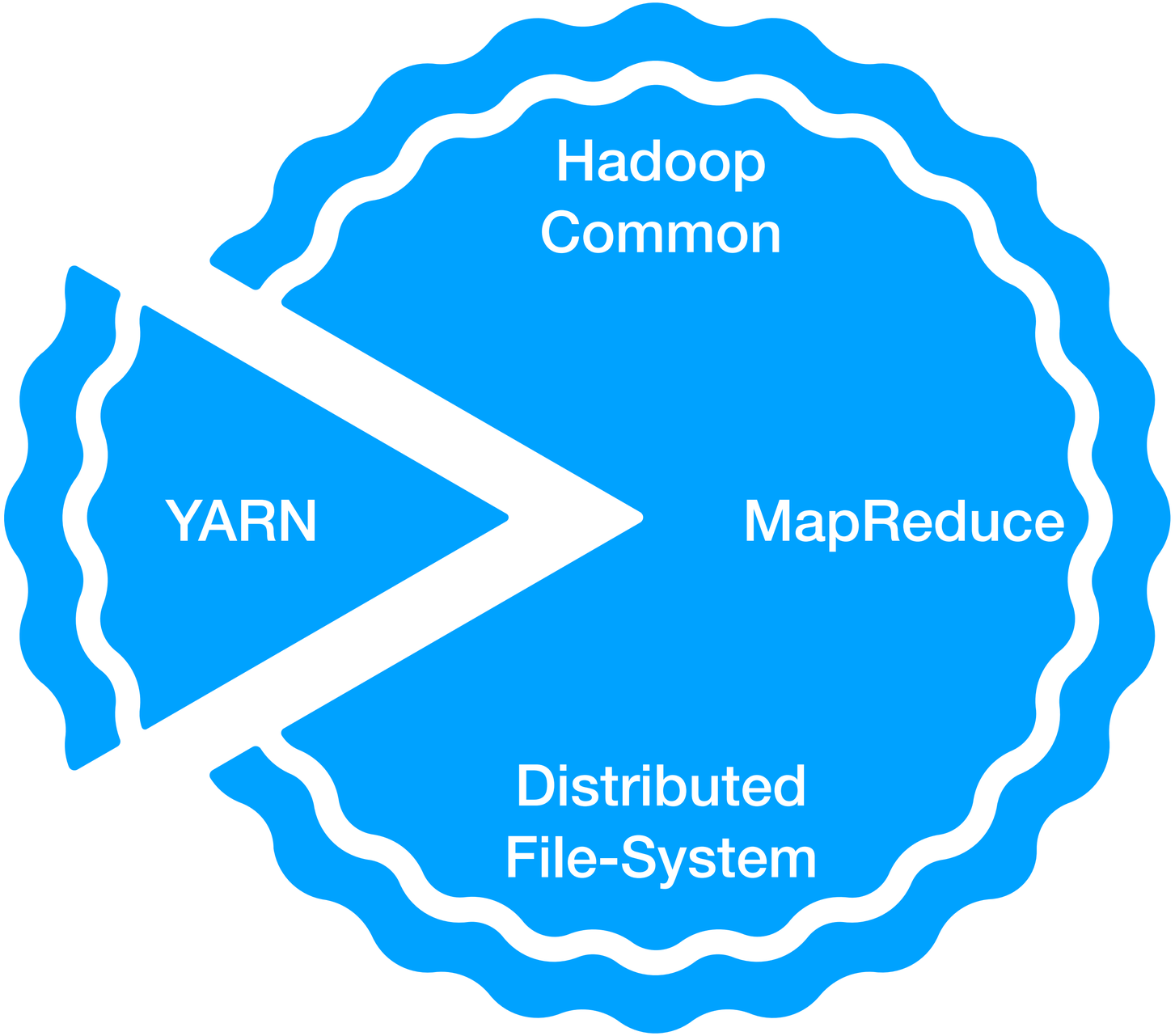

APACHE
hadoop
TM


APACHE
hadoop
TM


Distributed
File-System

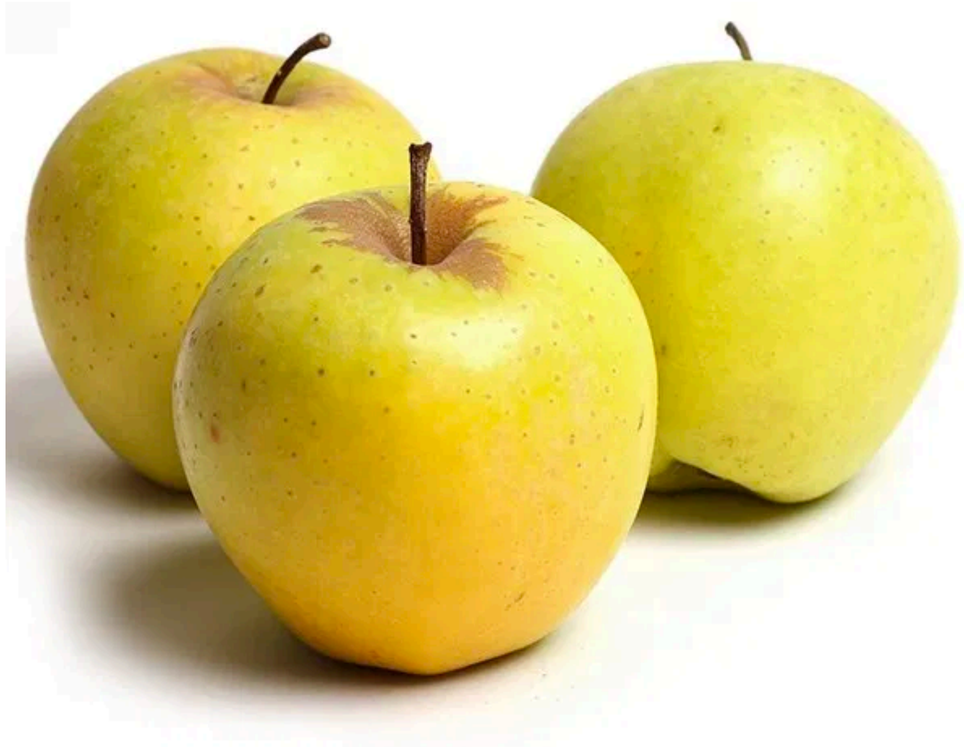

MapReduce
Distributed
File-System


Hadoop
Common
MapReduce
Distributed
File-System

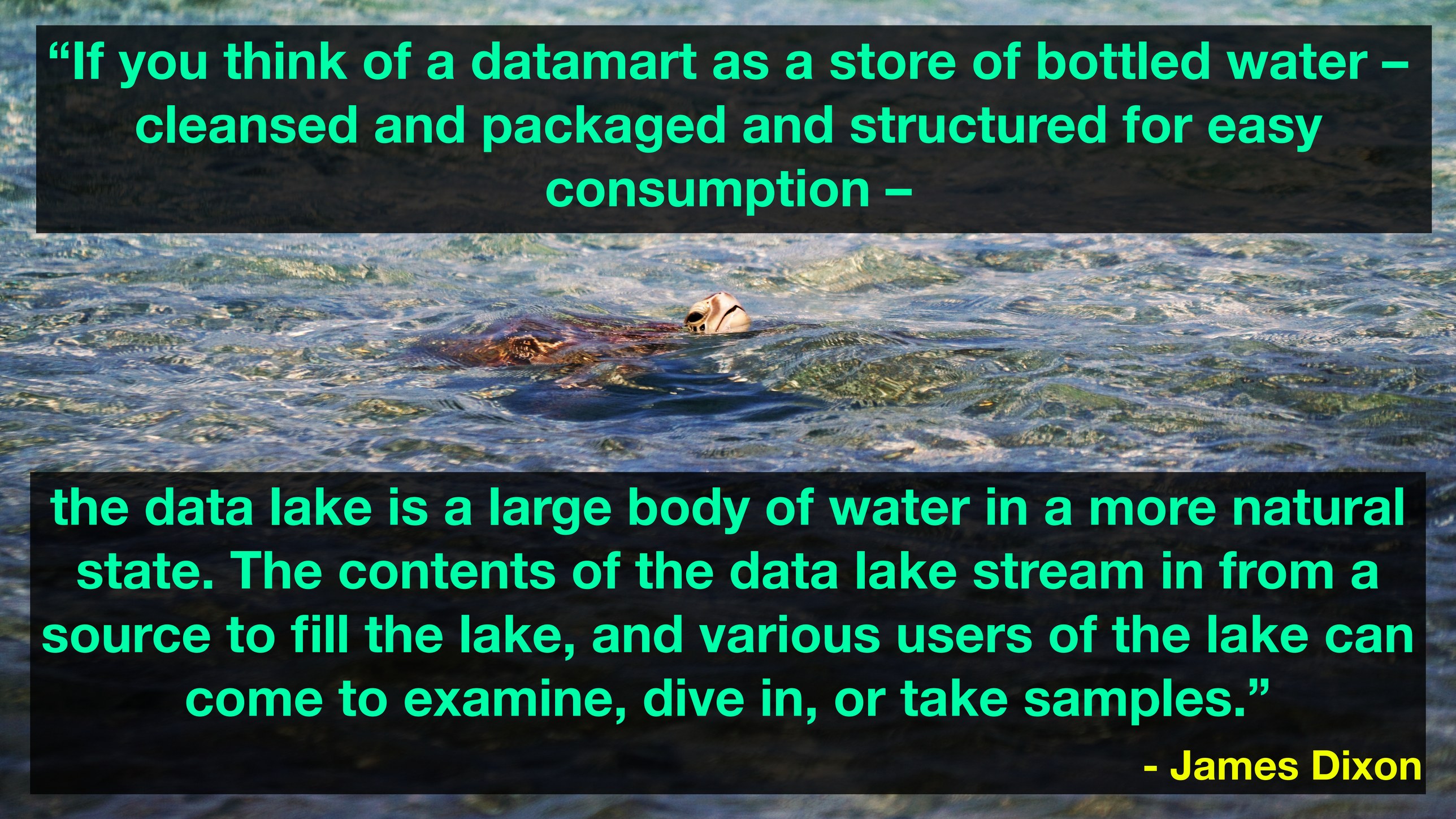

Hadoop
Common
YARN
MapReduce
Distributed
File-System

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 –
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

Database

Functions as a Service (aka Serverless)

AWS Lambda

Azure Functions

Google Cloud Functions

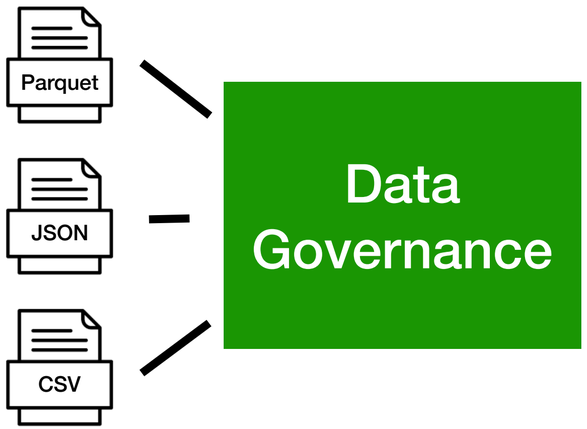
OPENFAAS

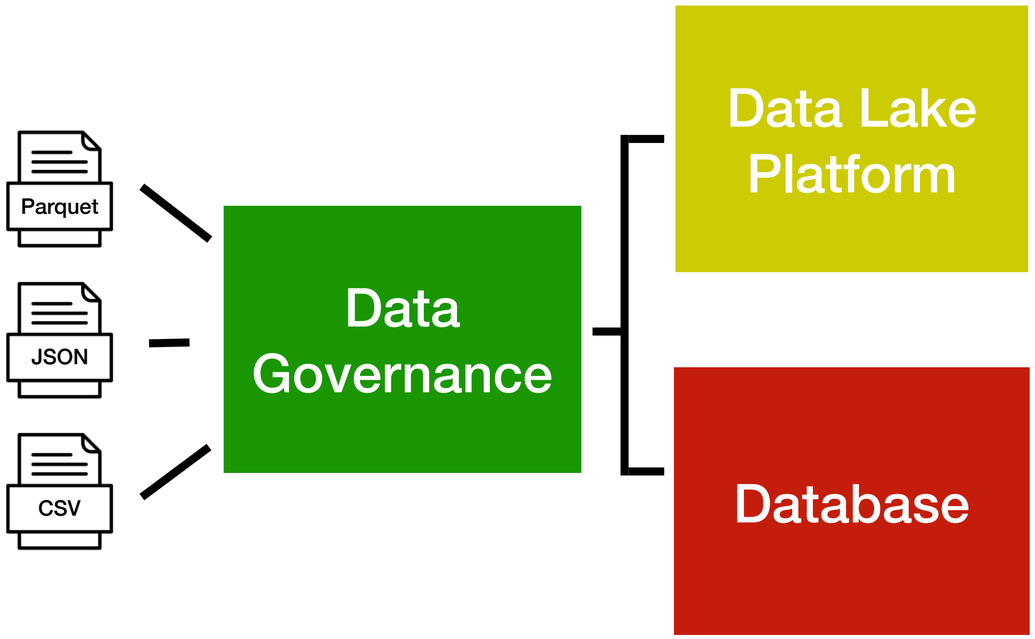
Data Governance

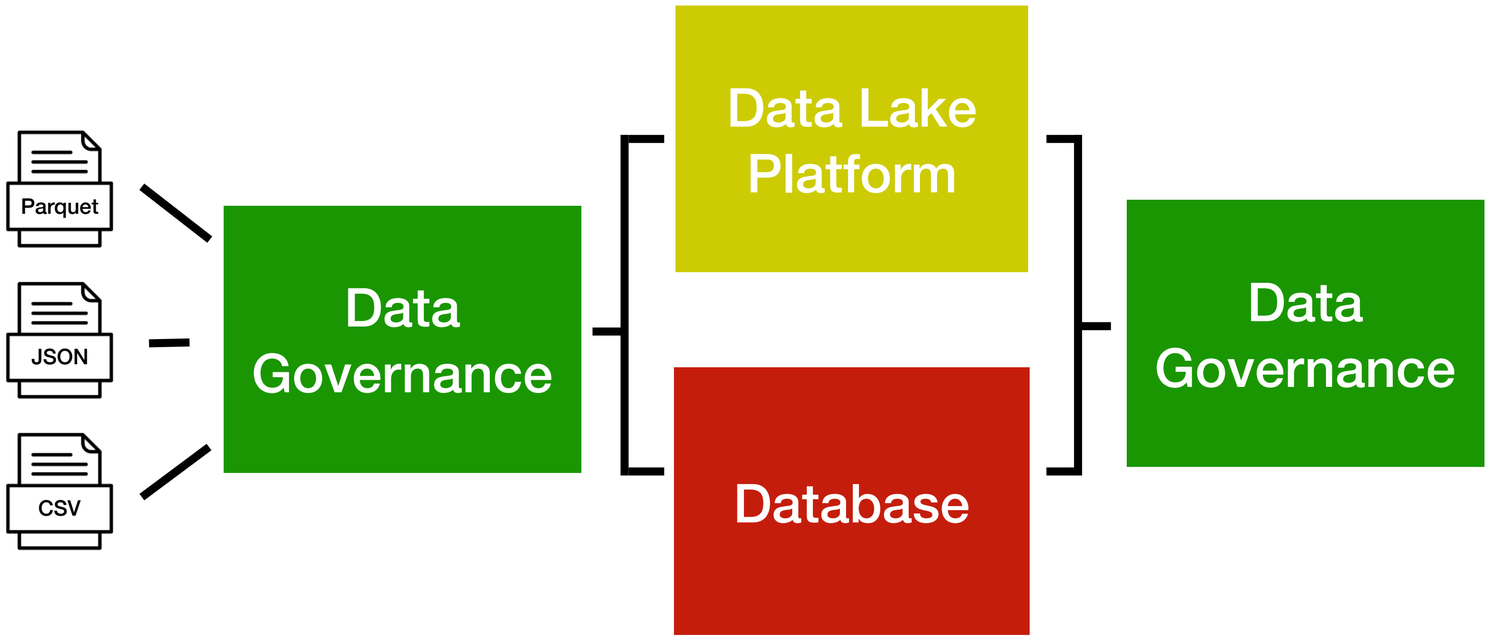
Monitoring

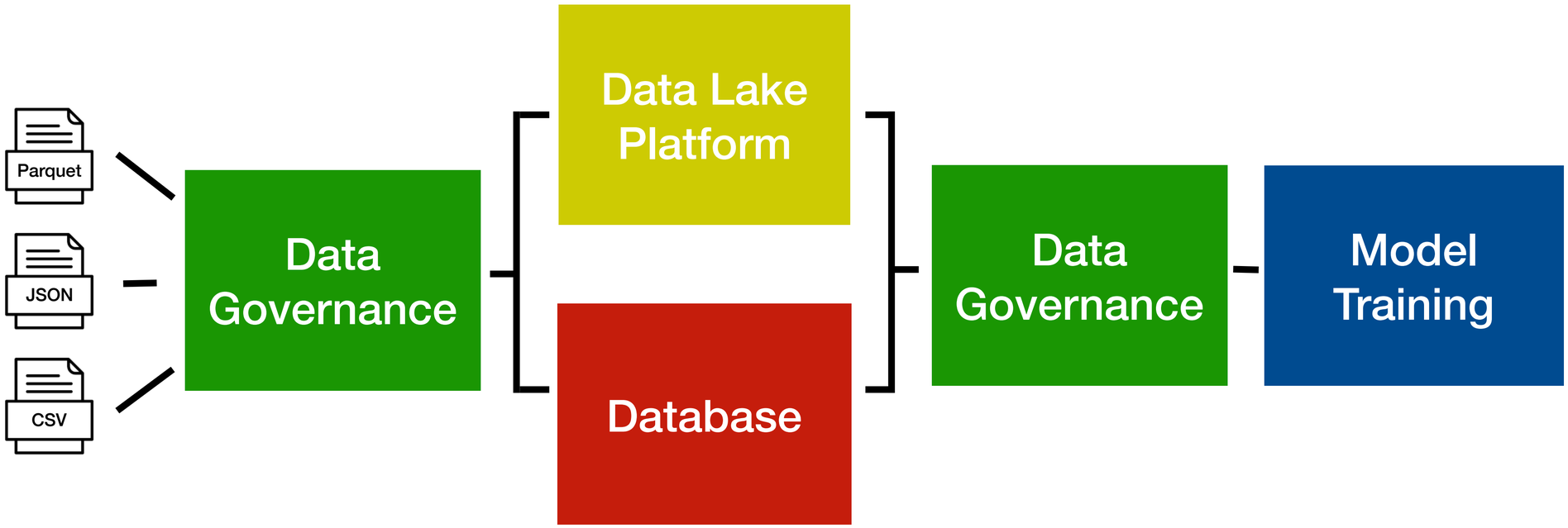
Putting it All Together: Bare Bones Infrastructure


Parquet
JSON
CSV

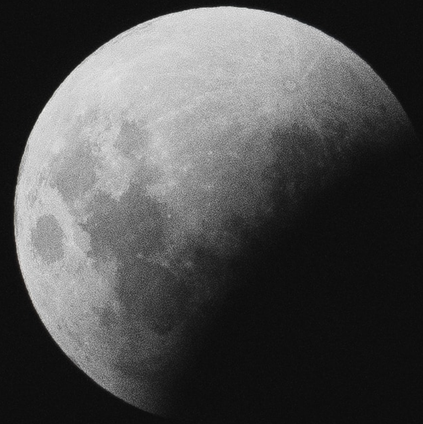

Parquet
JSON
CSV
Data
Governance


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance

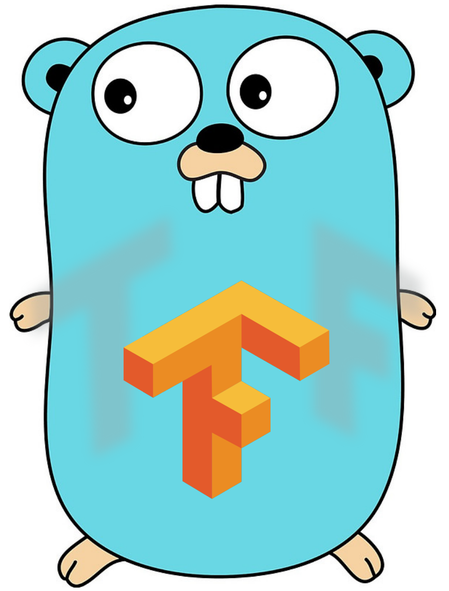

Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training

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**
   - assign random values
   - predict the train data
   - adjust weights
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



Types of Training

Static model (Offline) vs. Dynamic model (Online)

Static Model

+

Easier to build and test.

-

Can only predict things we know about.

Update latency likely measured in hours or days.

Dynamic Model

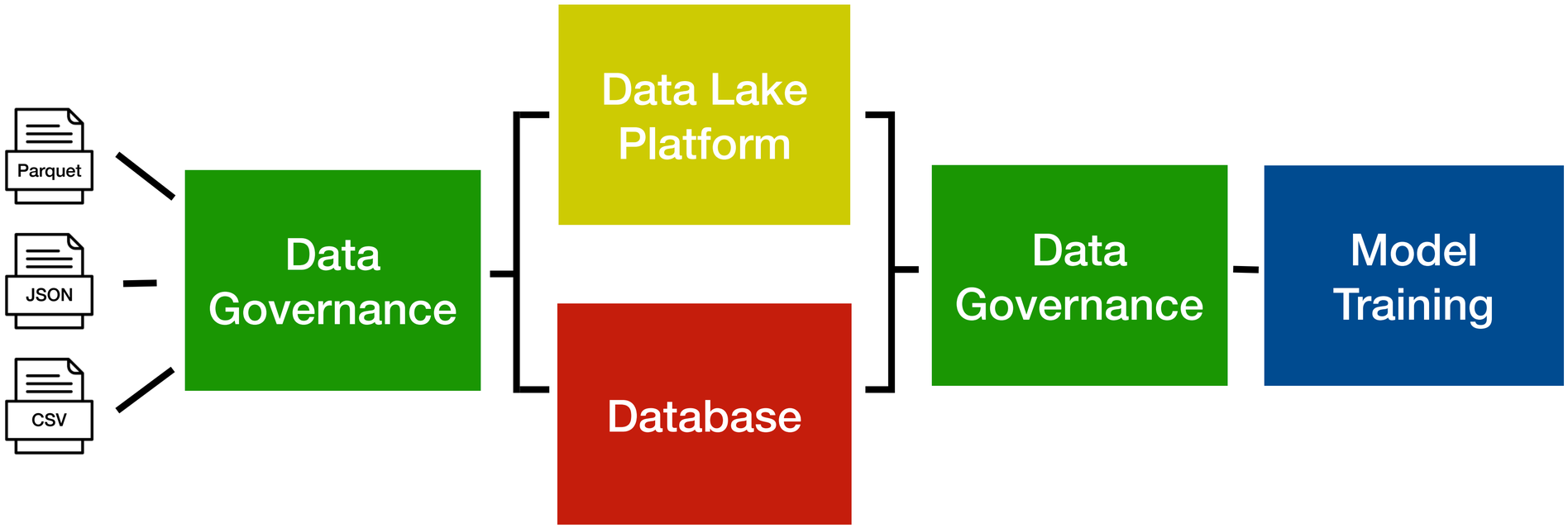
+

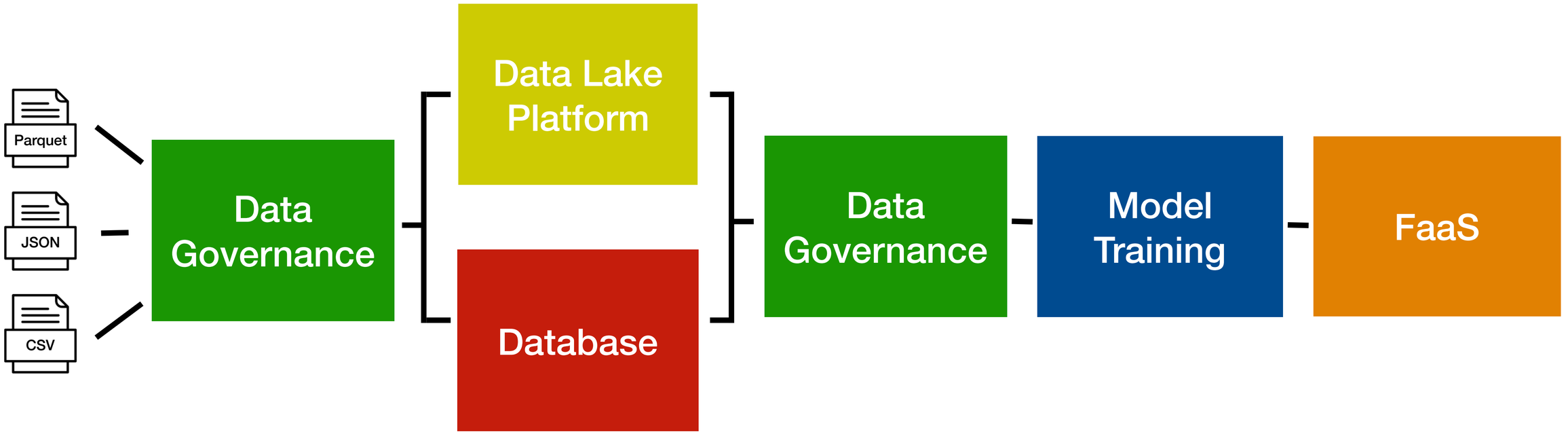
Adapted to changing data, hence more likely to make better predictions.

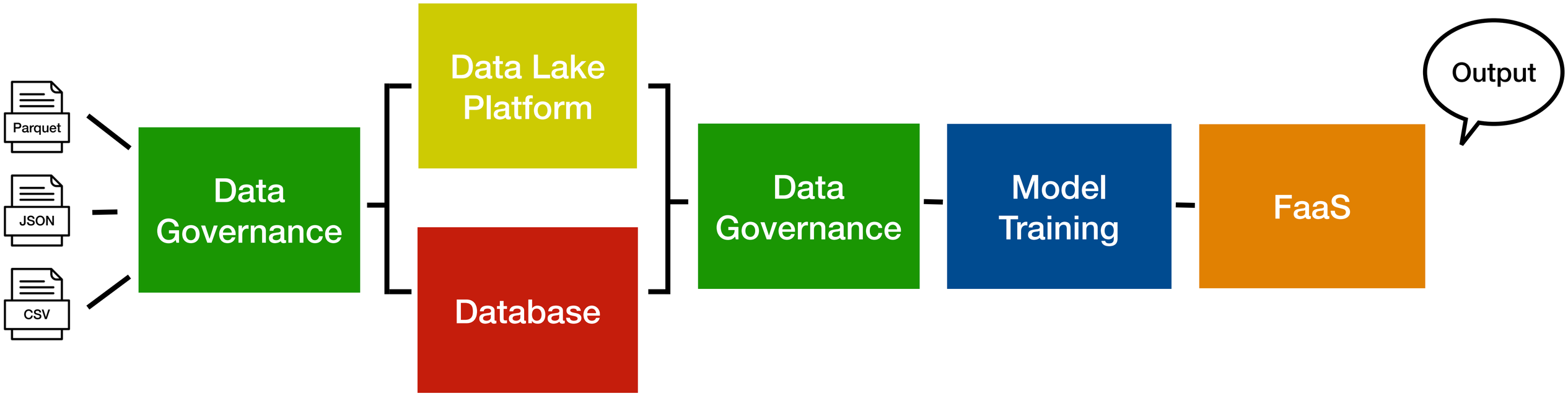
-

Compute intensive, latency sensitive, may limit model complexity.

Monitoring needs are more intensive: outputs and performance.


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training

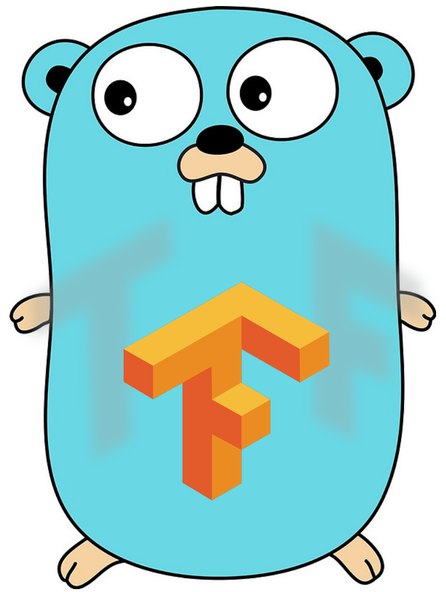

Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training
FaaS


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training
FaaS
Output

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

+

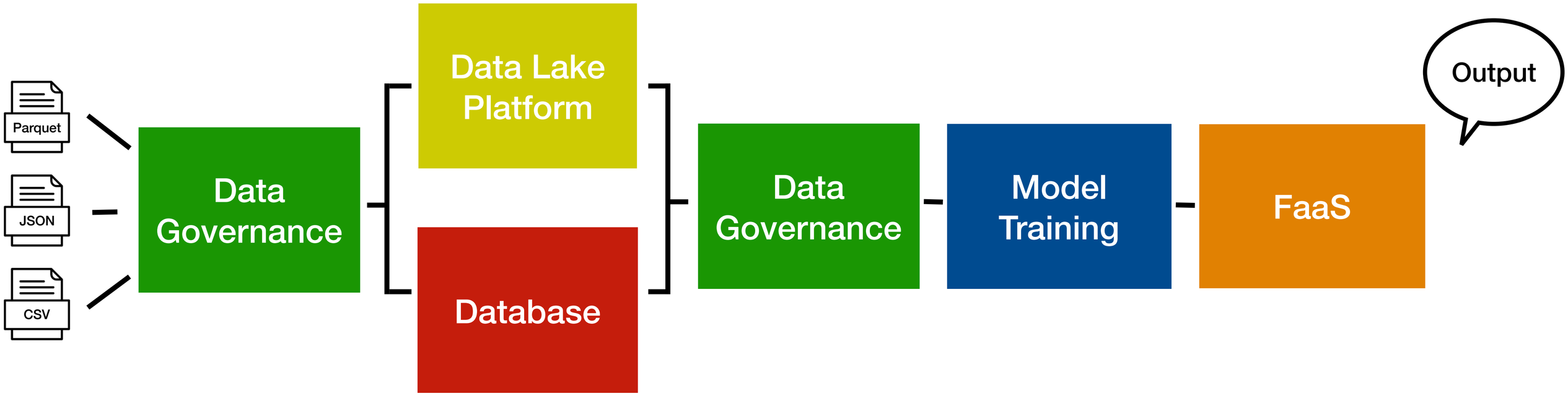
Don't worry much about cost of inference.

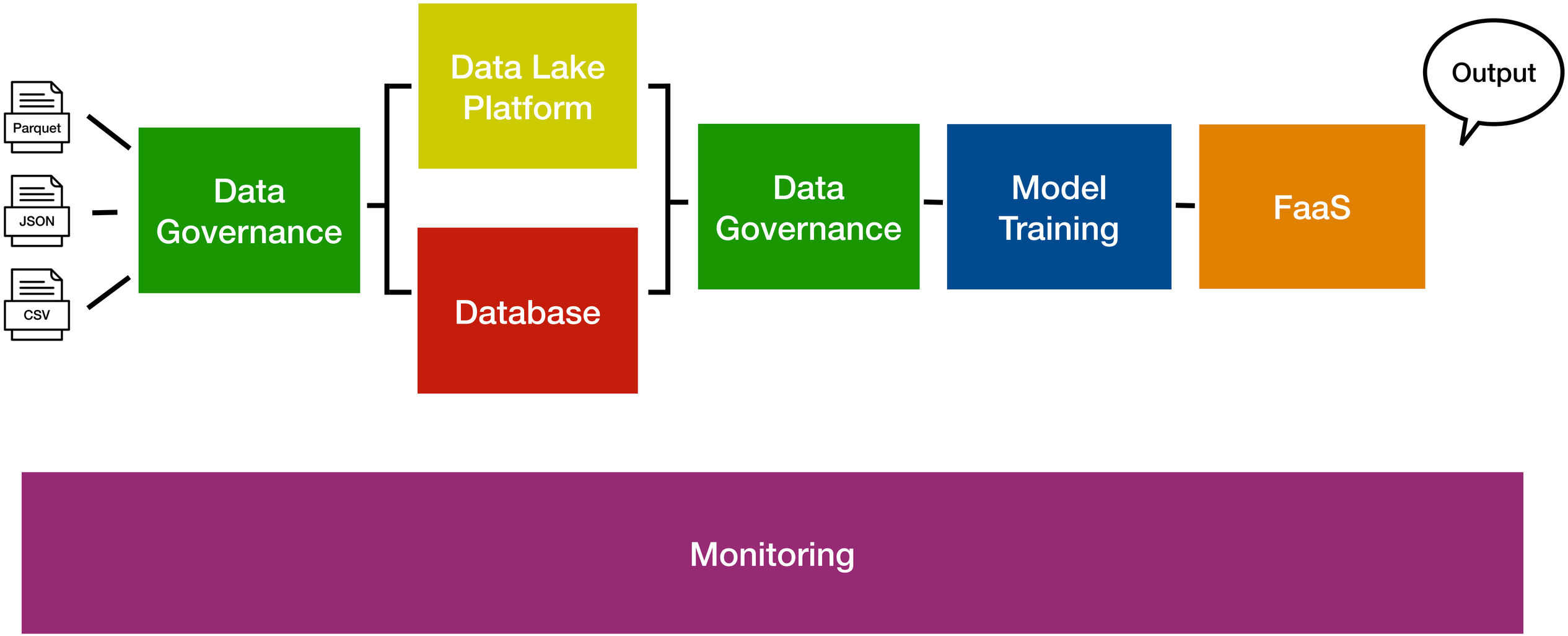
Likely to use batch quota.

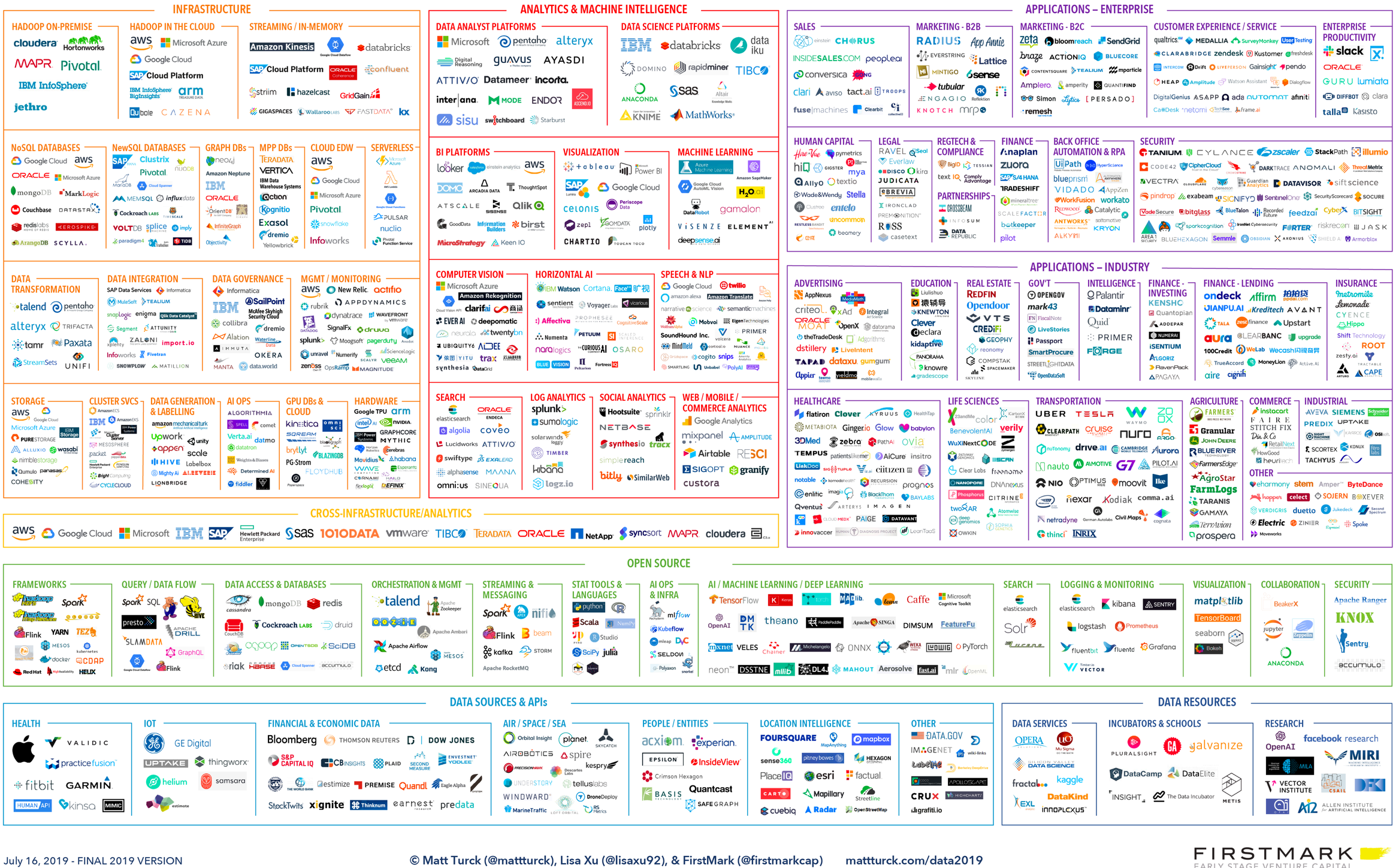
Can do post-verification of predictions on data before pushing.

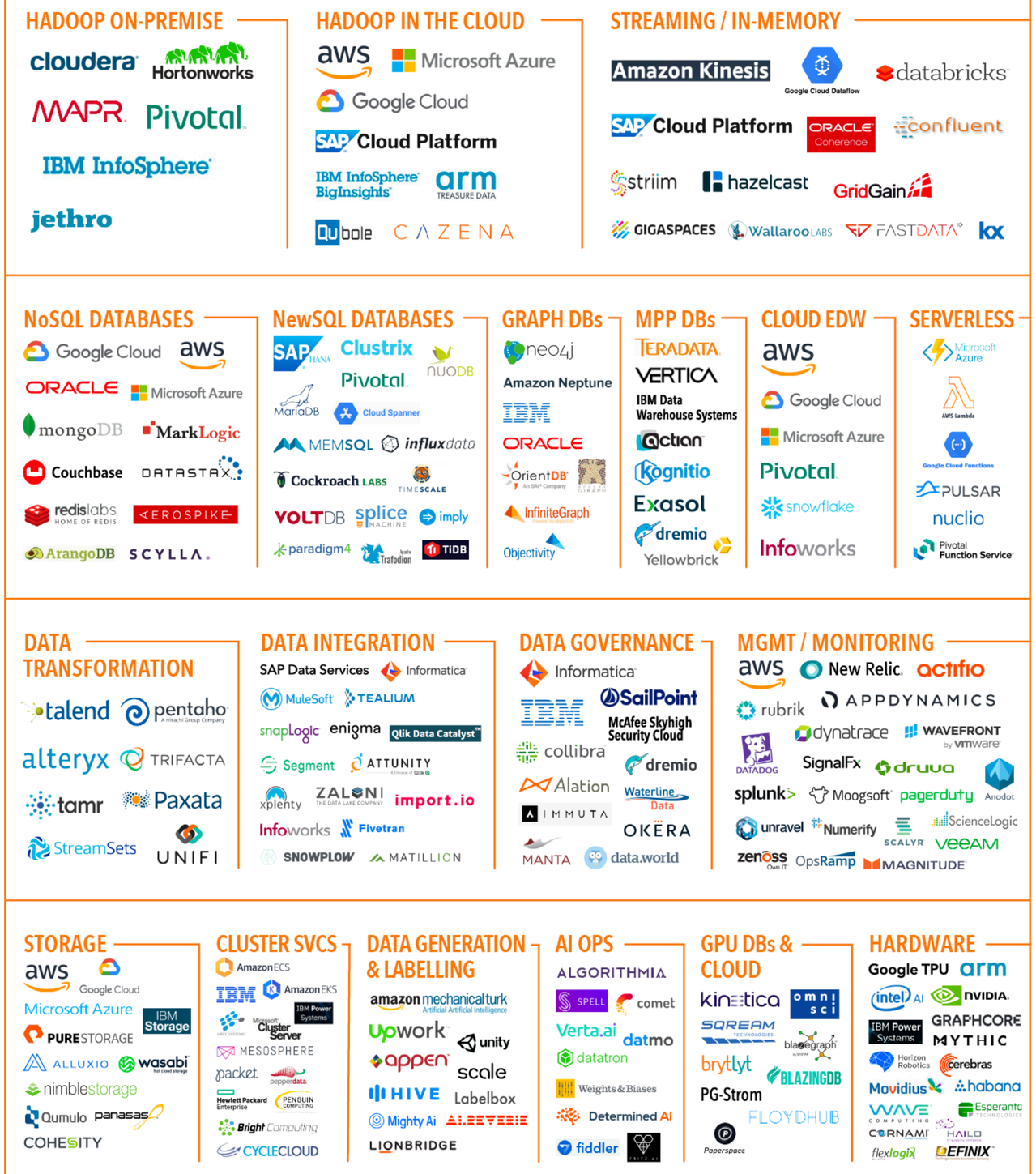
-

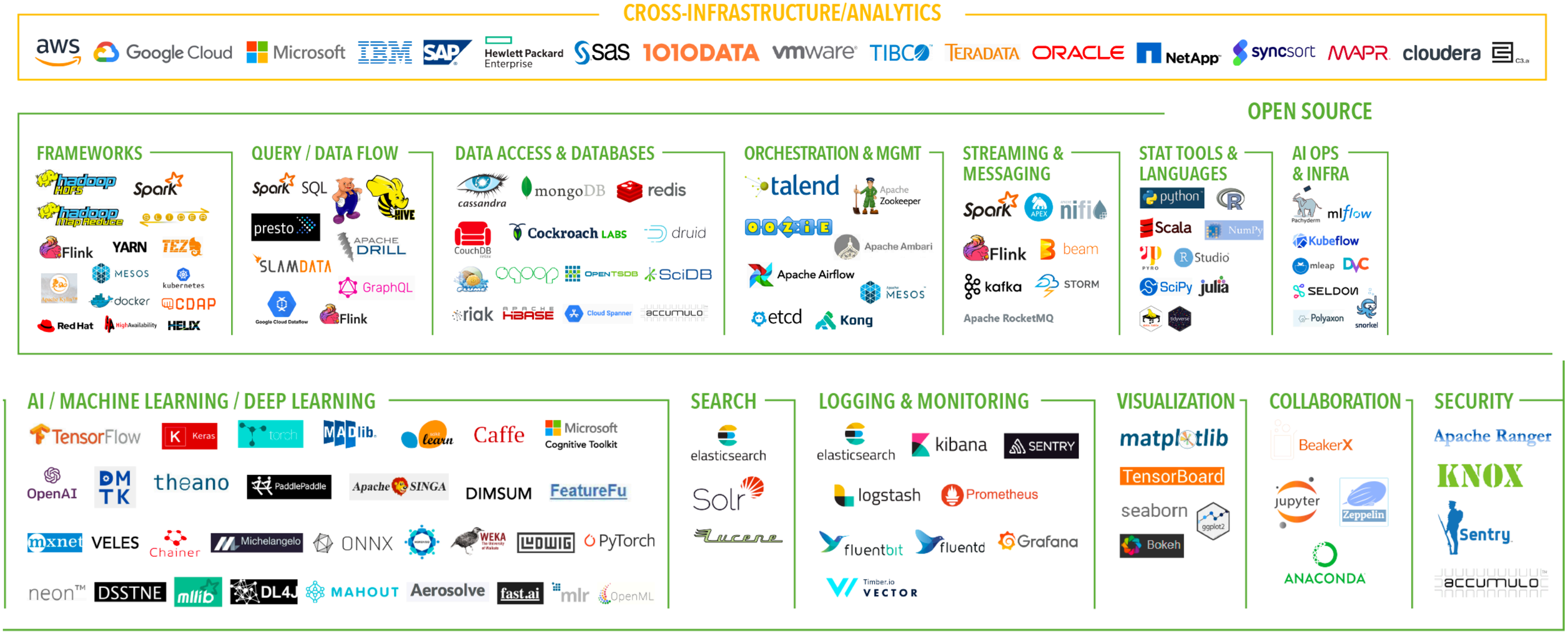
Can only predict things we know about.


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training
FaaS
Output


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training
FaaS
Output
Monitoring

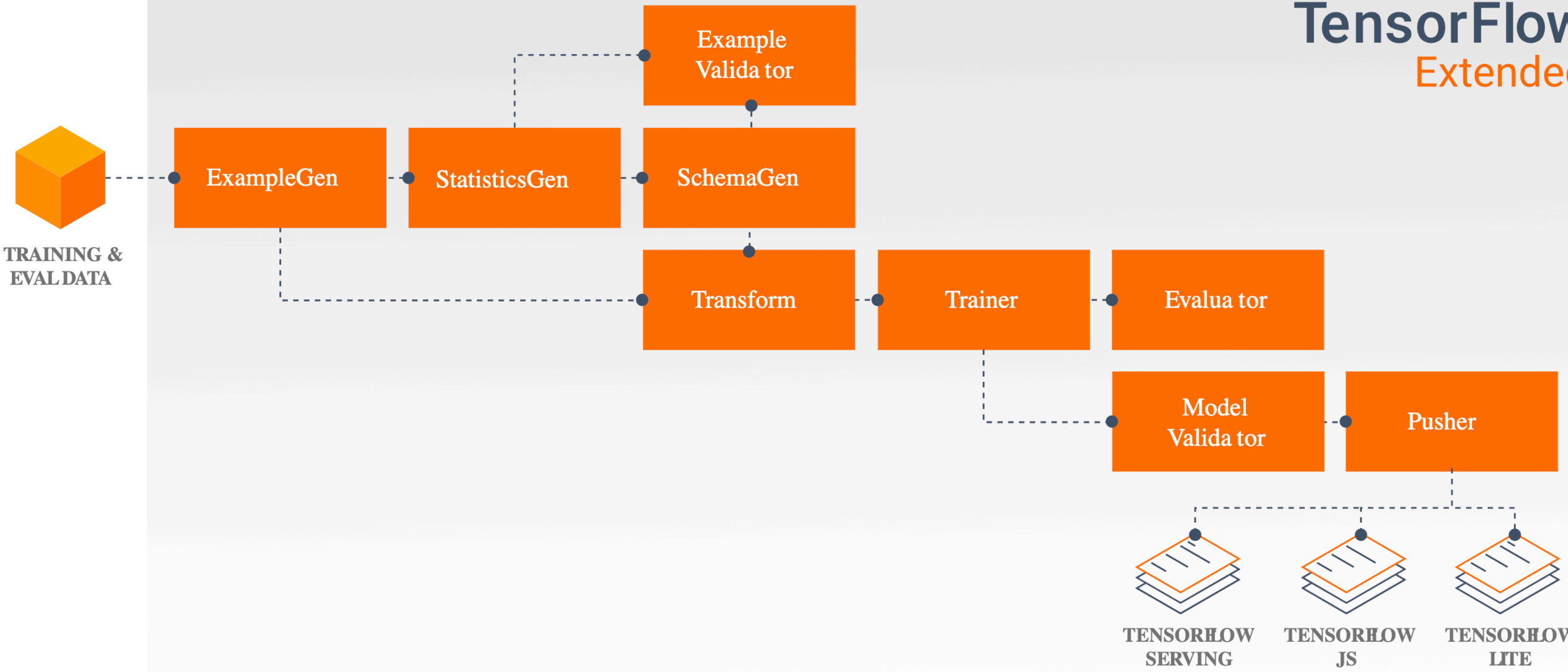



HADOOP ON-PREMISE
cloudera
Hortonworks
MAPR
Pivotal
IBM InfoSphere
jethro
HADOOP IN THE CLOUD
aws
Microsoft Azure
Google Cloud
SAP Cloud Platform
IBM InfoSphere BigInsights
arm TREASURE DATA
Qubole
CAZENA
STREAMING / IN-MEMORY
Amazon Kinesis
Google Cloud Dataflow
databricks
SAP Cloud Platform
ORACLE Coherence
confluent
striim
hazelcast
GridGain
GIGASPACES
Wallaroo LABS
FASTDATA
kx
NoSQL DATABASES
Google Cloud
aws
ORACLE
Microsoft Azure
mongoDB
MarkLogic
Couchbase
DATASTAX
redislabs HOME OF REDIS
AEROSPIKE
ArangoDB
SCYLLA.
NewSQL DATABASES
SAP HANA
Clustrix
NUODB
Pivotal
MariaDB
Cloud Spanner
MEMSQL
influxdata
Cockroach LABS
TIMESCALE
VOLTDB
splice MACHINE
imply
paradigm4
Trafodion
TIDB
GRAPH DBs
neo4j
Amazon Neptune
IBM
ORACLE
OrientDB
InfiniteGraph
Objectivity
MPP DBs
TERADATA
VERTICA
IBM Data Warehouse Systems
actian
Kognitio
Exasol
dremio
Yellowbrick
CLOUD EDW
aws
Google Cloud
Microsoft Azure
Pivotal
snowflake
Infoworks
SERVERLESS
Microsoft Azure
AWS Lambda
Google Cloud Functions
PULSAR
nuclio
Pivotal Function Service
DATA TRANSFORMATION
talend
pentaho
alteryx
TRIFACTA
tamr
Paxata
StreamSets
UNIFI
DATA INTEGRATION
SAP Data Services
Informatica
MuleSoft
TEALIUM
snaplogic
enigma
Qlik Data Catalyst
Segment
ATTUNITY
xplenty
ZALONI
import.io
Infoworks
Fivetran
SNOWPLOW
MATILLION
DATA GOVERNANCE
Informatica
IBM
SailPoint
McAfee Skyhigh Security Cloud
collibra
dremio
Alation
Waterline Data
IMMUTA
OKERA
MANTA
data.world
MGMT / MONITORING
aws
New Relic
actifio
rubrik
APPDYNAMICS
dynatrace
WAVEFRONT by vmware
DATADOG
SignalFx
druva
splunk>
Moogsoft
pagerduty
Anodot
unravel
Numerify
SCALYR
ScienceLogic
VeeAM
zenoss
OpsRamp
MAGNITUDE
STORAGE
aws
Google Cloud
Microsoft Azure
IBM Storage
PURESTORAGE
ALLUXIO
wasabi
nimblestorage
Qumulo
panasas
COHESITY
CLUSTER SVCS
Amazon ECS
IBM
Amazon EKS
IBM Power Systems
Microsoft Cluster Server
MESOSPHERE
packet
pepperdata
Hewlett Packard Enterprise
PENGUIN COMPUTING
Bright Computing
CYCLECLOUD
DATA GENERATION & LABELLING
amazon mechanical turk
Upwork
unity
appen
scale
HIVE
Labelbox
Mighty AI
LIONBRIDGE
AI OPS
ALGORITHMIA
SPELL
comet
Verta.ai
datmo
datatron
Weights & Biases
Determined AI
fiddler
GPU DBs & CLOUD
kinetica
omnisci
SQREAM TECHNOLOGIES
blazegraph
brytlyt
BLAZINGDB
PG-Strom
FLOYDHUB
Paperspace
HARDWARE
Google TPU
arm
intel AI
NVIDIA
IBM Power Systems
GRAPHCORE
MYTHIC
Horizon Robotics
cerebras
Movidius
habana
WAVE COMPUTING
Esperanto TECHNOLOGIES
CORNAMI
HAILO
flexlogix
EFINIX

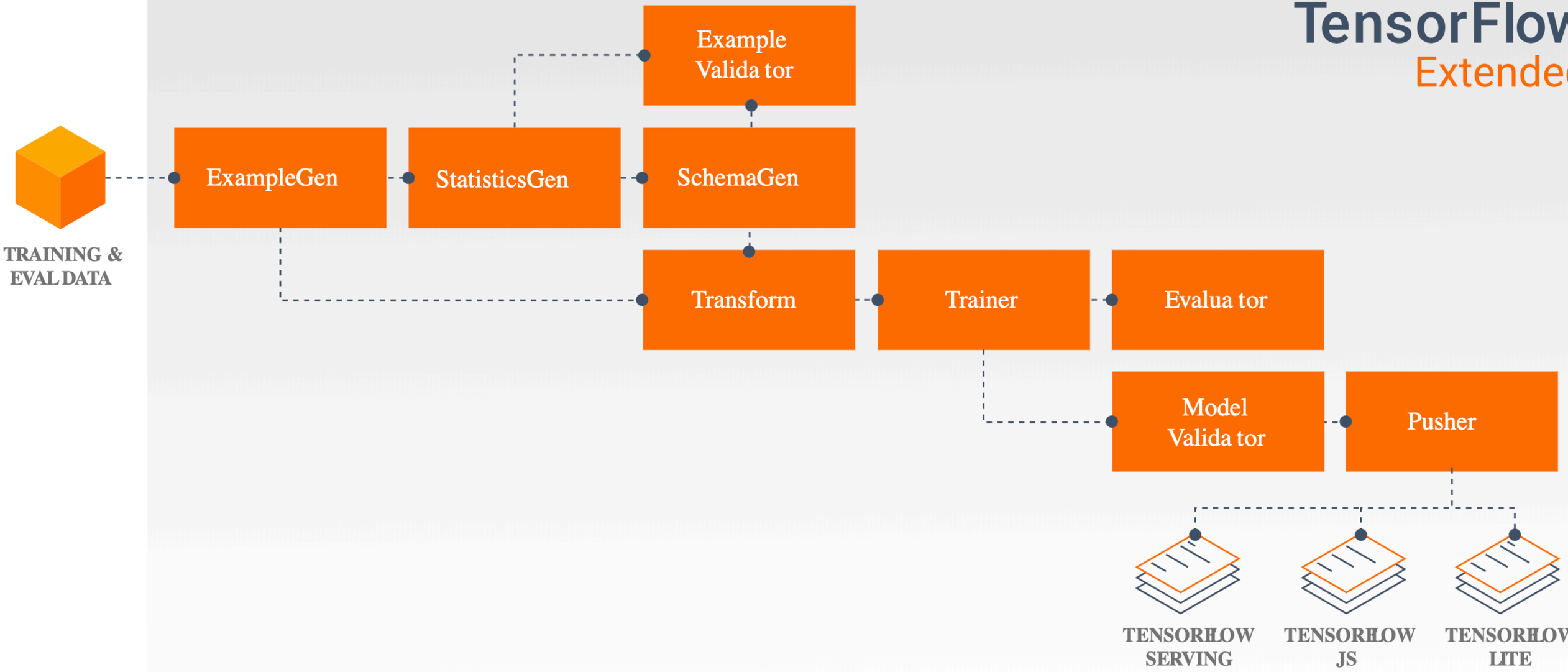

CROSS-INFRASTRUCTURE/ANALYTICS
aws
Google Cloud
Microsoft
IBM
SAP
Hewlett Packard Enterprise
sas
1010DATA
vmware
TIBCO
TERADATA
ORACLE
NetApp
syncsort
MAPR
cloudera
C3.ai
OPEN SOURCE
FRAMEWORKS
hadoop HDFS
Spark
hadoop MapReduce
Flink
YARN
TEZ
MESOS
kubernetes
docker
CDAP
Red Hat
HighAvailability
HELIX
QUERY / DATA FLOW
Spark SQL
HIVE
presto
APACHE DRILL
SLAMDATA
GraphQL
Google Cloud Dataflow
Flink
DATA ACCESS & DATABASES
cassandra
mongoDB
redis
CouchDB
Cockroach LABS
druid
OPENTSDB
SciDB
riak
APACHE HBASE
Cloud Spanner
accumulo
ORCHESTRATION & MGMT
talend
Apache Zookeeper
OOZIE
Apache Ambari
Apache Airflow
MESOS
etcd
Kong
STREAMING & MESSAGING
Spark
APEX
nifi
Flink
beam
kafka
STORM
Apache RocketMQ
STAT TOOLS & LANGUAGES
python
Scala
NumPy
PYRO
Studio
SciPy
julia
tidyverse
AI OPS & INFRA
Pachyderm
mlflow
Kubeflow
mleap
DVC
SELDON
Polyaxon
snorkel
AI / MACHINE LEARNING / DEEP LEARNING
TensorFlow
Keras
torch
MADlib
learn
Caffe
Microsoft Cognitive Toolkit
OpenAI
DMTK
theano
PaddlePaddle
Apache SINGA
DIMSUM
FeatureFu
mxnet
VELES
Chainer
Michelangelo
ONNX
WEKA
LUDWIG
PyTorch
neon™
DSSTNE
mllib
DL4J
MAHOUT
Aerosolve
fast.ai
mlr
OpenML
SEARCH
elasticsearch
Solr
lucene
LOGGING & MONITORING
elasticsearch
kibana
SENTRY
logstash
Prometheus
fluentbit
fluentd
Grafana
Timber.io
VECTOR
VISUALIZATION
matplotlib
TensorBoard
seaborn
ggplot2
Bokeh
COLLABORATION
BeakerX
jupyter
Zeppelin
ANACONDA
SECURITY
Apache Ranger
KNOX
Sentry
accumulo

End to End
Off the Shelf Solution

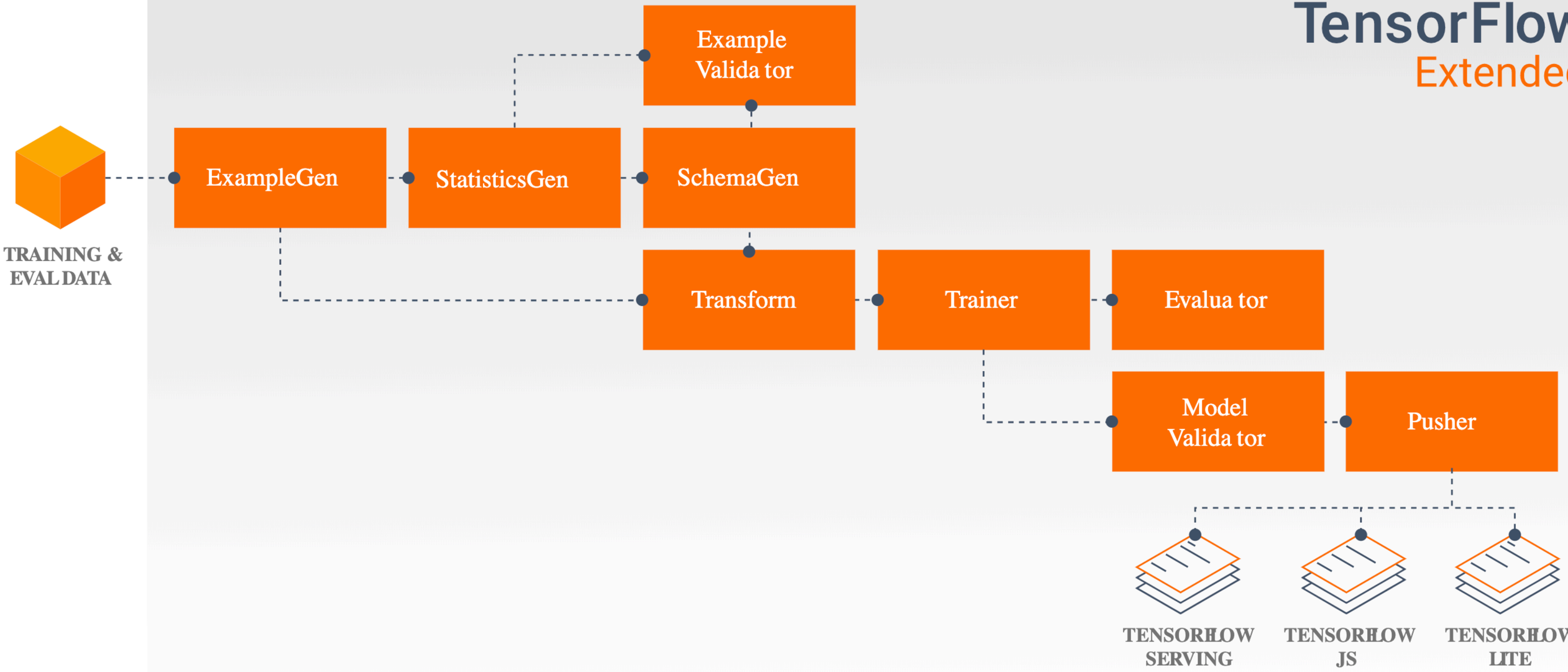
TensorFlow Extended



TensorFlow Extended



TensorFlow Extended



A Concrete Examples

Gmail

Search mail

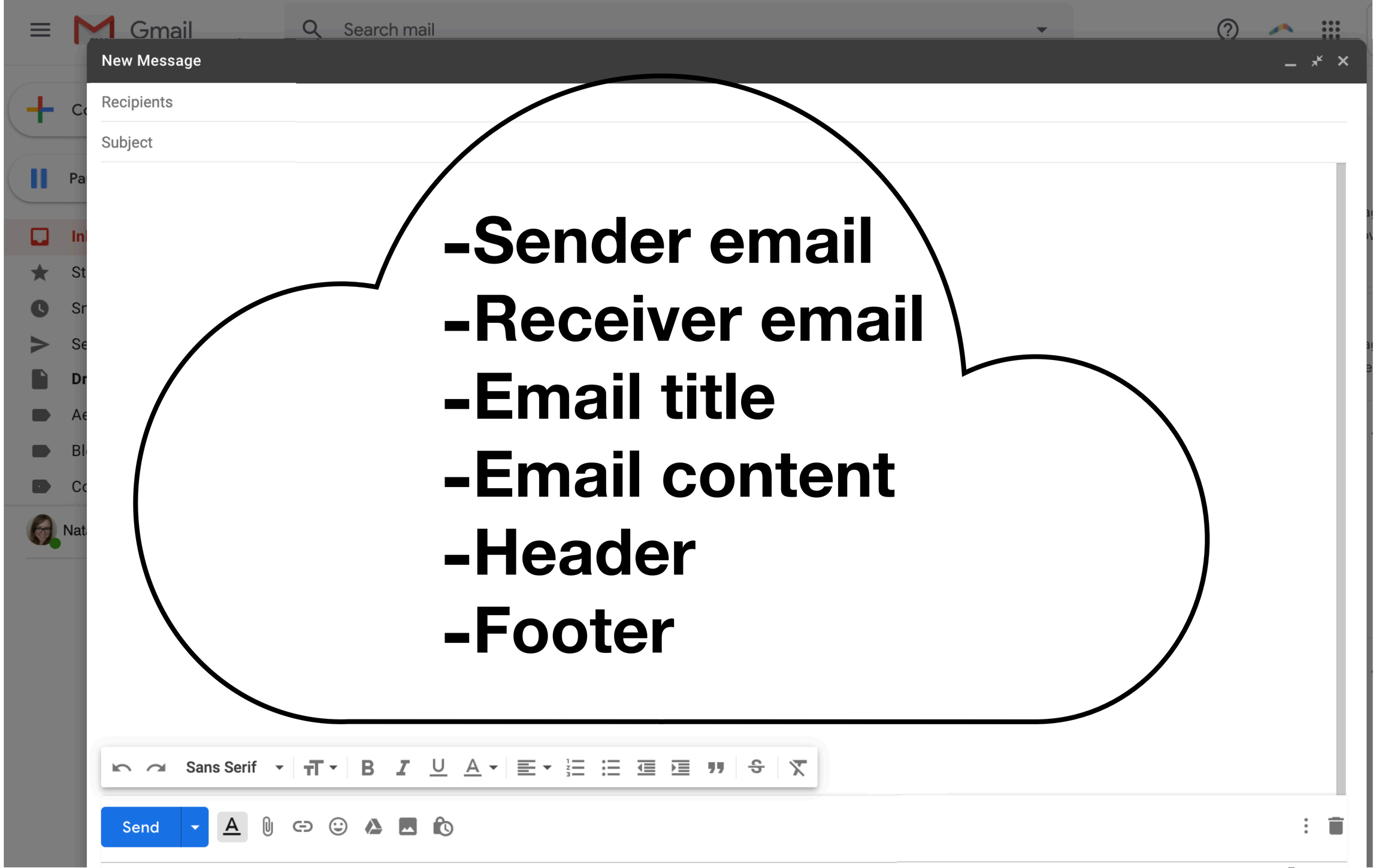
New Message

Recipients

Subject

Sans Serif

Send


Gmail
Search mail
New Message
Recipients
Subject
-Sender email
-Receiver email
-Email title
-Email content
-Header
-Footer
Sans Serif
Send


Gmail
New Message
Recipients
Subject
Sans Serif
Send
- Sender email
- Receiver email
- Email title
- Email content
- Header
- Footer


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

AEROSPIKE

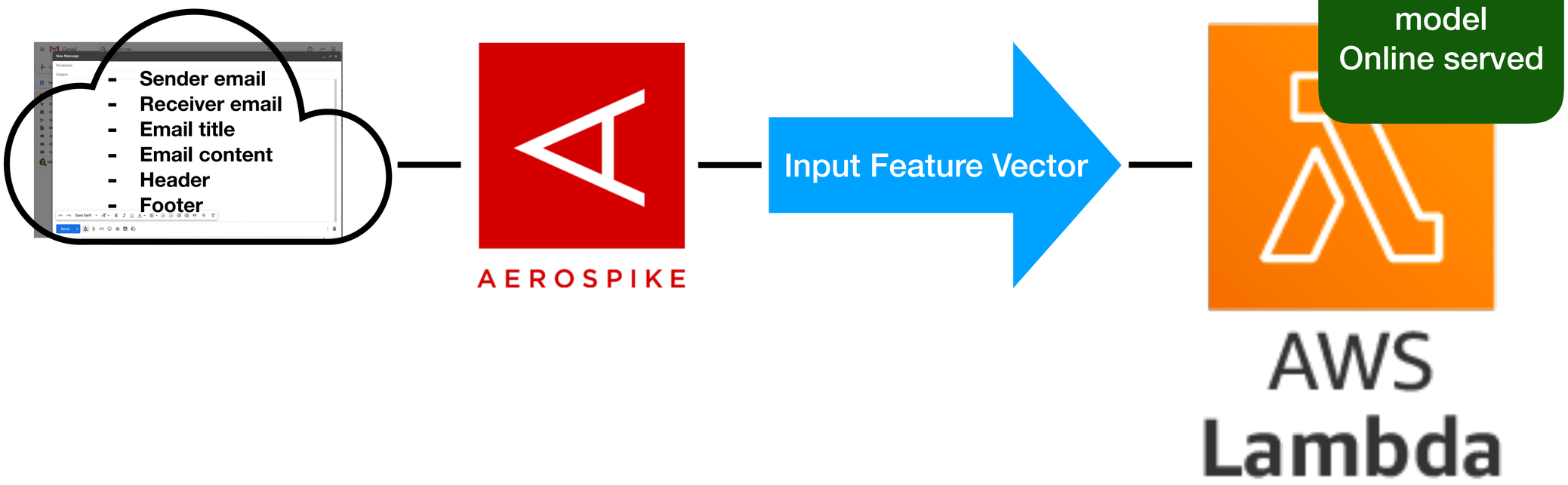
Presence of a recognised header	1
Presence of an official signature	0
Email structure	0.7
Language	0.78
Frequency of “special price”	0
Frequency of “prince”	0.9
Grammatical correctness	0.61

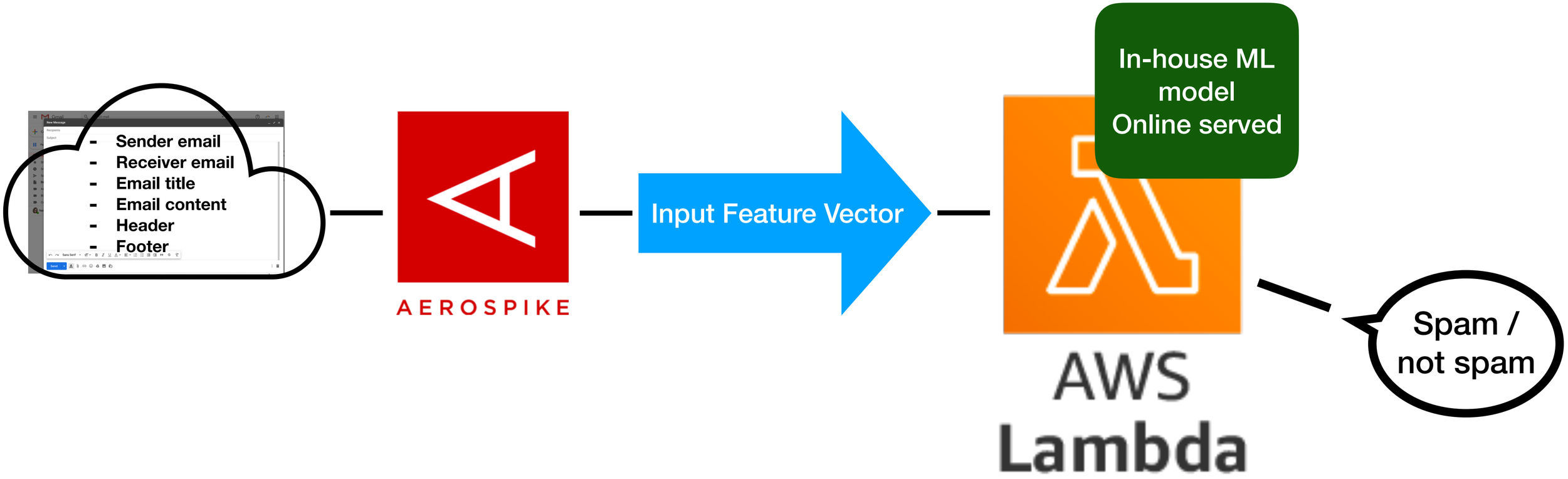

- Sender email
- Receiver email
- Email title
- Email content
- Header
- Footer
AEROSPIKE
Input Feature Vector

AEROSPIKE

Input Feature Vector

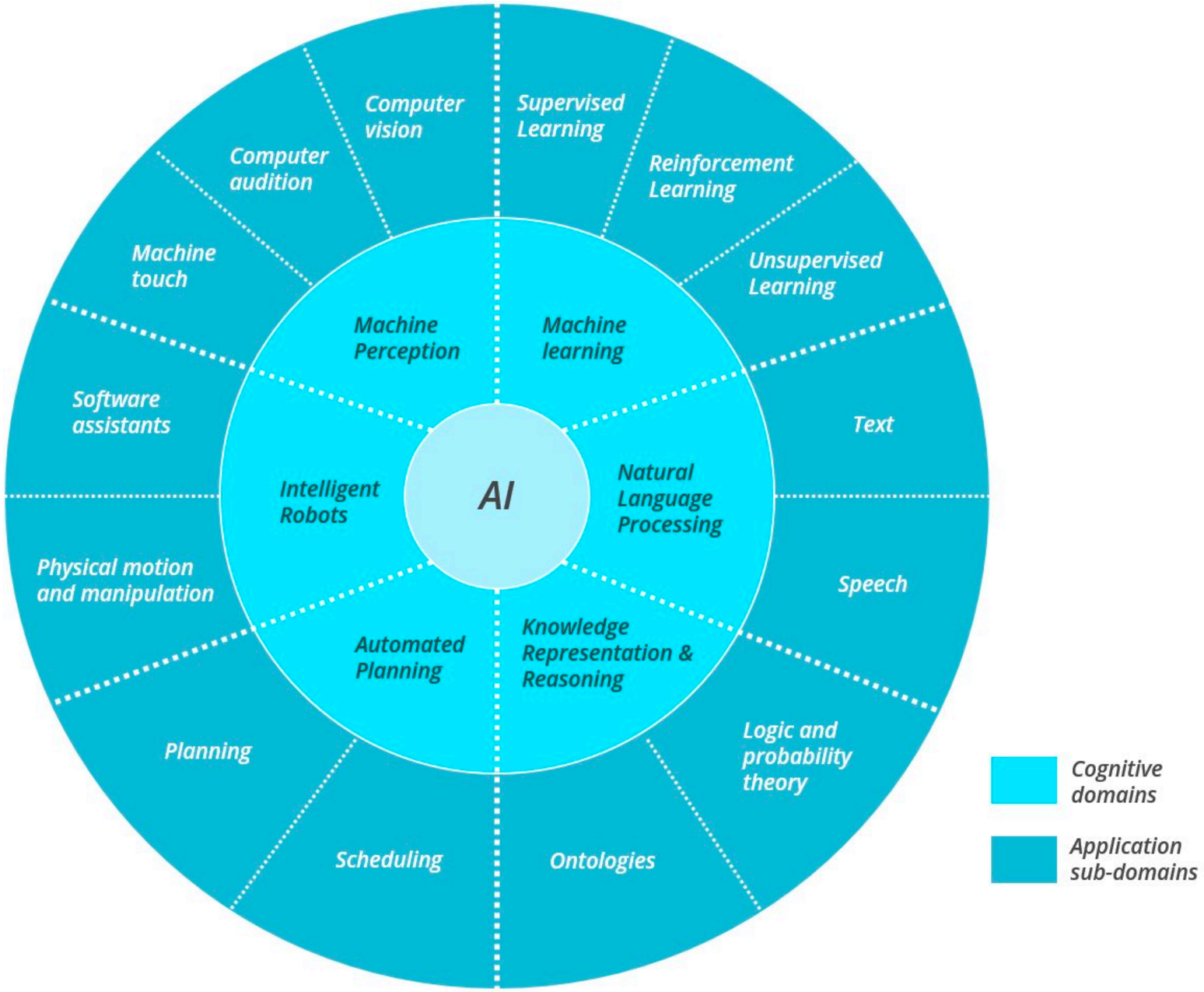
AWS Lambda


- Sender email
- Receiver email
- Email title
- Email content
- Header
- Footer
AEROSPIKE
Input Feature Vector
In-house ML model Online served
AWS Lambda


- Sender email
- Receiver email
- Email title
- Email content
- Header
- Footer
AEROSPIKE
Input Feature Vector
In-house ML model Online served
AWS Lambda
Spam / not spam

Recap

- What’s AI? What’s ML? How to ML?

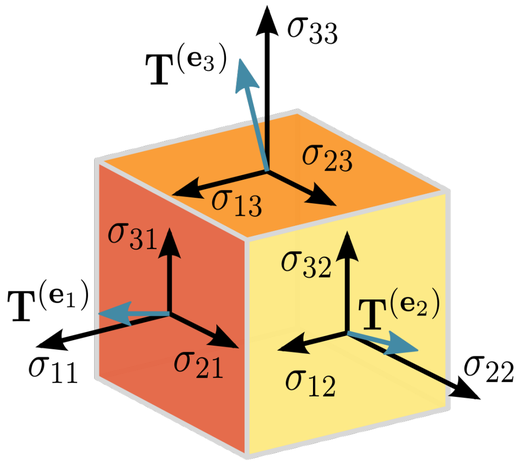


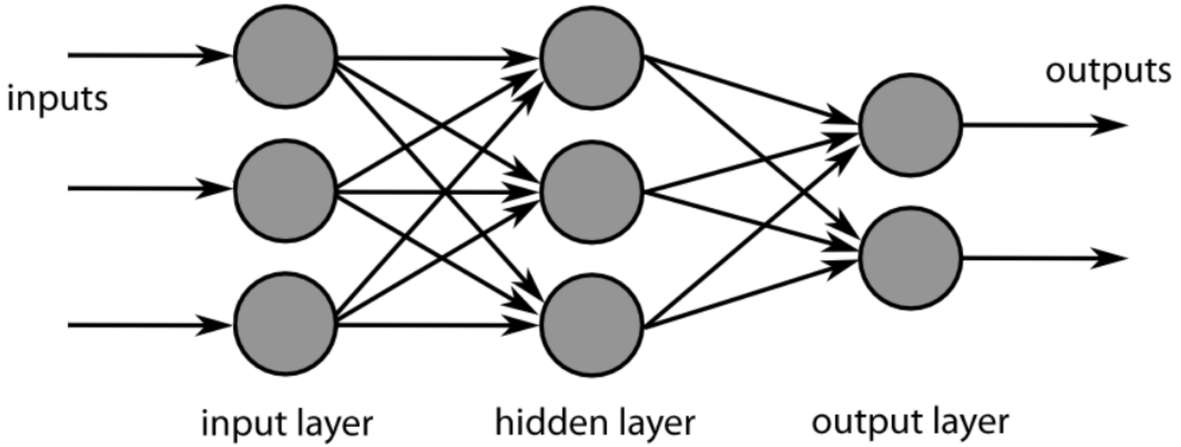
@theAlexFreeman

Recap

- What’s AI? What’s ML? How to ML?
- TensorFlow

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

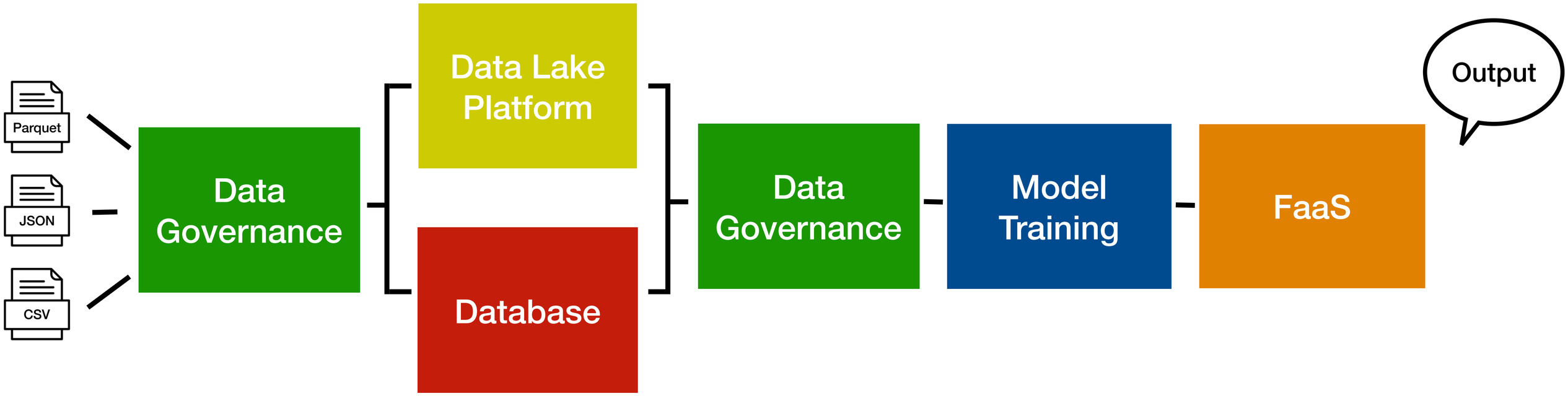
You can look at, and evaluate, any node of the graph

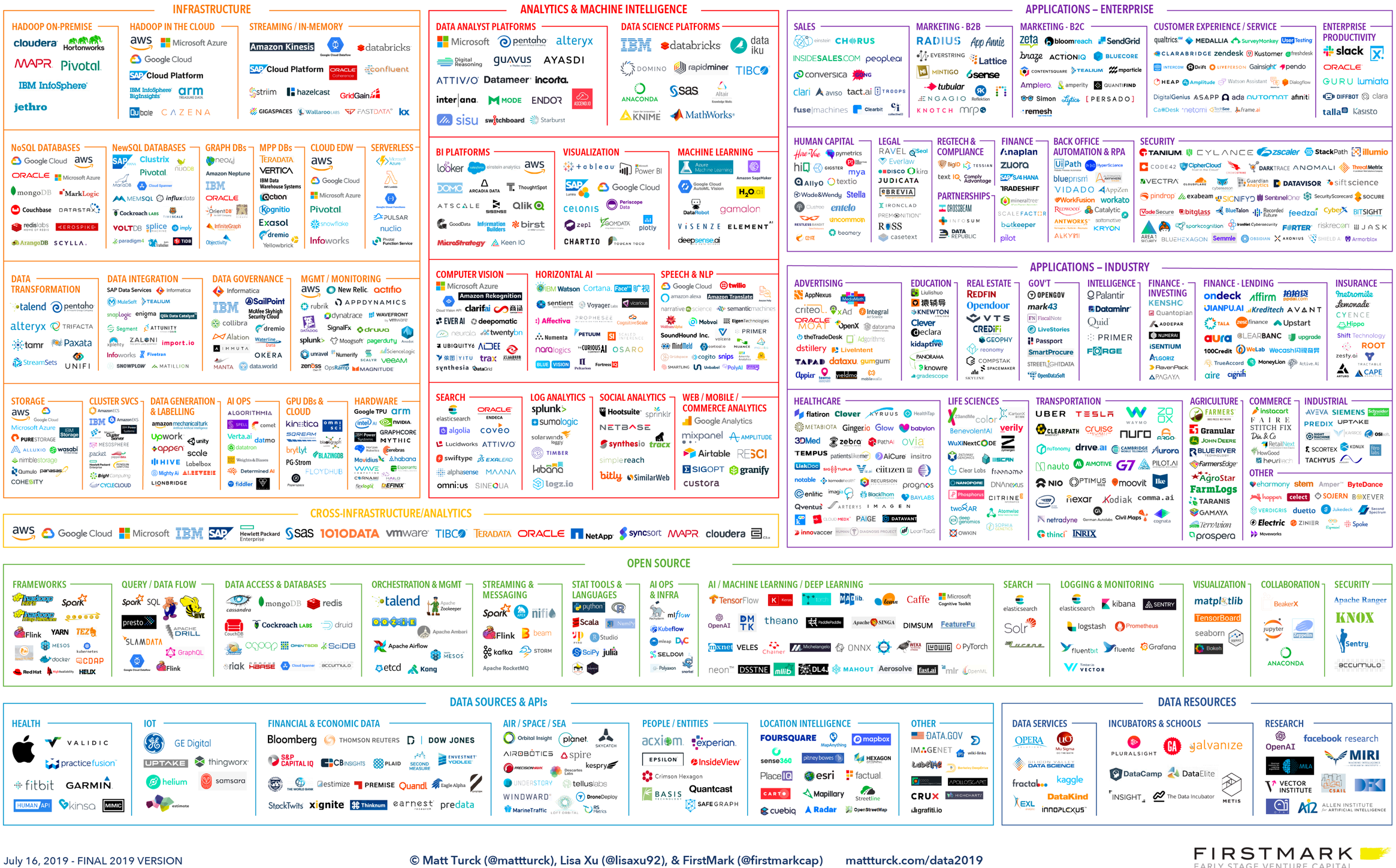
Recap

- What's AI? What's ML? How to ML?
- TensorFlow
- Training and Inference: Online vs. Offline

Recap

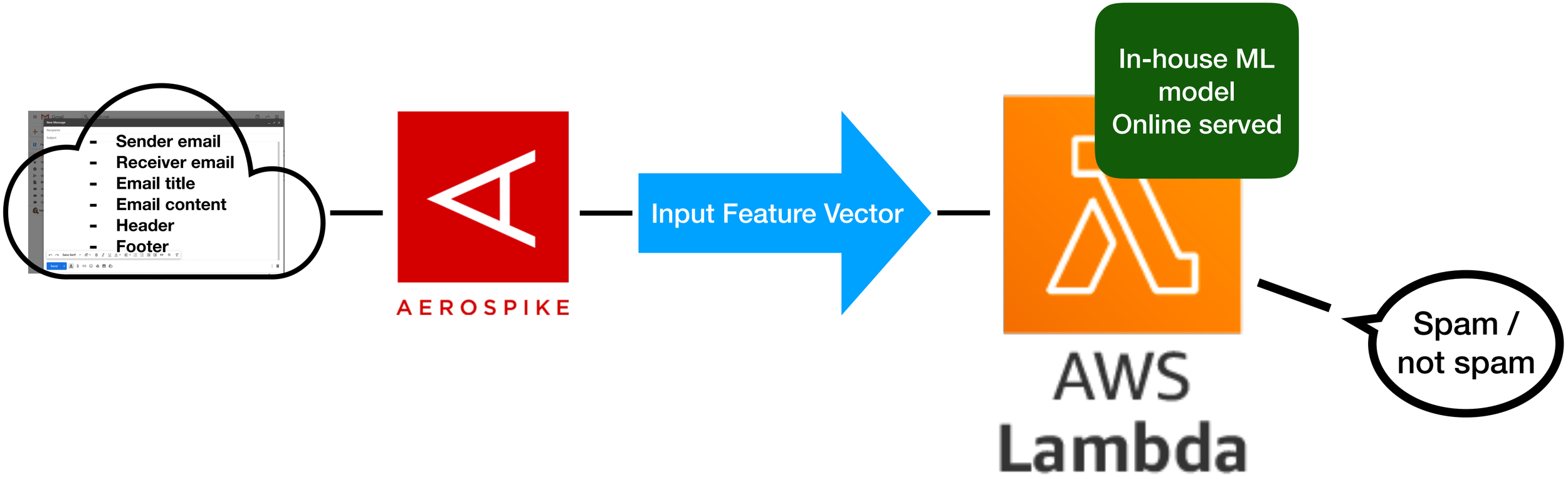
- What's AI? What's ML? How to ML?
- TensorFlow
- Training and Inference: Online vs. Offline
- Bare bones architecture components


Parquet
JSON
CSV
Data Governance
Data Lake Platform
Database
Data Governance
Model Training
FaaS
Output



Recap

- What's AI? What's ML? How to ML?
- TensorFlow
- Training and Inference: Online vs. Offline
- Bare bones architecture components
- A concrete architecture example


- Sender email
- Receiver email
- Email title
- Email content
- Header
- Footer
AEROSPIKE
Input Feature Vector
In-house ML model Online served
AWS Lambda
Spam / not spam

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

@golangwaw
#golang

Thank You!

@NataliePis

@golangwaw #golang