The New DBfication of ML/AI

Arun Kumar

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Computer Science and Engineering

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

San Diego ML Meetup: HDSI Data Science Insights September 22, 2021

Golden Age of ML/AI

Golden Age of ML/AI

amazon FACEBOOK Google Microsoft

MAYO CLINIC

Healthcare

AMERICAN FAMILY INSURANCE

Insurance

Walmart

Retail

Sciences

Golden Age of ML/AI

amazon FACEBOOK Google Microsoft

MAYO CLINIC Healthcare

AMERICAN FAMILY INSURANCE® Insurance

Walmart Retail

 Sciences

$ 38 billion in 2019*

→

$ 500 billion! by 2024*

*International Data Corporation

Golden Age of ML/AI

amazon FACEBOOK Google Microsoft

MAYO CLINIC

Healthcare

AMERICAN FAMILY INSURANCE

Insurance

Walmart

Retail

Sciences

$ 38 billion
in 2019*

$ 500 billion!
by 2024*

scikit learn dmlc XGBoost APACHE Spark ML

TensorFlow PyTorch

*International Data Corporation

Golden Age of ML/AI

amazon FACEBOOK Google Microsoft

MAYO CLINIC Healthcare

AMERICAN FAMILY INSURANCE Insurance

Walmart Retail

 Sciences

$ 38 billion
in 2019*

→

$ 500 billion!
by 2024*

scikit learn dmlc XGBoost APACHE Spark ML

TensorFlow PyTorch

amazon Google Microsoft

*International Data Corporation

Golden Age of ML/AI

amazon FACEBOOK Google Microsoft

MAYO CLINIC — Healthcare

AMERICAN FAMILY INSURANCE — Insurance

Walmart — Retail

 Sciences

$ 38 billion in 2019* → **$ 500 billion!** by 2024*

Still, fundamental **efficiency and usability bottlenecks** in the **end-to-end process** of building and deploying ML applications

*International Data Corporation

My Research

New abstractions, algorithms, and software systems to "***democratize***" ML/AI-based data analytics from a data management/systems standpoint

My Research

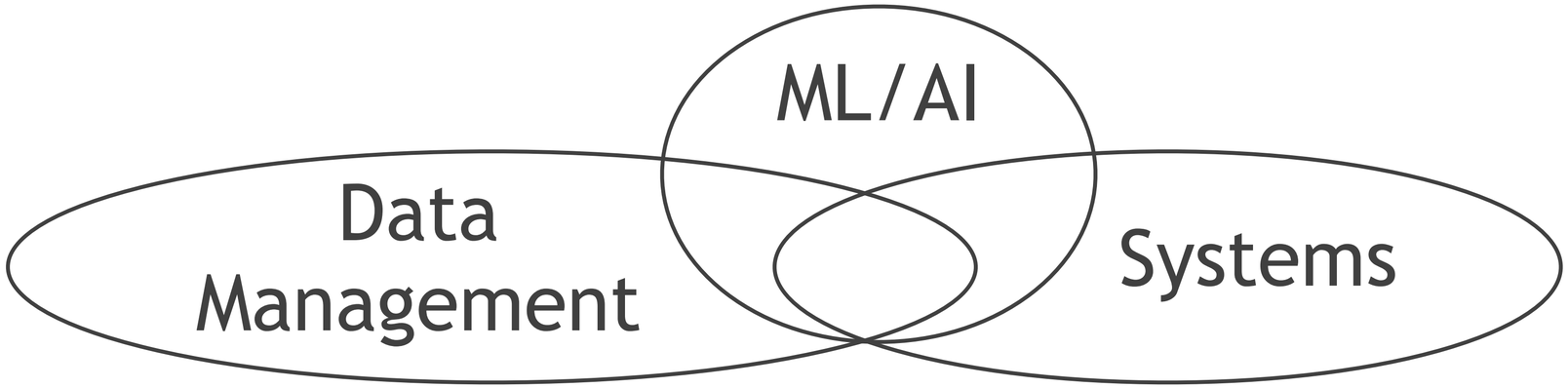
New abstractions, algorithms, and software systems to "***democratize***" ML/AI-based data analytics from a data management/systems standpoint

Democratization = System Efficiency (Reduce costs) + Human Efficiency (Improve productivity)

My Research

New abstractions, algorithms, and software systems to "***democratize***" ML/AI-based data analytics from a data management/systems standpoint

Democratization = System Efficiency (Reduce costs) + Human Efficiency (Improve productivity)



My Research

New abstractions, algorithms, and software systems to "***democratize***" ML/AI-based data analytics from a data management/systems standpoint

Democratization = System Efficiency (Reduce costs) + Human Efficiency (Improve productivity)

Practical and scalable data systems for ML/AI analytics

Inspired by *relational database systems* principles

Exploit insights from *learning theory* and *optimization theory*

End-to-End ML Application Lifecycle

Data Scientist/
ML Engineer

Source → Build → Deploy

Data + ML Systems Implementations

https://ADALabUCSD.github.io

End-to-End ML Application Lifecycle

Data Scientist/
ML Engineer

Source → Build → Deploy

Data + ML Systems Implementations

Research Approach: ***Abstract*** key steps + ***Formalize*** computation + ***Automate*** grunt work + ***Optimize*** execution

https://ADALabUCSD.github.io

Outline

- The New DBfication of ML/AI
- Two Examples from My Research
- Accelerating the DBfication of ML/AI

DB-style Practical Concerns in ML

DB-style Practical Concerns in ML

Key concerns in ML:

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Q: What if the dataset is larger than single-node RAM?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Q: What if the dataset is larger than single-node RAM?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Q: How are the features and models configured?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Q: How are the features and models configured?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Q: How does it fit within production systems and workflows?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Q: How does it fit within production systems and workflows?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Q: How to simplify the implementation of such systems?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Developability

Q: How to simplify the implementation of such systems?

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Developability

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Developability

} *Long-standing concerns in the **DB systems** world!*

DB-style Practical Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Developability

} *Long-standing concerns in the **DB systems** world!*

Can often trade off accuracy a bit to gain on the rest!

The Push to “Platformize” ML/AI

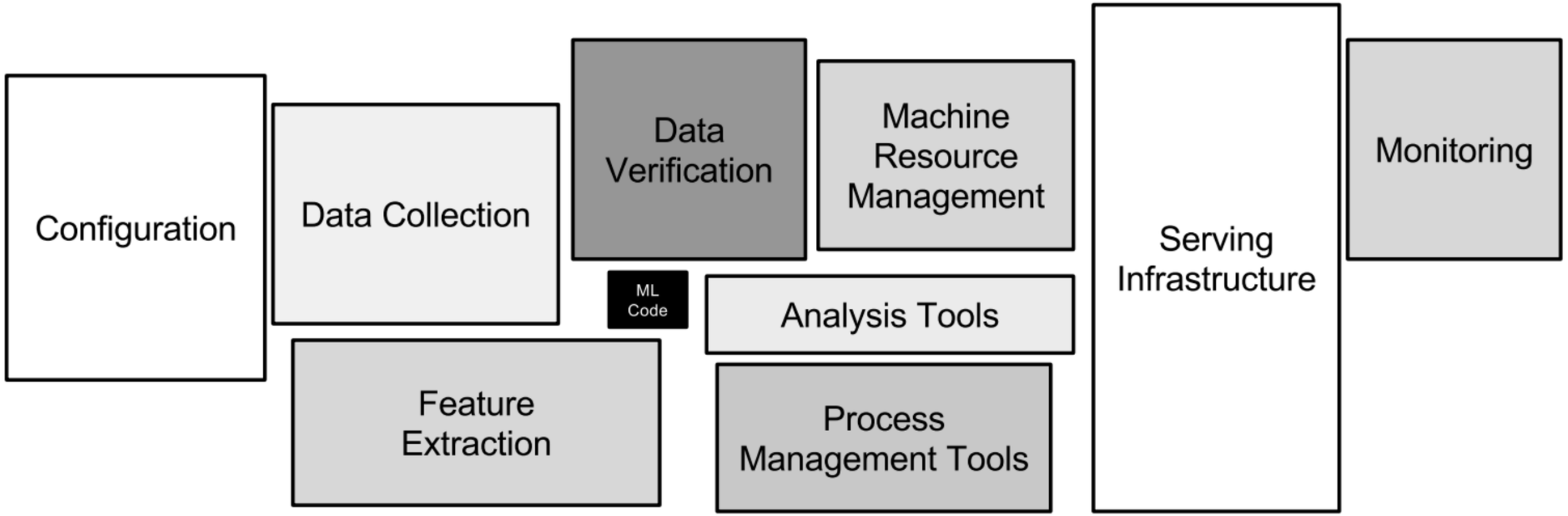


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Google

Hidden Technical Debt in Machine Learning Systems. ***NIPS’15***

A Brief History of “Platformizing” ML

A Brief History of “Platformizing” ML

1980s

S

A Brief History of “Platformizing” ML

Mid
1990s

1980s

S

A Brief History of “Platformizing” ML

Mid
1990s

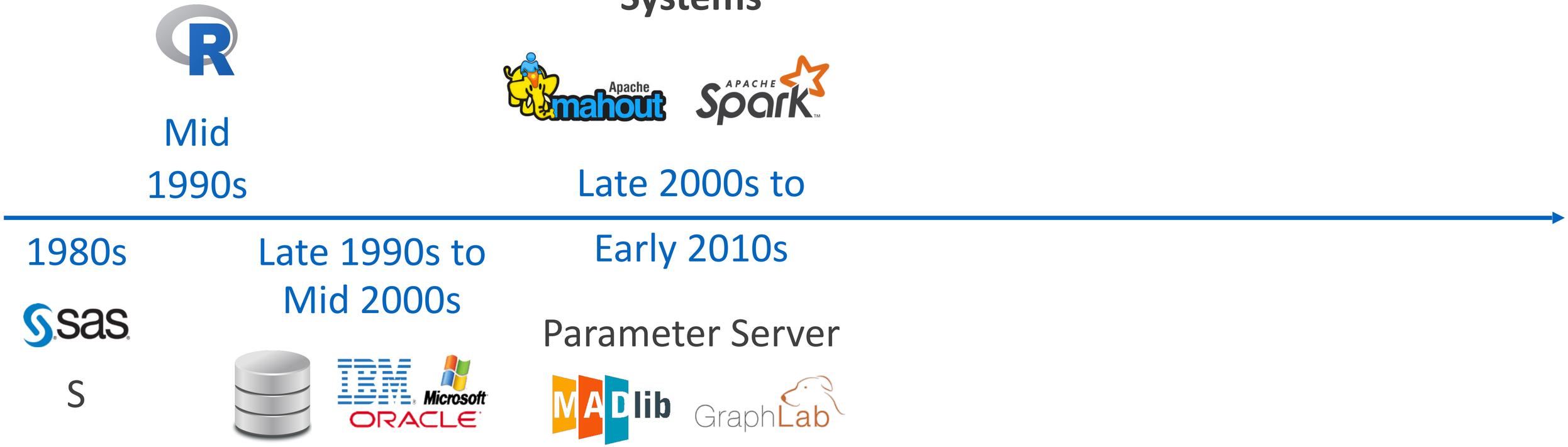
1980s

S

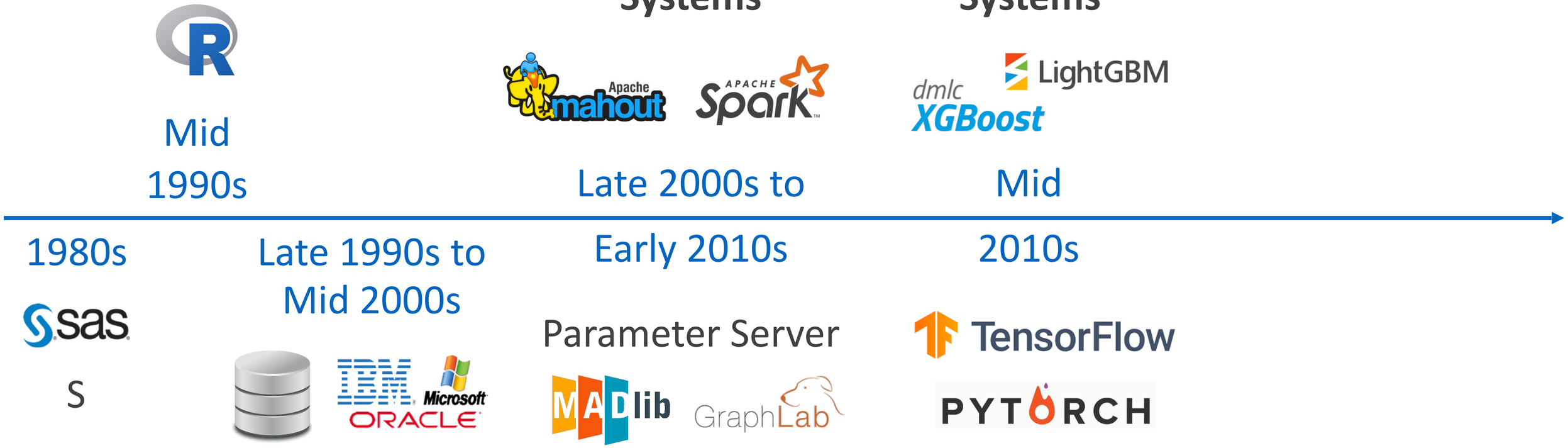
Late 1990s to
Mid 2000s

In-RDBMS
ML Systems

A Brief History of “Platformizing” ML



A Brief History of “Platformizing” ML



A Brief History of “Platformizing” ML

1980s

S

Mid 1990s

Late 1990s to Mid 2000s

In-RDBMS ML Systems

Late 2000s to Early 2010s

ML on Dataflow Systems

Parameter Server

ML System Abstractions

Mid 2010s

Tree Learning Systems

Deep Learning Systems

Late 2010s Onwards

Amazon SageMaker

Cloud ML/AI Services

ML Platforms and Feature Stores

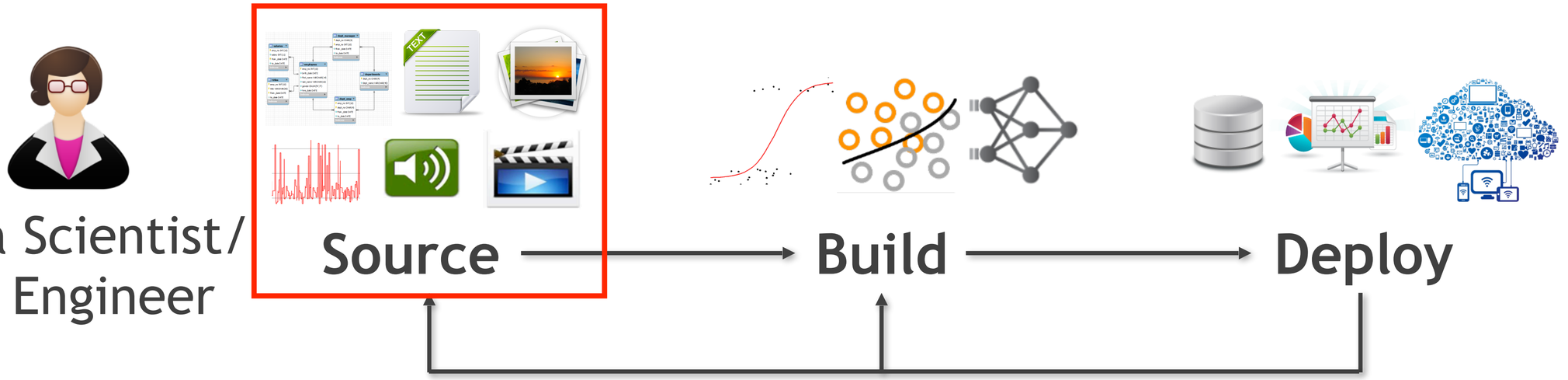
The way I see it, the rise of ML systems/ platforms today resembles the rise of RDBMSs circa early 1980s

The New DBfication of ML/AI

Data Scientist/
ML Engineer

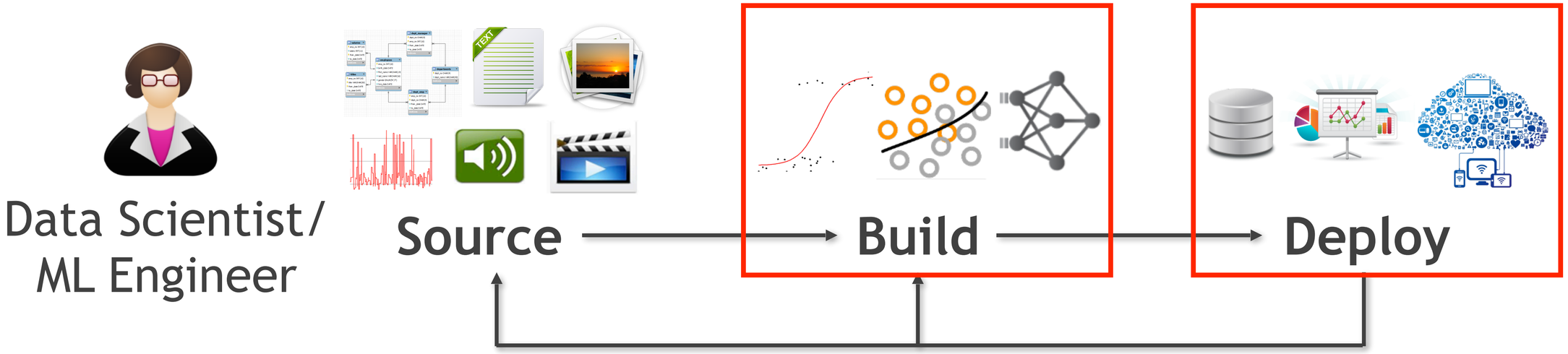
Source → Build → Deploy

The New DBfication of ML/AI



Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

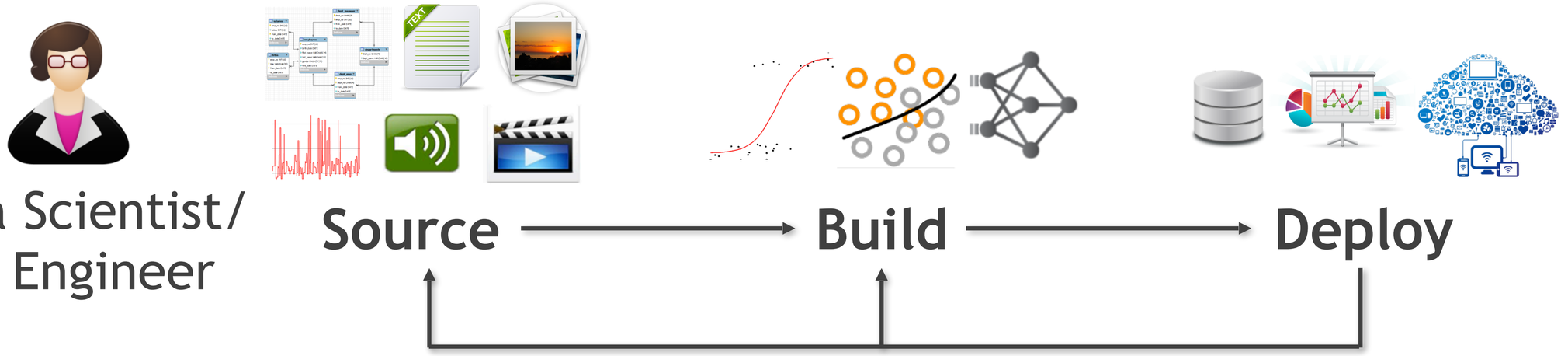
The New DBfication of ML/AI



Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

The New DBfication of ML/AI



Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

Benchmark Frameworks and Data
Fairness, Transparency, Privacy, etc.

Outline

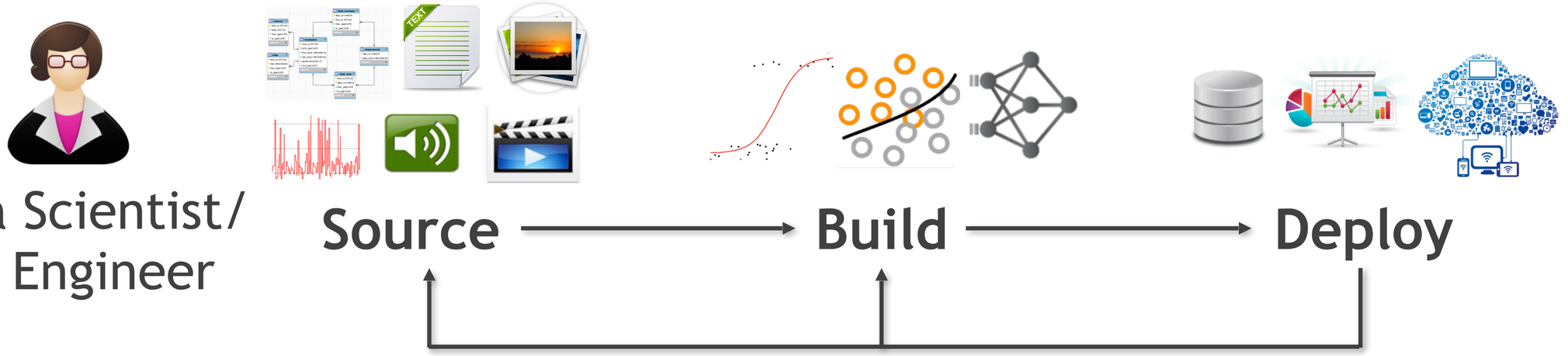
- The New DBfication of ML/AI
- Two Examples from My Research
- Accelerating the DBfication of ML/AI

The New DBfication of ML/AI

Data Scientist/
ML Engineer

Source → Build → Deploy

The New DBfication of ML/AI

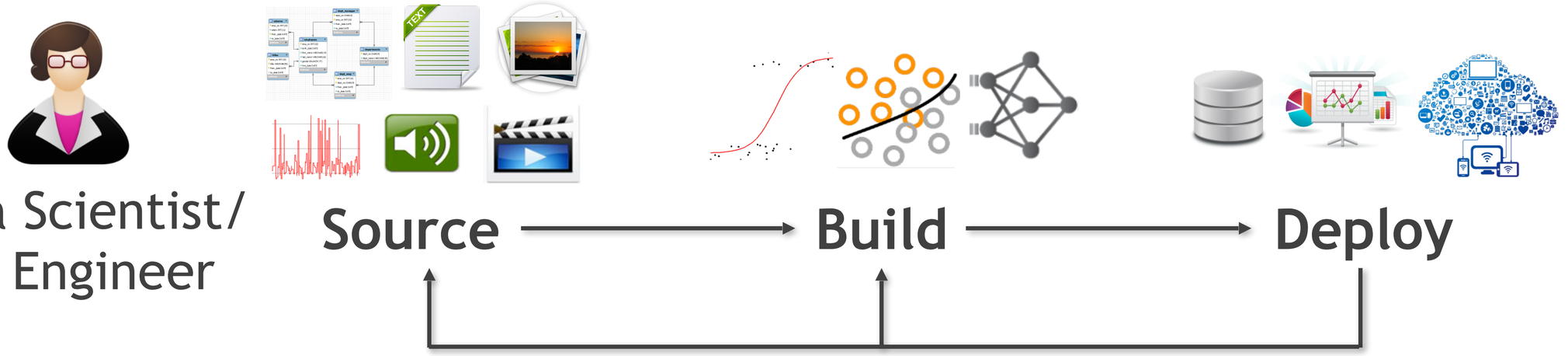


Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

Benchmark Frameworks and Data
Fairness, Transparency, Privacy, etc.

The New DBfication of ML/AI

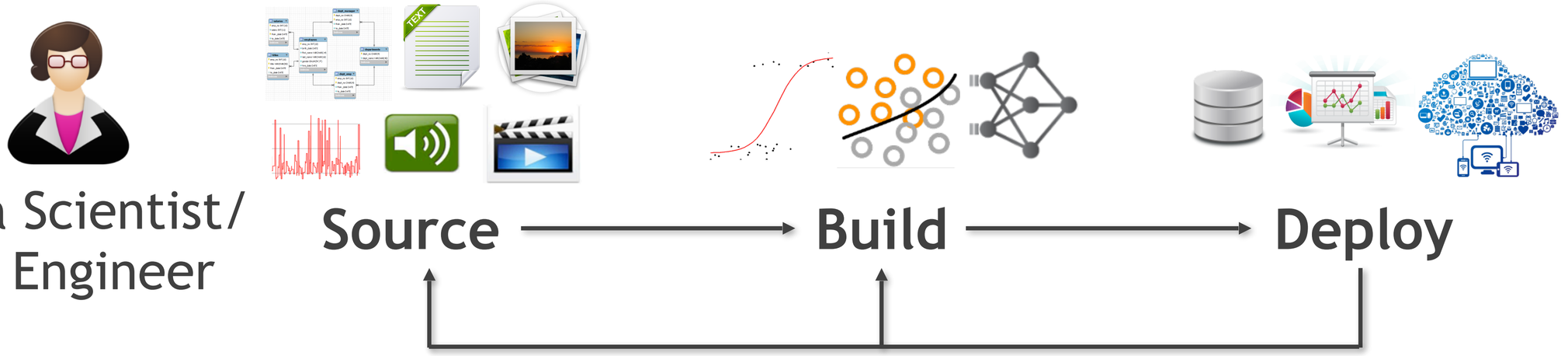


Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

Benchmark Frameworks and Data
Fairness, Transparency, Privacy, etc.

The New DBfication of ML/AI

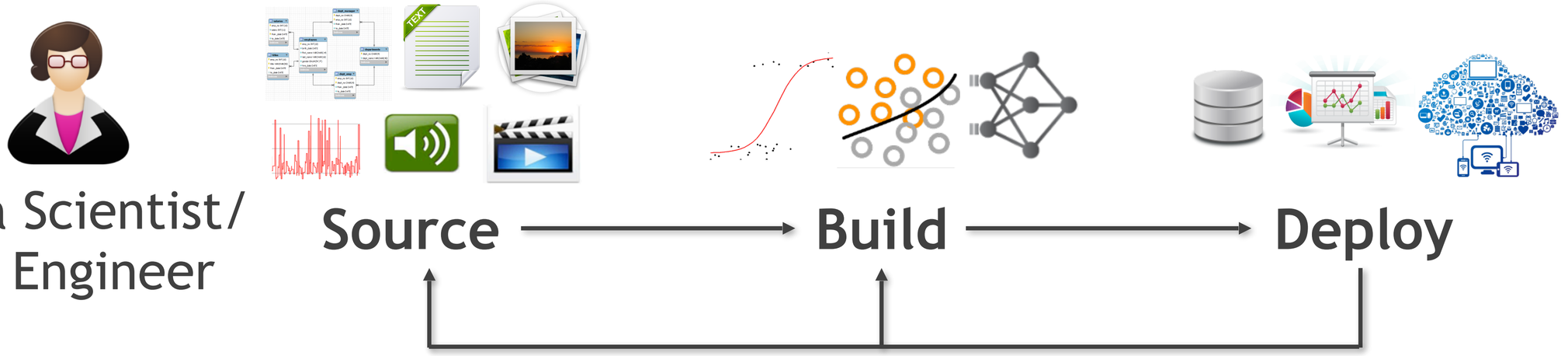


Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

Benchmark Frameworks and Data
Fairness, Transparency, Privacy, etc.

The New DBfication of ML/AI



Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

Benchmark Frameworks and Data

Fairness, Transparency, Privacy, etc.

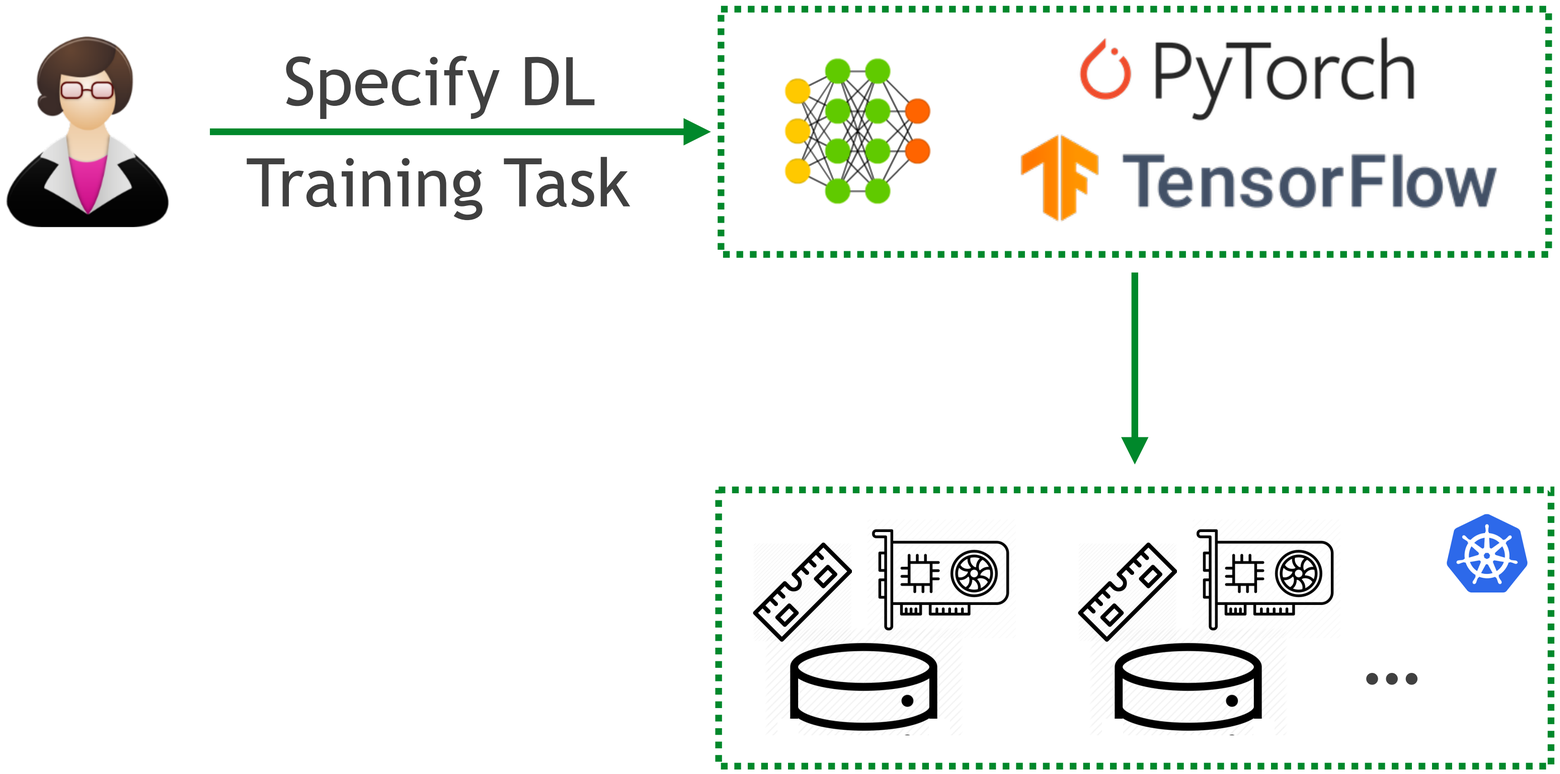
Outline

- The New DBfication of ML/AI
- Two Examples from My Research
  - Example 1: Scalable DL Systems
  - Example 2: Auto Data Prep for ML
- Accelerating the DBfication of ML/AI

Outline

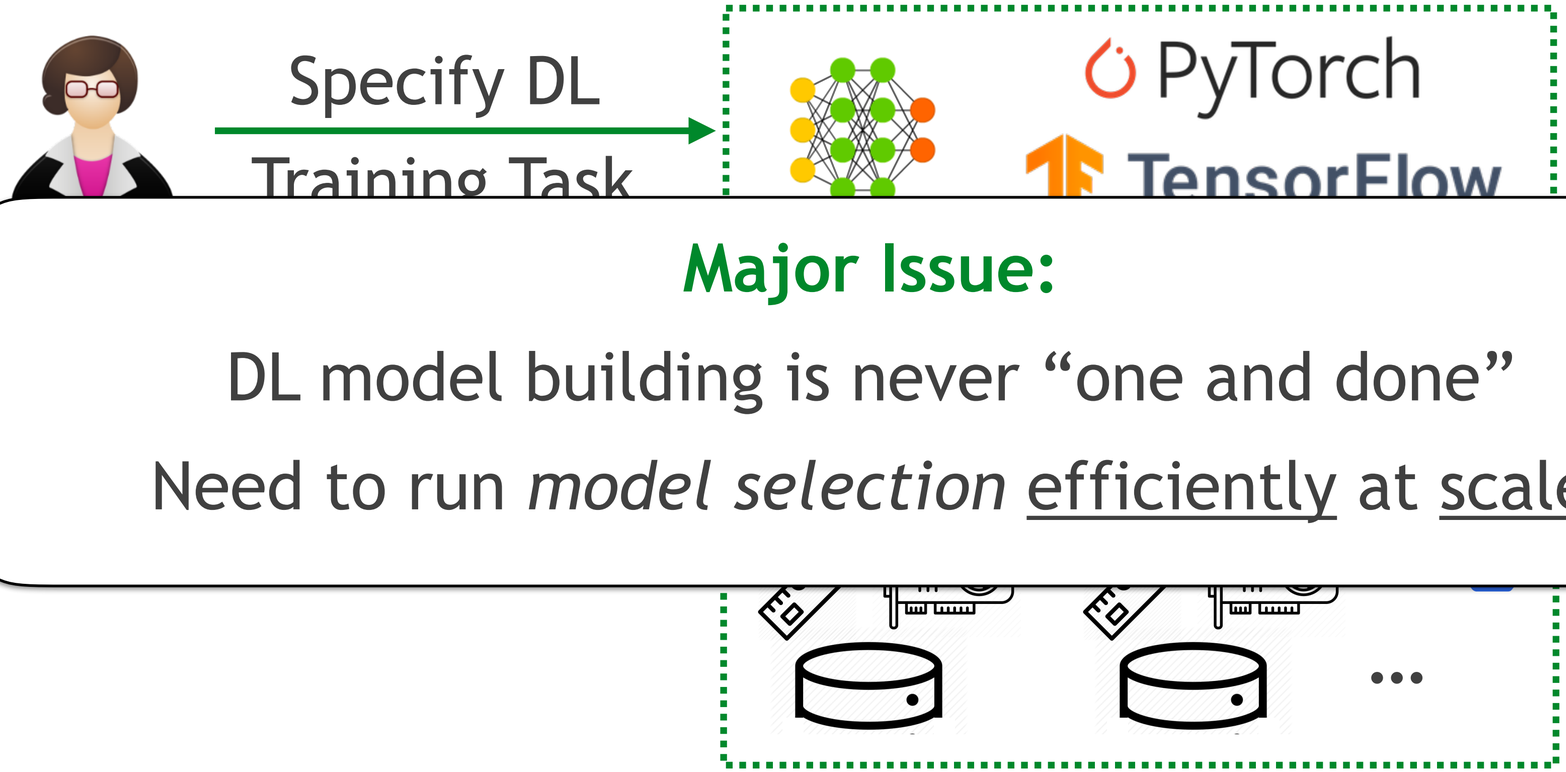
- The New DBfication of ML/AI
- Two Examples from My Research
  - Example 1: Scalable DL Systems
  - Example 2: Auto Data Prep for ML
- Accelerating the DBfication of ML/AI

Example 1: DL Systems at Scale



Cluster: GPU+CPU, memory, disks

Example 1: DL Systems at Scale



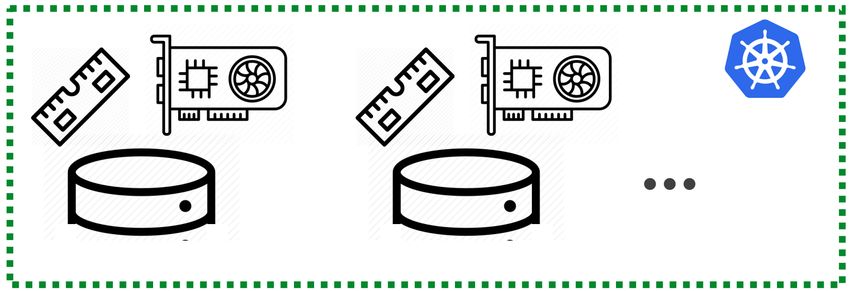
Major Issue:

DL model building is never “one and done”

Need to run *model selection* efficiently at scale

Cluster: GPU+CPU, memory, disks

Project Cerebro



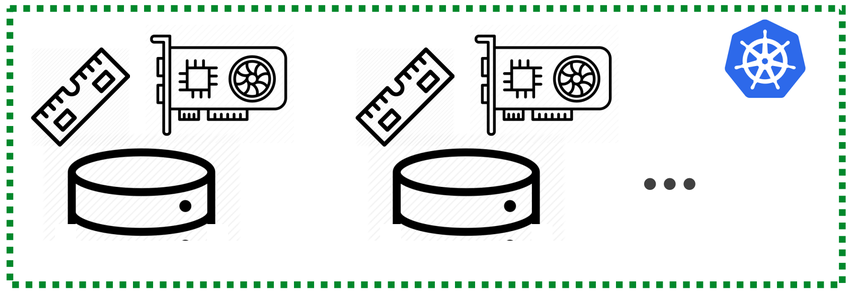
Plain/Kubernetes cluster

Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Project Cerebro

Specify DL Model

Selection Holistically



Plain/Kubernetes cluster

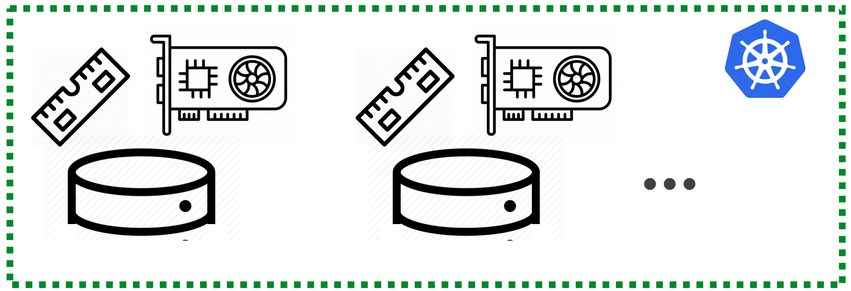
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Project Cerebro

Specify DL Model

Selection Holistically

Magic of Multi-Query Optimization for DL



Plain/Kubernetes cluster

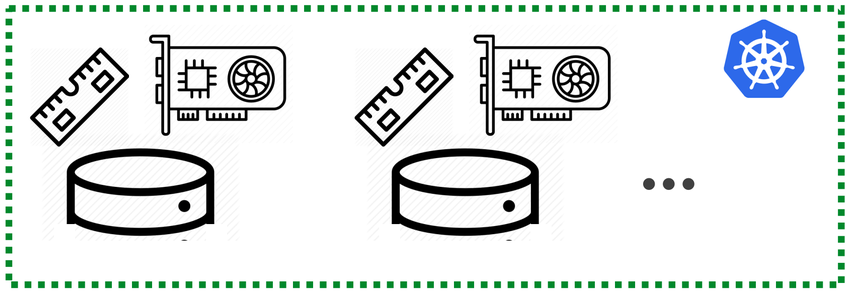
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Project Cerebro

Specify DL Model

Selection Holistically

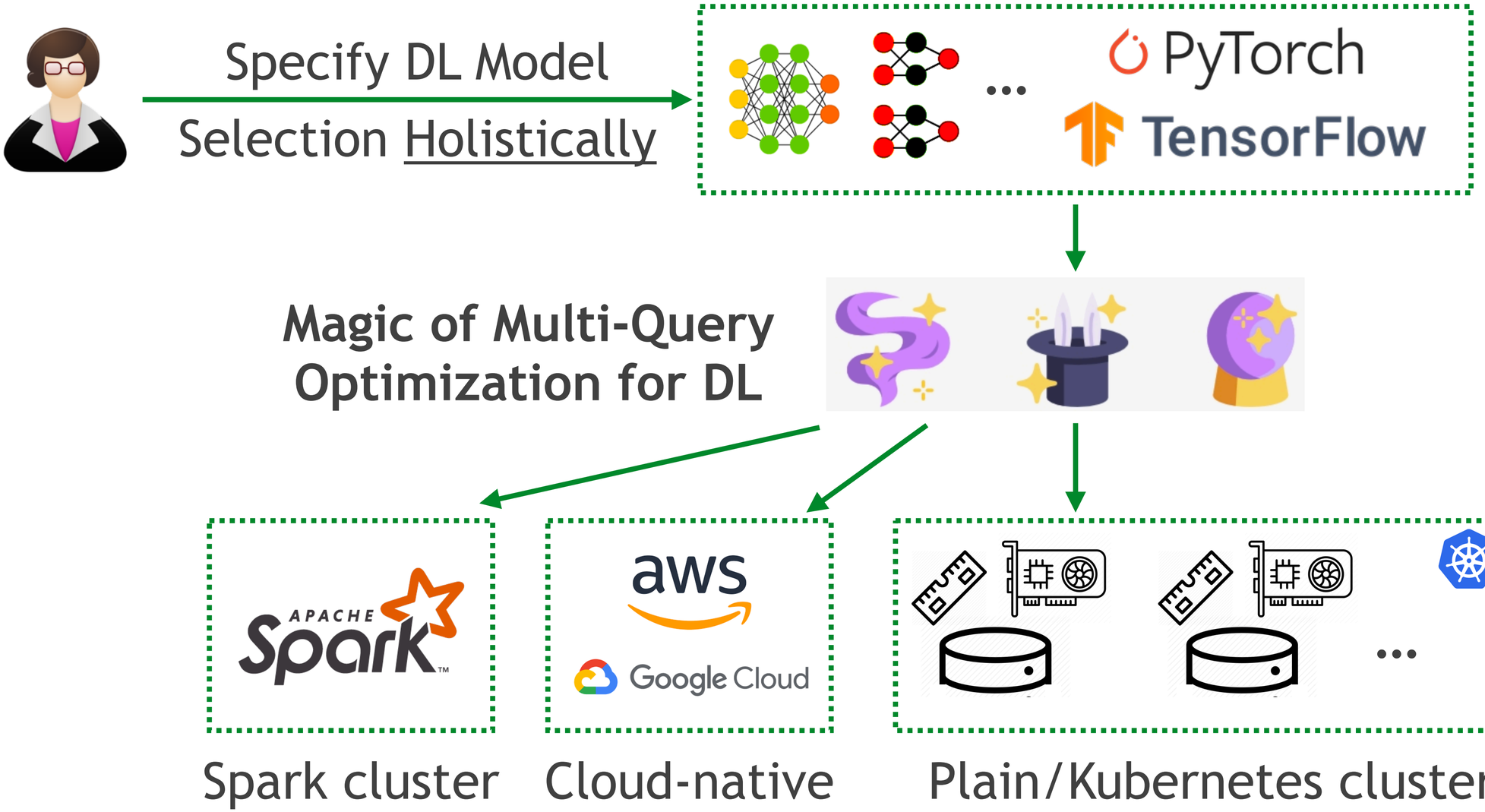
Magic of Multi-Query Optimization for DL



Plain/Kubernetes cluster

Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Project Cerebro

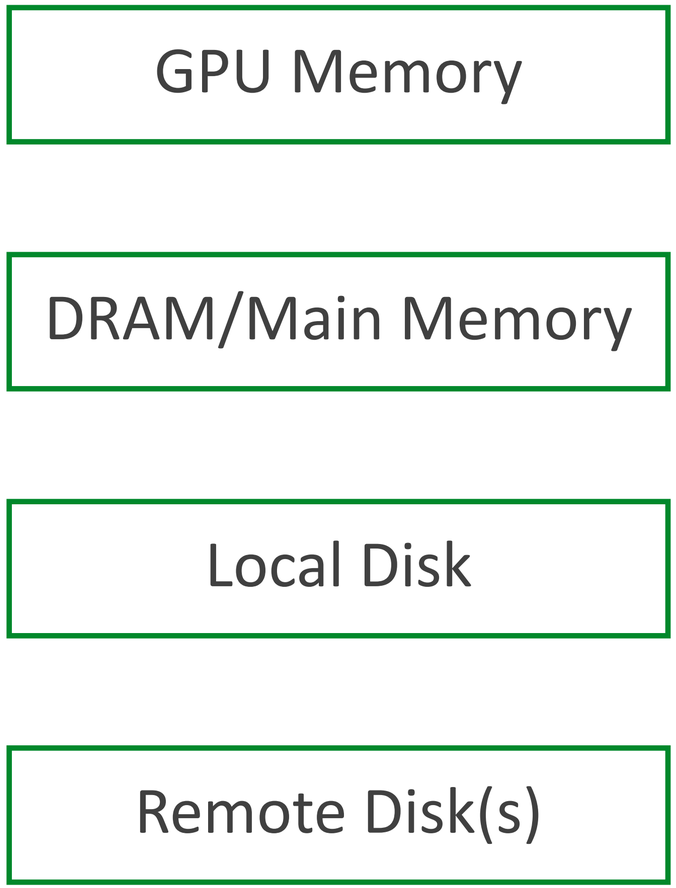


Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

So, what kinds of scalability issues arise?

What optimizations does Cerebro do?

DL Scalability Issues on Memory Hierarchy



DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

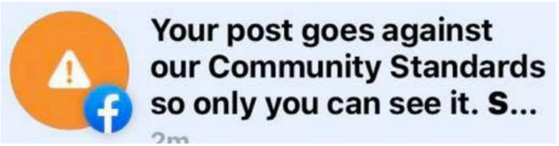
GPU Memory

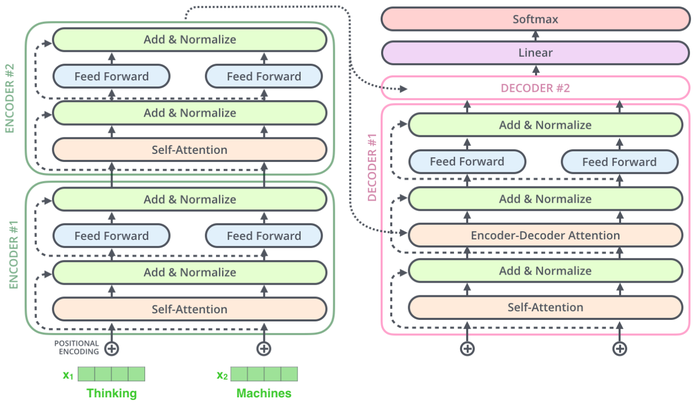
DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!





What Cerebro Does (Under Submission)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!

Hydra: A Scalable and Optimized Data System for Large Multi-Model Deep Learning. 2021

What Cerebro Does (Under Submission)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!

1. Automatic model *sharding* and *spilling*

Hydra: A Scalable and Optimized Data System for Large Multi-Model Deep Learning. 2021

What Cerebro Does (Under Submission)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!

1. Automatic model *sharding* and *spilling*

We rewrite computational graph to break it up on DRAM

Hydra: A Scalable and Optimized Data System for Large Multi-Model Deep Learning. 2021

What Cerebro Does (Under Submission)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!

1. Automatic model *sharding* and *spilling*

We rewrite computational graph to break it up on DRAM

Interleave multiple model parts across GPUs to raise throughput

Hydra: A Scalable and Optimized Data System for Large Multi-Model Deep Learning. 2021

What Cerebro Does (Under Submission)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

1. DL model and/or data batch does not fit in GPU memory!

1. Automatic model *sharding* and *spilling*

We rewrite computational graph to break it up on DRAM

Interleave multiple model parts across GPUs to raise throughput

Seamlessly scales to any size

Hydra: A Scalable and Optimized Data System for Large Multi-Model Deep Learning. 2021

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

2. Dataset does not fit in DRAM; GNNs do random I/Os to disk!

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

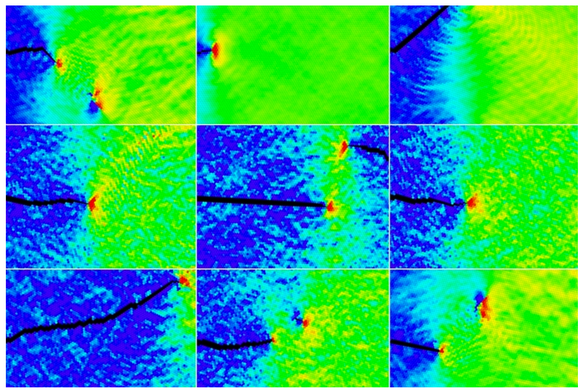
GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

2. Dataset does not fit in DRAM; GNNs do random I/Os to disk!



What Cerebro Will Do (In the Works)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

2. Dataset does not fit in DRAM and random accesses to disk!

What Cerebro Will Do (In the Works)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

2. Dataset does not fit in DRAM and random accesses to disk!

2. I/O optimization with staged reads and caching

What Cerebro Will Do (In the Works)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

2. Dataset does not fit in DRAM and random accesses to disk!

2. I/O optimization with staged reads and caching

We rewrite DL computation to reduce random I/Os

Memory manager to cache data and reuse across many models

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

3. Dataset is too big for one node to finish on time!
Copying full data across all cluster nodes wastes DRAM/disk

DL Scalability Issues on Memory Hierarchy

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

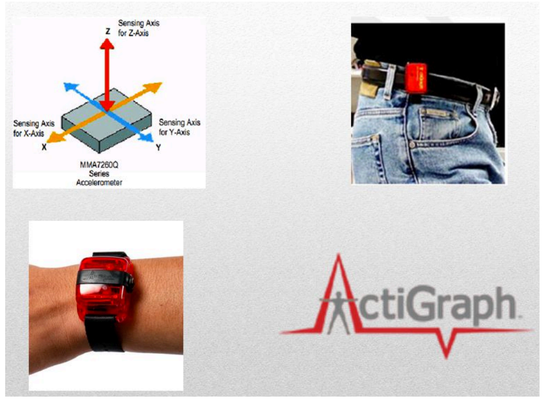
Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)



UC San Diego SCHOOL OF MEDICINE

3. Dataset is too big for one node to finish on time!
Copying full data across all cluster nodes wastes DRAM/disk

What Cerebro Does (Already Published)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

3. Dataset is too big for one node to finish on time!
Copying full data across all cluster nodes wastes DRAM/disk
Copying wastes DRAM/disk

What Cerebro Does (Already Published)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

3. Dataset is too big for one node to finish on time!
Copying full data across all cluster nodes wastes DRAM/disk
Copying wastes DRAM/disk

3. MQO for DL Model Selection

What Cerebro Does (Already Published)

Fastest data access
Tiny; Costliest

Slowest data access
Large; Cheaper

Multiplicity

GPU Memory

DRAM/Main Memory

Local Disk

Remote Disk(s)

3. Dataset is too big for one node to finish on time!
Copying full data across all cluster nodes wastes DRAM/disk
Copying wastes DRAM/disk

3. MQO for DL Model Selection

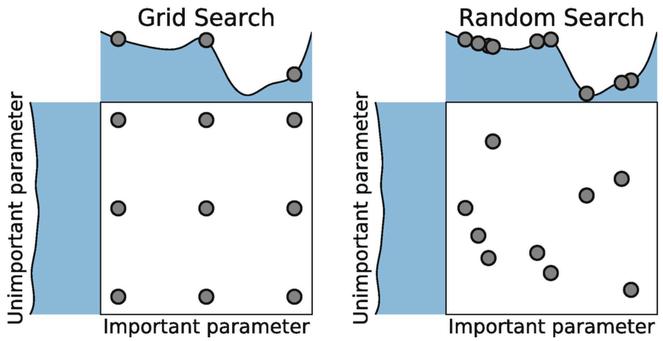
Training multiple models at scale on sharded data

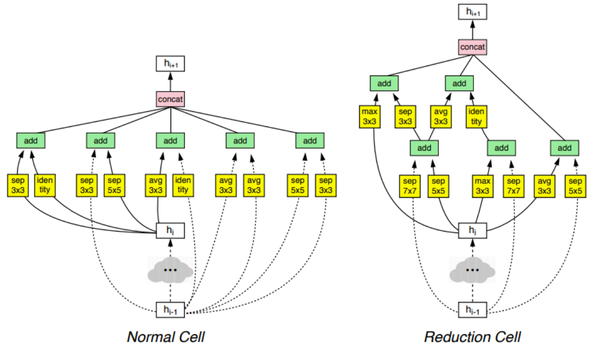
MQO for DL Model Selection

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

MQO for DL Model Selection

Workload: Hyper-parameter tuning and neural architecture exploration yield numerous (even 100s) of DL training configs



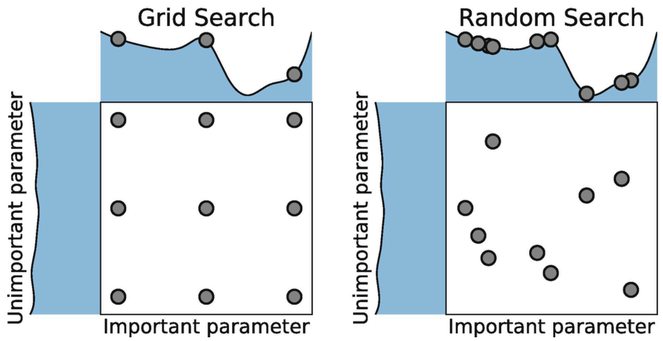


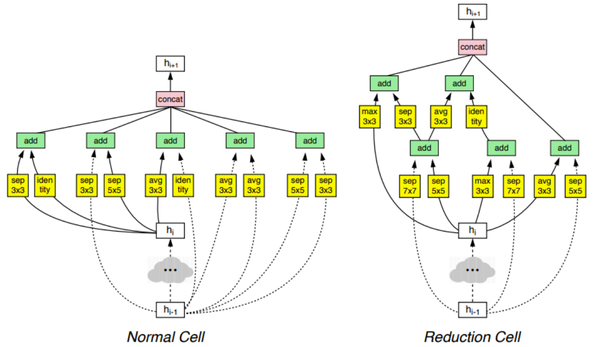
HyperBand, AutoKeras, ASHA, PBT, HyperOpt, NAS, etc.

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

MQO for DL Model Selection

Workload: Hyper-parameter tuning and neural architecture exploration yield numerous (even 100s) of DL training configs





HyperBand, AutoKeras, ASHA, PBT, HyperOpt, NAS, etc.

Multi-Query: Run multiple configs on large dataset in one go

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

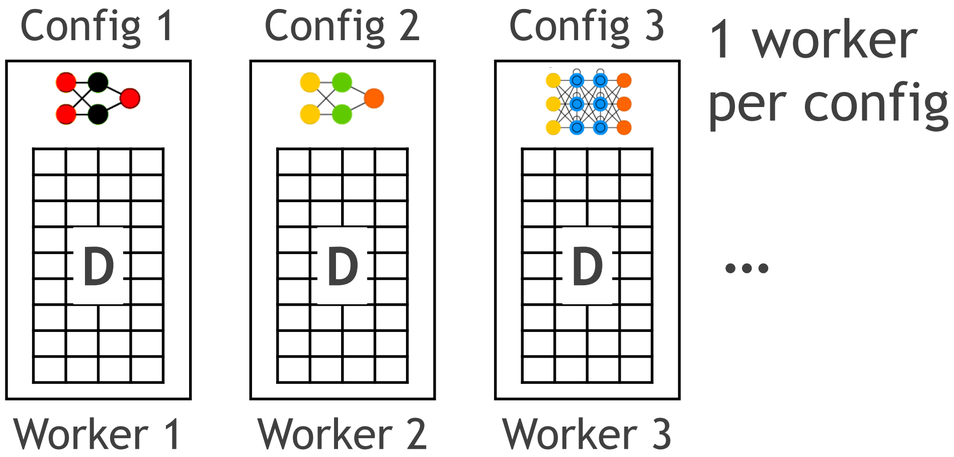
Positioning Cerebro's Technique vs Prior Art

We devise a novel execution strategy called **Model Hopper Parallelism (MOP)**

Positioning Cerebro's Technique vs Prior Art

We devise a novel execution strategy called **Model Hopper Parallelism (MOP)**

Task Parallelism:



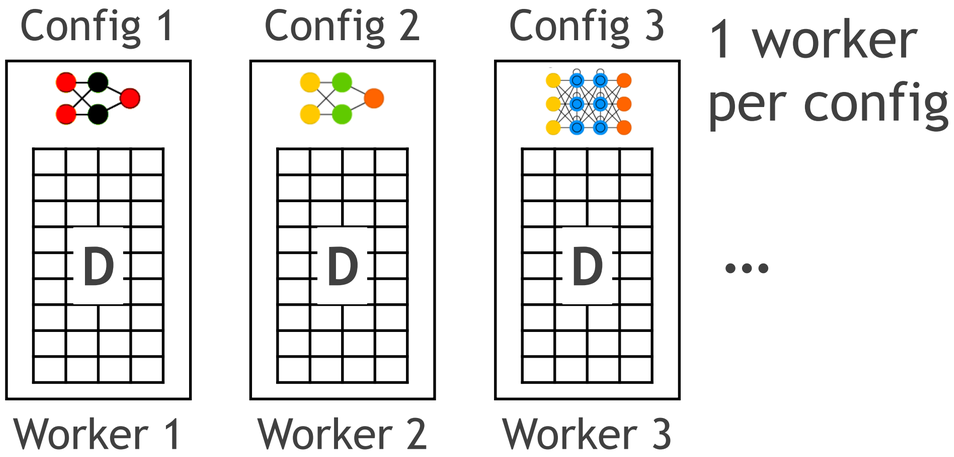
1 worker per config

...

Positioning Cerebro's Technique vs Prior Art

We devise a novel execution strategy called **Model Hopper Parallelism (MOP)**

Task Parallelism:

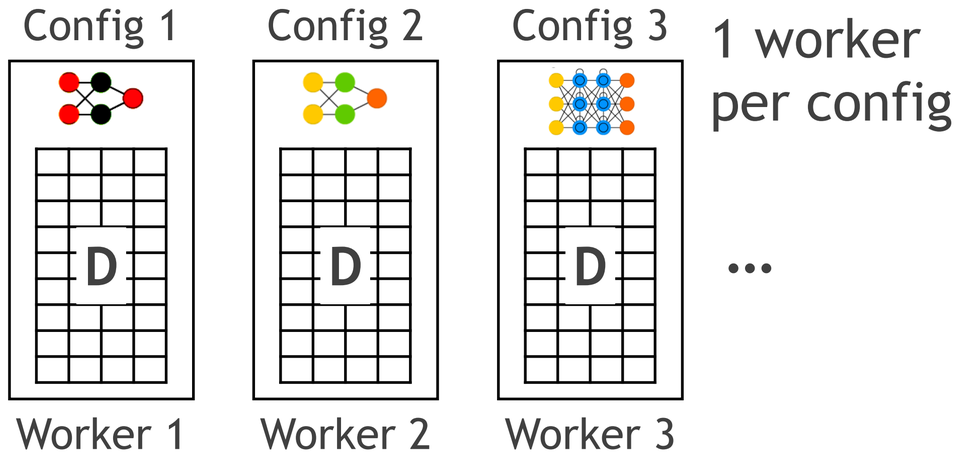


+ High throughput model selection
+ Best accuracy from Sequential SGD
— Low data scalability; wastes space (copy) or network (remote read)

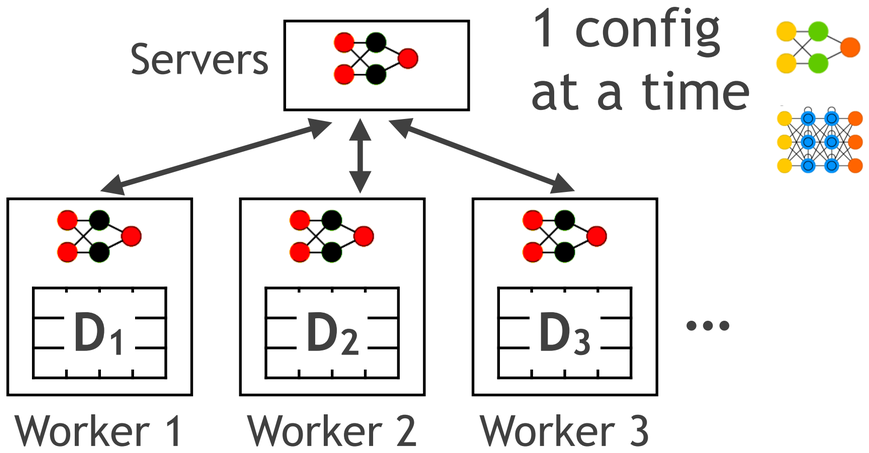
Positioning Cerebro's Technique vs Prior Art

We devise a novel execution strategy called **Model Hopper Parallelism (MOP)**

Task Parallelism:



Data Parallelism:

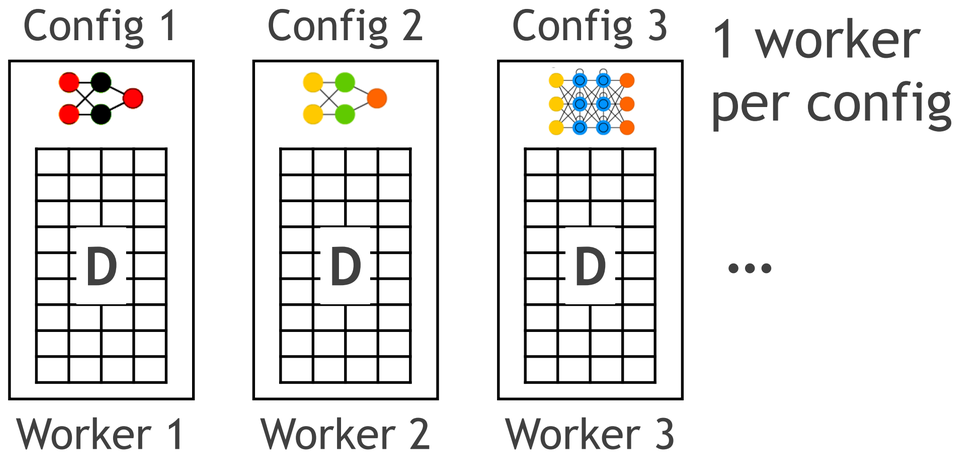


+ High throughput model selection
+ Best accuracy from Sequential SGD
— Low data scalability; wastes space (copy) or network (remote read)

Positioning Cerebro's Technique vs Prior Art

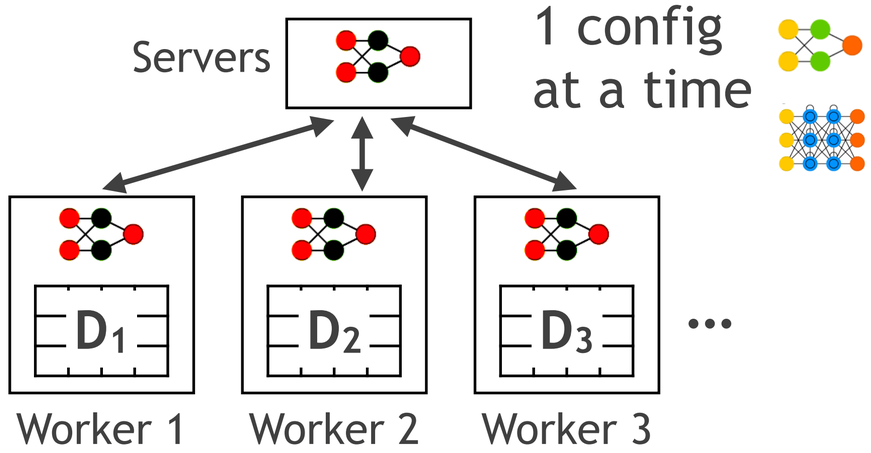
We devise a novel execution strategy called **Model Hopper Parallelism (MOP)**

Task Parallelism:



+ High throughput model selection
+ Best accuracy from Sequential SGD
— Low data scalability; wastes space (copy) or network (remote read)

Data Parallelism:

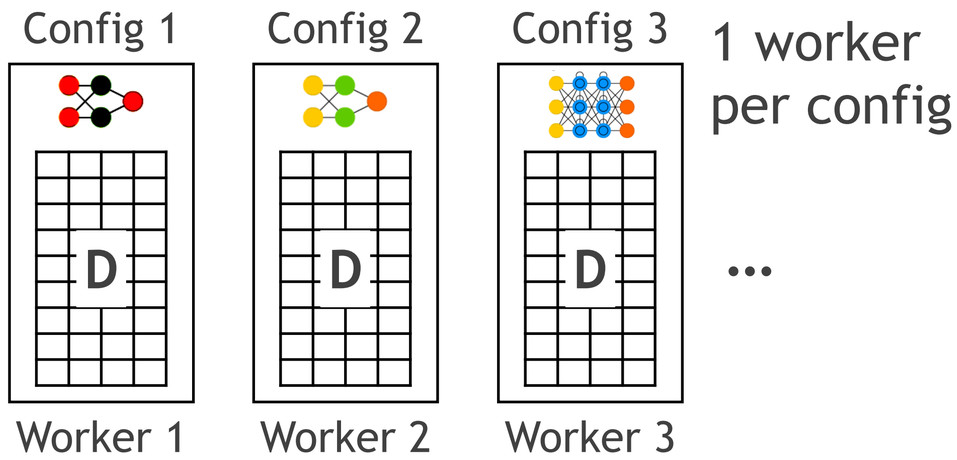


+ High data scalability via sharding
— Shard-level SGD does not converge; mini-batch level is too slow
— Very low throughput overall

Positioning Cerebro's Technique vs Prior Art

We devise a novel execution strategy called **Model Hopper Parallelism (MOP)**

Task Parallelism:



1 worker per config

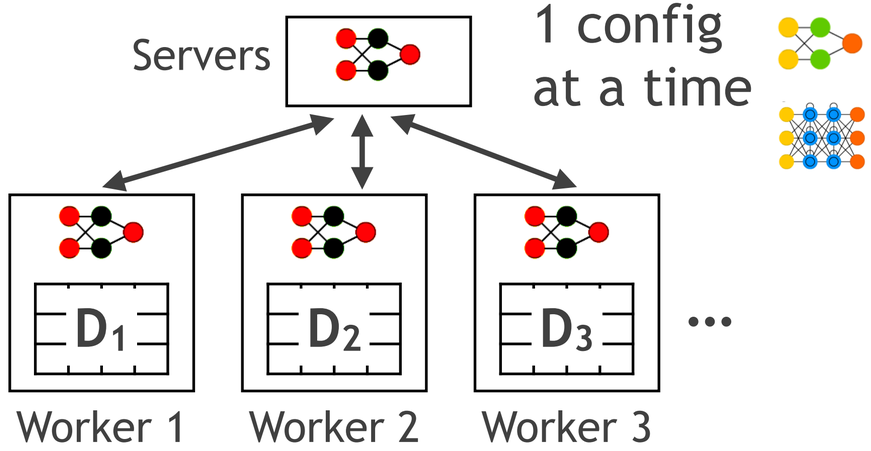
...

+ High throughput model selection
+ Best accuracy from Sequential SGD
— Low data scalability; wastes space (copy) or network (remote read)

MOP:

New hybrid of data and task parallelism
Best of both worlds!

Data Parallelism:



...

+ High data scalability via sharding
— Shard-level SGD does not converge; mini-batch level is too slow
— Very low throughput overall

Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

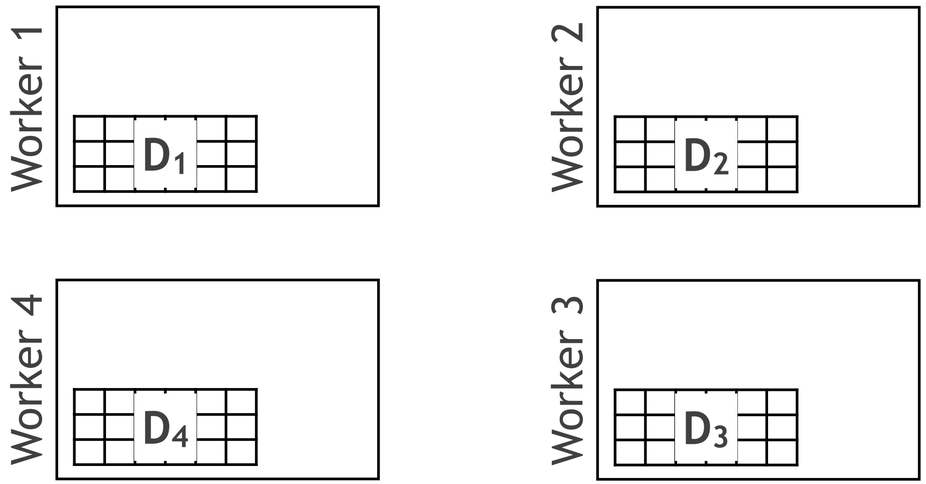
Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

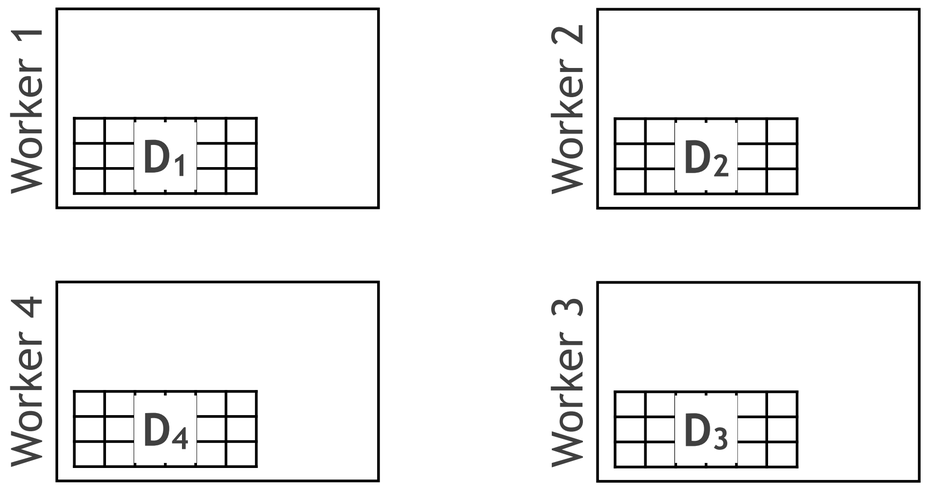
Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Epoch 1.1 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

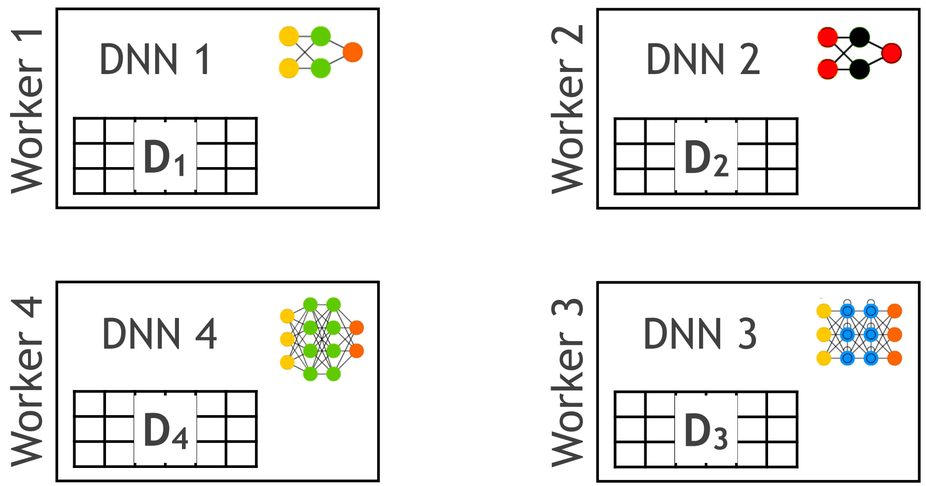
Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Epoch 1.1 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

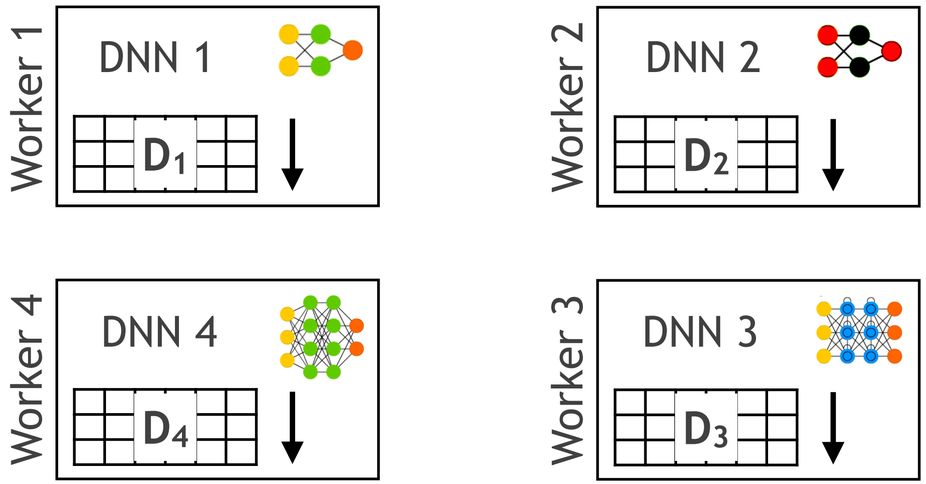
Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Epoch 1.1 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

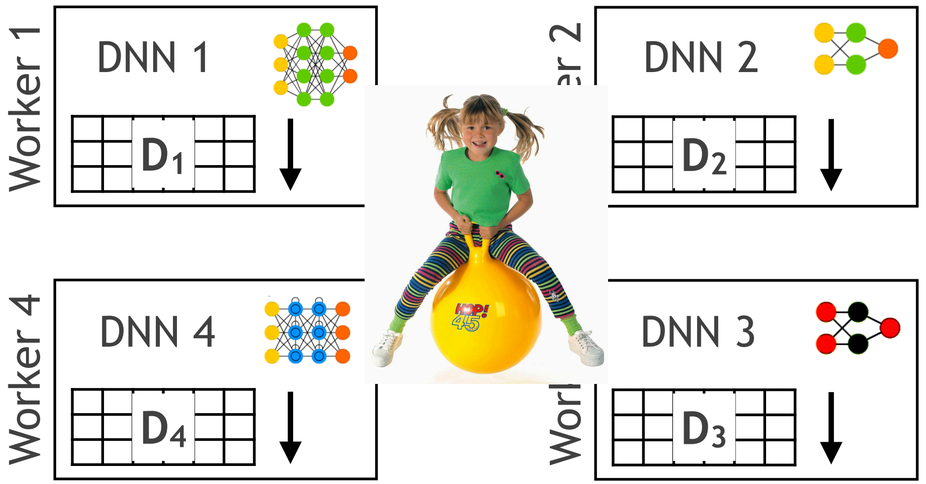
Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Epoch 1.1 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

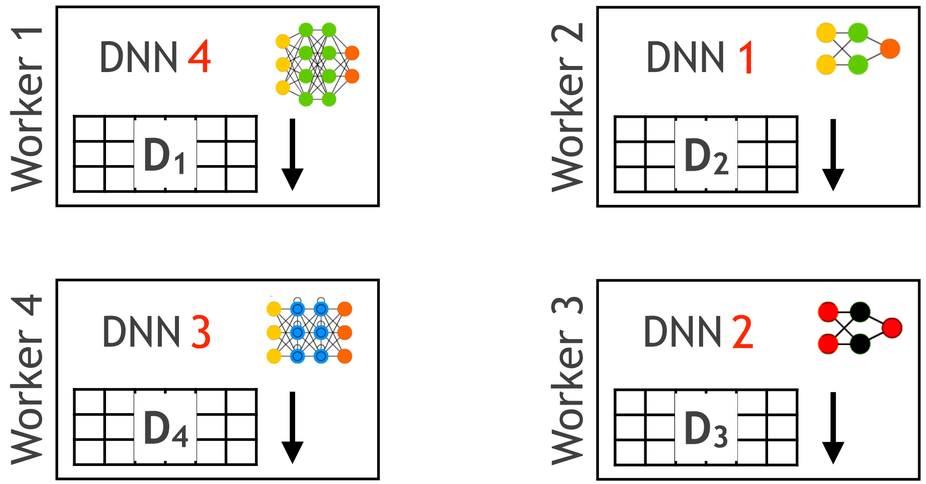
Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Epoch 1.1 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

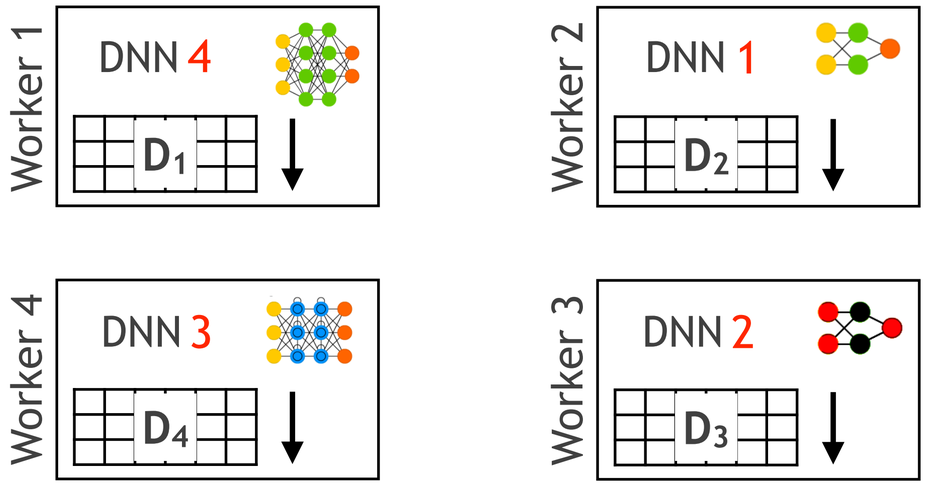
Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Epoch 1.2 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Model Hopper Parallelism (MOP)

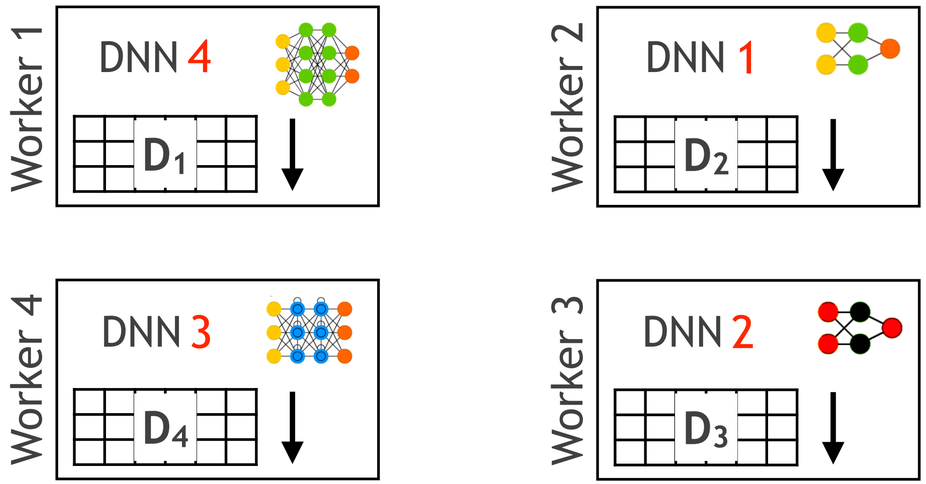
Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Shuffle and partition dataset
Run *n* DNNs on *n* workers

Each model keeps "hopping" until it sees all of D

Epoch 1.2 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

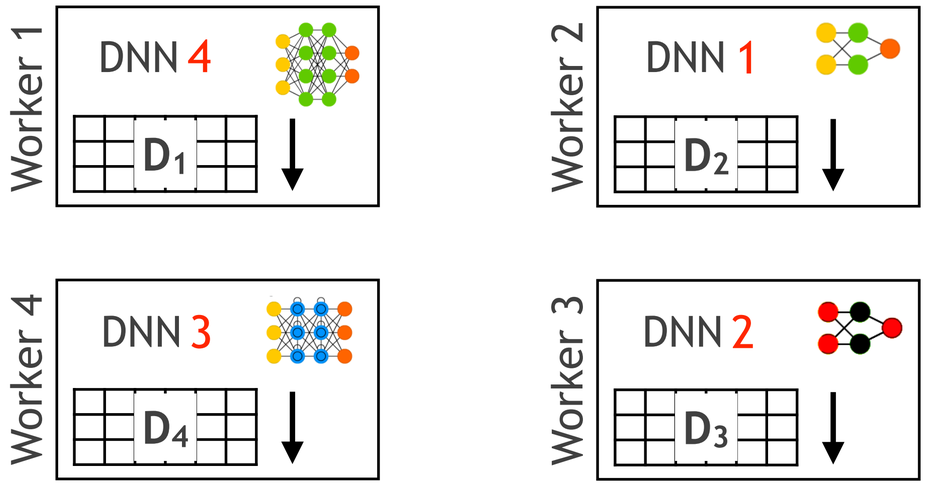
Shuffle and partition dataset
Run *n* DNNs on *n* workers

Each model keeps "hopping" until it sees all of D

Strong theoretical guarantees:

1. *Equivalence* to sequential SGD
2. Hits lower bound on comm. cost

Epoch 1.2 starts in parallel



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Model Hopper Parallelism (MOP)

Insight from Optimization Theory:

SGD is robust to *data ordering randomness*

Technical Challenges (See paper for details):

Resource-aware scheduling of evolving model configs' hops

Support data replication, fault tolerance, and elasticity

Non-disruptive integration with existing DL systems

1. *Equivalence* to sequential SGD
2. Hits lower bound on comm. cost

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

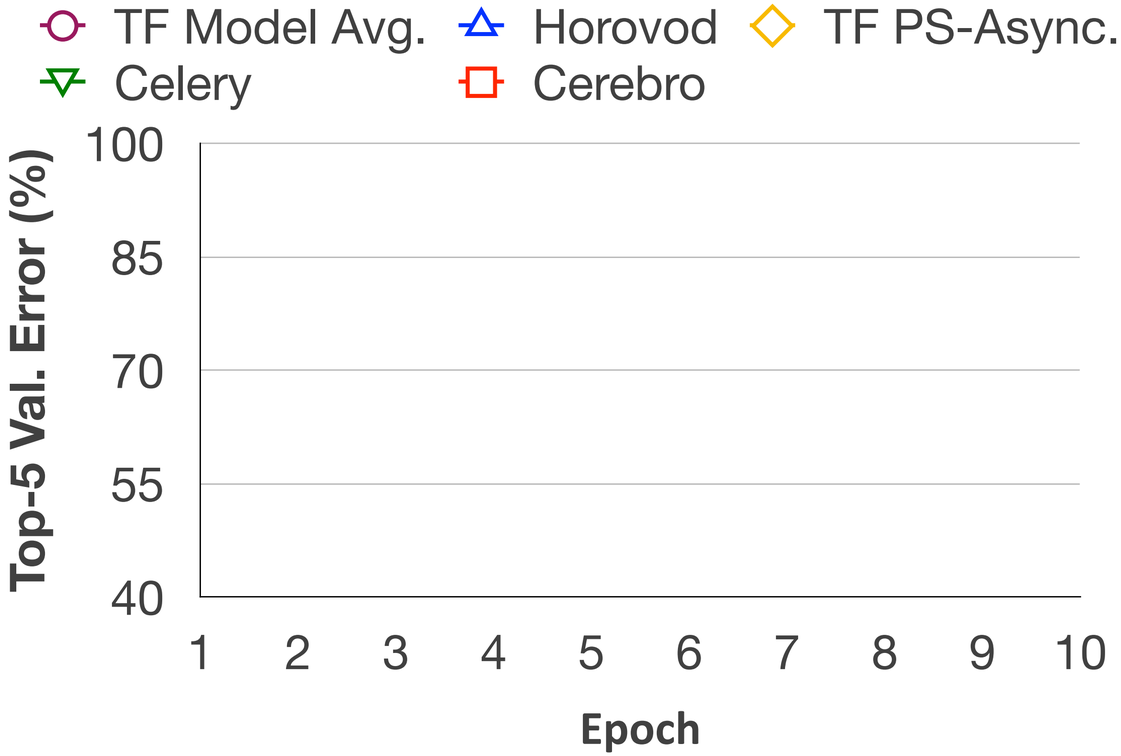
Experimental Evaluation

Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

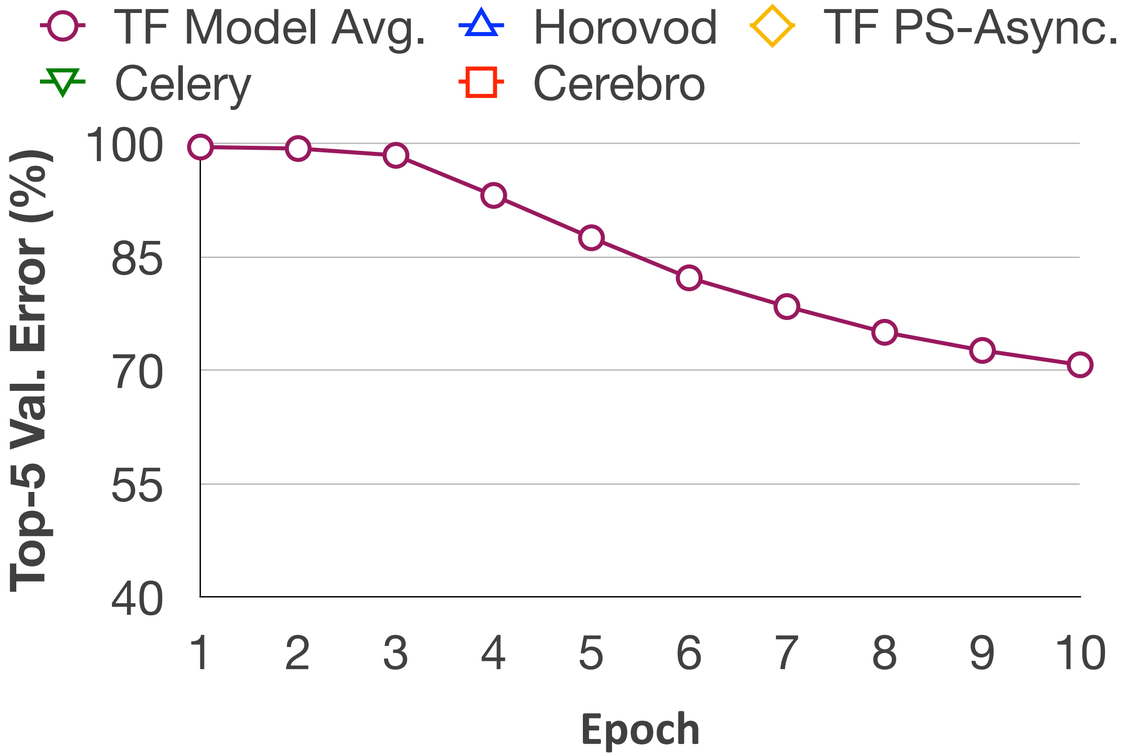
Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

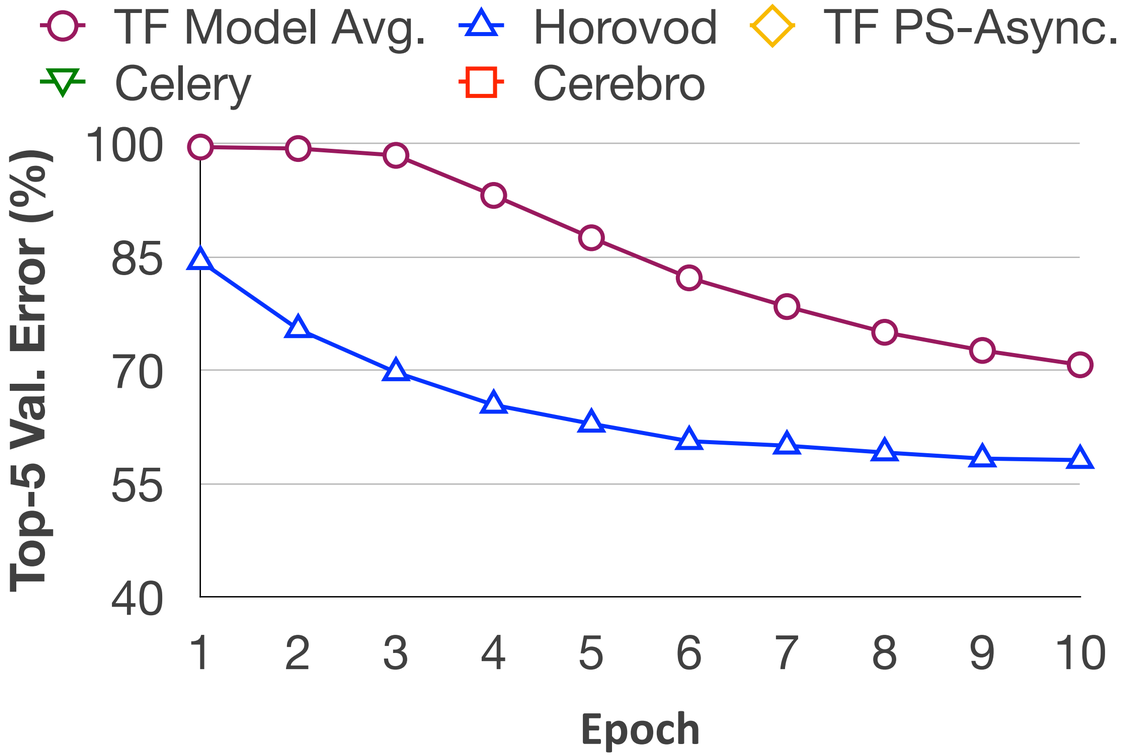
Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

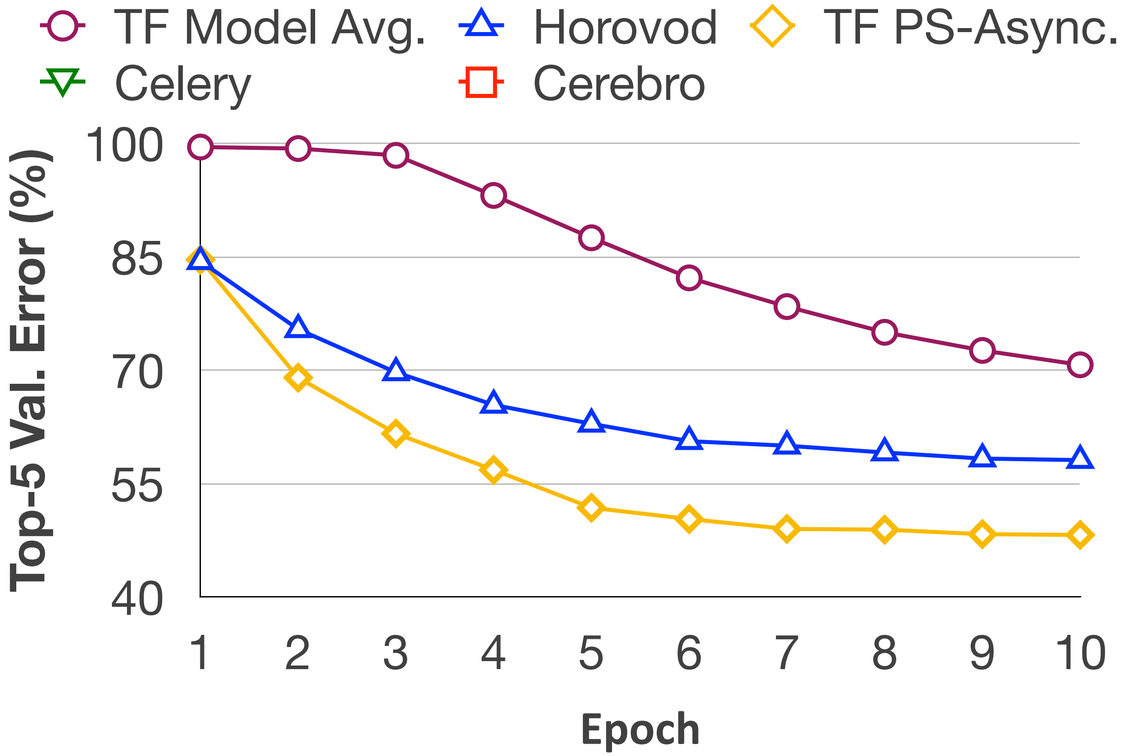
Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

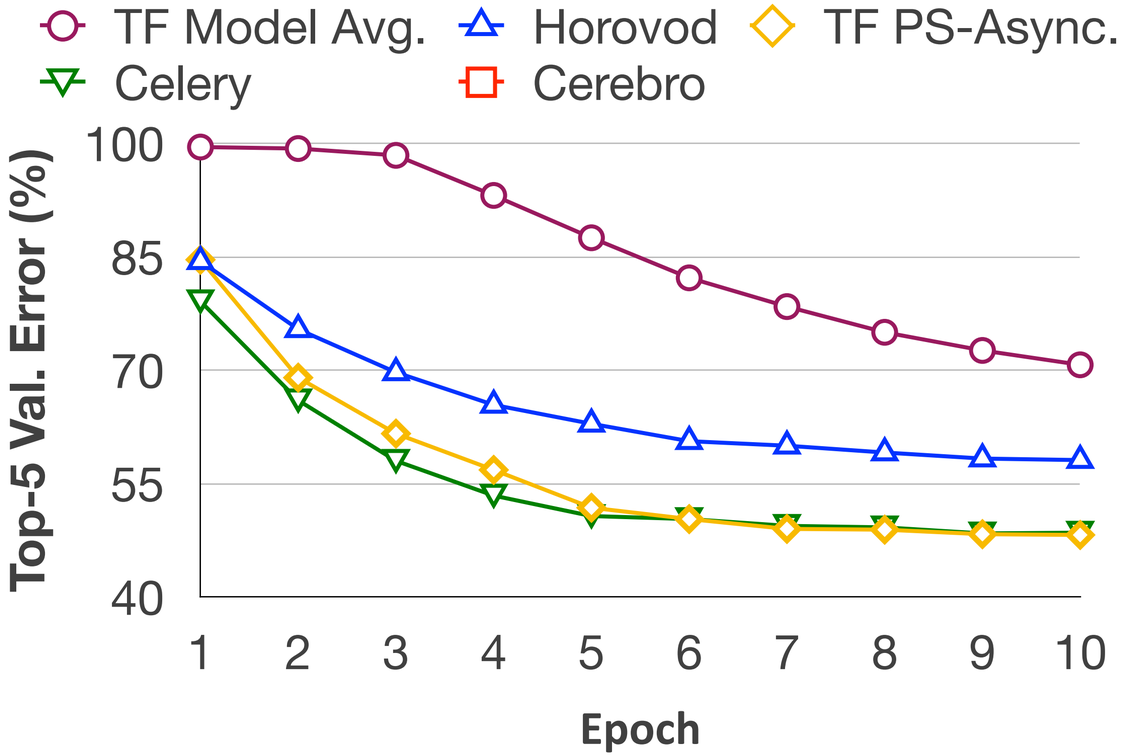
Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

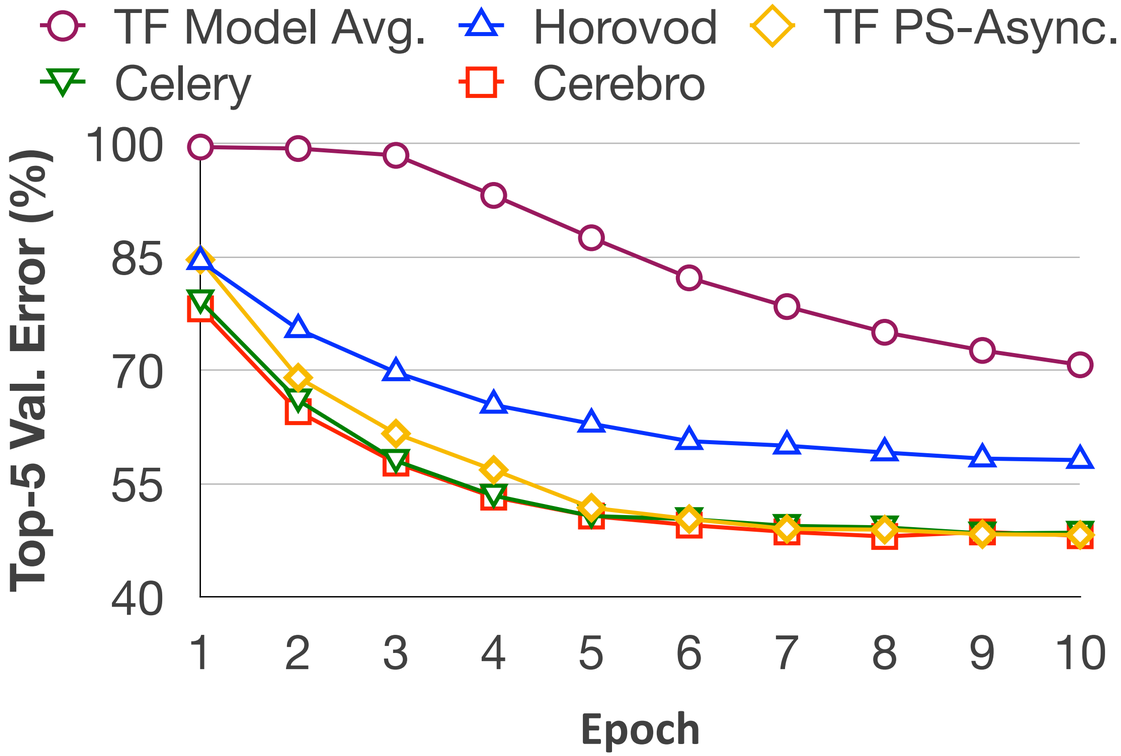
Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

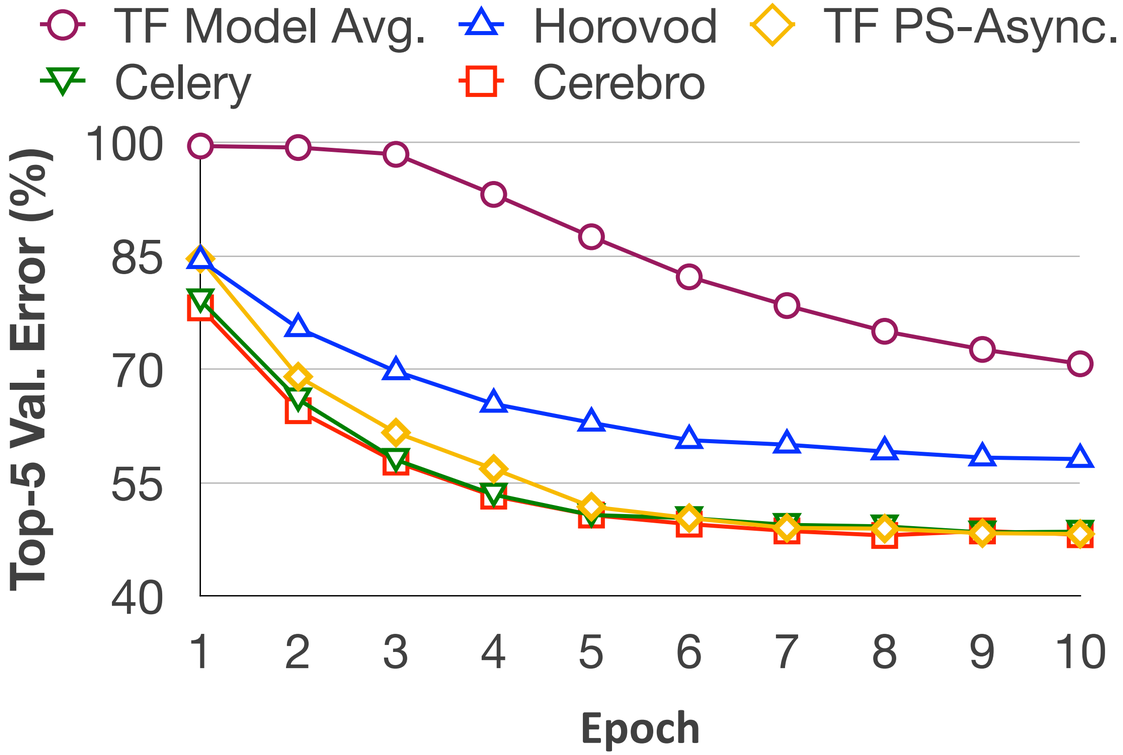
Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes



Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes

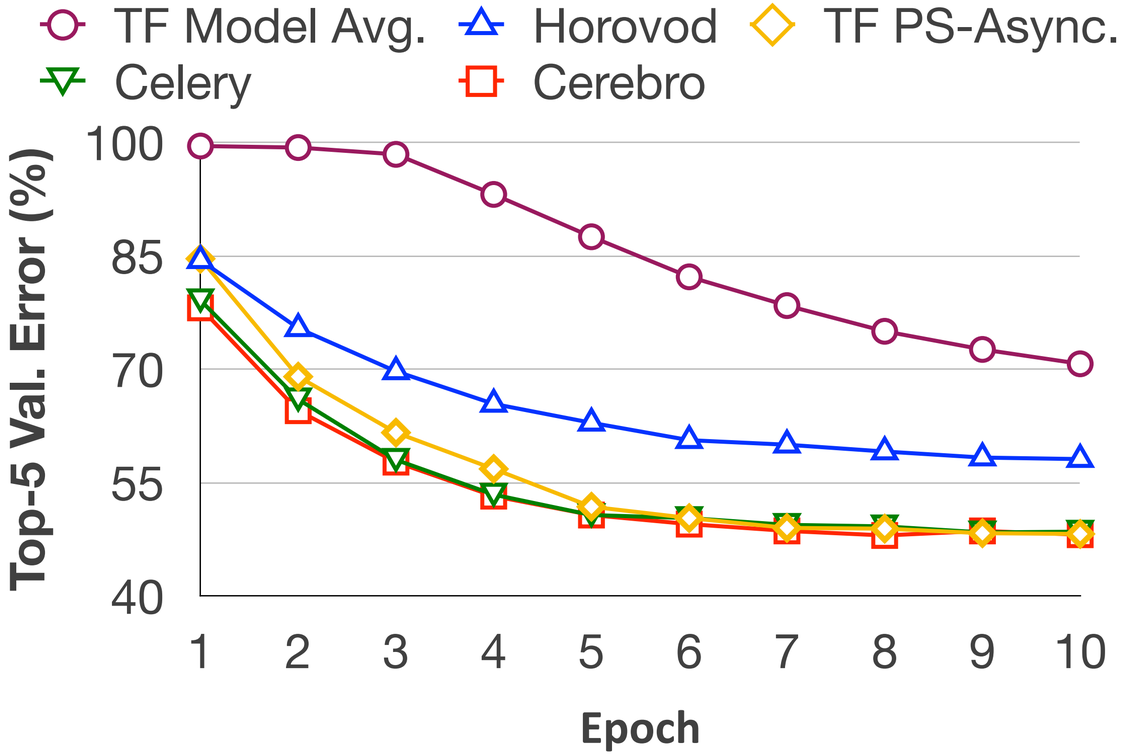


Method	Runtime (hrs)	Memory
TF PS-Async.	190.0 (!)	200 GB
Horovod	54.2	200 GB
TF Model Averaging	19.7	200 GB
Celery (Task-Parallel)	17.2-19.0*	1600 GB (!)
MOP/Cerebro	17.7	200 GB

Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Experimental Evaluation

Setup: ImageNet; 16 CNN configurations; TensorFlow; 8 GPU nodes

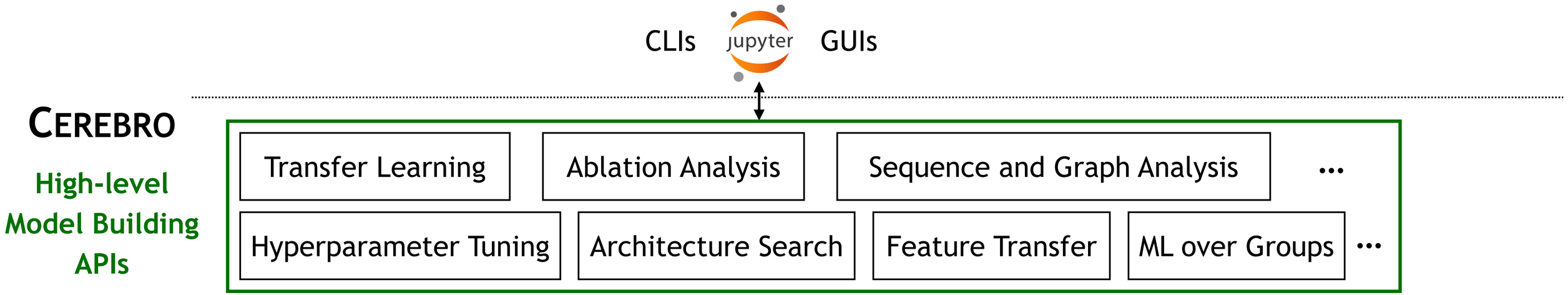


Method	Runtime (hrs)	Memory
TF PS-Async.	190.0 (!)	200 GB
Horovod	54.2	200 GB
TF Model Averaging	19.7	200 GB
Celery (Task-Parallel)	17.2-19.0*	1600 GB (!)
MOP/Cerebro	17.7	200 GB

Cerebro offers overall best combination of resource efficiency and accuracy

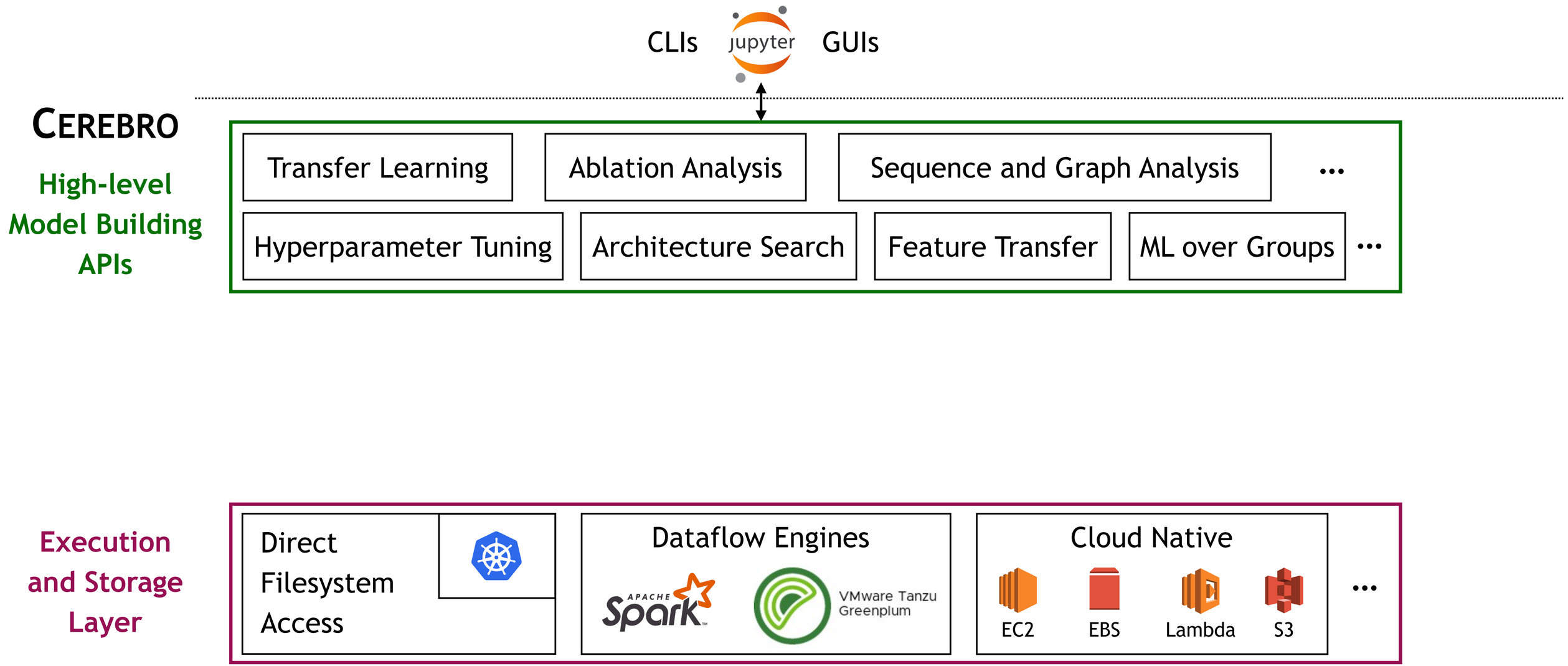
Cerebro: Efficient and Reproducible Model Selection on Deep Learning Systems. ***SIGMOD DEEM'19***
Cerebro: A Data System for Optimized Deep Learning Model Selection. ***VLDB'20***

Full Vision of the Cerebro Platform



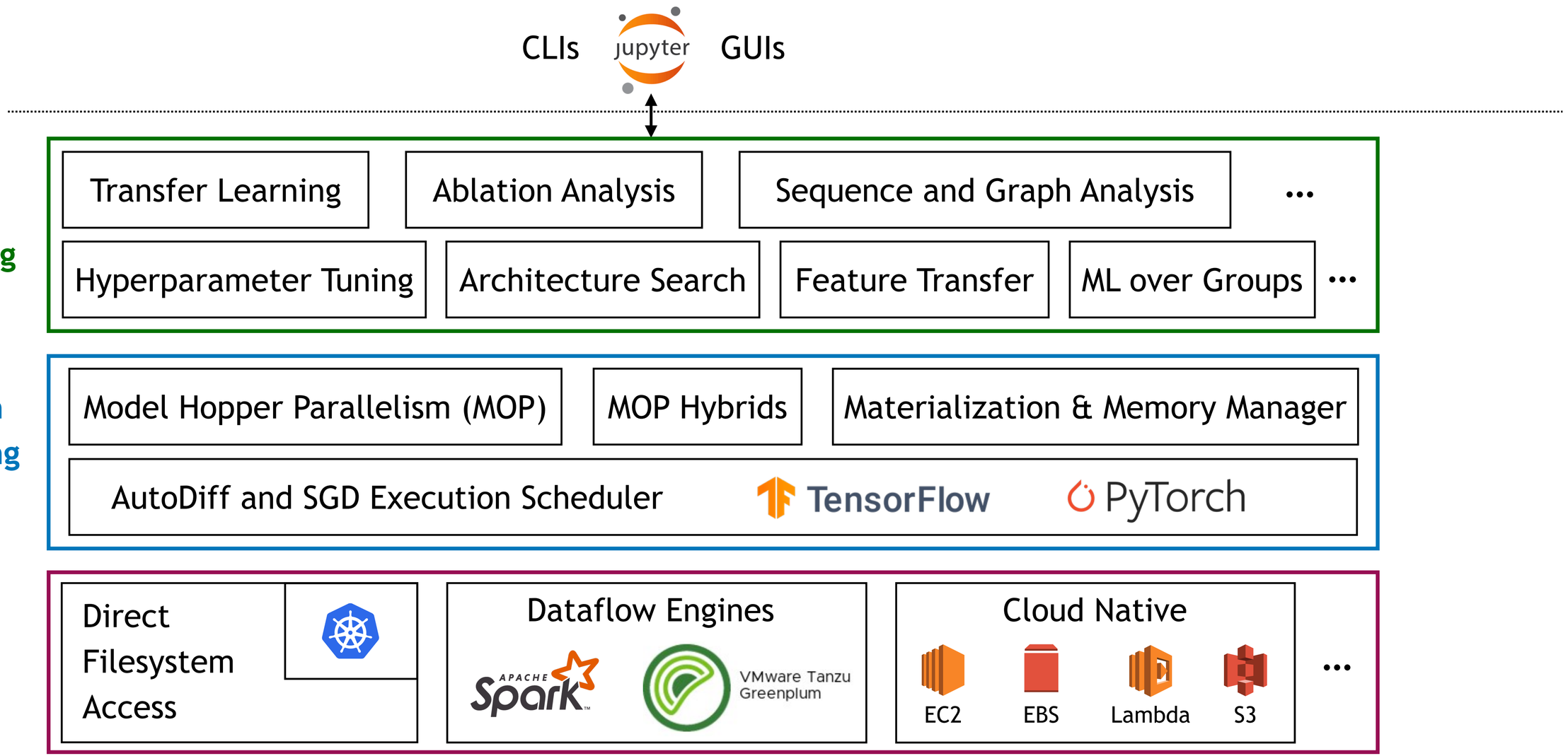
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



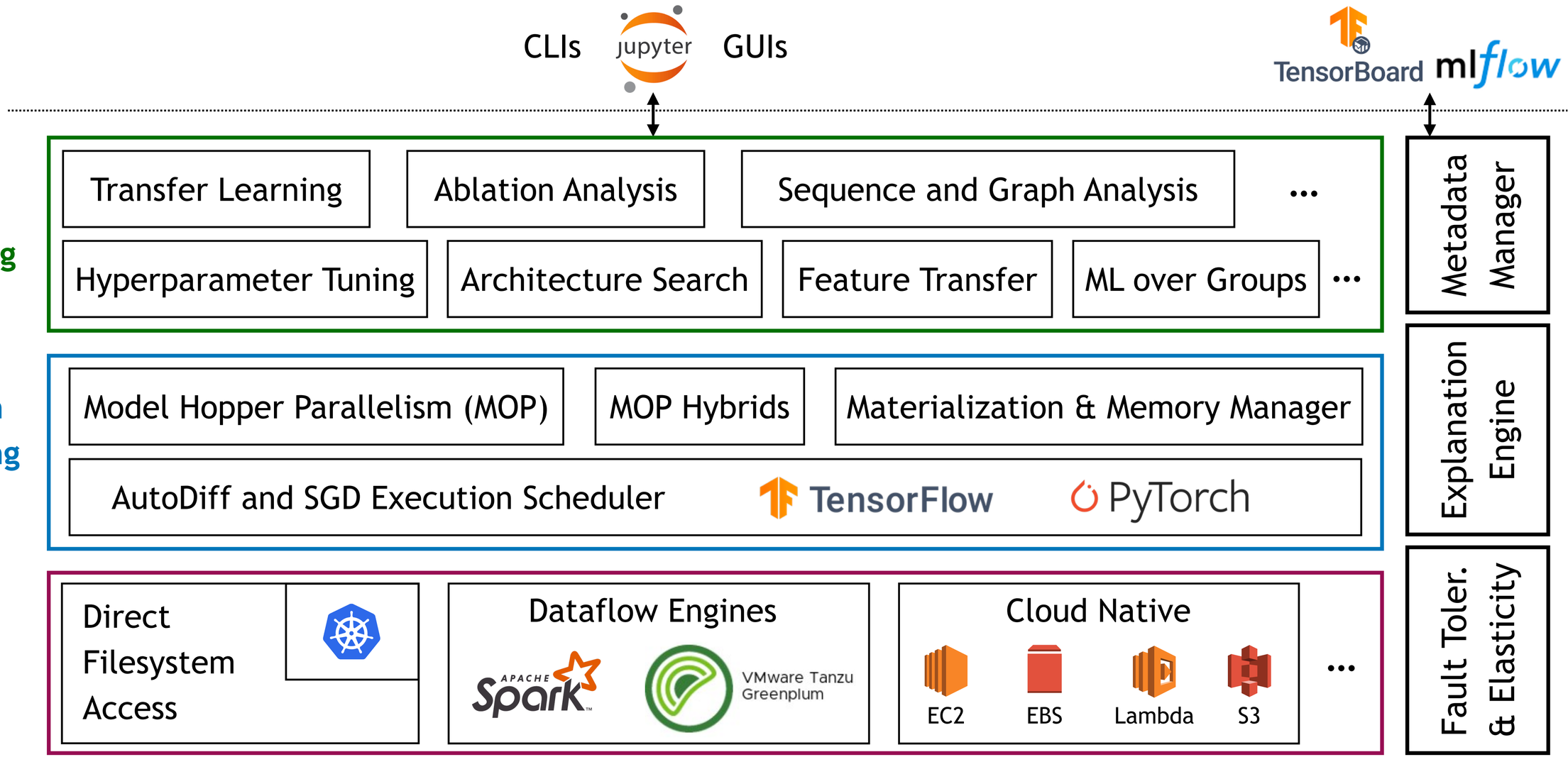
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



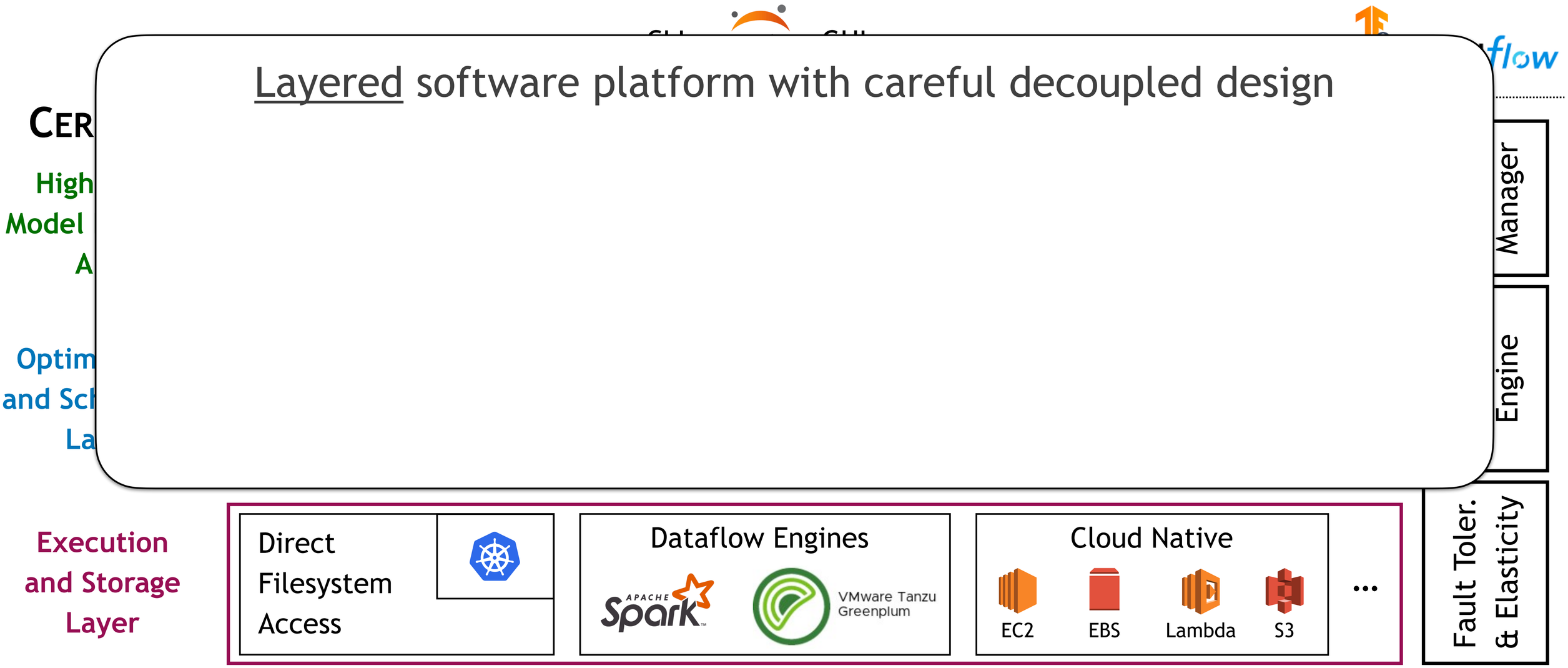
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



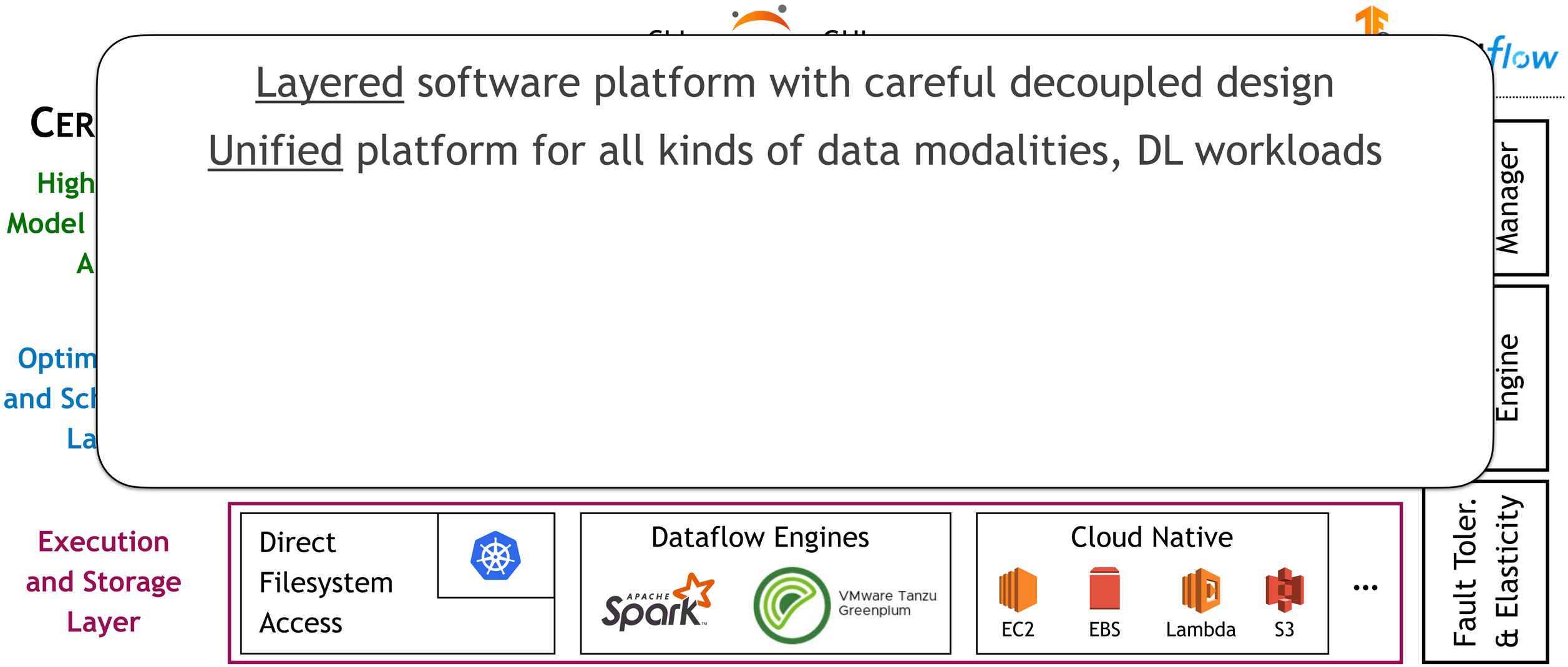
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



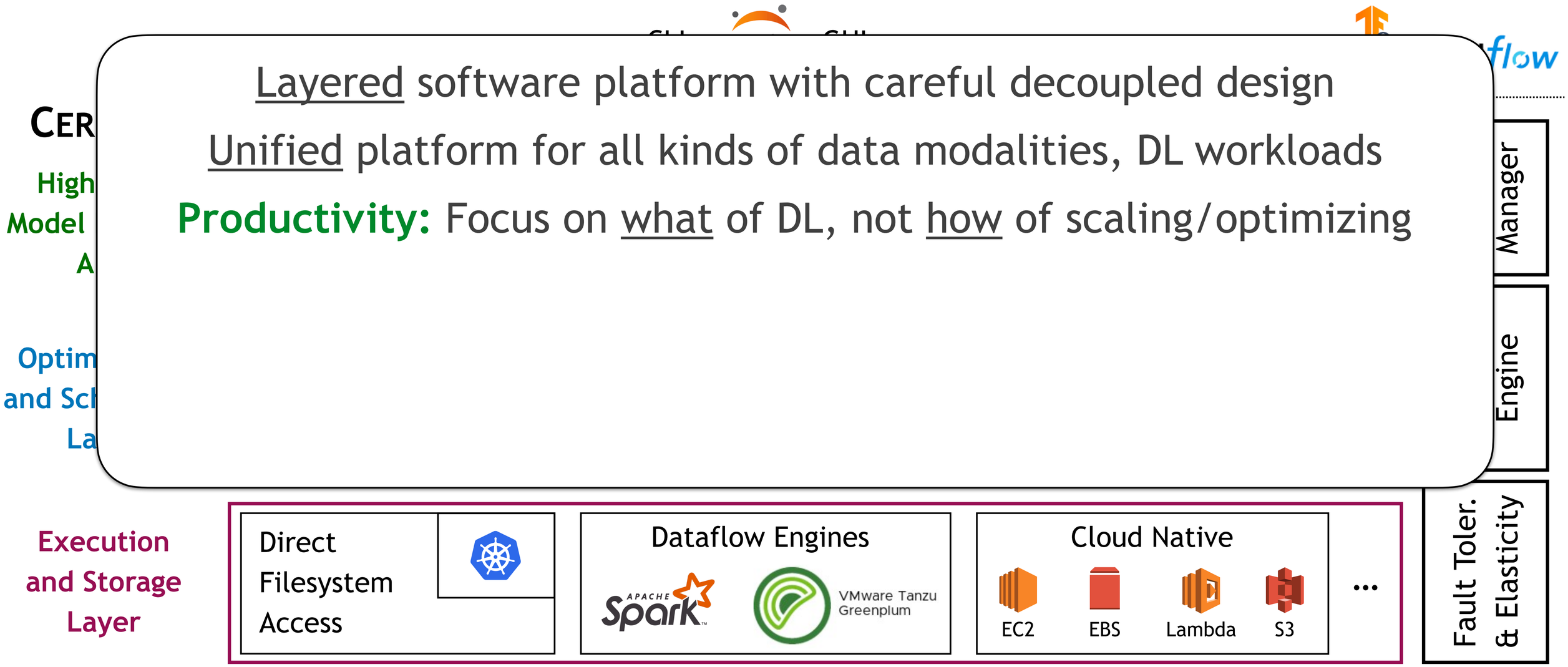
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



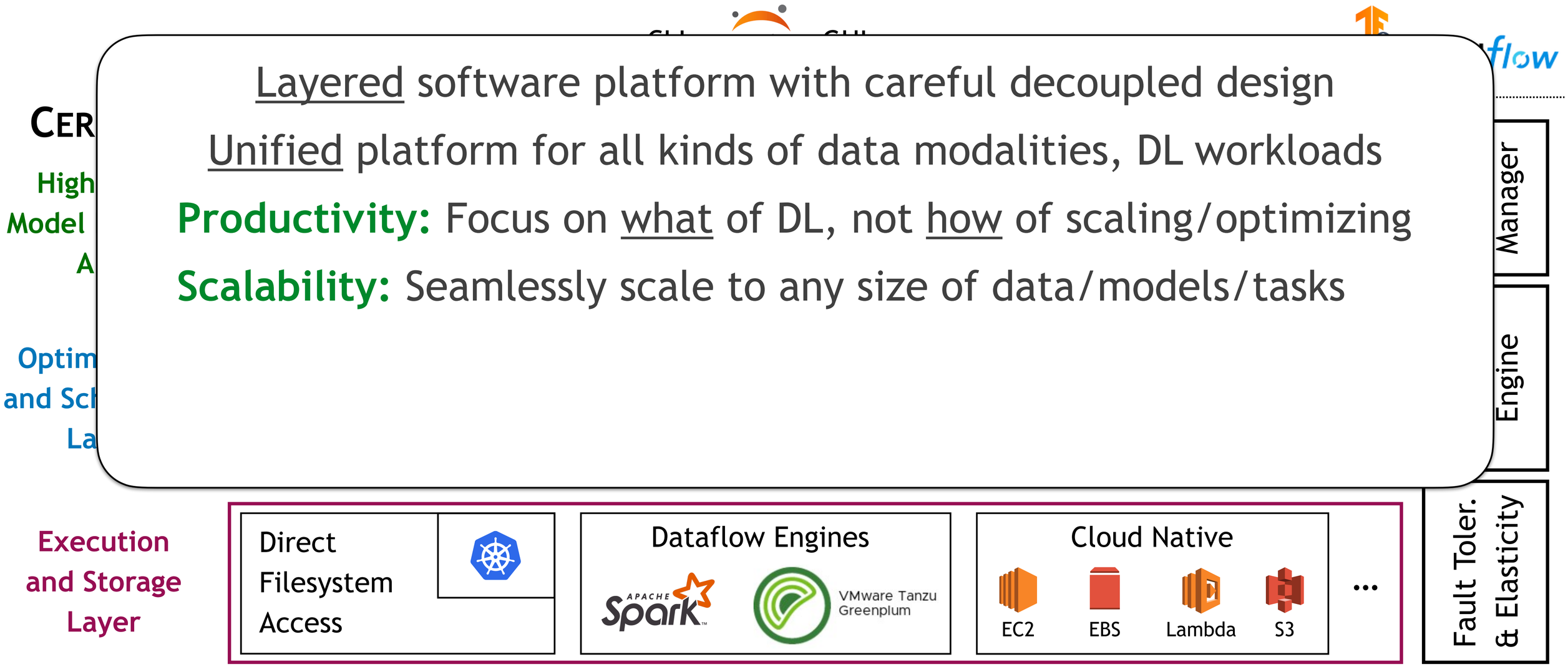
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



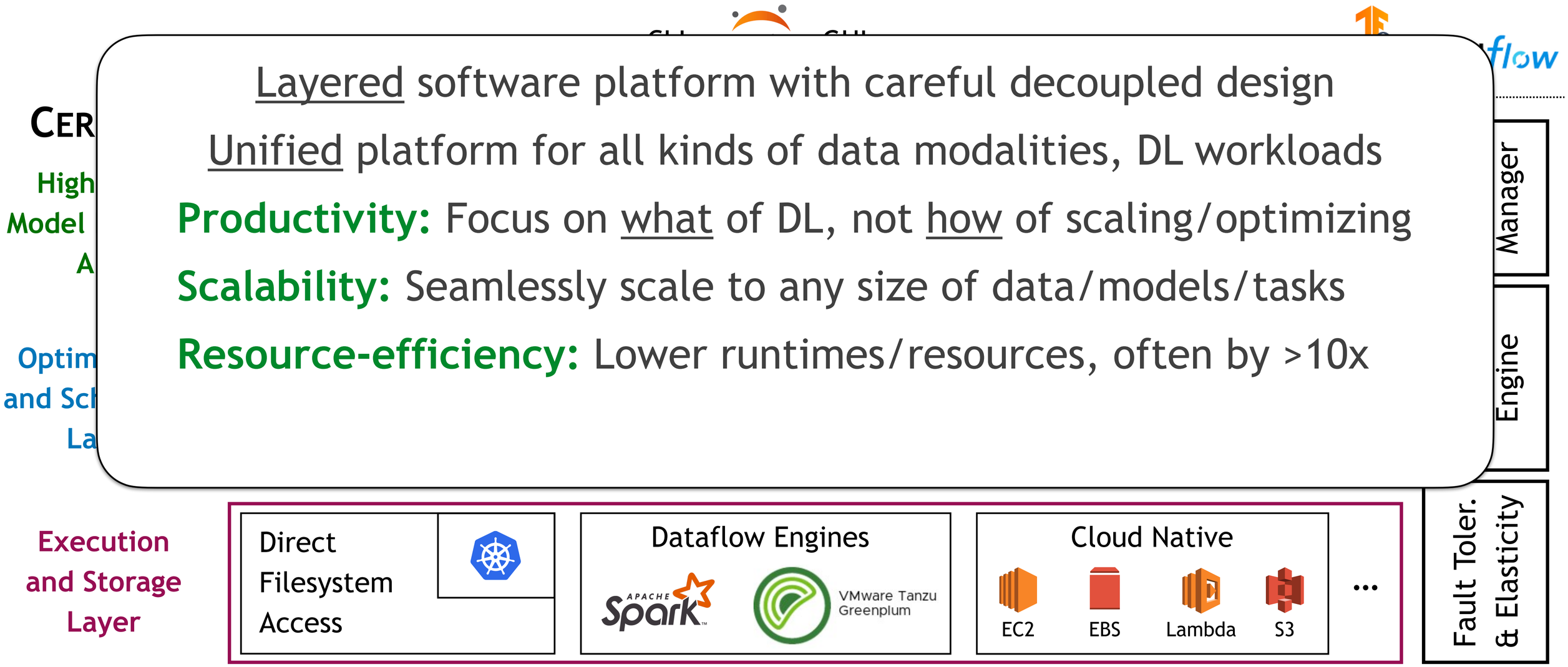
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



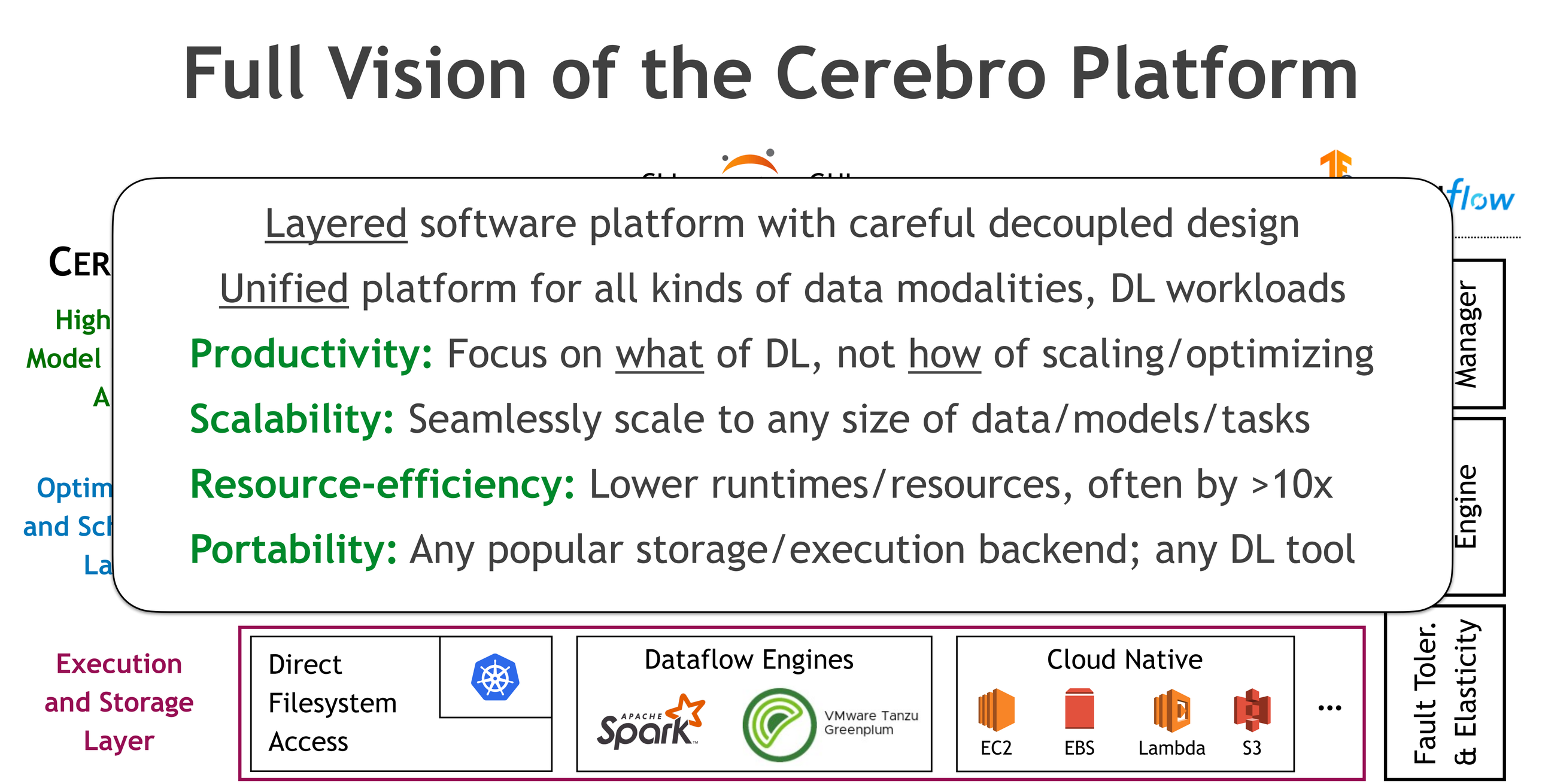
Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Full Vision of the Cerebro Platform



Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Cerebro: Early Impact and Trajectory

Used on TBs by UCSD Public Health scientists; Physics & Materials Eng. next

Shipped with Apache MADlib by Pivotal; VMware collaboration

Cerebro+Spark released; all open sourced under Apache License v2.0

Coming soon: More workloads + optimizations + domain science use cases

Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Cerebro: Early Impact and Trajectory

Used on TBs by UCSD Public Health scientists; Physics & Materials Eng. next

Shipped with Apache MADlib by Pivotal; VMware collaboration

Cerebro+Spark released; all open sourced under Apache License v2.0

Coming soon: More workloads + optimizations + domain science use cases

Cerebro: A Layered Data Platform for Scalable Deep Learning. ***CIDR'21***

Cerebro: Open Source Release and Demo

Table of Contents

Quick search

Go

Cerebro Documentation

`Cerebro` is a data system for optimized deep learning model selection. It uses a novel parallel execution strategy called **Model Hopper Parallelism (MOP)** to execute end-to-end deep learning model selection workloads in a more resource-efficient manner.

Installation

The best way to install the `Cerebro` is via pip.

```
pip install -U cerebro-dl
```

Alternatively, you can git clone and run the provided Makefile script to install the master branch

```
git clone https://github.com/ADALabUCSD/cerebro-system.git && cd cere
```

Note:

You MUST be running on **Python >= 3.6** with **Tensorflow >= 2.2** and **Apache Spark >= 2.4**

Table of Contents

Links on project webpage: https://adalabucsd.github.io/cerebro.html

Cerebro: Open Source Release and Demo

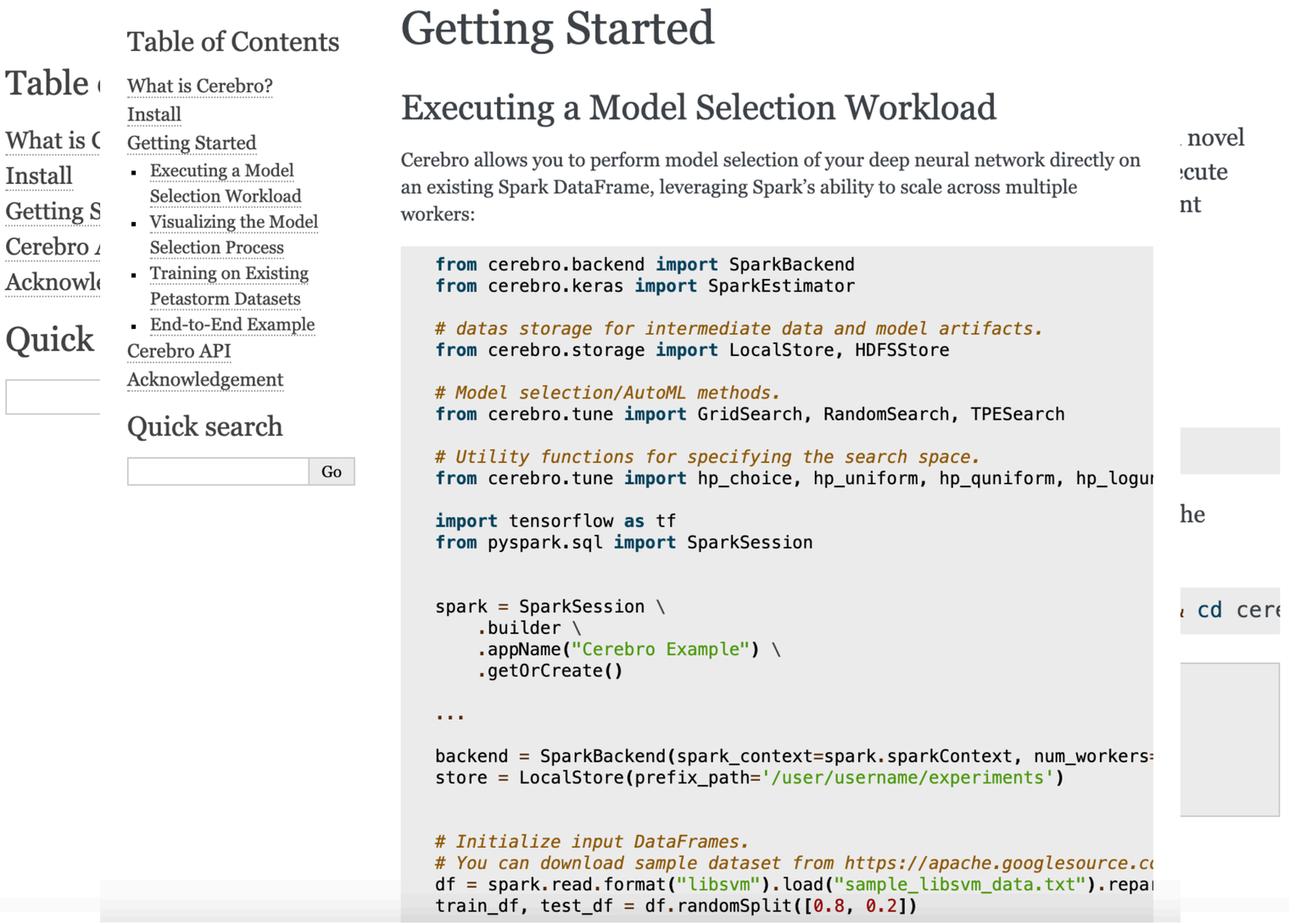


Table of Contents

What is Cerebro?
Install
Getting Started
- Executing a Model Selection Workload
- Visualizing the Model Selection Process
- Training on Existing Petastorm Datasets
- End-to-End Example

Cerebro API
Acknowledgement

Quick search

Go

Getting Started

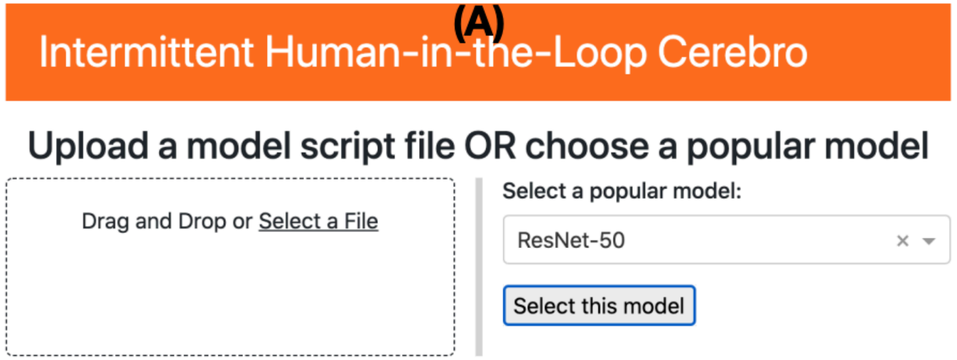
Executing a Model Selection Workload

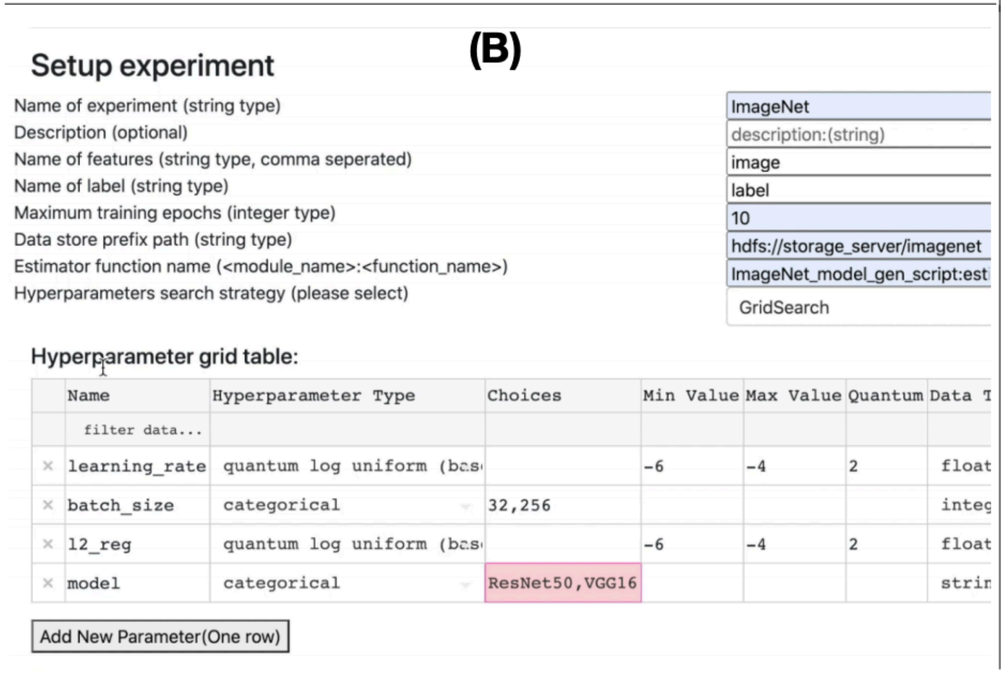
Cerebro allows you to perform model selection of your deep neural network directly on an existing Spark DataFrame, leveraging Spark's ability to scale across multiple workers:

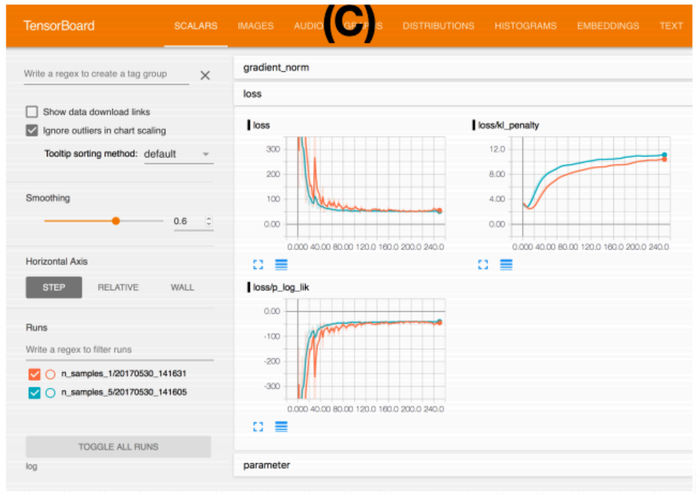
```
from cerebro.backend import SparkBackend
from cerebro.keras import SparkEstimator

# datas storage for intermediate data and model artifacts.
from cerebro.storage import LocalStore, HDFSStore

# Model selection/AutoML methods.
from cerebro.tune import GridSearch, RandomSearch, TPESearch

# Utility functions for specifying the search space.
from cerebro.tune import hp_choice, hp_uniform, hp_quniform, hp_logu

import tensorflow as tf
from pyspark.sql import SparkSession


spark = SparkSession \
    .builder \
    .appName("Cerebro Example") \
    .getOrCreate()

...

backend = SparkBackend(spark_context=spark.sparkContext, num_workers=
store = LocalStore(prefix_path='/user/username/experiments')


# Initialize input DataFrames.
# You can download sample dataset from https://apache.googlesource.co
df = spark.read.format("libsvm").load("sample_libsvm_data.txt").repa
train_df, test_df = df.randomSplit([0.8, 0.2])
```

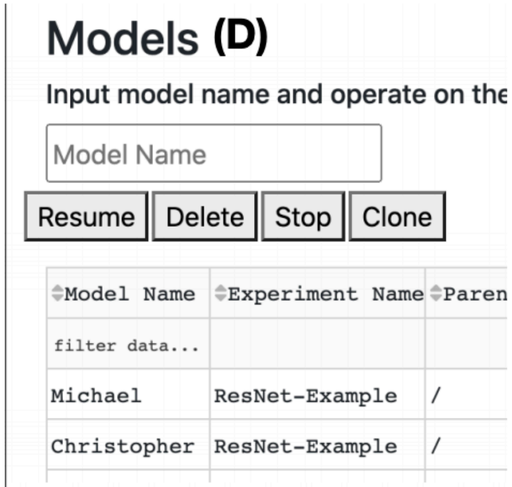
Links on project webpage: https://adalabucsd.github.io/cerebro.html

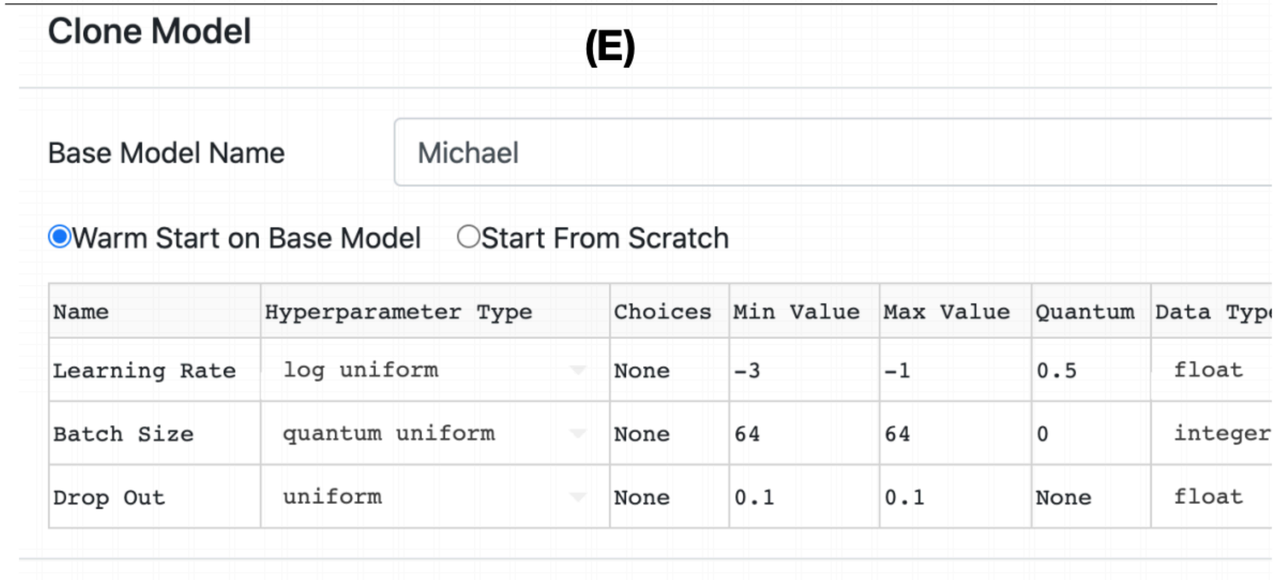
Cerebro: Open Source Release and Demo











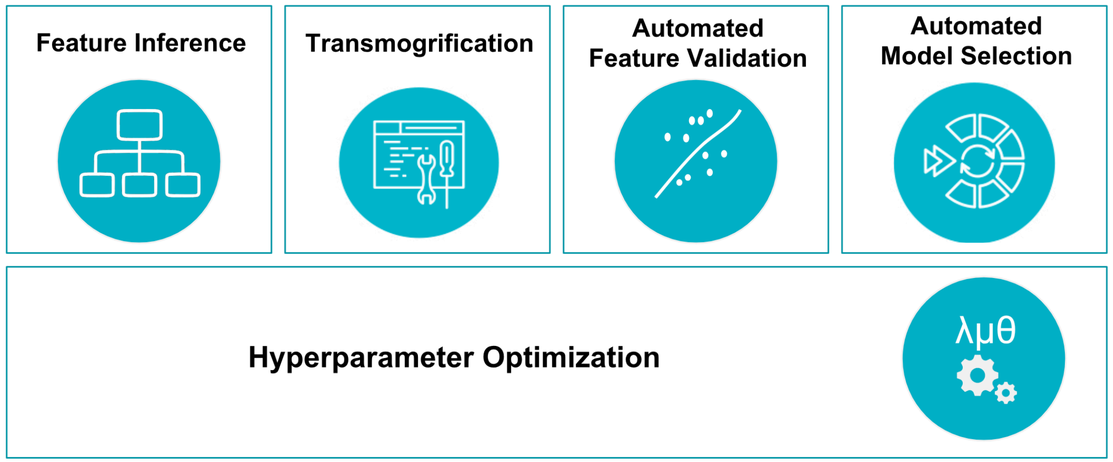
Links on project webpage: https://adalabucsd.github.io/cerebro.html

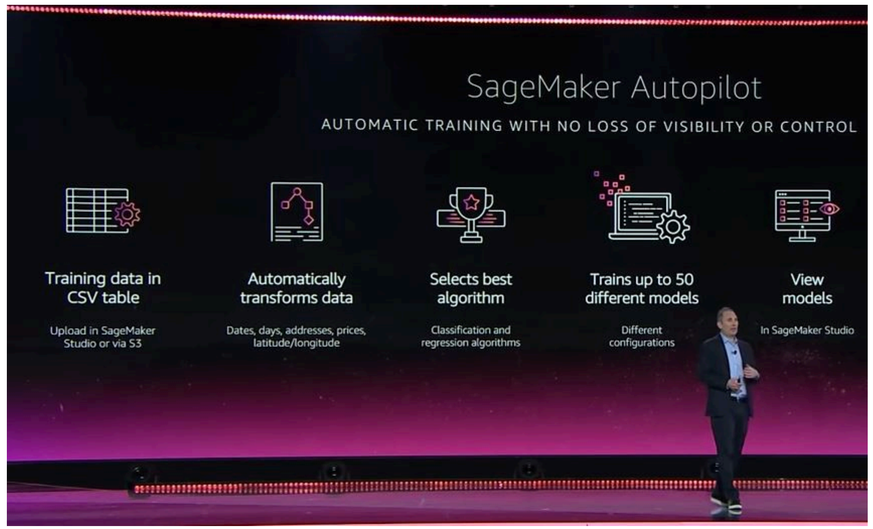
Outline

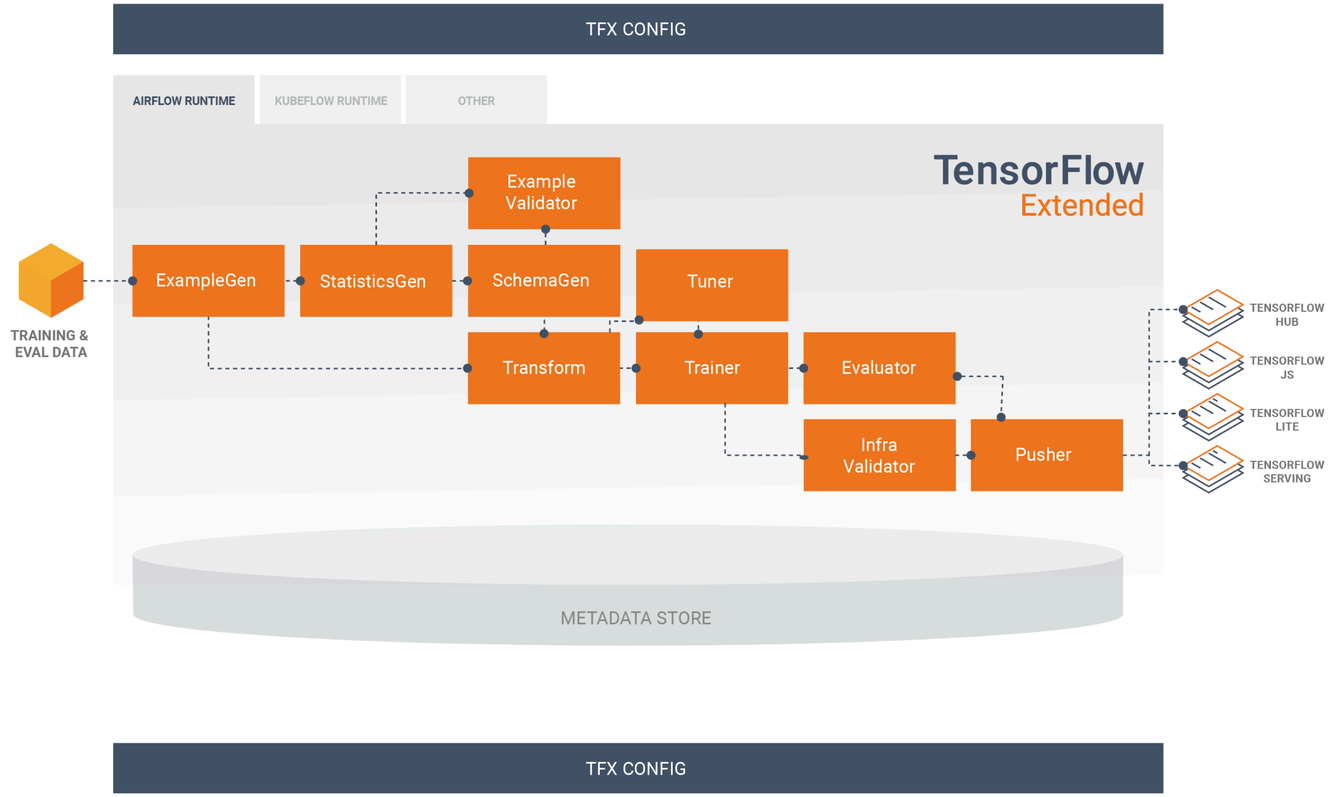
- The New DBfication of ML/AI
- Two Examples from My Research
  - Example 1: Scalable DL Systems
  - Example 2: Auto Data Prep for ML
- Accelerating the DBfication of ML/AI

Example 2: Auto Data Prep for ML

TransmogrifAI

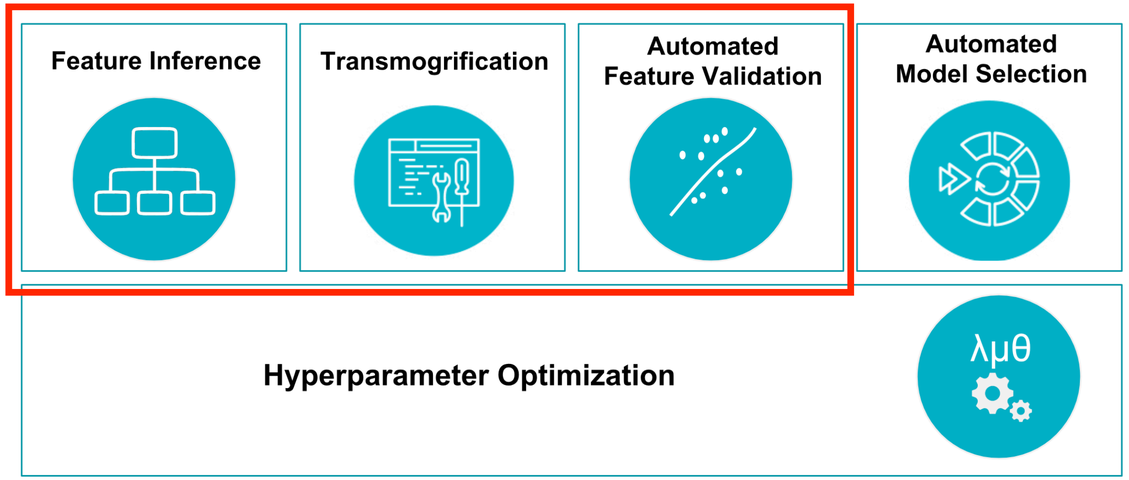


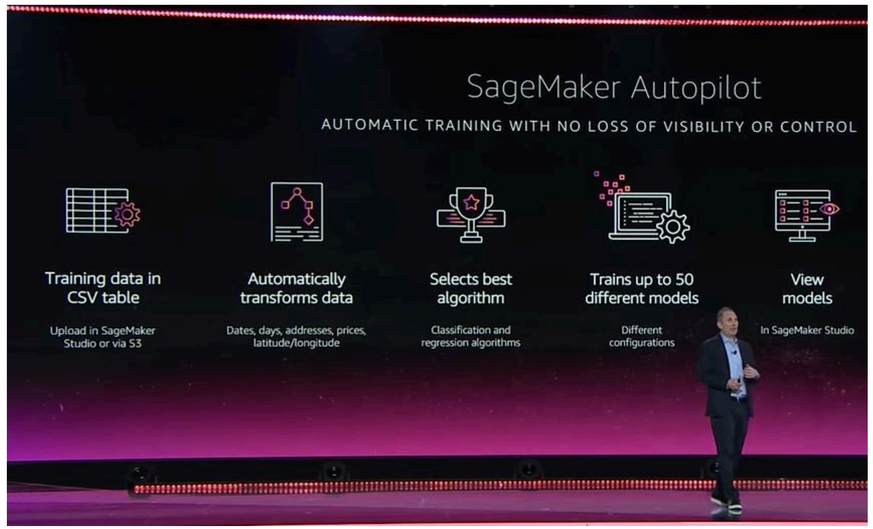


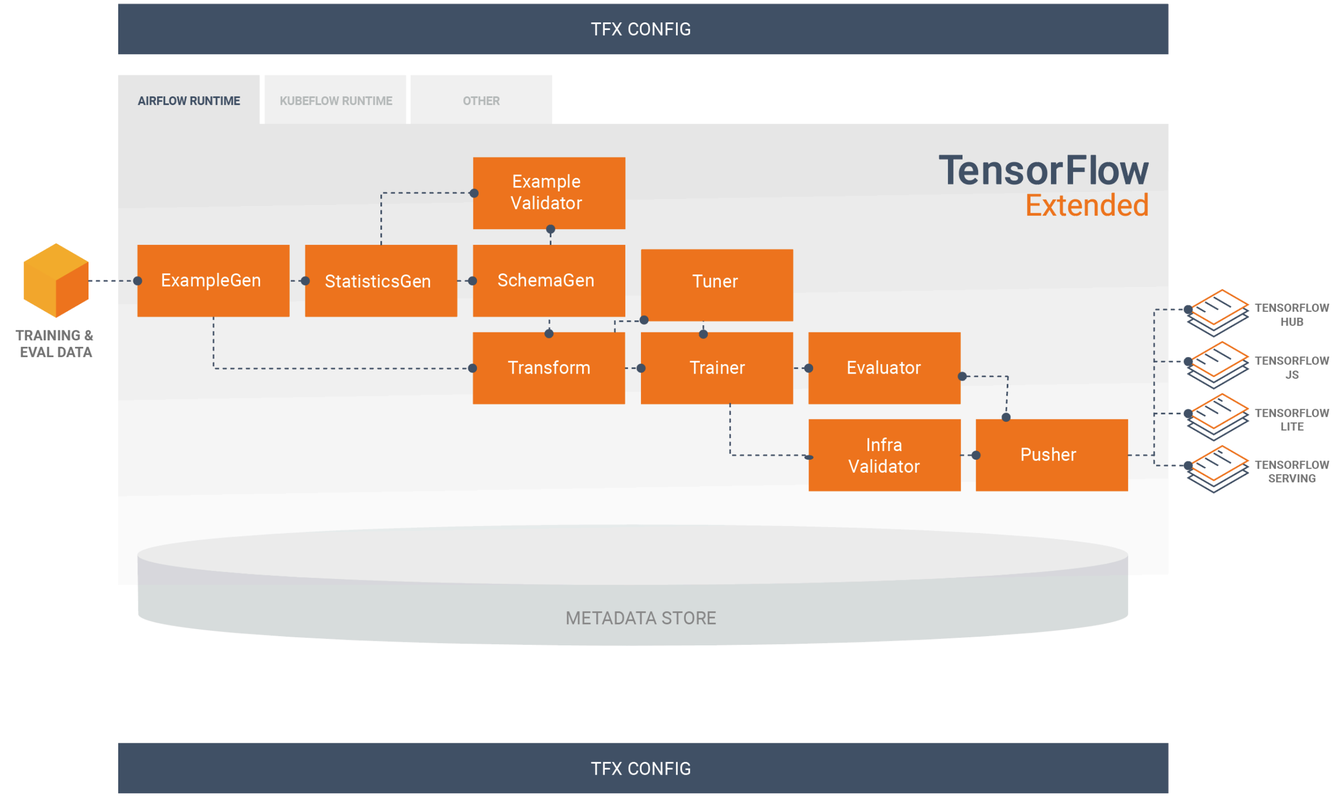


Example 2: Auto Data Prep for ML

TransmogrifAI

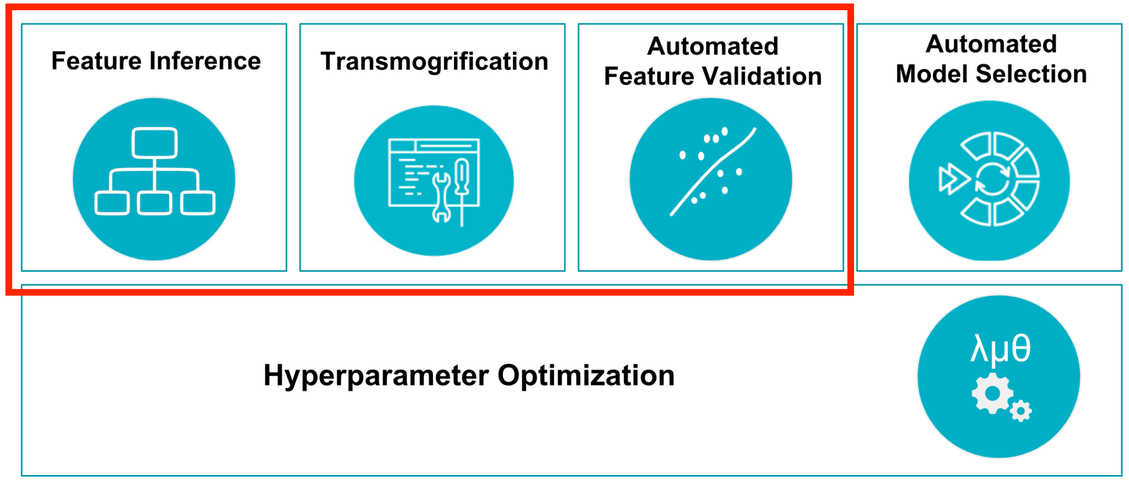


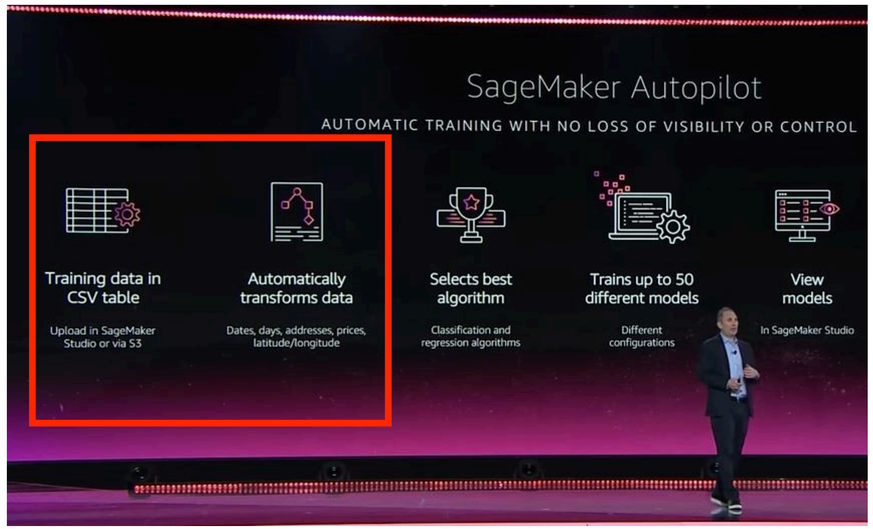


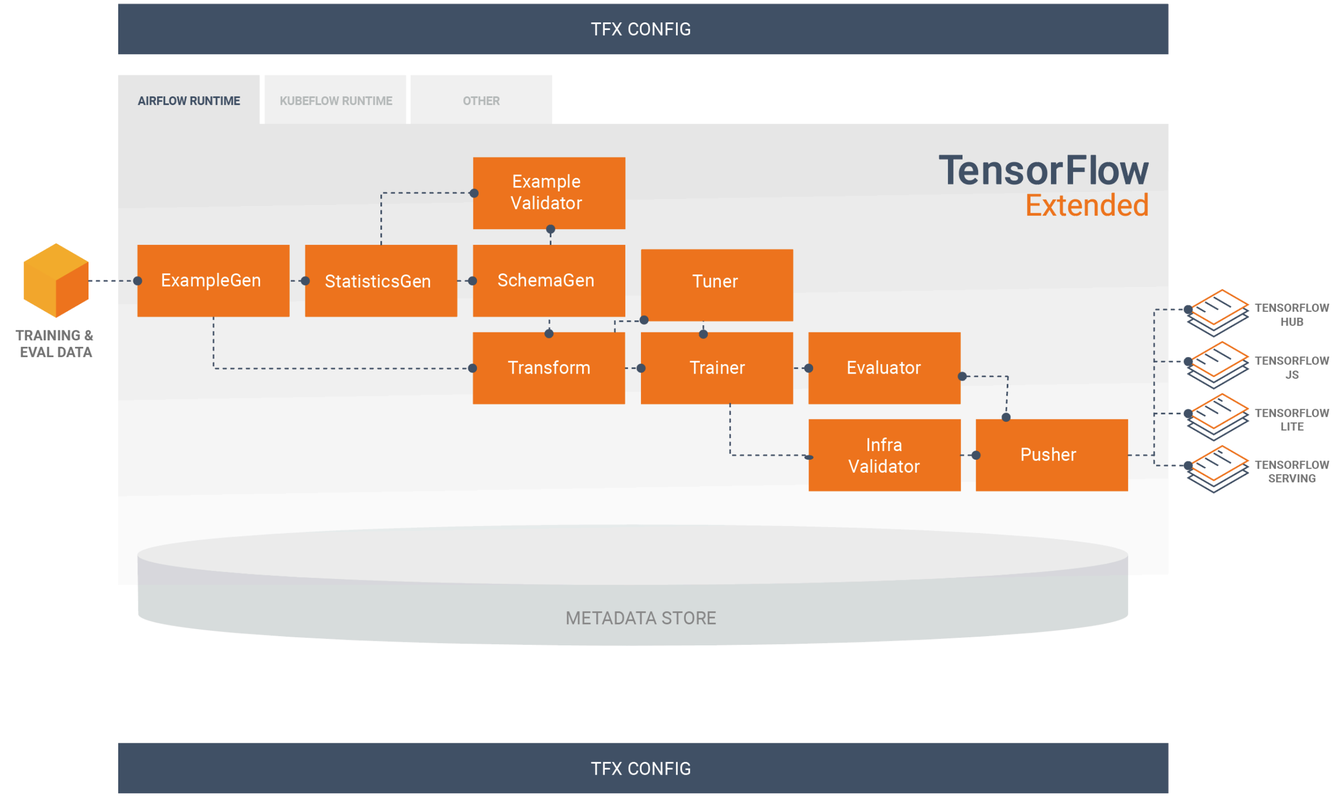


Example 2: Auto Data Prep for ML

TransmogrifAI

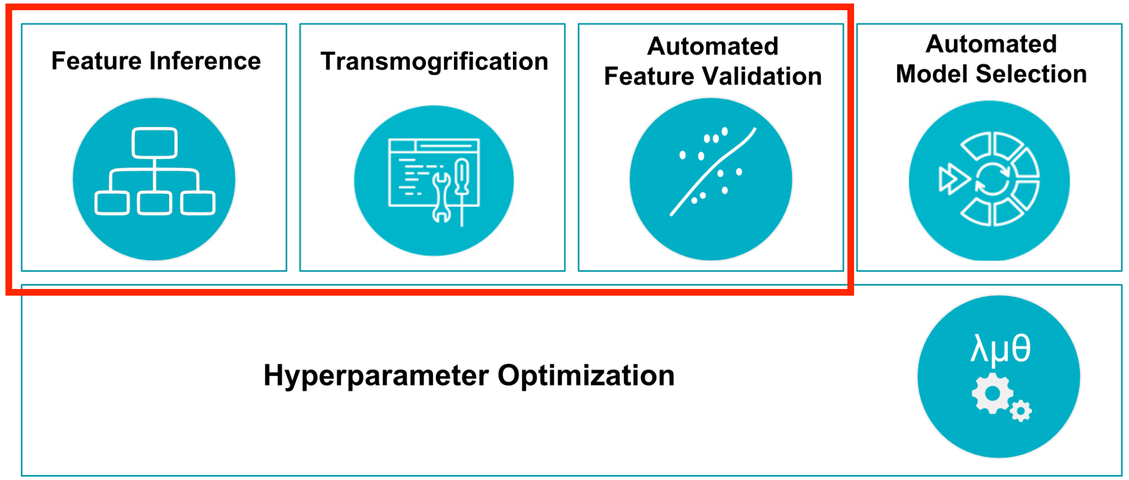


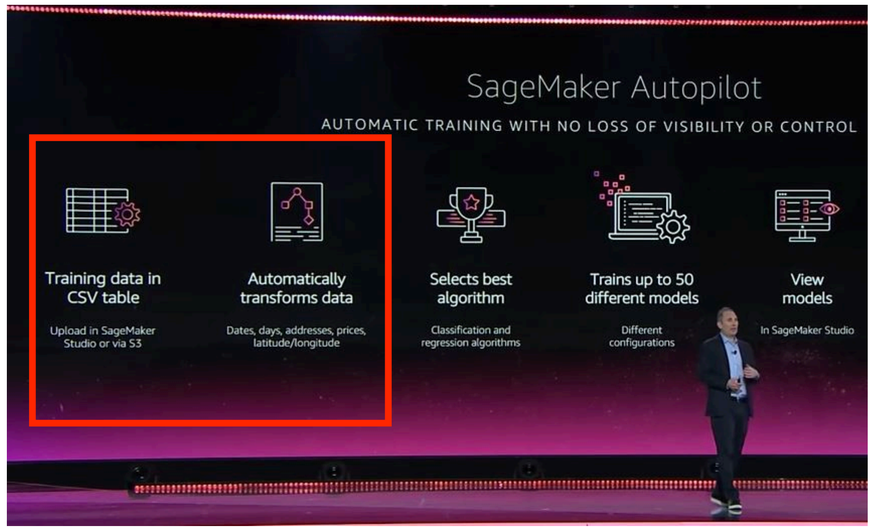


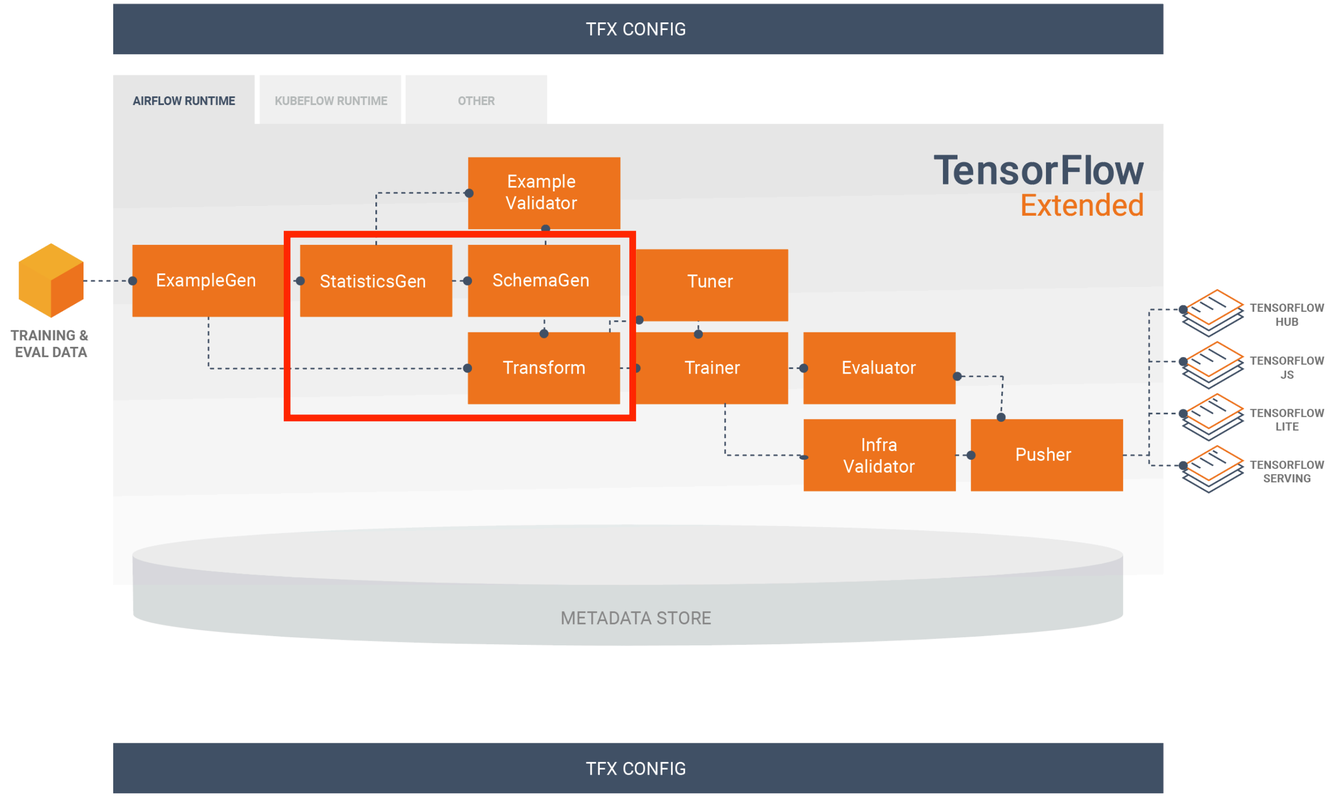


Example 2: Auto Data Prep for ML

TransmogrifAI

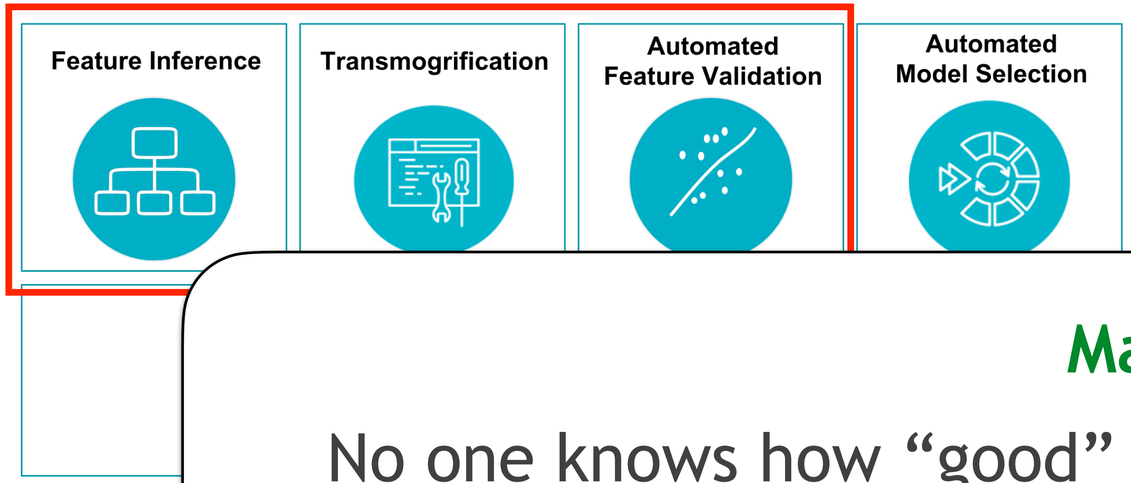


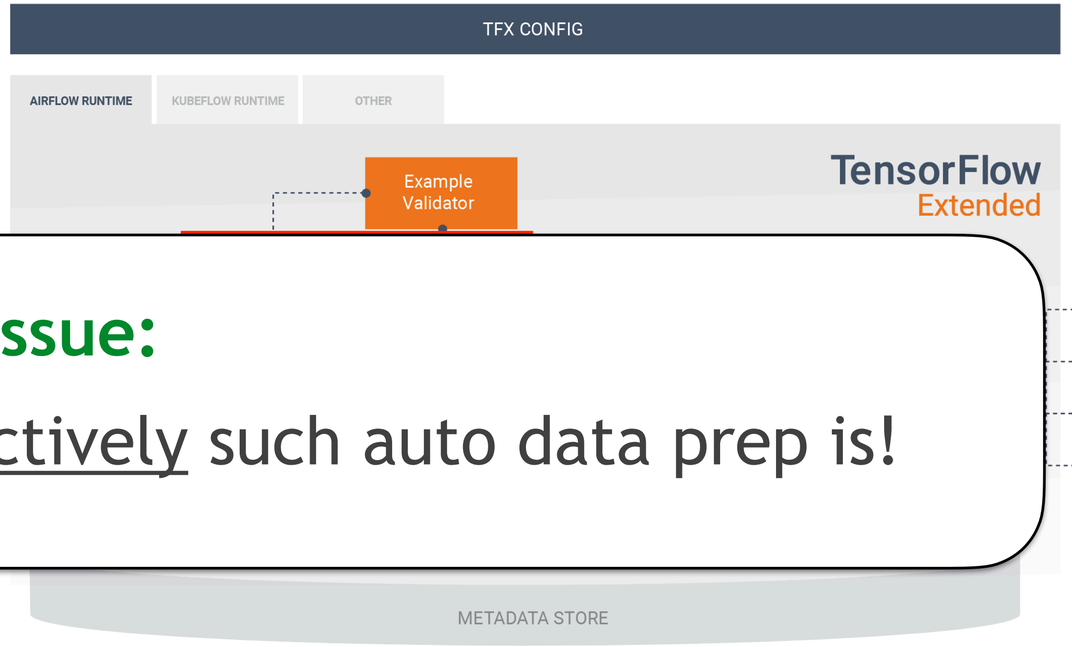




Example 2: Auto Data Prep for ML

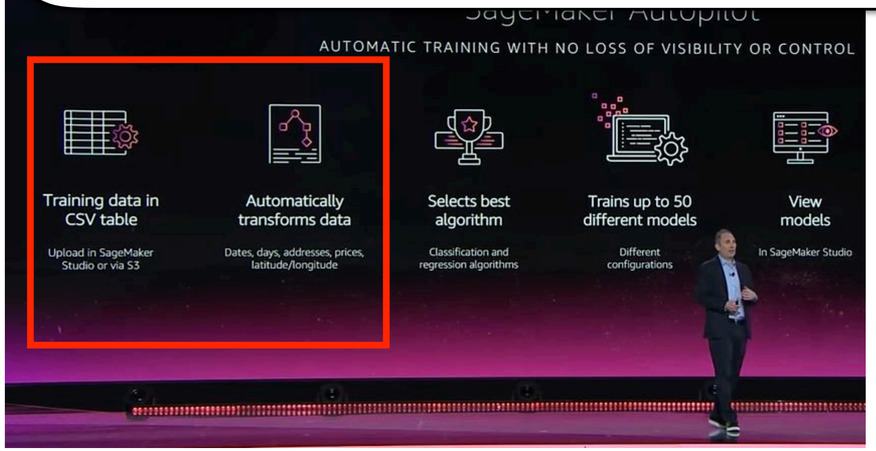
TransmogrifAI





Major Issue:

No one knows how "good" objectively such auto data prep is!



Project SortingHat

Benchmarks for ML Data Prep =

The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM'19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD'21***

Project SortingHat

Benchmarks for ML Data Prep = **Formalized task definition**

What are the inputs?
What are the outputs?

The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM'19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD'21***

Project SortingHat

Benchmarks for ML Data Prep = **Formalized task definition** + **Labeled dataset(s)**

What are the inputs?
What are the outputs?

(Input,Output) pairs to assess accuracy

The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM'19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD'21***

Project SortingHat

Benchmarks for ML Data Prep	=	**Formalized task definition**	+	**Labeled dataset(s)**	+	**Downstream predictions**
		What are the inputs? What are the outputs?		(Input,Output) pairs to assess accuracy		Does it really help?

The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM'19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD'21***

Project SortingHat

Benchmarks for ML Data Prep = **Formalized task definition** + **Labeled dataset(s)** + **Downstream predictions**

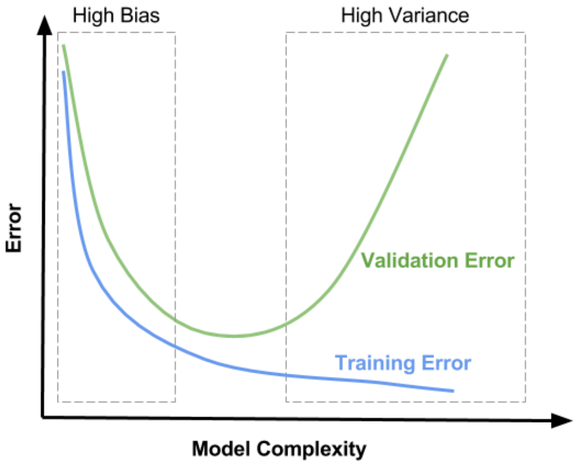
What are the inputs?
What are the outputs?

(Input,Output) pairs to assess accuracy

Does it really help?

How exactly does data prep affect ML accuracy?

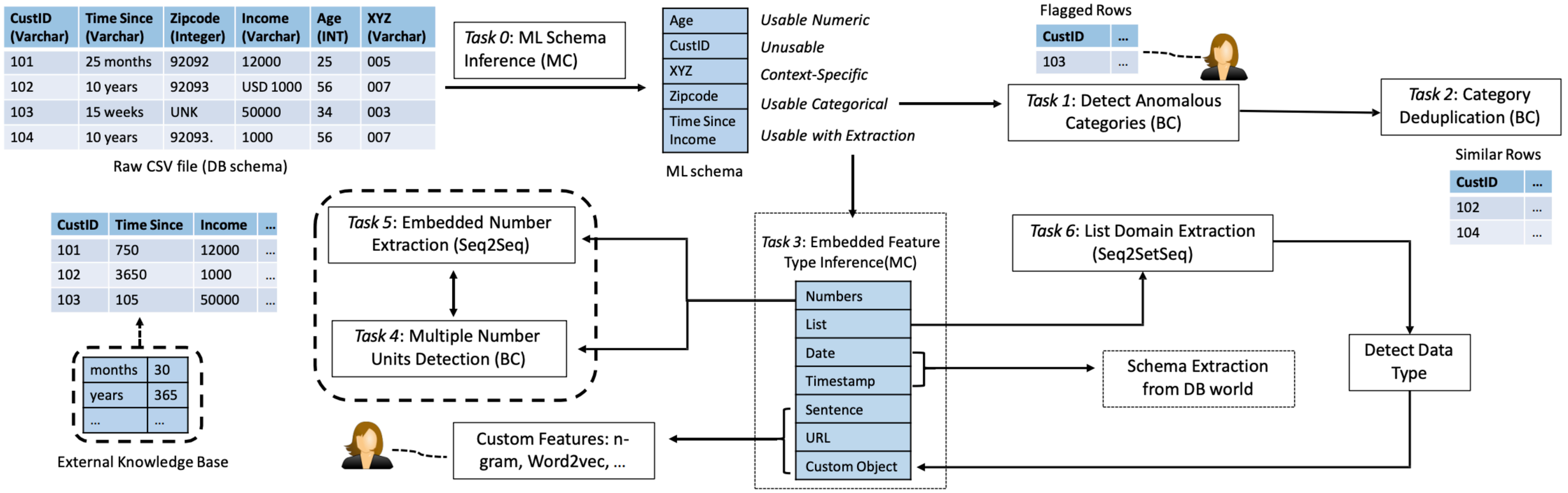
Are “failsafes” possible for errors in auto data prep?



The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM’19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD’21***

Project SortingHat

Current major tasks covered in our **ML Data Prep Zoo** benchmarks:



The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM'19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD'21***

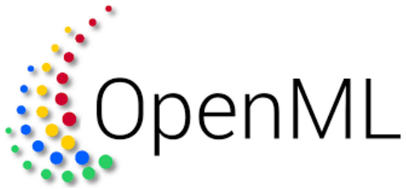
SortingHat: Early Impact and Trajectory

Integrated with TFDV in TFX in collaboration with Google

Being tested on Google's internal AutoML benchmarks

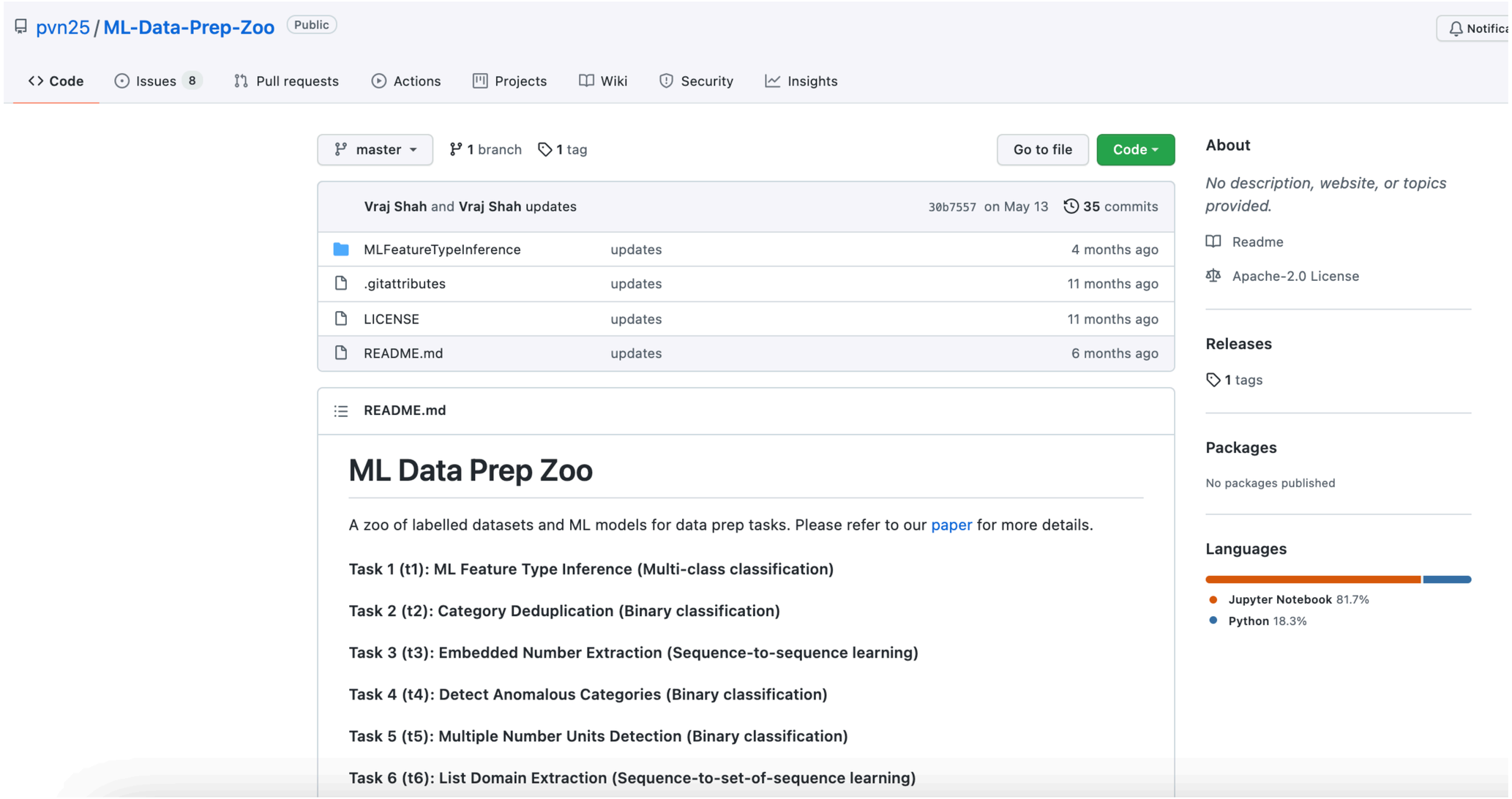
Our model used in NSF-funded "AI Maker" model + dataset search tool

Coming soon: AWS DBMS and ML/AI products; OpenML; catalog search



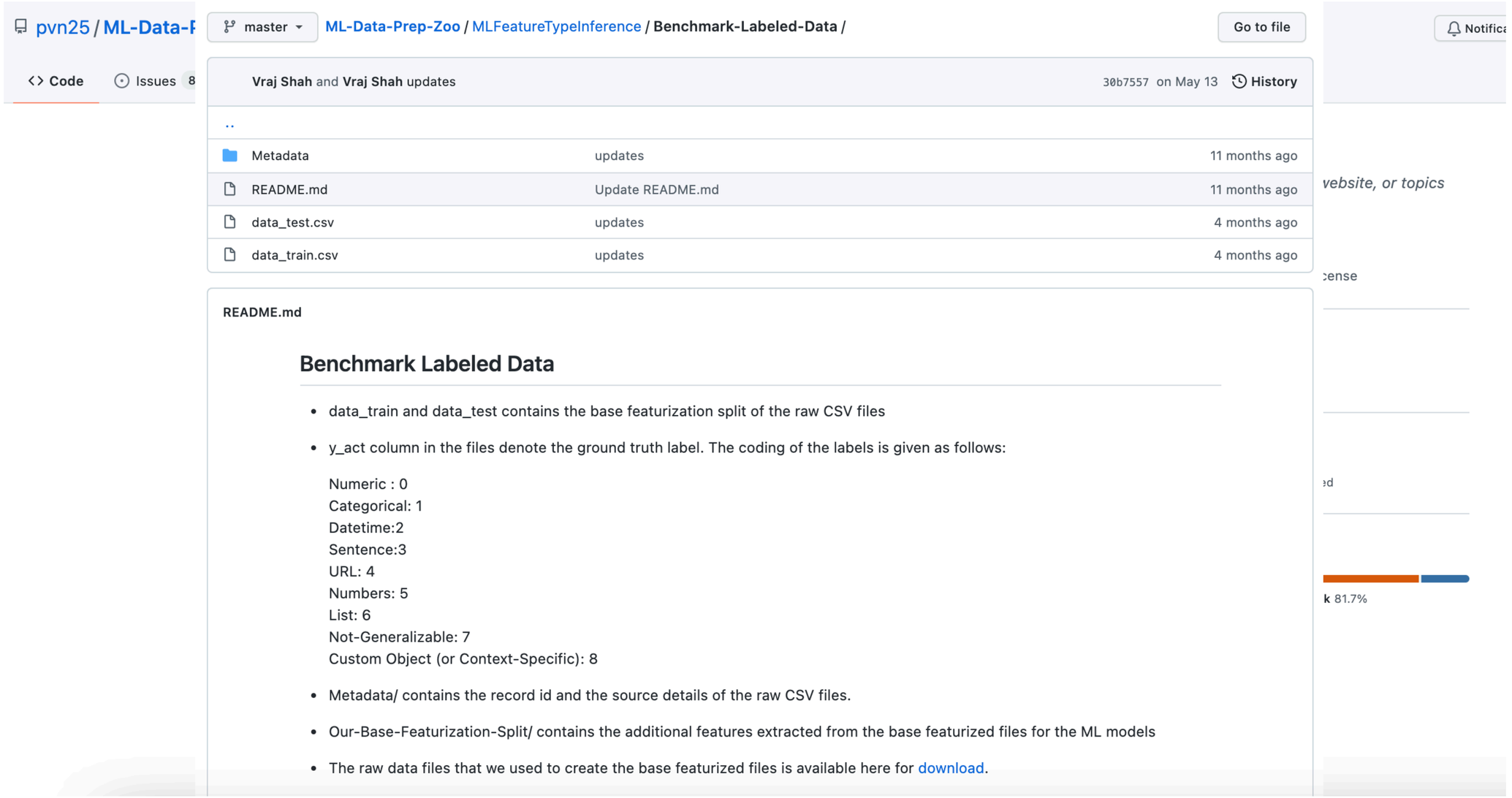
The ML Data Prep Zoo: Towards Semi-Automatic Data Preparation for ML. ***SIGMOD DEEM'19***
Towards Benchmarking Feature Type Inference for AutoML Platforms. ***SIGMOD'21***

SortingHat: Open Source Benchmark Repo



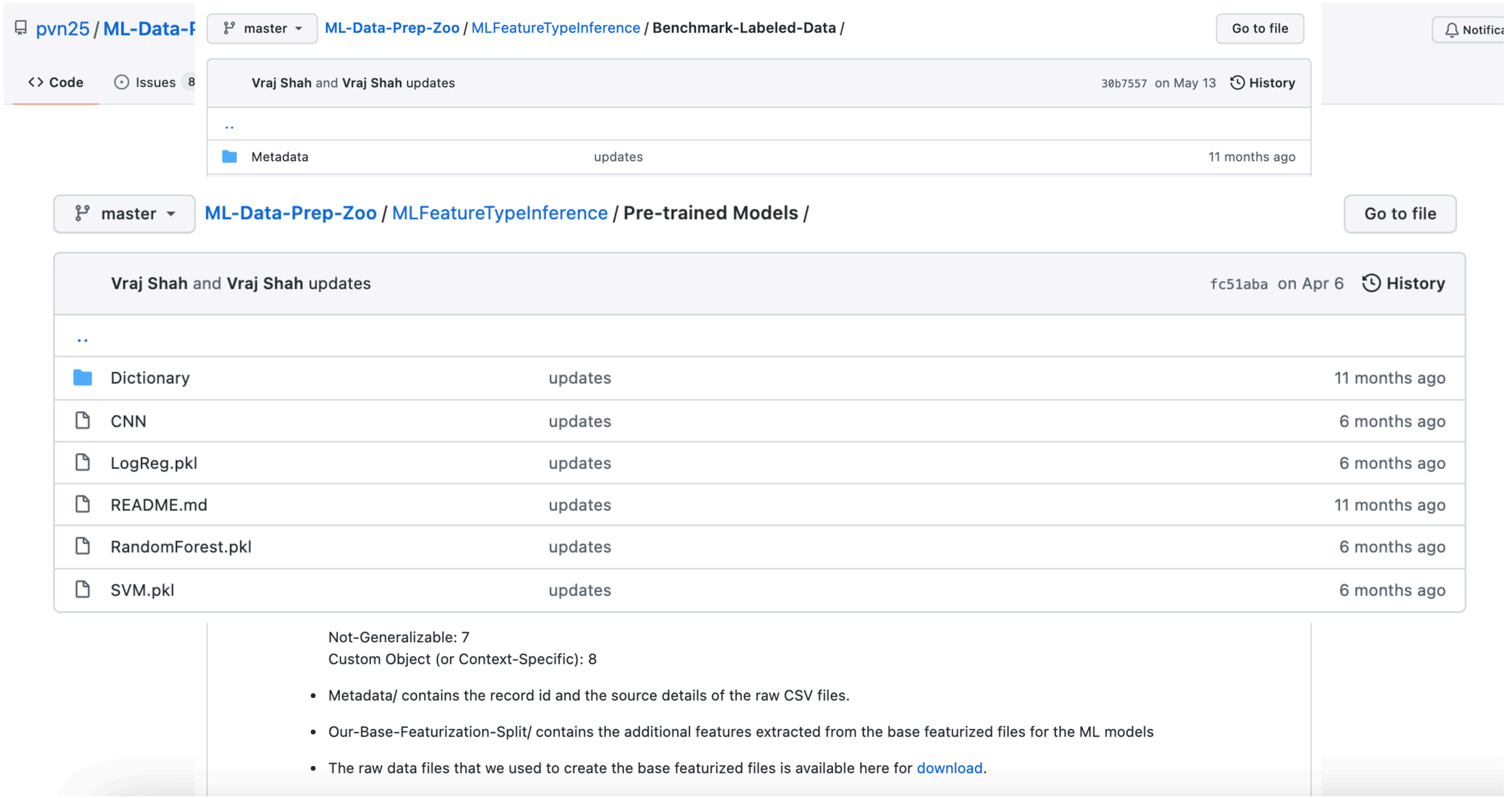
Links on project webpage: https://adalabucsd.github.io/sortinghat.html

SortingHat: Open Source Benchmark Repo



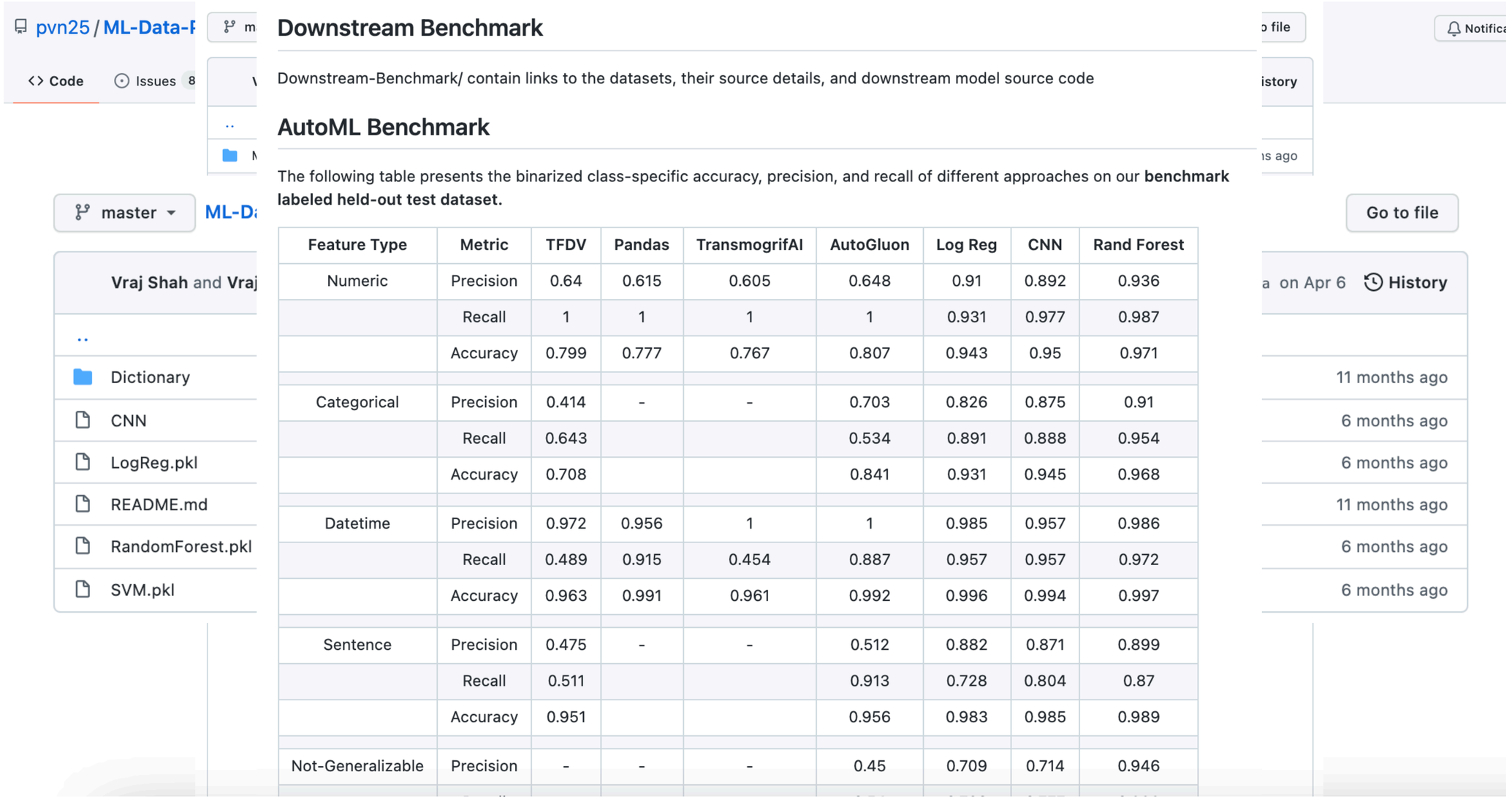
Links on project webpage: https://adalabucsd.github.io/sortinghat.html

SortingHat: Open Source Benchmark Repo



Links on project webpage: https://adalabucsd.github.io/sortinghat.html

SortingHat: Open Source Benchmark Repo



Downstream Benchmark

Downstream-Benchmark/ contain links to the datasets, their source details, and downstream model source code

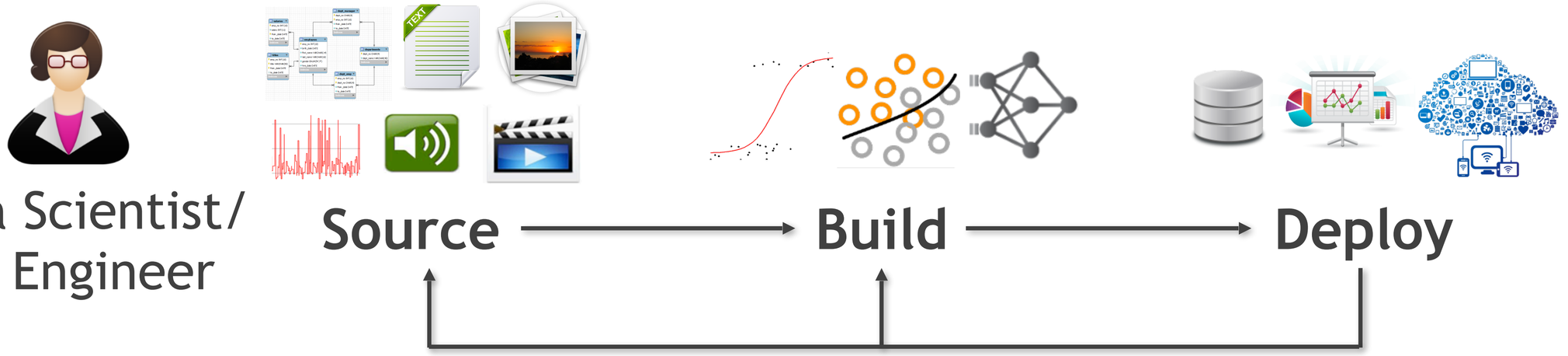
AutoML Benchmark

The following table presents the binarized class-specific accuracy, precision, and recall of different approaches on our **benchmark labeled held-out test dataset.**

Feature Type	Metric	TFDV	Pandas	TransmogrifAI	AutoGluon	Log Reg	CNN	Rand Forest
Numeric	Precision	0.64	0.615	0.605	0.648	0.91	0.892	0.936
	Recall	1	1	1	1	0.931	0.977	0.987
	Accuracy	0.799	0.777	0.767	0.807	0.943	0.95	0.971
Categorical	Precision	0.414	-	-	0.703	0.826	0.875	0.91
	Recall	0.643			0.534	0.891	0.888	0.954
	Accuracy	0.708			0.841	0.931	0.945	0.968
Datetime	Precision	0.972	0.956	1	1	0.985	0.957	0.986
	Recall	0.489	0.915	0.454	0.887	0.957	0.957	0.972
	Accuracy	0.963	0.991	0.961	0.992	0.996	0.994	0.997
Sentence	Precision	0.475	-	-	0.512	0.882	0.871	0.899
	Recall	0.511			0.913	0.728	0.804	0.87
	Accuracy	0.951			0.956	0.983	0.985	0.989
Not-Generalizable	Precision	-	-	-	0.45	0.709	0.714	0.946

Links on project webpage: https://adalabucsd.github.io/sortinghat.html

The New DBfication of ML/AI



Metadata Management for ML
Data Prep/Cleaning for ML
Multimodal ML Query Models
Data Search, Labeling, etc.
…

Scalable Data Systems for ML
Query Optimization for ML
Cloud and Streaming Infra.
Provenance and Debugging
…

Benchmark Frameworks and Data
Fairness, Transparency, Privacy, etc.

Leaving these problems open => Huge waste of time/ effort/money/energy/etc. by users of ML/AI!

Leaving these problems open => Huge waste of time/ effort/money/energy/etc. by users of ML/AI!

It is high time for DB, ML/AI, and more areas to join forces to accelerate the DBfication of ML/AI!

Outline

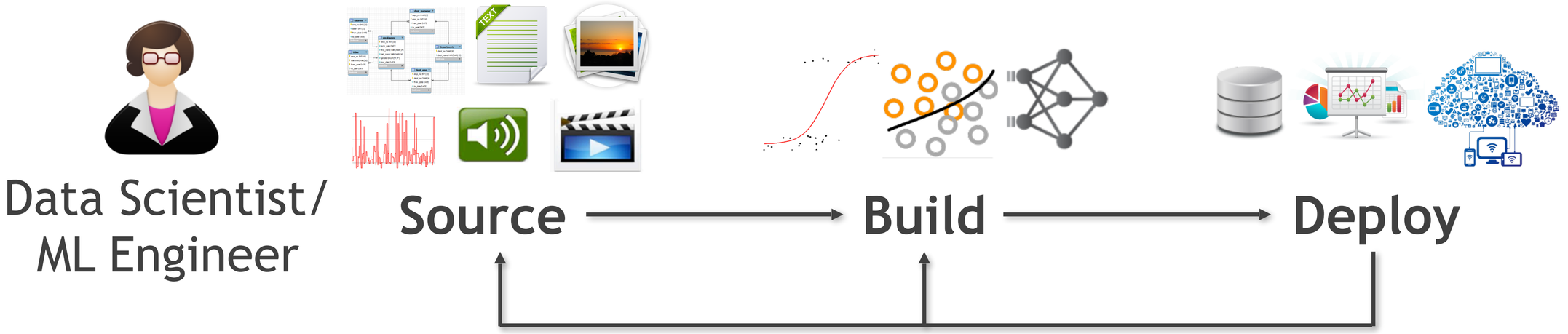
- The New DBfication of ML/AI
- Two Examples from My Research
- Accelerating the DBfication of ML/AI

DBfication of ML/AI is NOT just a DB Concern

Data Scientist/
ML Engineer

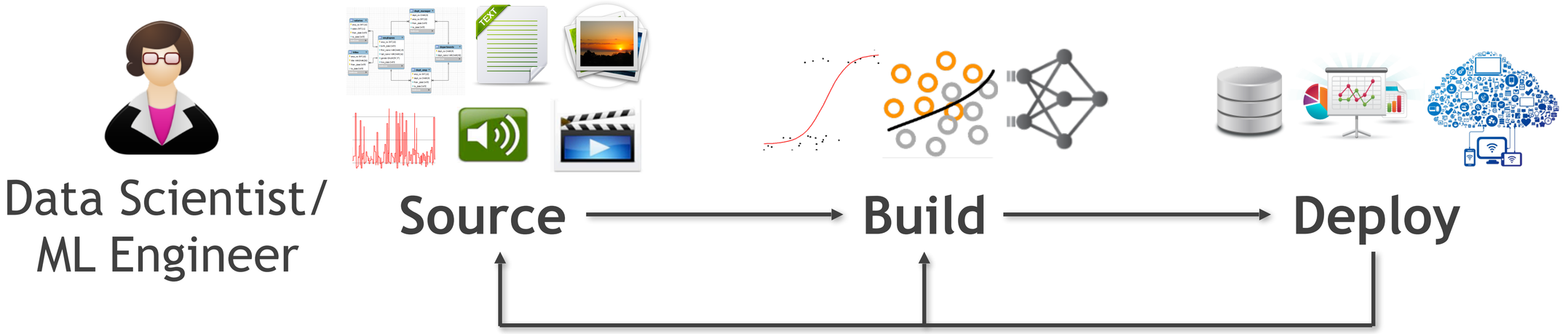
Source → Build → Deploy

DBfication of ML/AI is NOT just a DB Concern



1a. Applications of ML, IR, and data mining in this arena:

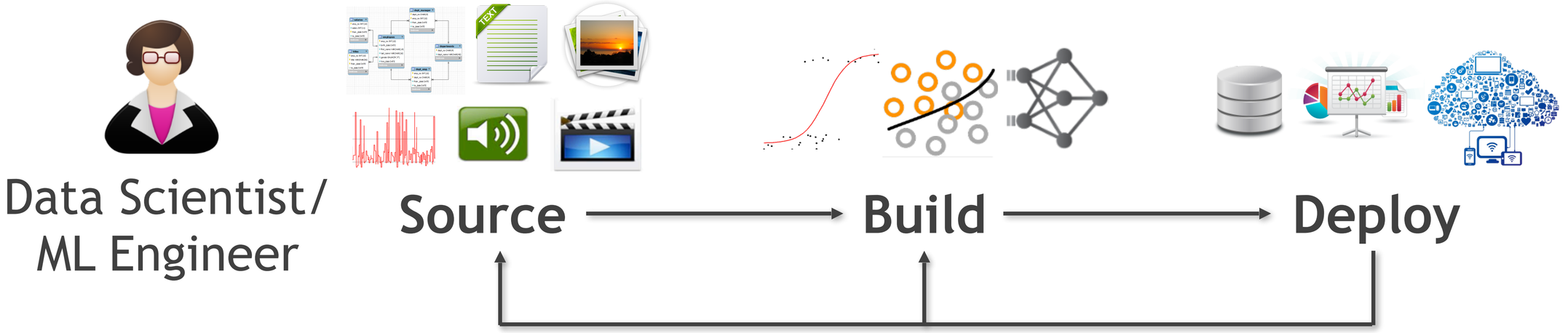
DBfication of ML/AI is NOT just a DB Concern



1a. Applications of ML, IR, and data mining in this arena:

Programmatic and learned label engineering

DBfication of ML/AI is NOT just a DB Concern

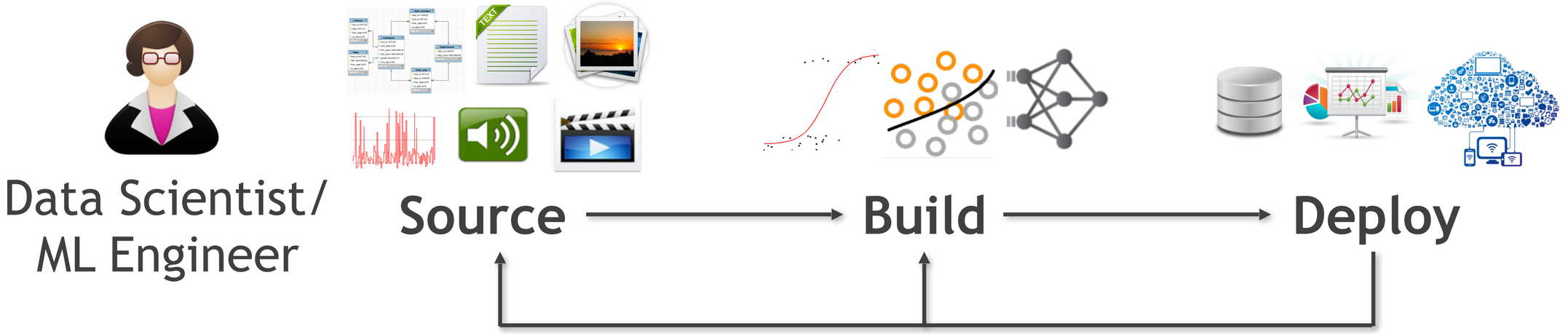


1a. Applications of ML, IR, and data mining in this arena:

Programmatic and learned label engineering

Dataset discovery and search systems

DBfication of ML/AI is NOT just a DB Concern



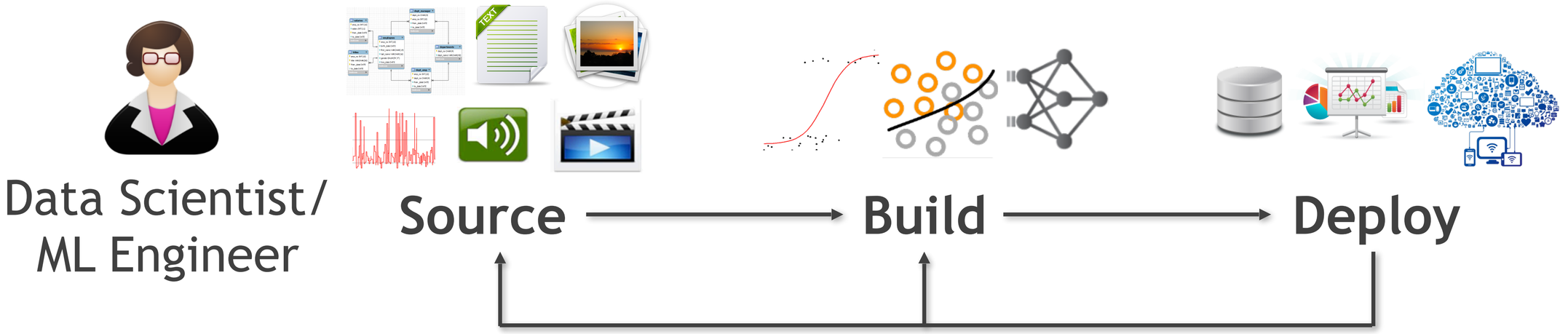
1a. Applications of ML, IR, and data mining in this arena:

Programmatic and learned label engineering

Dataset discovery and search systems

Recommendation systems for Data Science pipelines

DBfication of ML/AI is NOT just a DB Concern



1a. Applications of ML, IR, and data mining in this arena:

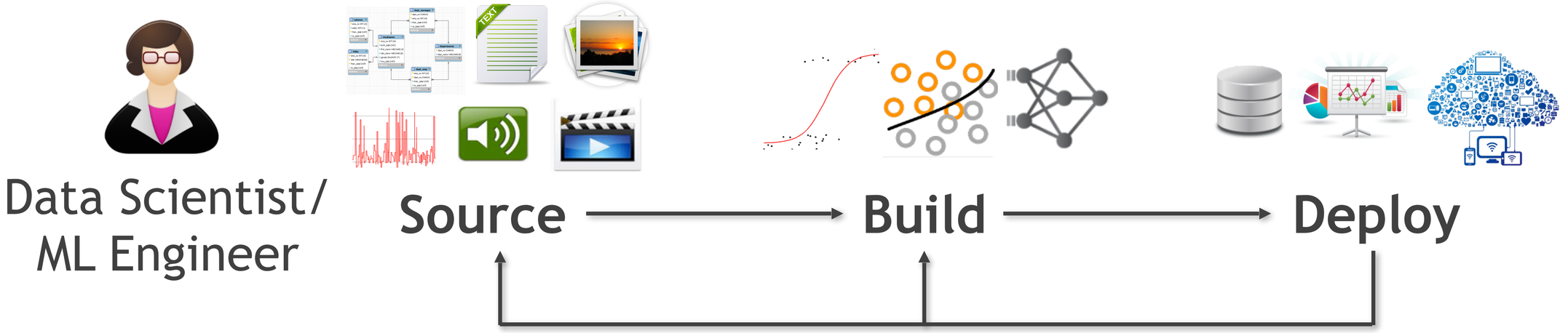
Programmatic and learned label engineering

Dataset discovery and search systems

Recommendation systems for Data Science pipelines

Automatic prediction monitoring and actionable recourse

DBfication of ML/AI is NOT just a DB Concern



1a. Applications of ML, IR, and data mining in this arena:

Programmatic and learned label engineering

Dataset discovery and search systems

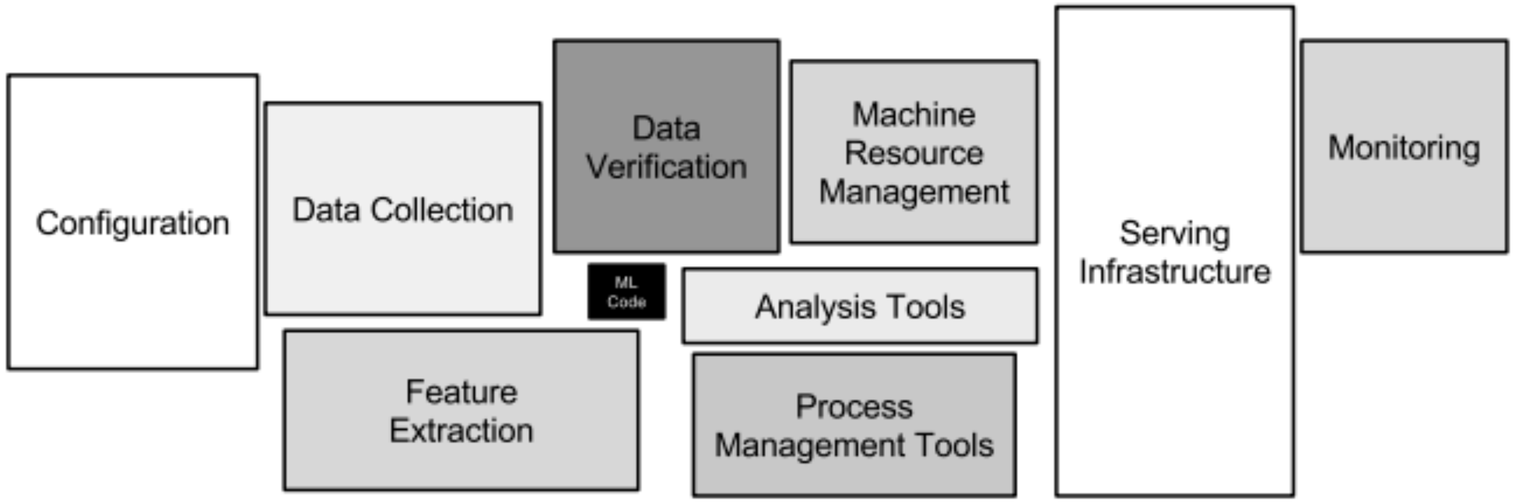
Recommendation systems for Data Science pipelines

Automatic prediction monitoring and actionable recourse

Hybrid human-in-the-loop AutoML platforms

DBfication of ML/AI is NOT just a DB Concern

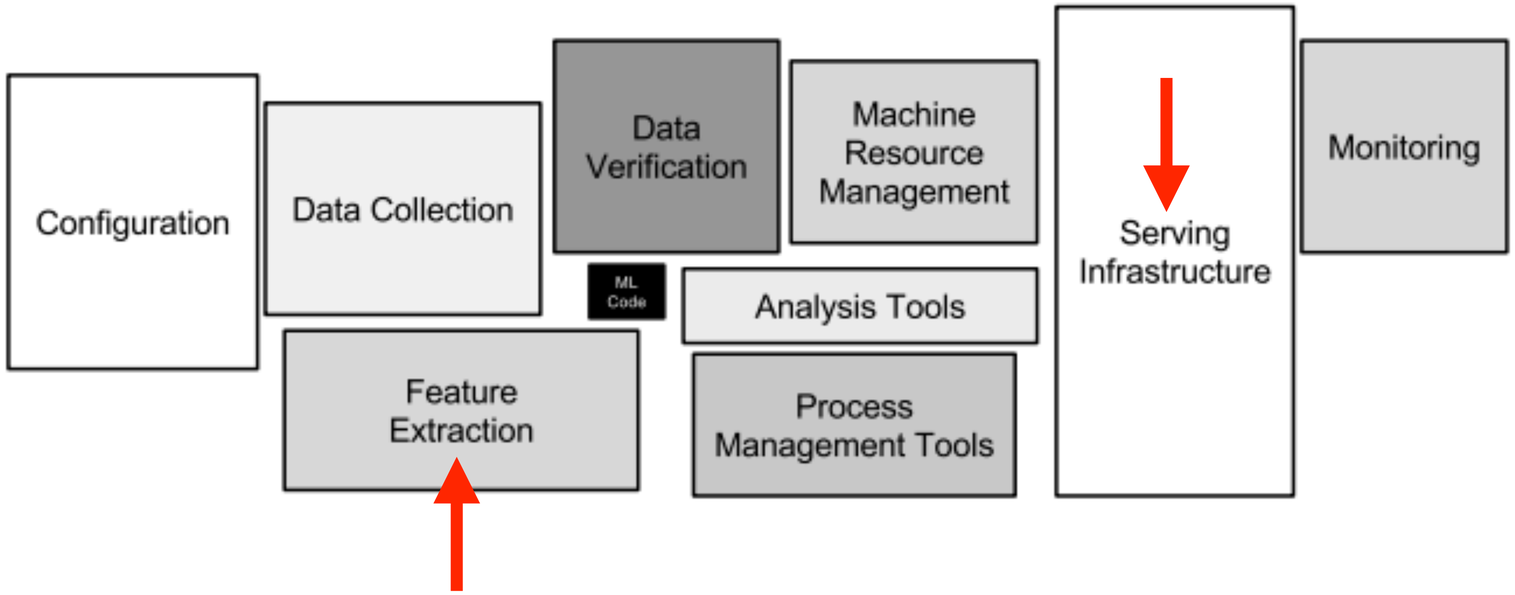
1b. More intersection with systems, PL/software engineering, etc:



DBfication of ML/AI is NOT just a DB Concern

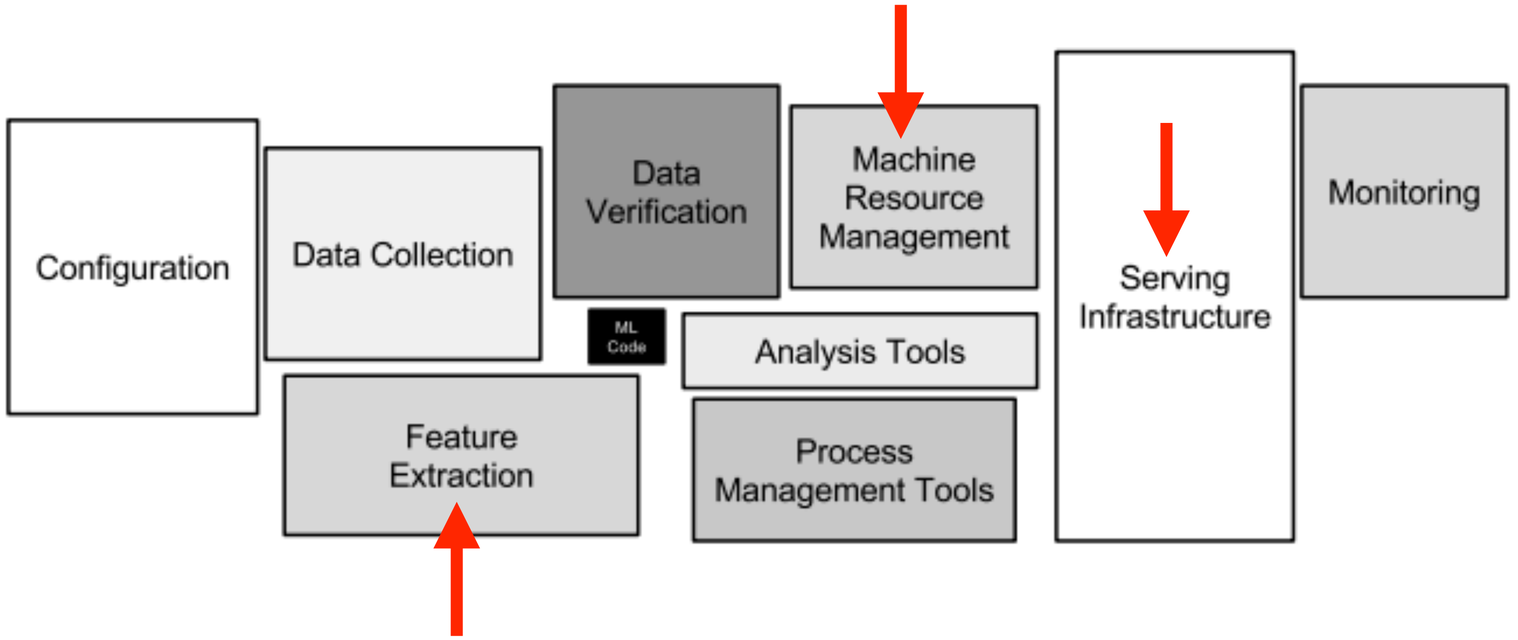
1b. More intersection with systems, PL/software engineering, etc:

Feature stores



DBfication of ML/AI is NOT just a DB Concern

1b. More intersection with systems, PL/software engineering, etc:

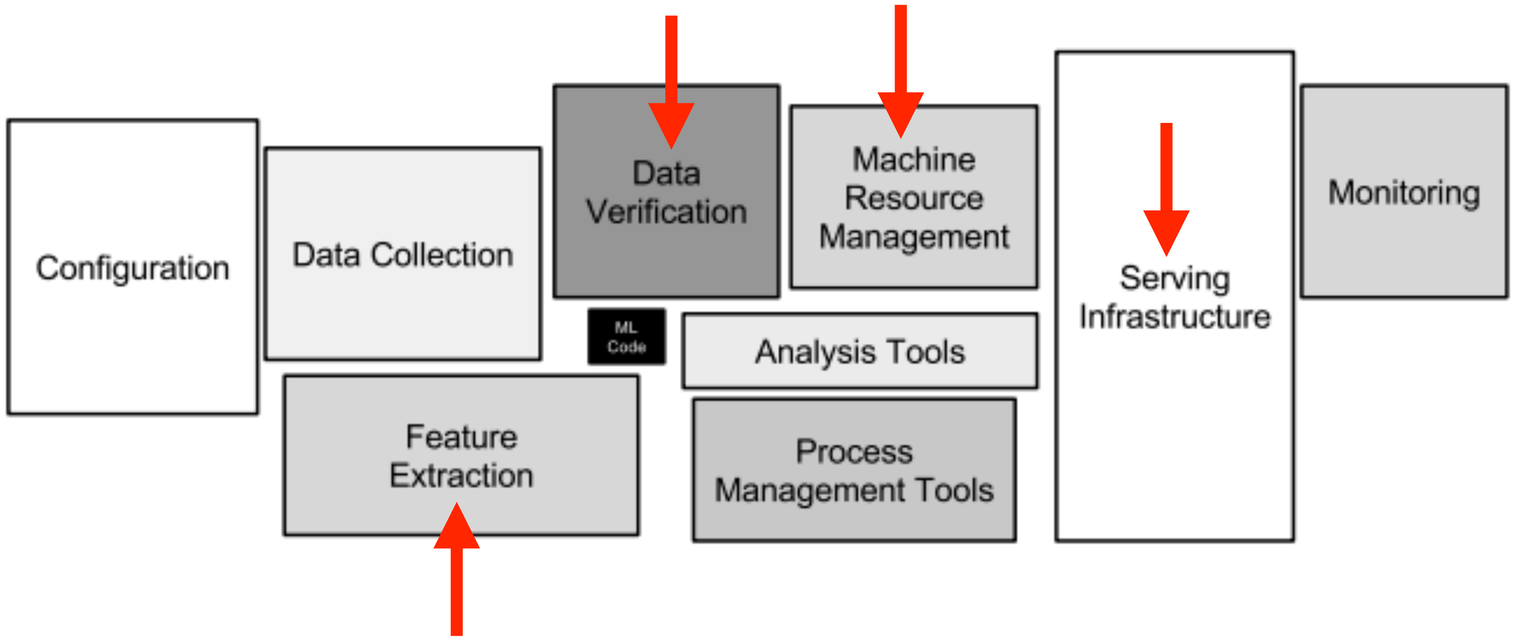


Feature stores

Provisioning & scheduling, incl. learned systems

DBfication of ML/AI is NOT just a DB Concern

1b. More intersection with systems, PL/software engineering, etc:



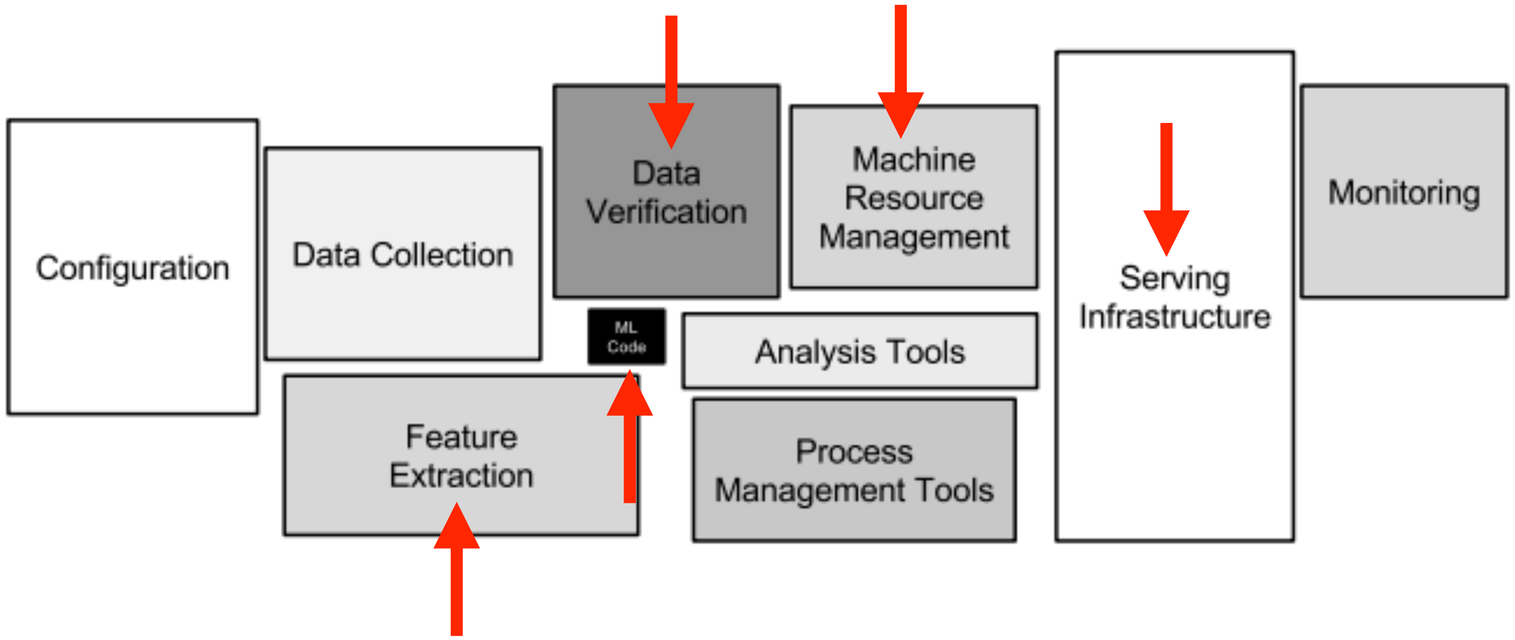
Feature stores

Provisioning & scheduling, incl. learned systems

Data quality verification

DBfication of ML/AI is NOT just a DB Concern

1b. More intersection with systems, PL/software engineering, etc:



Feature stores

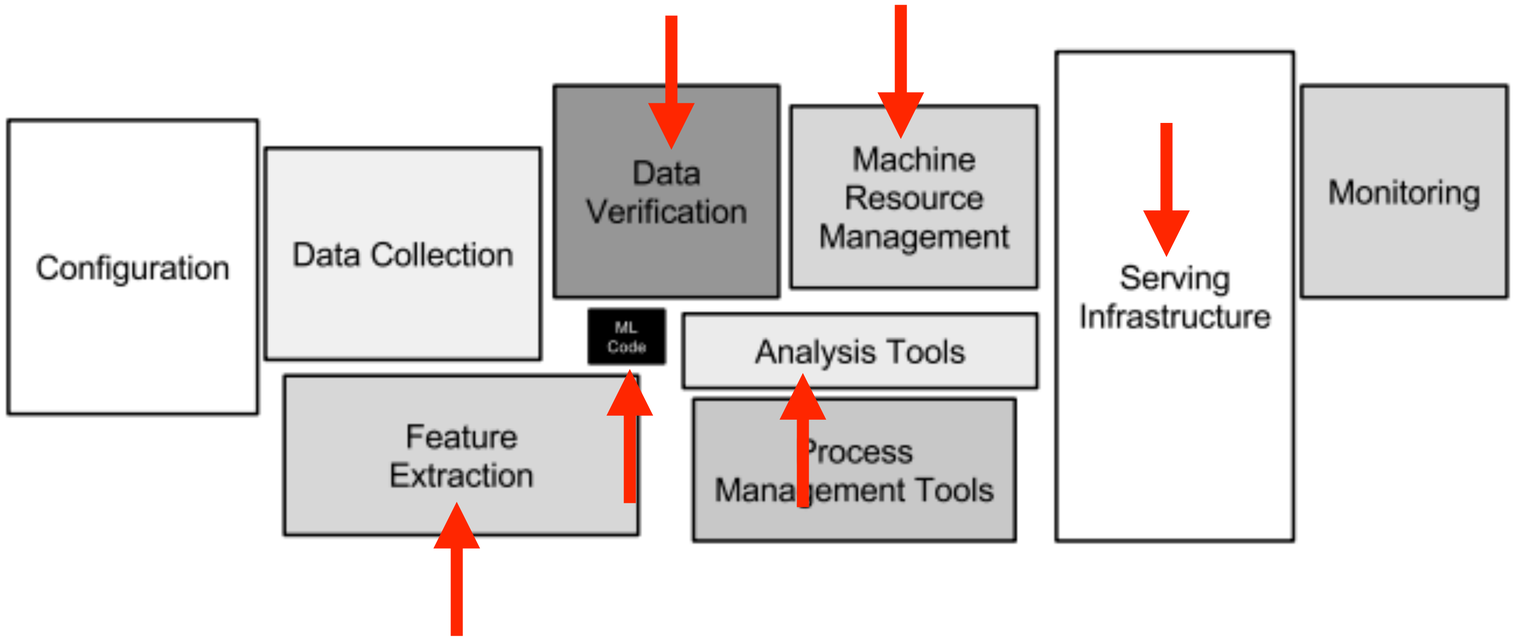
Provisioning & scheduling, incl. learned systems

Data quality verification

ML compilers and learned operator placement

DBfication of ML/AI is NOT just a DB Concern

1b. More intersection with systems, PL/software engineering, etc:



Feature stores

Provisioning & scheduling, incl. learned systems

Data quality verification

ML compilers and learned operator placement

CI/CD for ML engineering

…

Accelerating the DBfication of ML/AI

2. Check out my "DB for ML" grad course and research book? :)

CSE 234/291: Data Systems for Machine Learning. UC San Diego. 2020.

Data Management in Machine Learning Systems. Morgan & Claypool Publishers. 2019.

Accelerating the DBfication of ML/AI

2. Check out my “DB for ML” grad course and research book? :)

CSE 234/291: Data Systems for Machine Learning. UC San Diego. 2020.

Data Management in Machine Learning Systems. Morgan & Claypool Publishers. 2019.

3. Check out SIGMOD DEEM and HILDA Workshops. Check out MLSys.

Accelerating the DBfication of ML/AI

2. Check out my "DB for ML" grad course and research book? :)

CSE 234/291: Data Systems for Machine Learning. UC San Diego. 2020.

Data Management in Machine Learning Systems. Morgan & Claypool Publishers. 2019.

3. Check out SIGMOD DEEM and HILDA Workshops. Check out MLSys.

4. Check out recent "DB for ML" tutorials and papers.

Data Management in Machine Learning: Challenges, Techniques, and Systems. SIGMOD 2017.

Data Management Challenges in Production Machine Learning. SIGMOD 2017.

Data Collection and Quality Challenges for Deep Learning. VLDB 2020.

Accelerating the DBfication of ML/AI

Accelerating the DBfication of ML/AI

5. Check out topical panel discussions on “DB for ML” stuff.

Accelerating the DBfication of ML/AI

5. Check out topical panel discussions on “DB for ML” stuff.

ACM SIGMOD Blog

ML/AI SYSTEMS AND APPLICATIONS: IS THE SIGMOD/VLDB COMMUNITY LOSING RELEVANCE?

Arun Kumar

AUGUST 21, 2018

Databases , Machine Learning

Photos from the workshop. L to R: (1) The DEEM audience. (2) The Panelists: Matei, Joaquin, Jens, Joey, and Manasi (Martin not pictured). (3) Advocatus Diaboli.

SIGMOD DEEM’18

Accelerating the DBfication of ML/AI

5. Check out topical panel discussions on “DB for ML” stuff.

ACM SIGMOD Blog

ML/AI SYSTEMS AND APPLICATIONS: IS THE SIGMOD/VLDB COMMUNITY LOSING RELEVANCE?

Arun Kumar

AUGUST 21, 2018

Databases , Machine Learning

Photos from the workshop. L to R: (1) The DEEM audience. (2) The Panelists: Matei, Joaquin, Jens, Joey, and Manasi (Martin not pictured). (3) Advocatus Diaboli.

SIGMOD DEEM’18

ACM SIGMOD

Online. Tue, June 22, 2021. 8:30am US PT.

Hot Button Panel Discussion

Automation of Data Prep, ML, and Data Science:
New Cure or Snake Oil?

Panel of Experts:

Felix Naumann
University of Potsdam &
Hasso Plattner Institute

Ihab Ilyas
Inductiv, Tamr, &
University of Waterloo

Joe Hellerstein
Trifacta &
UC Berkeley

Luna Dong
Amazon

Sarah Catanzaro
Amplify Partners

Panel Chair(s):

Arun Kumar
UC San Diego

&

Dr. A. Diaboli
At large

SIGMOD’21

Accelerating the DBfication of ML/AI

6. MOST IMPORTANT: Build REAL stuff and help transfer research to practice via partnerships/collaboration with ML/AI users.

Data scientists, Data analysts, ML engineers, MLOps engineers, etc.

Domain sciences, enterprises, Web companies, cloud vendors, policy, etc.

Create open source artifacts, both software and datasets

Attend both research and industry venues: Spark+AI Summit, FOSDEM, etc.

…

New Publication Avenues

Association For Computing Machinery

ACM SIGMOD Blog

Alon Halevy, Arun Kumar and Nesime Tatbul

FEBRUARY 10, 2020

SCALABLE DATA SCIENCE: A NEW RESEARCH TRACK CATEGORY AT PVLDB VOL 14 / VLDB 2021

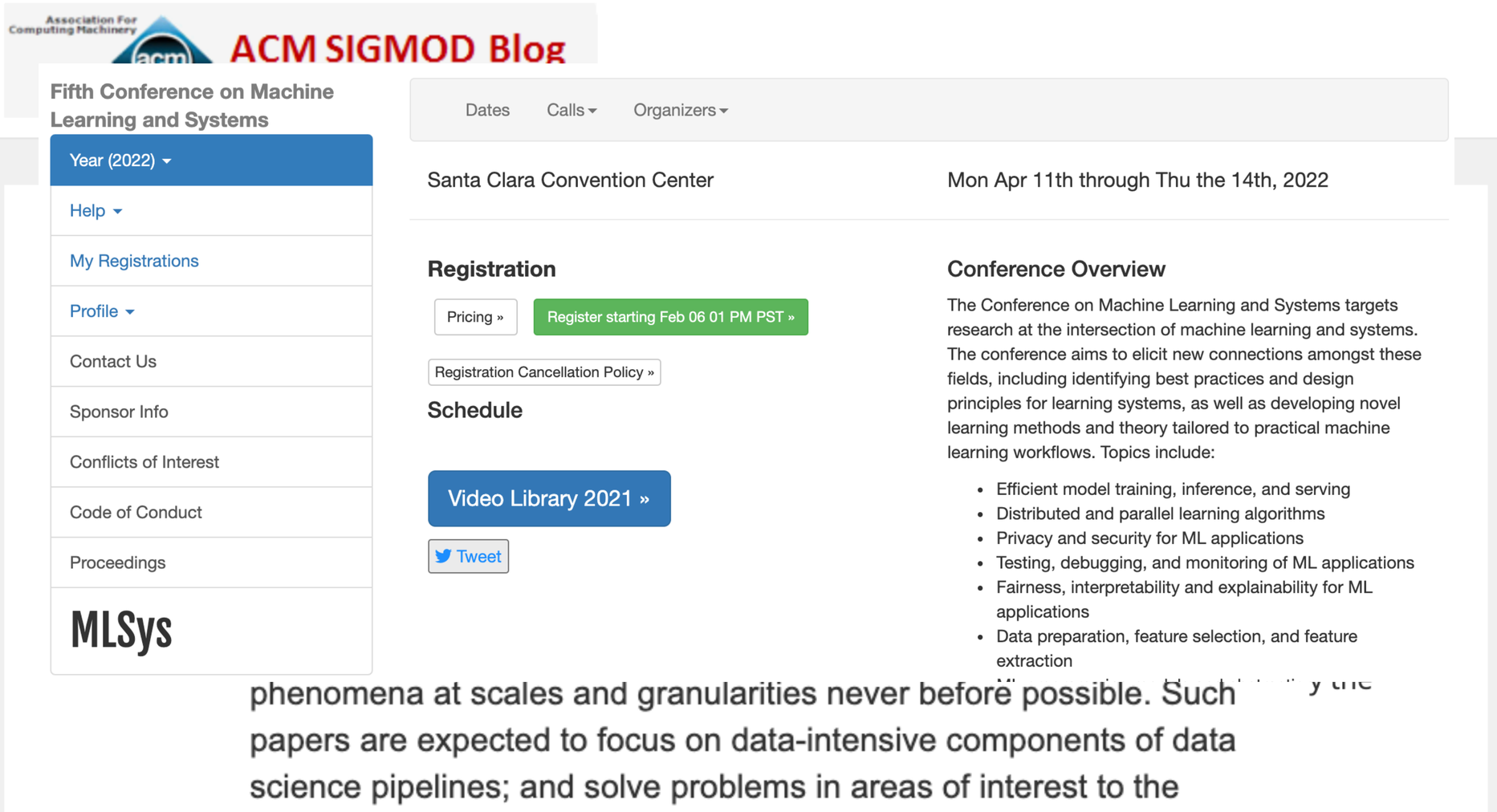
Uncategorized

This post introduces and explains the newly created category of "Scalable Data Science" within the Research Track of PVLDB. This category comes into effect for volume 14, i.e., submissions starting April 1, 2020, which will be evaluated by the Review Board of PVLDB vol 14 for presentation at VLDB 2021.

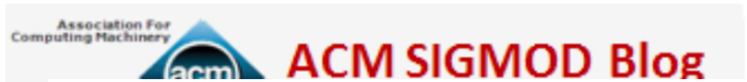
The Growth of Data Science

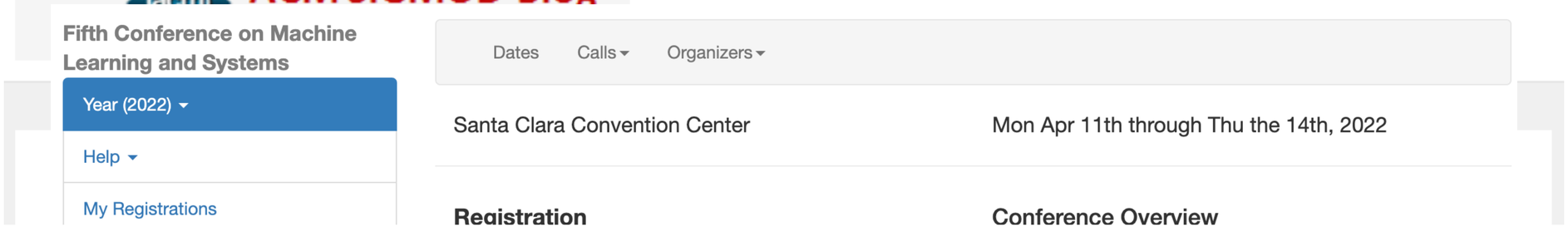
New Publication Avenues

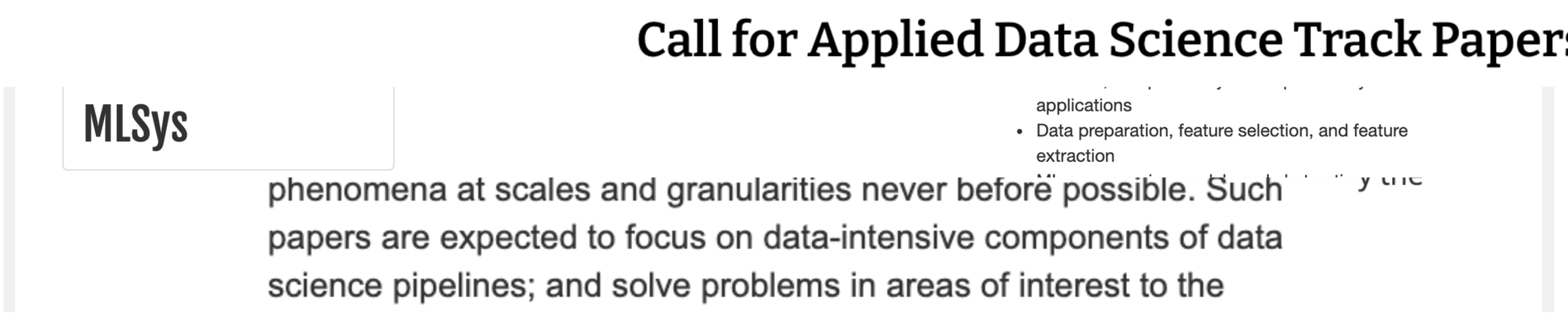
New Publication Avenues



New Publication Avenues







The DBfication of ML/AI is a grand challenge for all of computing to help democratize ML/AI.

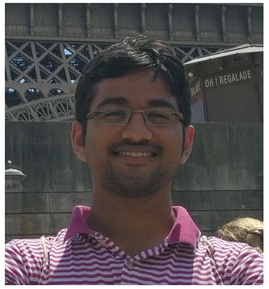
The DBfication of ML/AI is a grand challenge for all of computing to help democratize ML/AI.

Intersection of DB, ML/AI, systems, HCI, etc.; work with ML/AI users in industry and domain sciences.

My Terrific Advisees

Supun Nakandala
PhD

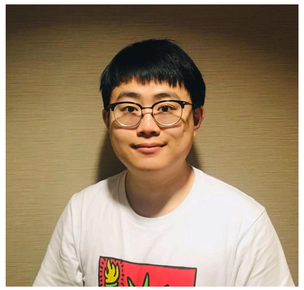
Tara Mirmira
PhD



Vraj Shah
PhD & MS

Liangde Li
Kyle Luoma

Xiuwen Zheng
PhD & MS



Yuhao Zhang
PhD & MS

Kabir Nagrecha
BS & PhD

Alumni:
Side Li
Advitya Gemawat
Shaoqing Yi
David Justo
Kevin Yang

https://ADALabUCSD.github.io

https://ADALabUCSD.github.io

arunkk@eng.ucsd.edu

github.com/ADALabUCSD

@TweetAtAKK

ACKS: