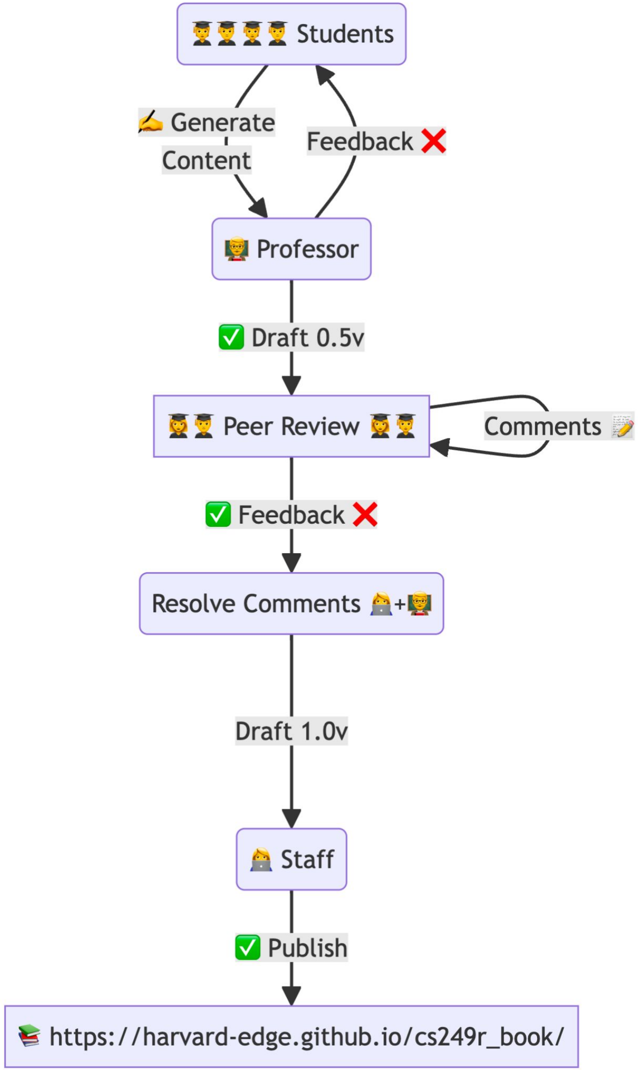
CS249r: MLOps

Oct 30

Course Logistics

Assignment Schedule Updates

- Assignment 2
  - Due: October 23rd (Monday)
- Mid-Project Review
  - Due: October 30th (Monday)
- Assignment 3
  - Due: November 6th (Monday)
- Assignment 4 Part 1
  - Due: November 20th (Monday)
- Assignment 4 Part 2
  - Due: November 27th (Monday)
- Project Presentations
  - Due: December 4th (Monday)
- Final Report
  - Due: December 11th (Monday)



Assignment 2

- Grading
    - Please reach out via email if you have any questions or concerns (cs249r-fa23-tinyml@googlegroups.com)

- Rubric

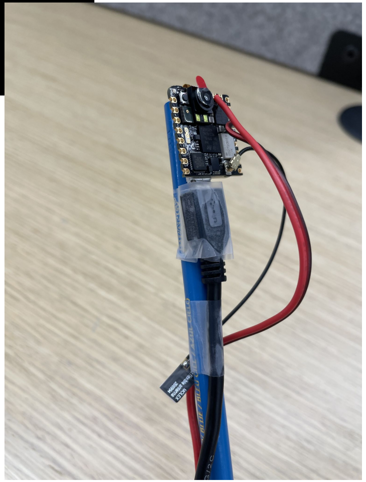
Assignment 3: Magic Nicla Wand

Due: November 6 at 11:59 pm

Objective:

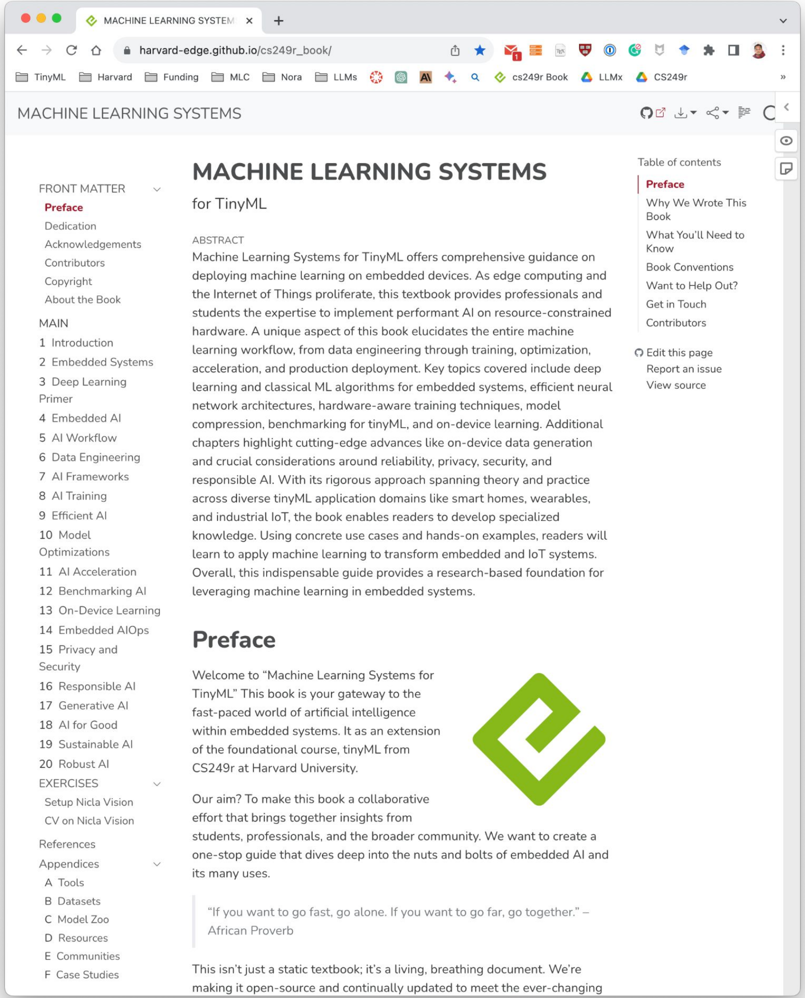
- Explore Tensorflow ecosystem (Tensorflow -> Tensorflow Lite -> Tensorflow Lite Micro)
- Model Optimization (quantization/pruning) using IMU data from Arduino Nicla Vision

Extra Credit: Deployment of model on Nicla



Scribing (again!)

- This week
  - Model Optimization will be reviewed and merged by EOD
    1. If you haven't taken part, look through it today ASAP.
    2. One detailed review
  - Benchmarking AI is out!
  - On-device Learning peer review starts today
- Next week
  - Hardware acceleration coming soon!
  - MLOps starts today! Meet after class.



Nov 6	Secure and Privacy-Preserving On-Device ML	Emanuel Moss, Research Scientist at Intel Labs	Required • Machine Learning Sensors (paper) • Security of Neural Networks from Hardware Perspective: A Survey and Beyond (paper) Optional • Robust Machine Learning Systems: Reliability and Security for Deep Neural Networks (paper) • On Safeguarding Privacy and Security in the Framework of Federated Learning (paper)	None	Assignment 3 Due

November 6th

Title: Codesigning Computing Systems for Artificial Intelligence

Abstract: The rapid advancement of artificial intelligence (AI) has ushered in an era of unprecedented computational demands, necessitating continuous innovation in computing systems. In this talk, we will highlight how codesign has been a key paradigm in enabling innovative solutions and state-of-the-art performance in Google's AI computing systems, namely Tensor Processing Units (TPUs). We present several codesign case studies across different layers of the stack, spanning hardware, systems, software, algorithms, all the way up to the datacenter. We discuss how TPUs have made judicious, yet opinionated bets in our design choices, and how these design choices have not only kept pace with the blistering rate of change, but also enabled many of the breakthroughs.

Project Updates

Project Updates

Lightning talks

- 13 groups, 3 mins each
- Slides here
- Real-time feedback - https://tinyurl.com/249r-projects
    - One clarifying question or
    - One tangible suggestion

Lecture

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

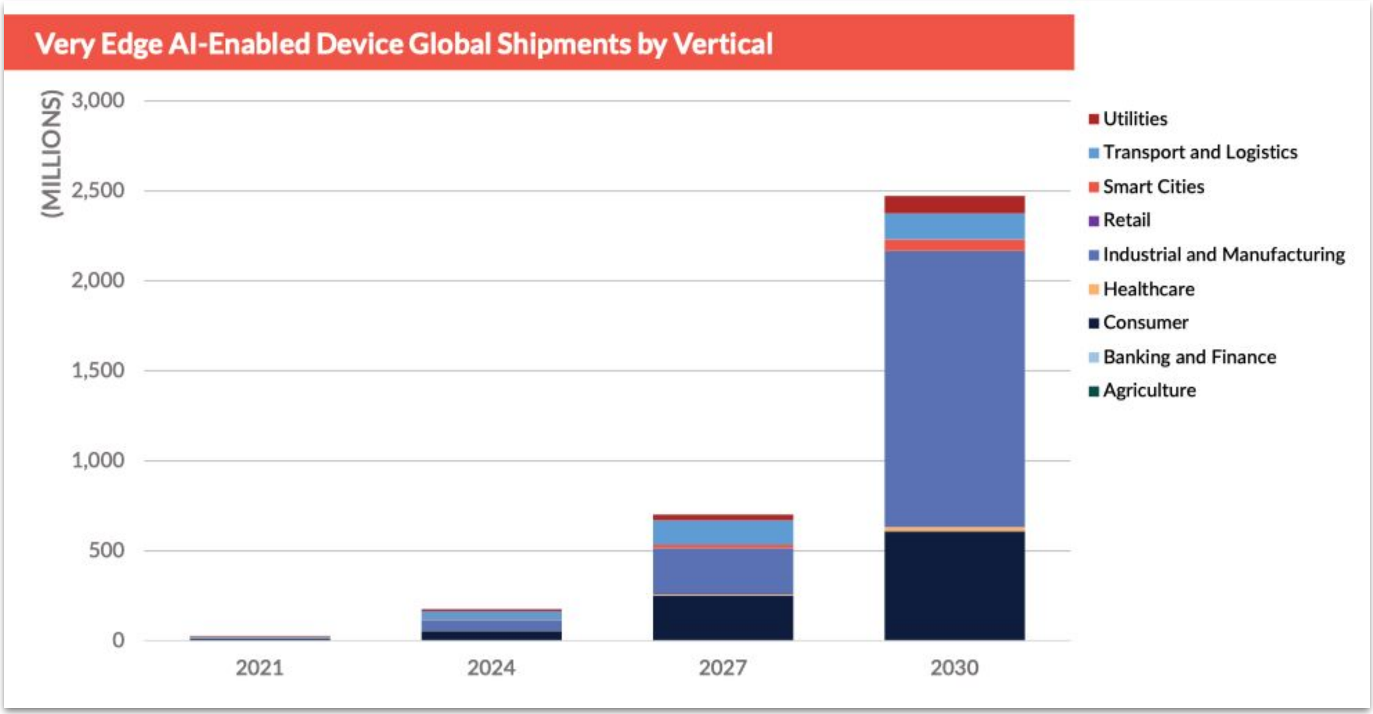
On-device sensor analytics

Low power consumption

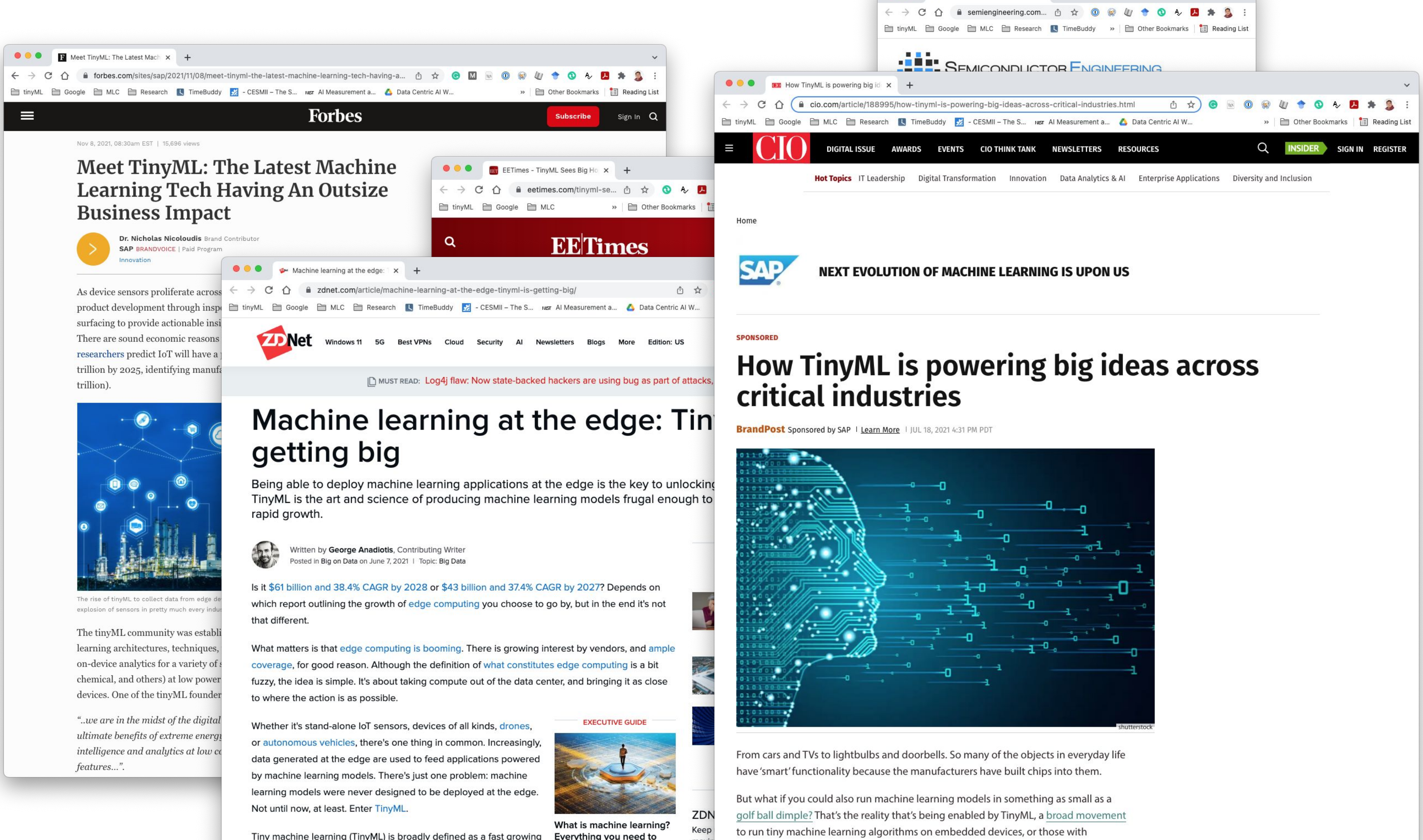
Always-on ML

Battery-operated

Market Forecast



Source: ABI Research: TinyML


forbes.com/sites/sap/2021/11/08/meet-tinyml-the-latest-machine-learning-tech-having-a...

Forbes

Nov 8, 2021, 08:30am EST | 15,696 views

Meet TinyML: The Latest Machine Learning Tech Having An Outsize Business Impact

Dr. Nicholas Nicoloudis Brand Contributor
SAP BRANDVOICE | Paid Program
Innovation

As device sensors proliferate across
product development through insp
surfacing to provide actionable insi
There are sound economic reasons
researchers predict IoT will have a
trillion by 2025, identifying manufa
trillion).

The rise of tinyML to collect data from edge de
explosion of sensors in pretty much every indu

The tinyML community was establi
learning architectures, techniques,
on-device analytics for a variety of
chemical, and others) at low power
devices. One of the tinyML founder

"..we are in the midst of the digital
ultimate benefits of extreme energ
intelligence and analytics at low c
features...".

eetimes.com/tinyml-se...

EETimes

zdnet.com/article/machine-learning-at-the-edge-tinyml-is-getting-big/

ZDNet

MUST READ: Log4j flaw: Now state-backed hackers are using bug as part of attacks,

Machine learning at the edge: Tin getting big

Being able to deploy machine learning applications at the edge is the key to unlocking
TinyML is the art and science of producing machine learning models frugal enough to
rapid growth.

Written by George Anadiotis, Contributing Writer
Posted in Big on Data on June 7, 2021 | Topic: Big Data

Is it $61 billion and 38.4% CAGR by 2028 or $43 billion and 37.4% CAGR by 2027? Depends on which report outlining the growth of edge computing you choose to go by, but in the end it's not that different.

What matters is that edge computing is booming. There is growing interest by vendors, and ample coverage, for good reason. Although the definition of what constitutes edge computing is a bit fuzzy, the idea is simple. It's about taking compute out of the data center, and bringing it as close to where the action is as possible.

Whether it's stand-alone IoT sensors, devices of all kinds, drones, or autonomous vehicles, there's one thing in common. Increasingly, data generated at the edge are used to feed applications powered by machine learning models. There's just one problem: machine learning models were never designed to be deployed at the edge. Not until now, at least. Enter TinyML.

Tiny machine learning (TinyML) is broadly defined as a fast growing

EXECUTIVE GUIDE

What is machine learning? Everything you need to

semiengineering.com...

Semiconductor Engineering

cio.com/article/188995/how-tinyml-is-powering-big-ideas-across-critical-industries.html

CIO

Hot Topics IT Leadership Digital Transformation Innovation Data Analytics & AI Enterprise Applications Diversity and Inclusion

Home

SAP NEXT EVOLUTION OF MACHINE LEARNING IS UPON US

SPONSORED

How TinyML is powering big ideas across critical industries

BrandPost Sponsored by SAP | Learn More | JUL 18, 2021 4:31 PM PDT

shutterstock

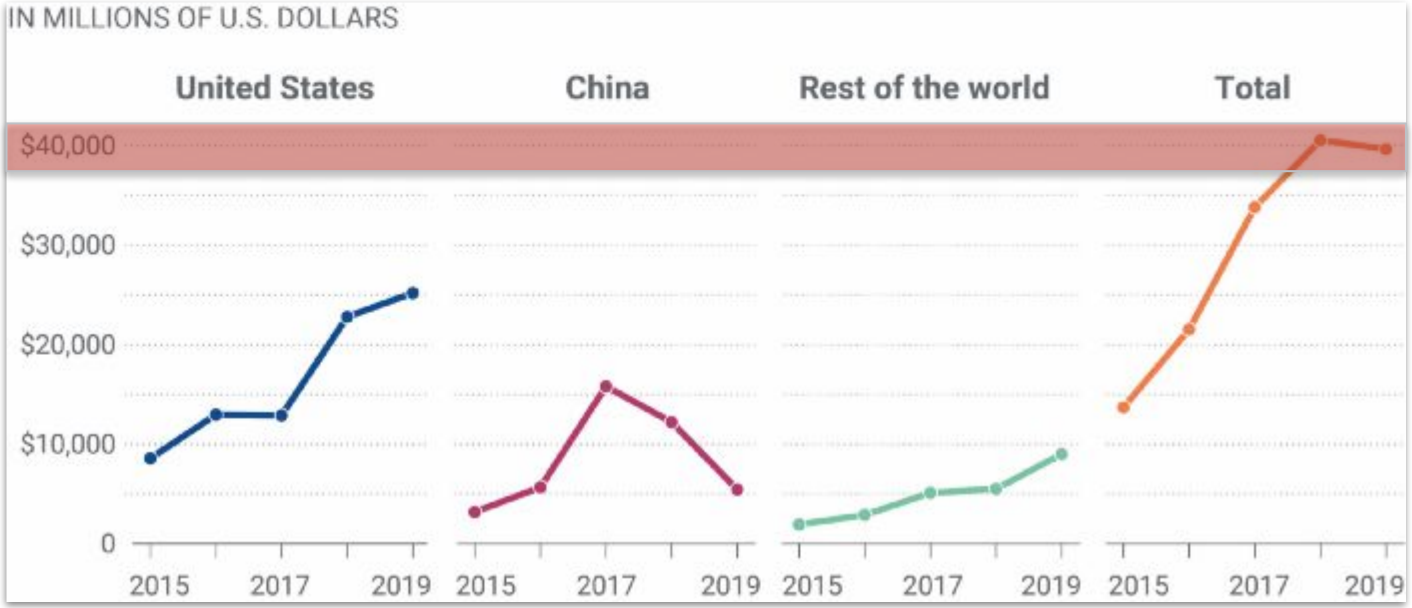
From cars and TVs to lightbulbs and doorbells. So many of the objects in everyday life have 'smart' functionality because the manufacturers have built chips into them.

But what if you could also run machine learning models in something as small as a golf ball dimple? That's the reality that's being enabled by TinyML, a broad movement to run tiny machine learning algorithms on embedded devices, or those with

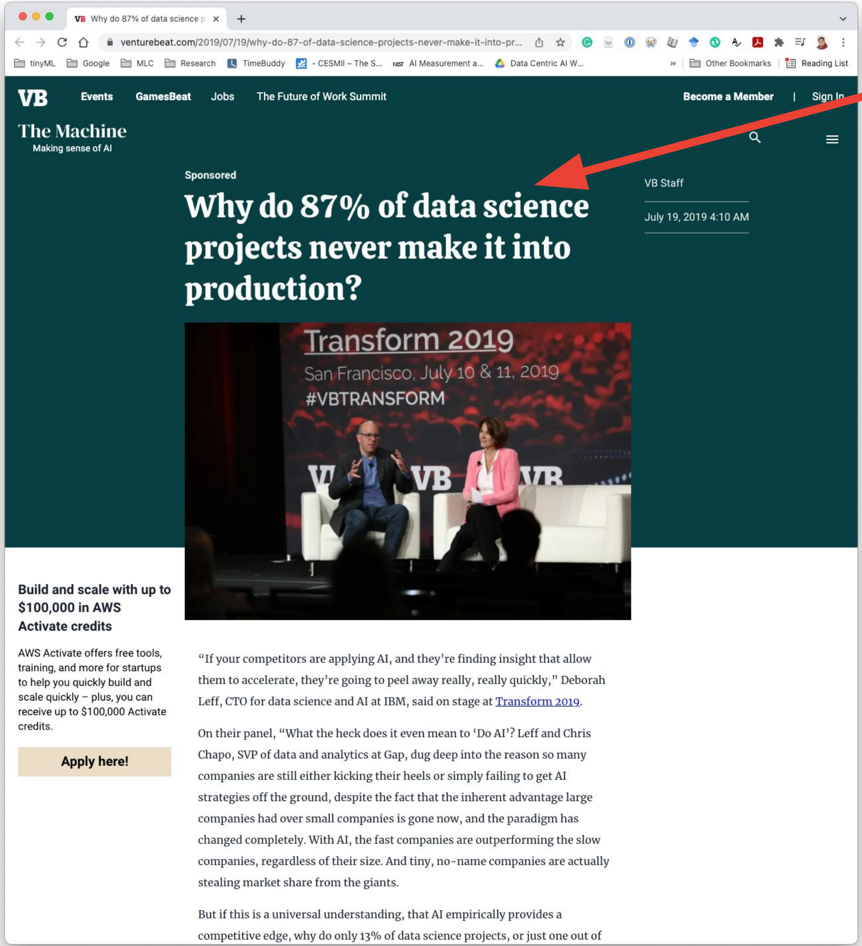
As device sensors proliferate across every company's value chain – from new product development through inspection, tracking, and delivery – tinyML is surfacing to provide actionable insights, transforming business as we know it. There are sound economic reasons for all this interest and activity. McKinsey researchers predict IoT will have a potential economic impact of US $4-11 trillion by 2025, identifying manufacturing as the largest vertical (US $1.2-3.7 trillion).

Source: https://www.forbes.com/sites/sap/2021/11/08/meet-tinyml-the-latest-machine-learning-tech-having-an-outsize-business-impact/

AI Investments



Source: Brookings Tech Stream



Sponsored

Why do 87% of data science projects never make it into production?

VB Staff

July 19, 2019 4:10 AM

“If your competitors are applying AI, and they’re finding insight that allow them to accelerate, they’re going to peel away really, really quickly,” Deborah Leff, CTO for data science and AI at IBM, said on stage at Transform 2019.

On their panel, “What the heck does it even mean to ‘Do AI’? Leff and Chris Chapo, SVP of data and analytics at Gap, dug deep into the reason so many companies are still either kicking their heels or simply failing to get AI strategies off the ground, despite the fact that the inherent advantage large companies had over small companies is gone now, and the paradigm has changed completely. With AI, the fast companies are outperforming the slow companies, regardless of their size. And tiny, no-name companies are actually stealing market share from the giants.

But if this is a universal understanding, that AI empirically provides a competitive edge, why do only 13% of data science projects, or just one out of

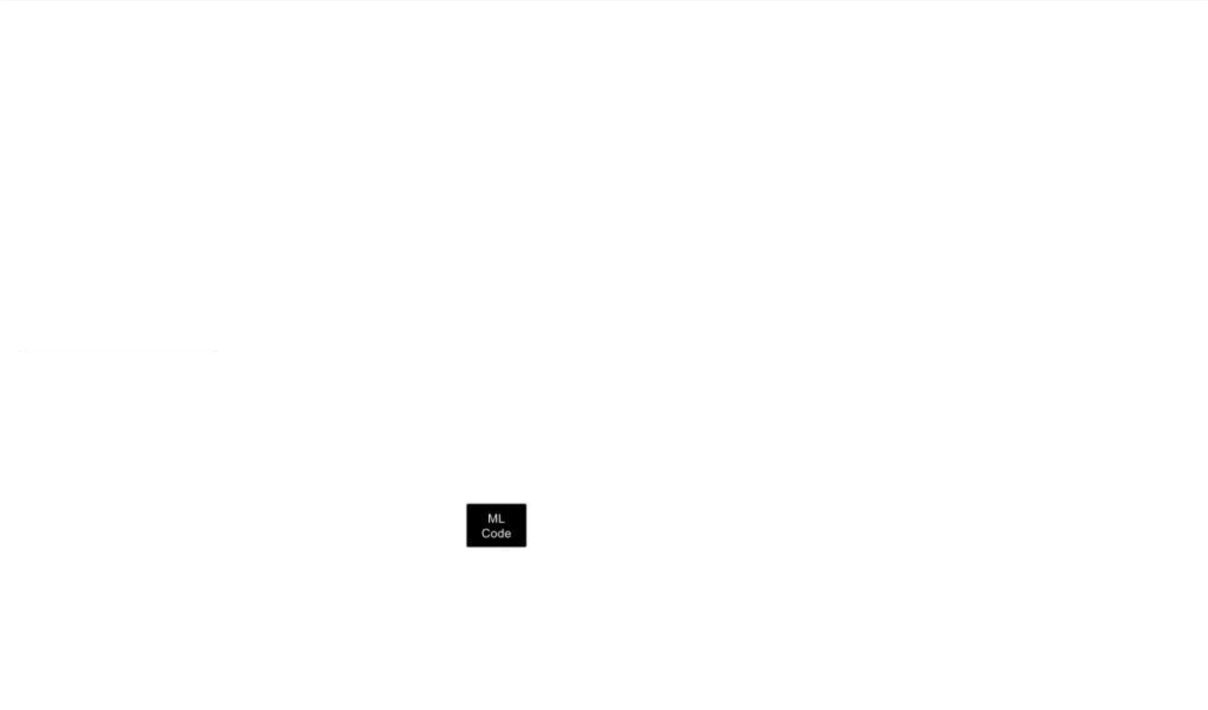
Predicts 2019: Analytics and BI Solutions

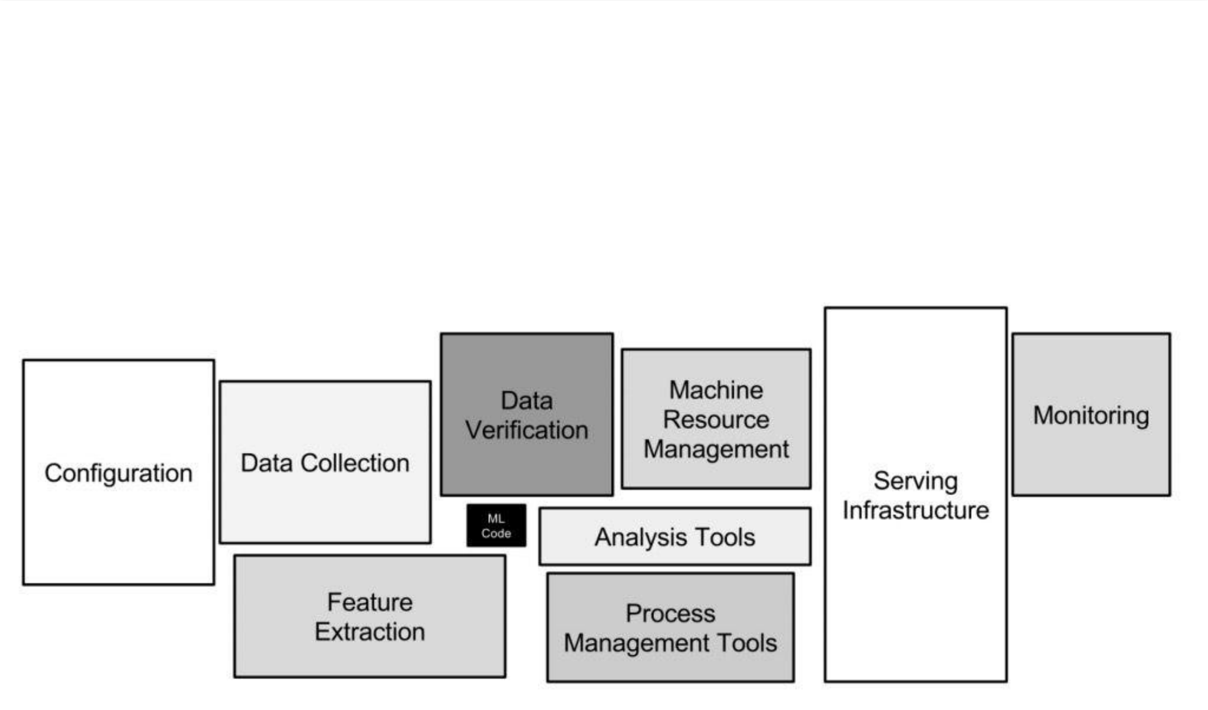
- Through 2020, 80% of AI projects will remain alchemy, run by wizards whose talents will not scale in the organization.
- Through 2022, only 20% of analytic insights will deliver business outcomes.
- By 2021, proof-of-concept analytic projects using quantum computing infrastructure will have outperformed traditional analytic approaches in multiple domains by at least a factor of 10

Source: https://blogs.gartner.com/andrew_white/2019/01/03/our-top-data-and-analytics-predicts-for-2019/

Let's quantify this a bit. In 2019 alone, approximately **USD 40 billions** were invested into privately held AI companies. If we extrapolate this and throw the approximated success rate of AI projects into these figures (and completely exclude intracompany ML investments), we reach the conclusion that in 2019, around **USD 38 billions were wasted due to unsuccessful Machine Learning projects.**

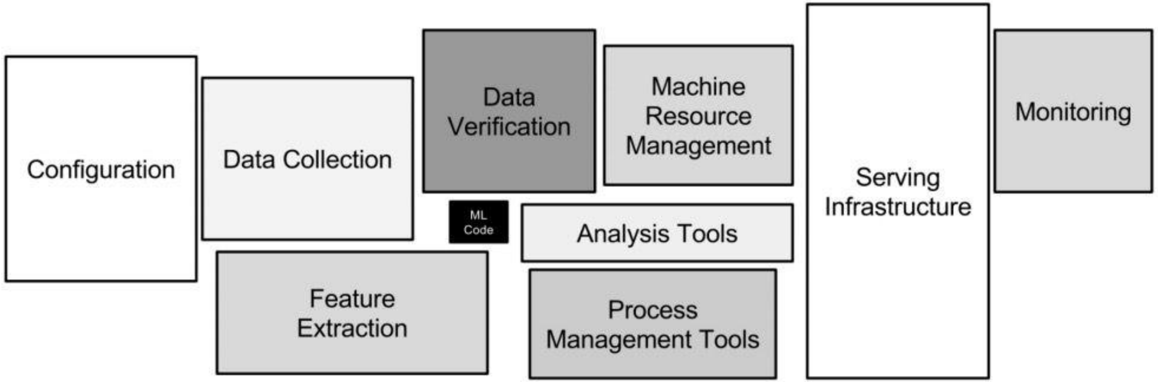
?


ML
Code

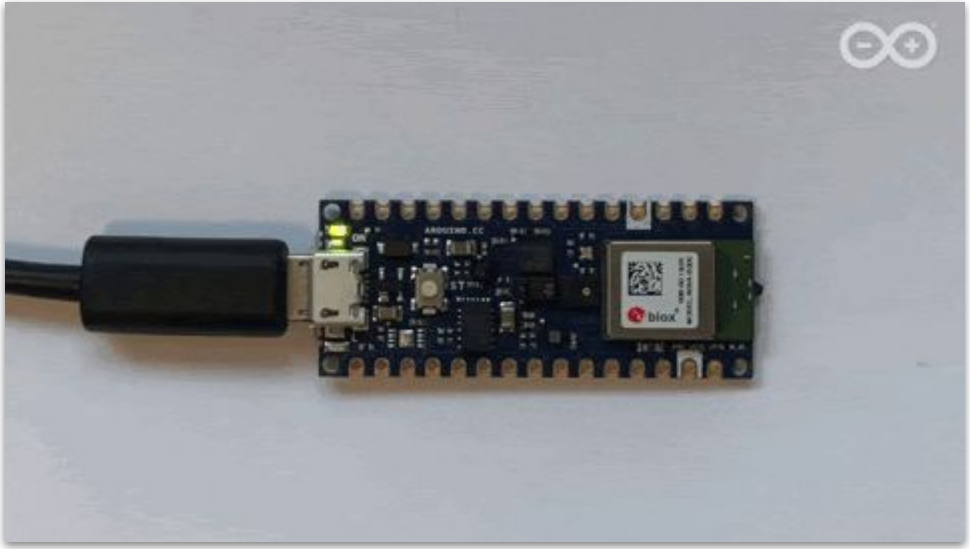

Configuration
Data Collection
Data Verification
Machine Resource Management
Serving Infrastructure
Monitoring
ML Code
Analysis Tools
Feature Extraction
Process Management Tools

Hidden Technical Debt in Machine Learning Systems

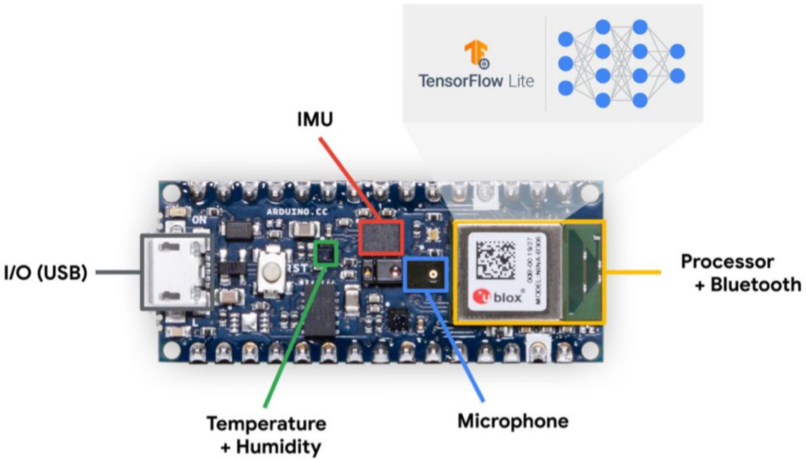
D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.



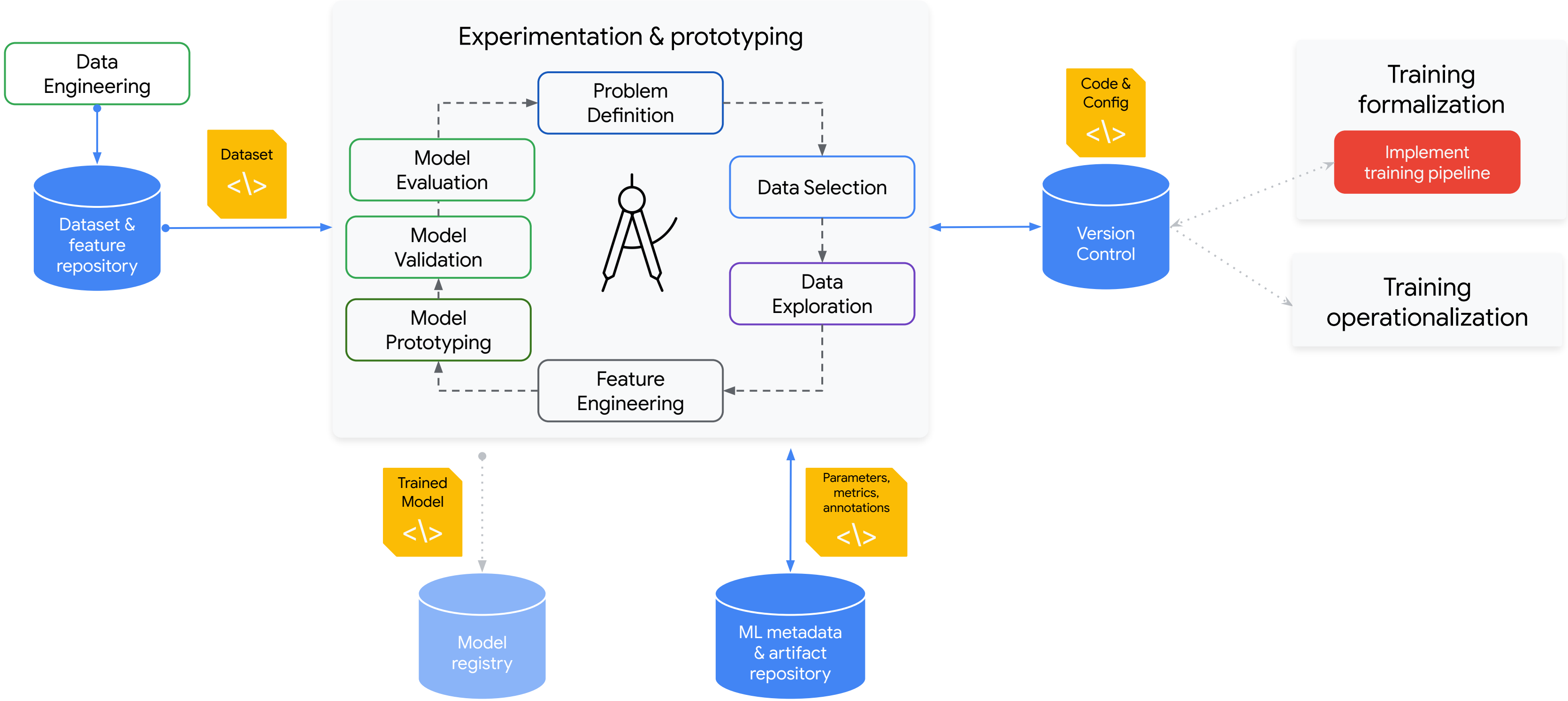
Missing the Forest for the Trees

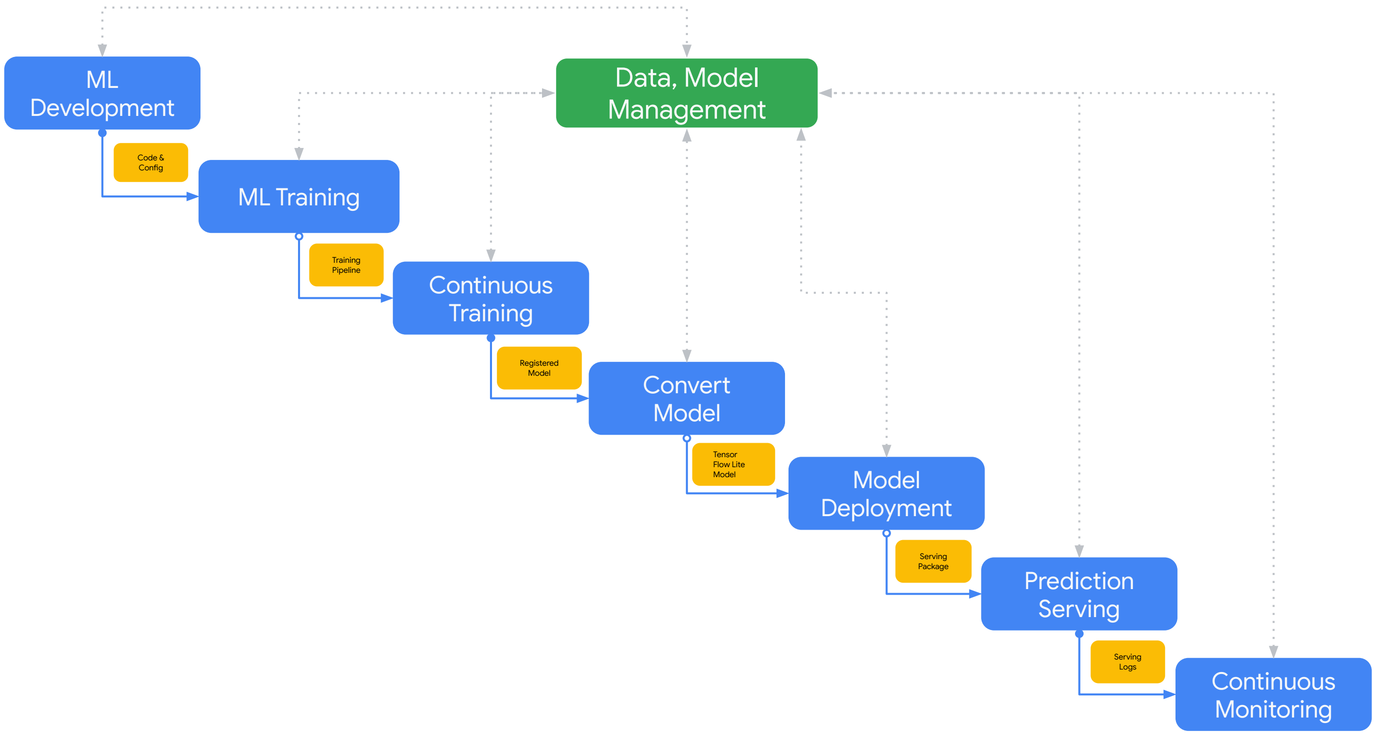

ARDUINO.CC
ON
blox


ARDUINO
TINY MACHINE
LEARNING KIT
edX
TensorFlow

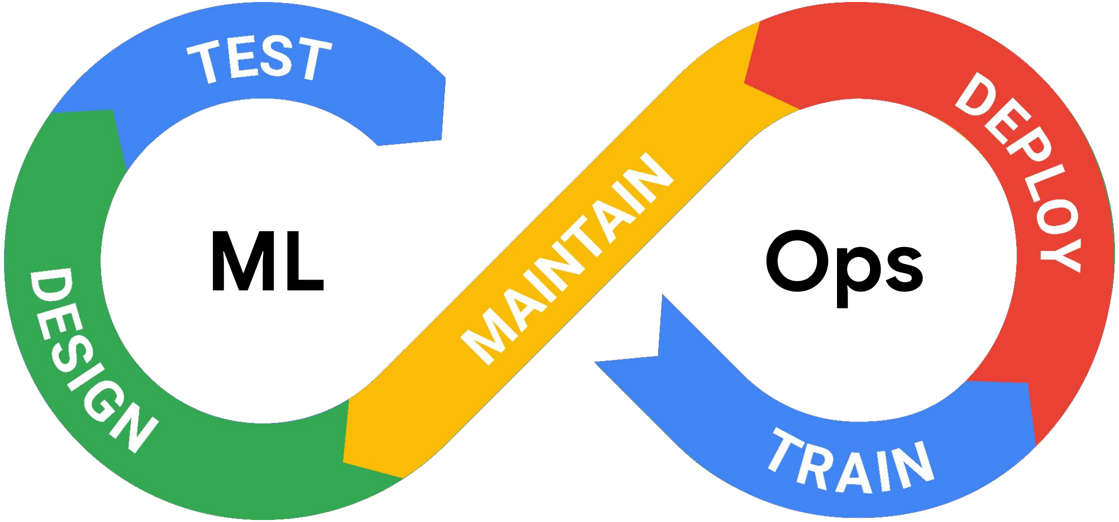

TensorFlow Lite
IMU
I/O (USB)
Processor
+ Bluetooth
Temperature
+ Humidity
Microphone

ML Development

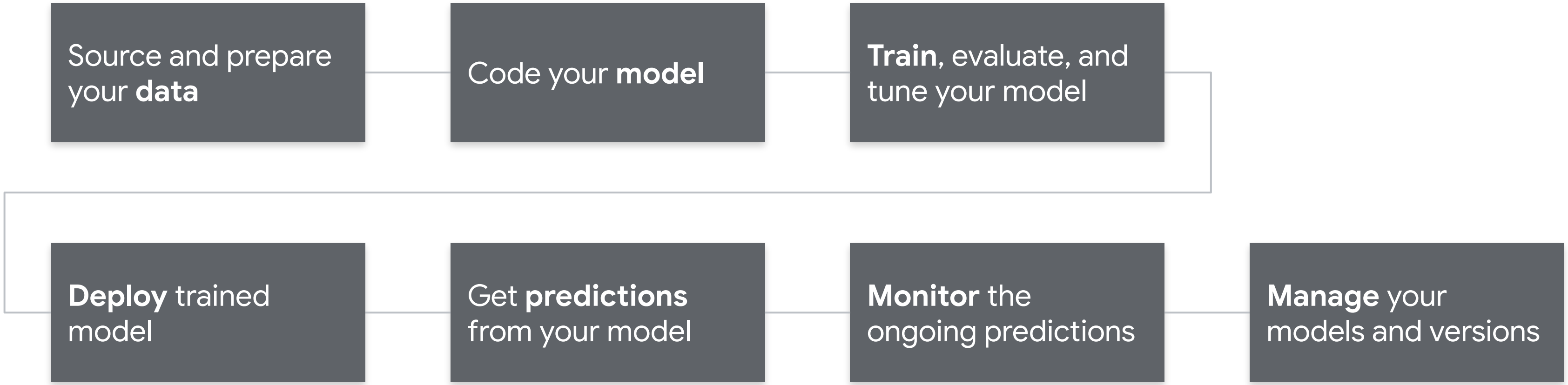



ML Development
Code & Config
ML Training
Training Pipeline
Continuous Training
Registered Model
Convert Model
Tensor Flow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring
Data, Model Management

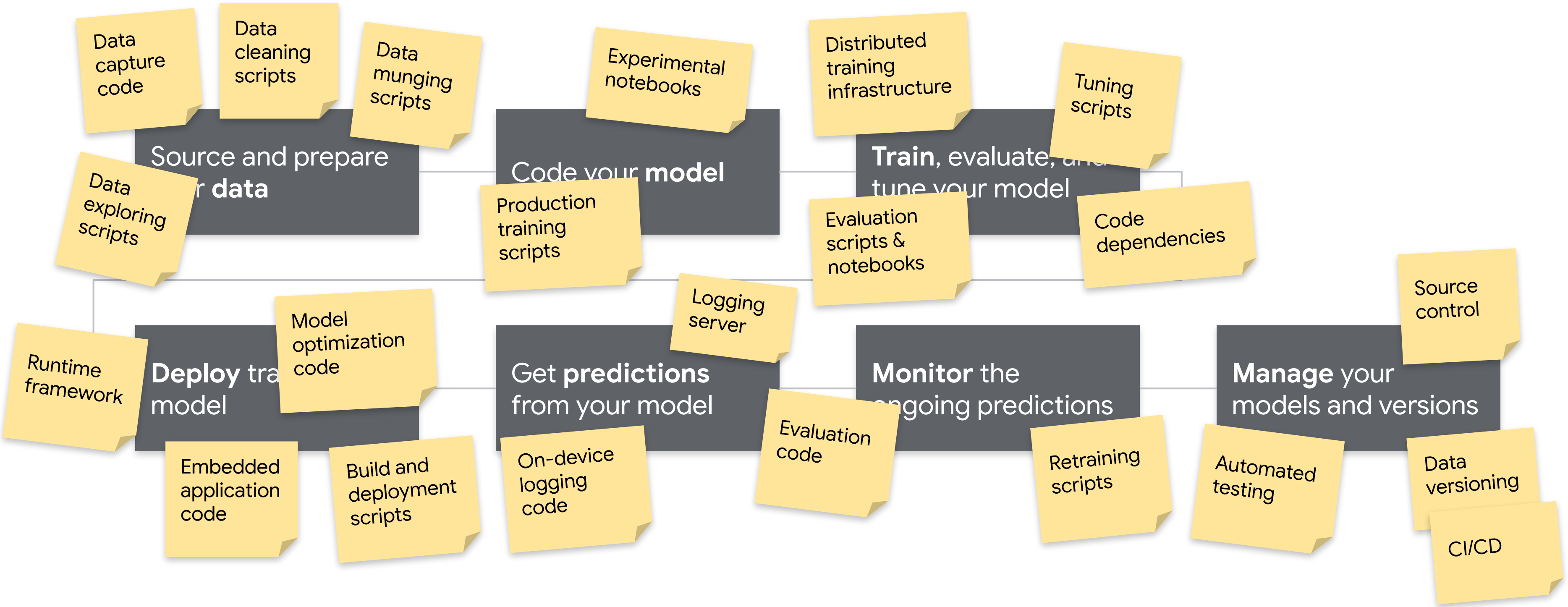
MLOps


ML
Ops
TEST
DESIGN
MAINTAIN
DEPLOY
TRAIN

The Machine Learning Workflow



The Machine Learning Workflow

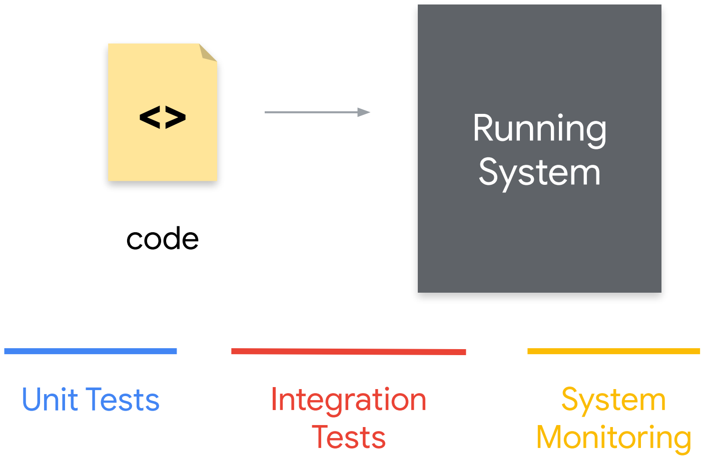


MLOps = ML Workflow + Automation

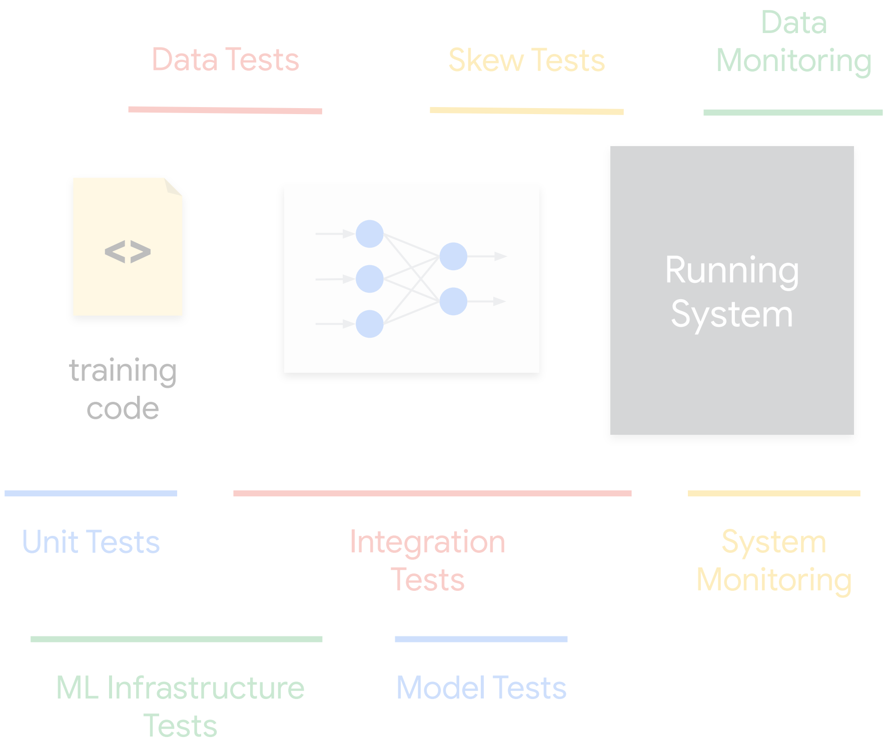
MLOps means...

5 Focus Areas

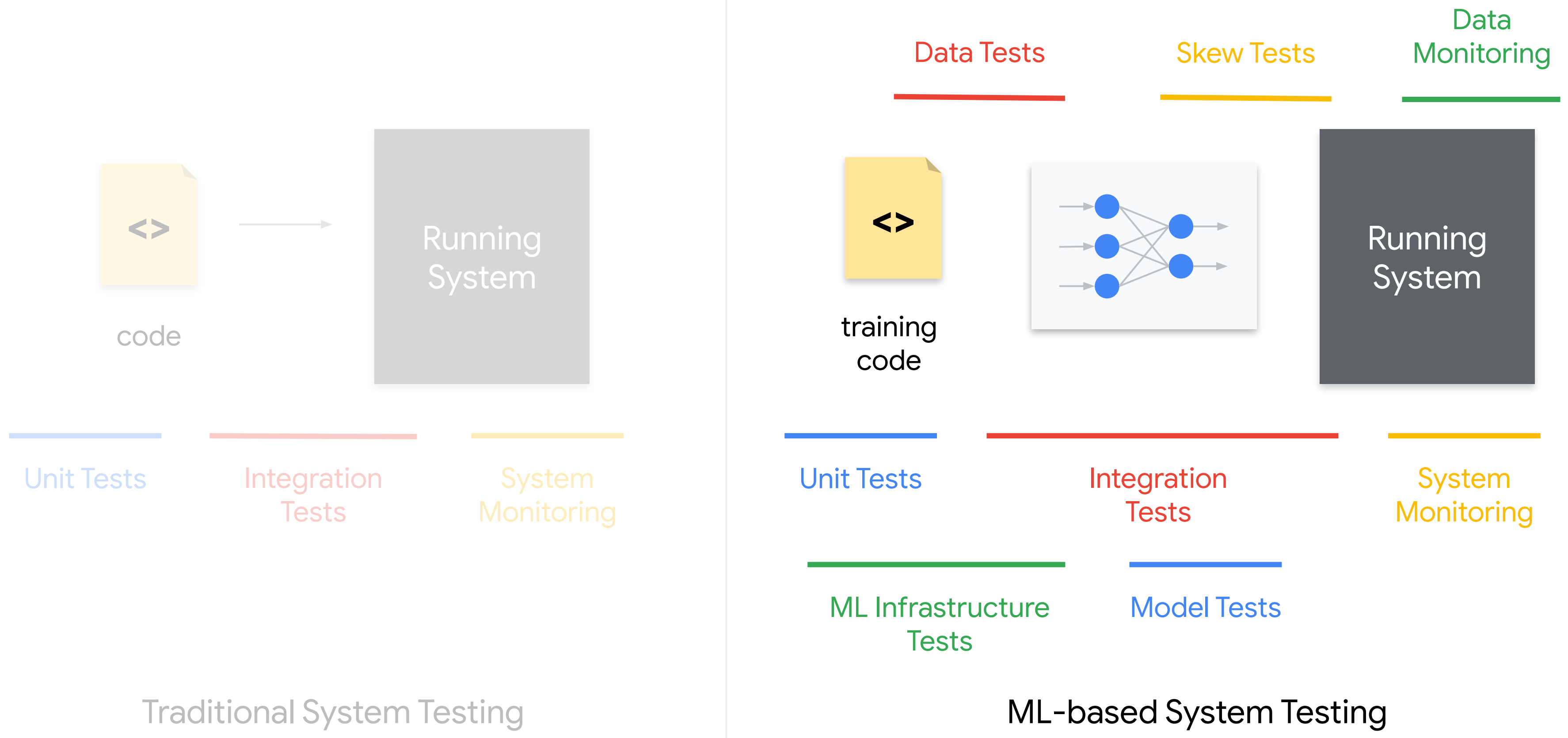
Running end-to-end	**Managing Complexity**	**Evaluating Results**	**Improving Models**	**Tracking deployments**



Traditional System Testing



ML-based System Testing



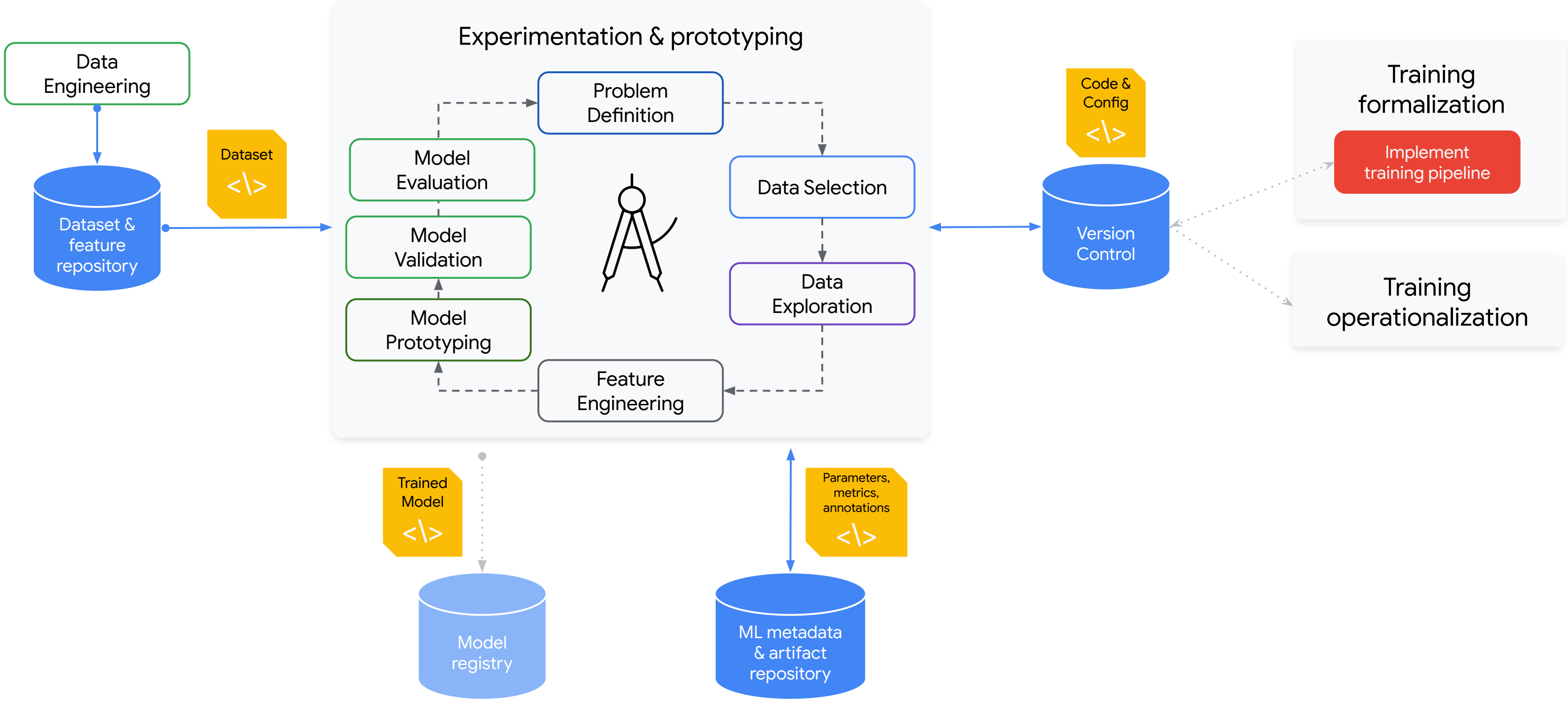
code
Running System
Unit Tests
Integration Tests
System Monitoring
Traditional System Testing
Data Tests
Skew Tests
Data Monitoring
training code
Running System
Unit Tests
Integration Tests
System Monitoring
ML Infrastructure Tests
Model Tests
ML-based System Testing


Data & model management
ML development
Training operationalization
Continuous training
Model deployment
Prediction serving
Continuous monitoring

ML Development

ML development entails experimenting with and establishing a dependable and repeatable model training procedure.

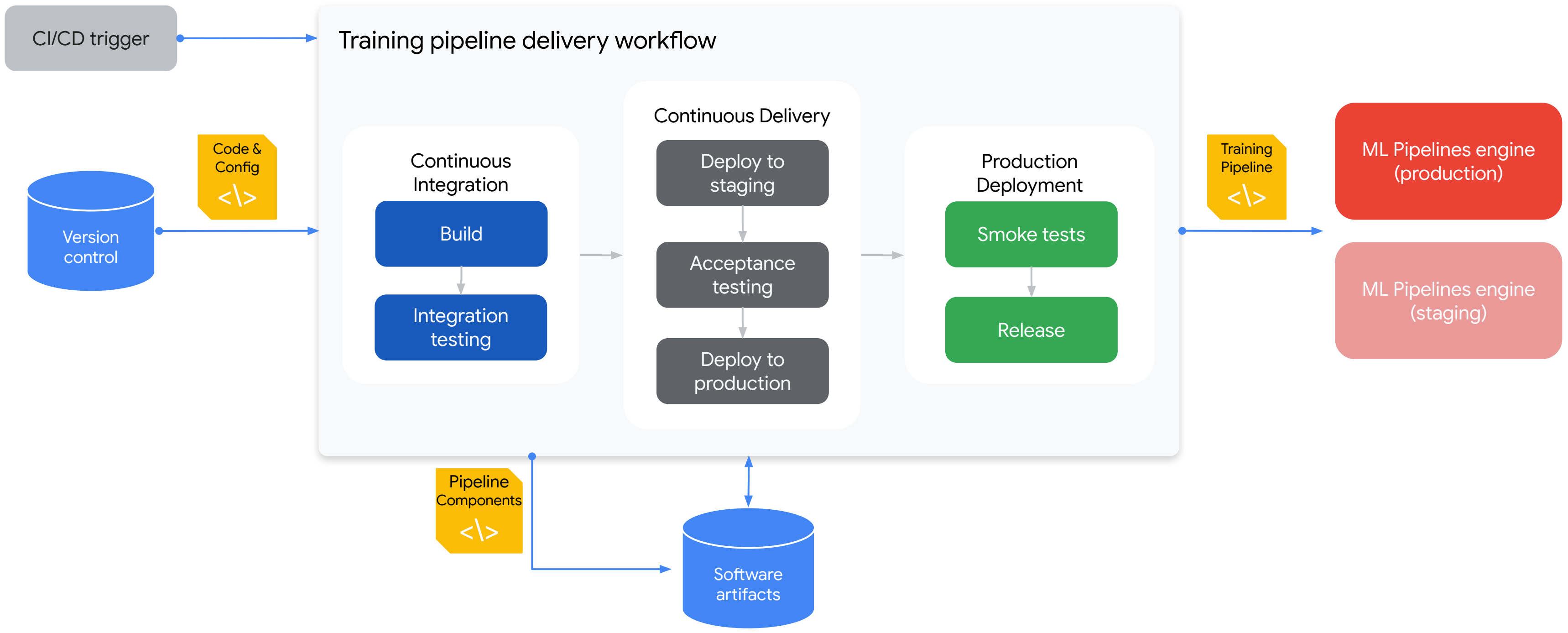
MLOps: ML Development



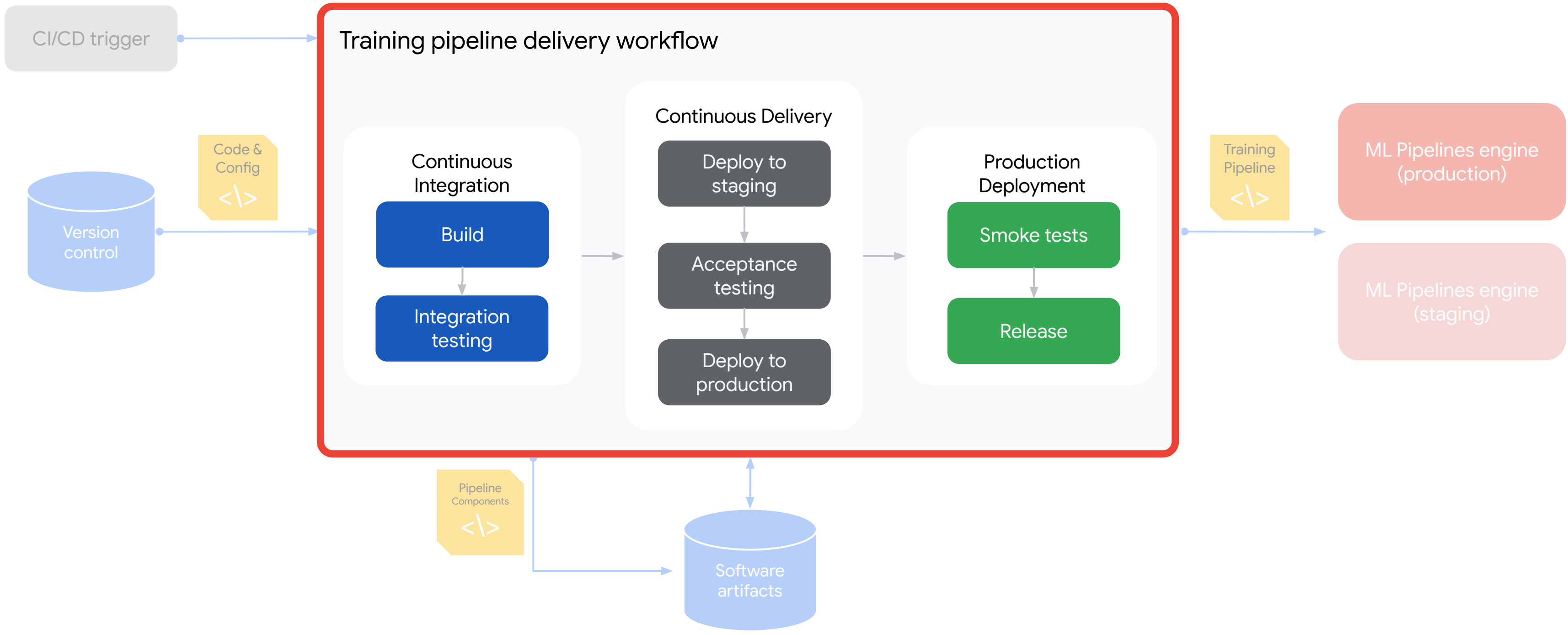
Training Operationalization

Training operationalization is all about automating the packaging, testing, and deployment of repeatable and dependable training pipelines.

MLOps: Training Operationalization

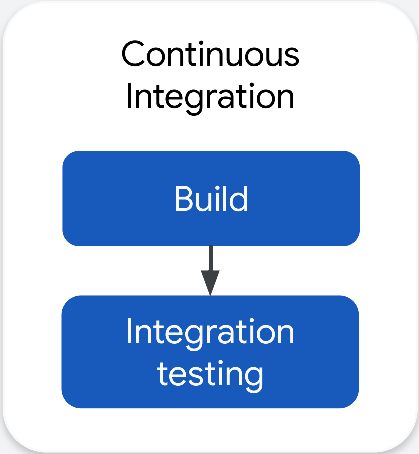


MLOps: Training Operationalization



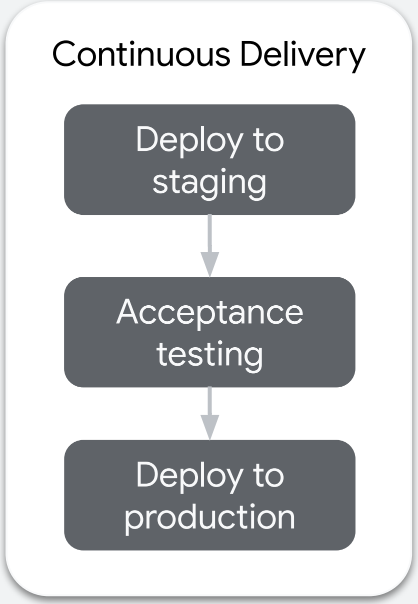
TinyML CI Questions

- What does the **build environment** look like?
- What **types of assets** do I need to consider writing for testing?



TinyML CD Questions

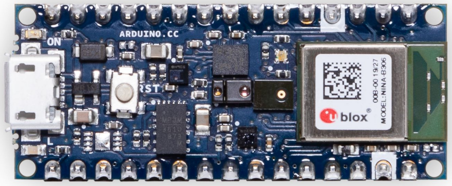
- What does the **staging environment** look like?
- What is considered as **accepted testing**?
- What and **how do I deploy** into production?

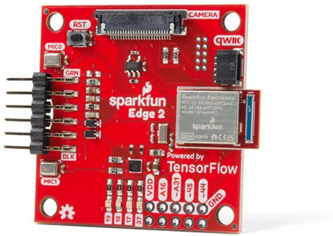


TinyML Production Deployment Questions

- What does it mean to do a **smoke test** for embedded machine learning systems?
- How can you do a **production release** with TinyML devices?

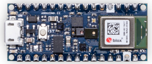
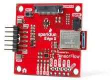
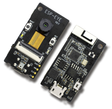

Himax


ARDUINO.CC
ON
RST
blox
MODEL:NINA-B306


CAMERA
RST
qwiic
sparkfun
Edge 2
Powered by
TensorFlow


ESP-EYE
V2.1

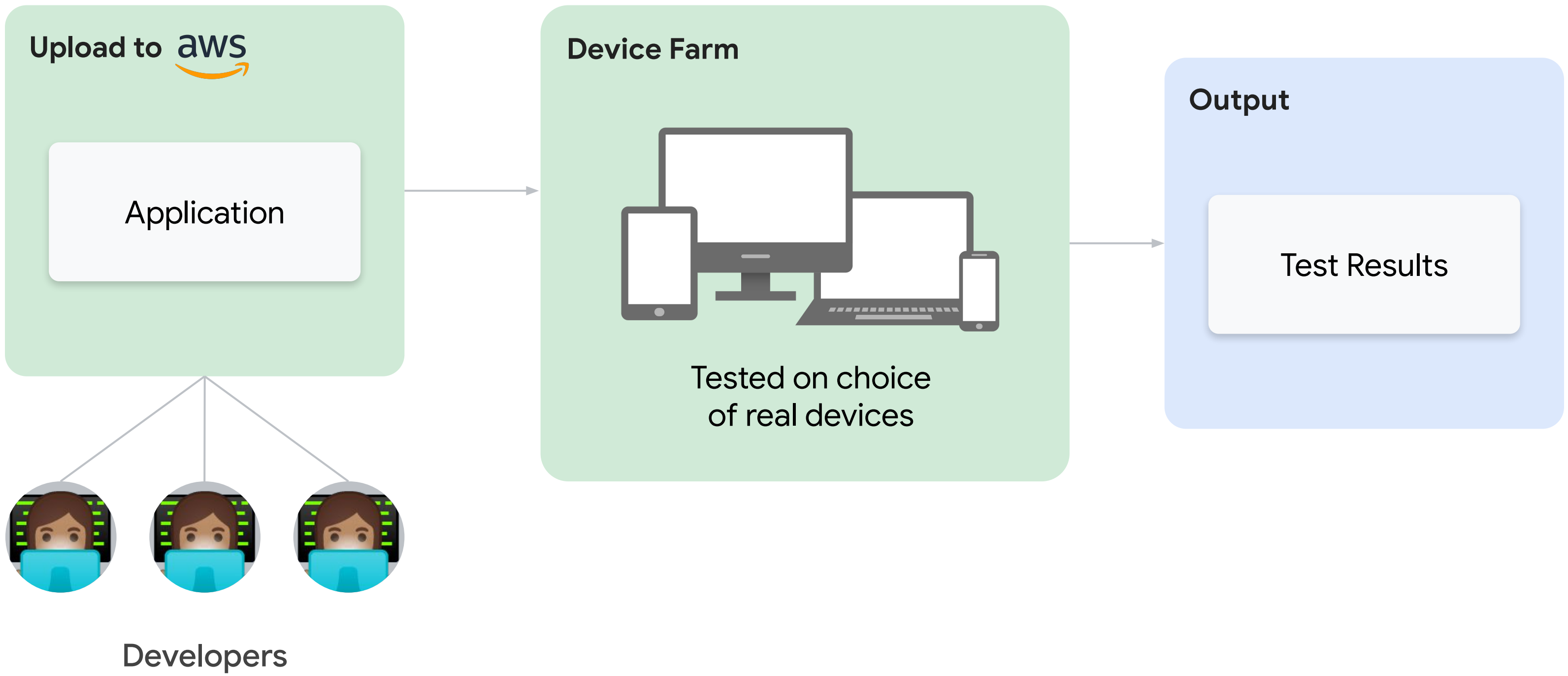
Deployment Challenges

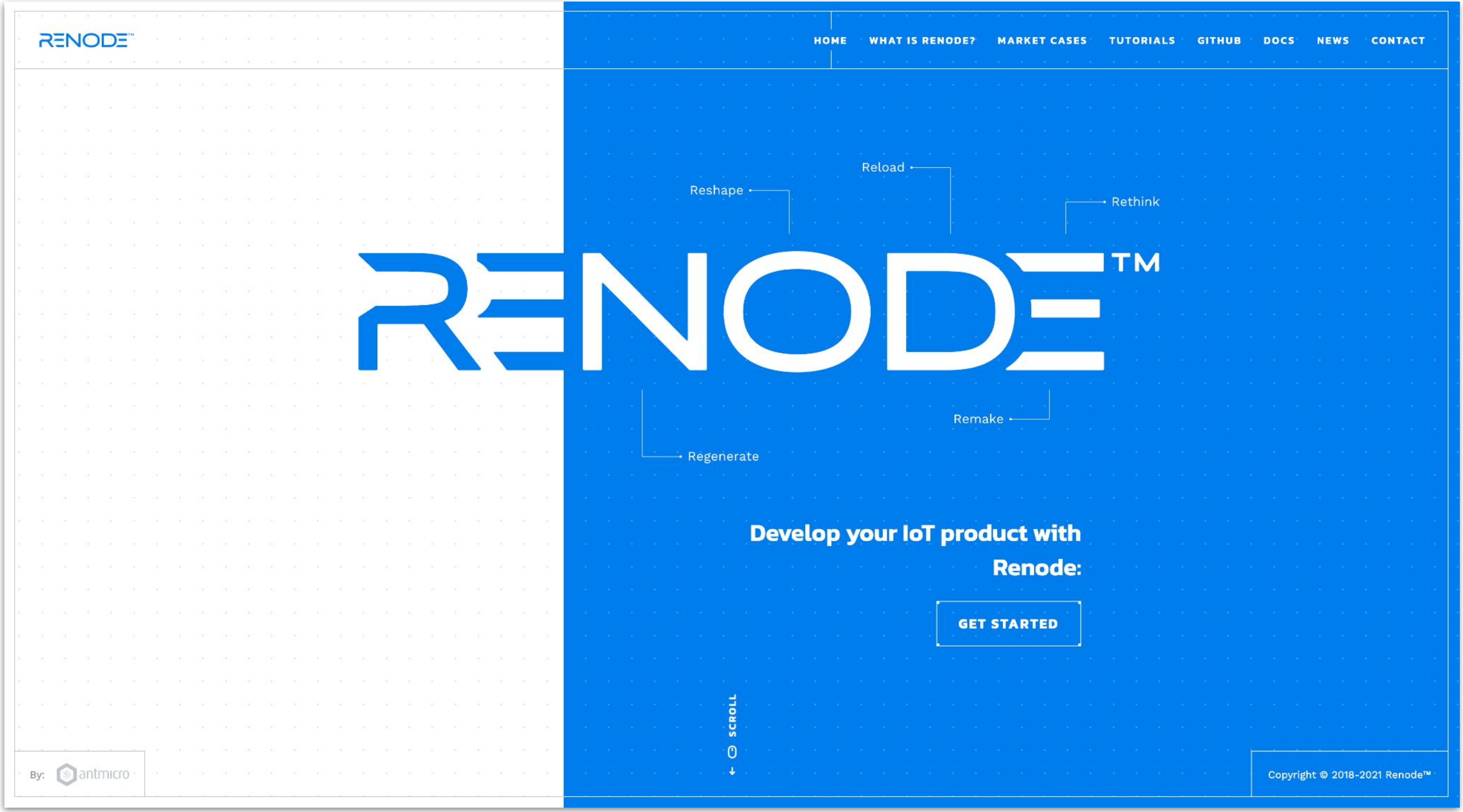





Board	MCU / ASIC	Clock	Memory	Sensors	Radio
Himax WE-I Plus EVB	HX6537-A 32-bit EM9D DSP	400 MHz	2MB flash 2MB RAM	Accelerometer, Mic, Camera	None
Arduino Nano 33 BLE Sense	32-bit nRF52840	64 MHz	1MB flash 256kB RAM	Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color	BLE
SparkFun Edge 2	32-bit ArtemisV1	48 MHz	1MB flash 384kB RAM	Accelerometer, Mic, Camera	BLE
Espressif EYE	32-bit ESP32-D0WD	240 MHz	4MB flash 520kB RAM	Mic, Camera	WiFi, BLE


100X
SAMSUNG

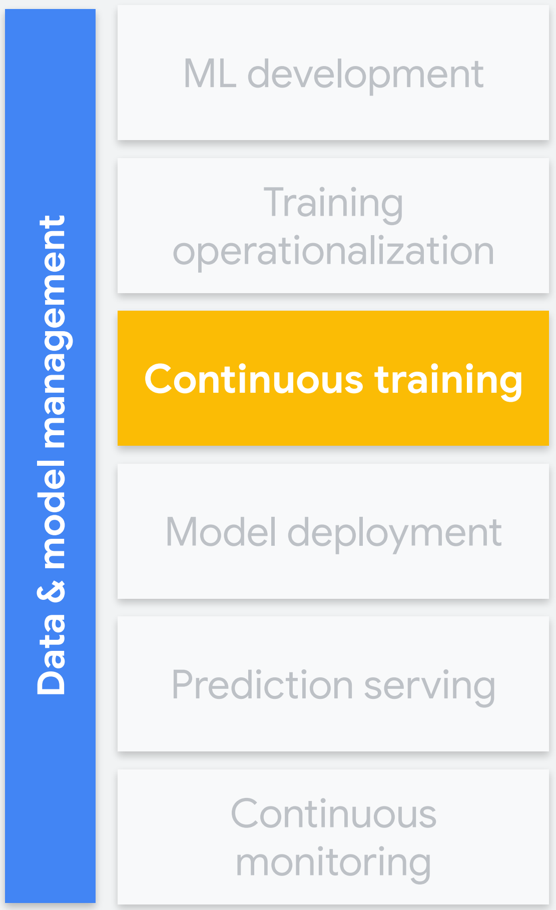
Device Farm


Upload to aws
Application
Developers
Device Farm
Tested on choice
of real devices
Output
Test Results

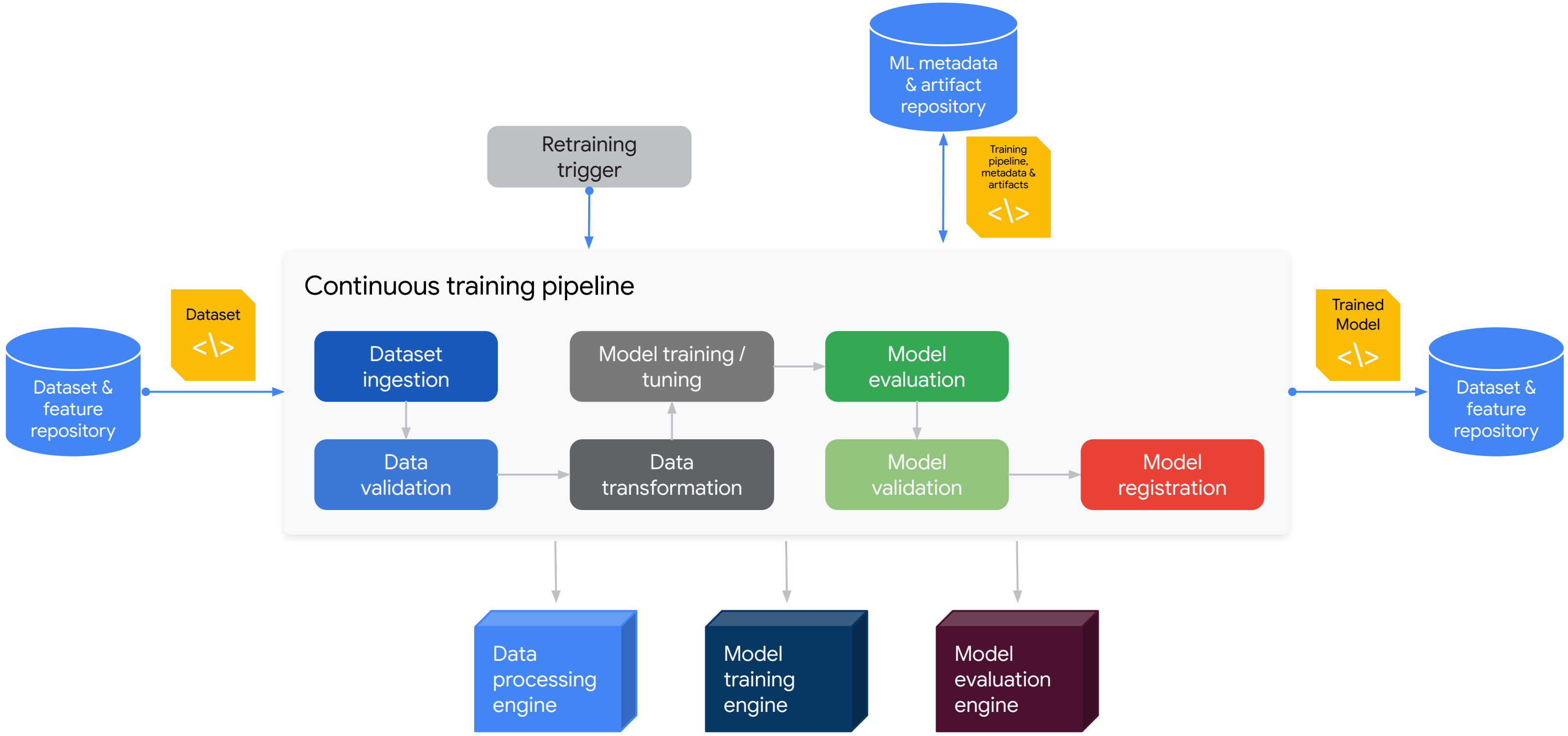

RENODE™
HOME
WHAT IS RENODE?
MARKET CASES
TUTORIALS
GITHUB
DOCS
NEWS
CONTACT
Reload
Reshape
Rethink
RENODE™
Remake
Regenerate
Develop your IoT product with Renode:
GET STARTED
SCROLL
By: antmicro
Copyright © 2018-2021 Renode™

Continuous Training

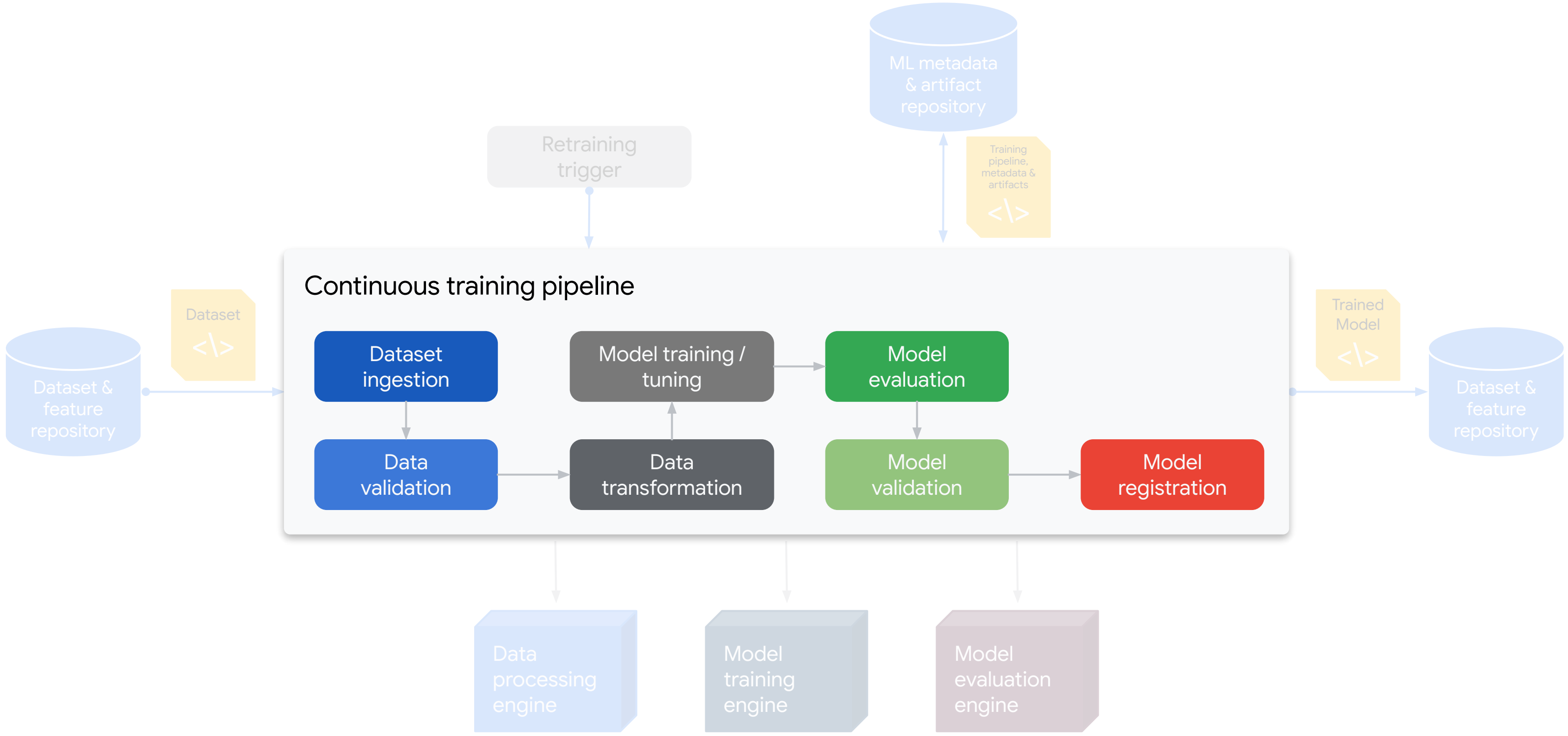
Continuous training entails running the training pipeline on a regular basis, maybe with fresh training settings, in response to new data or code modifications.



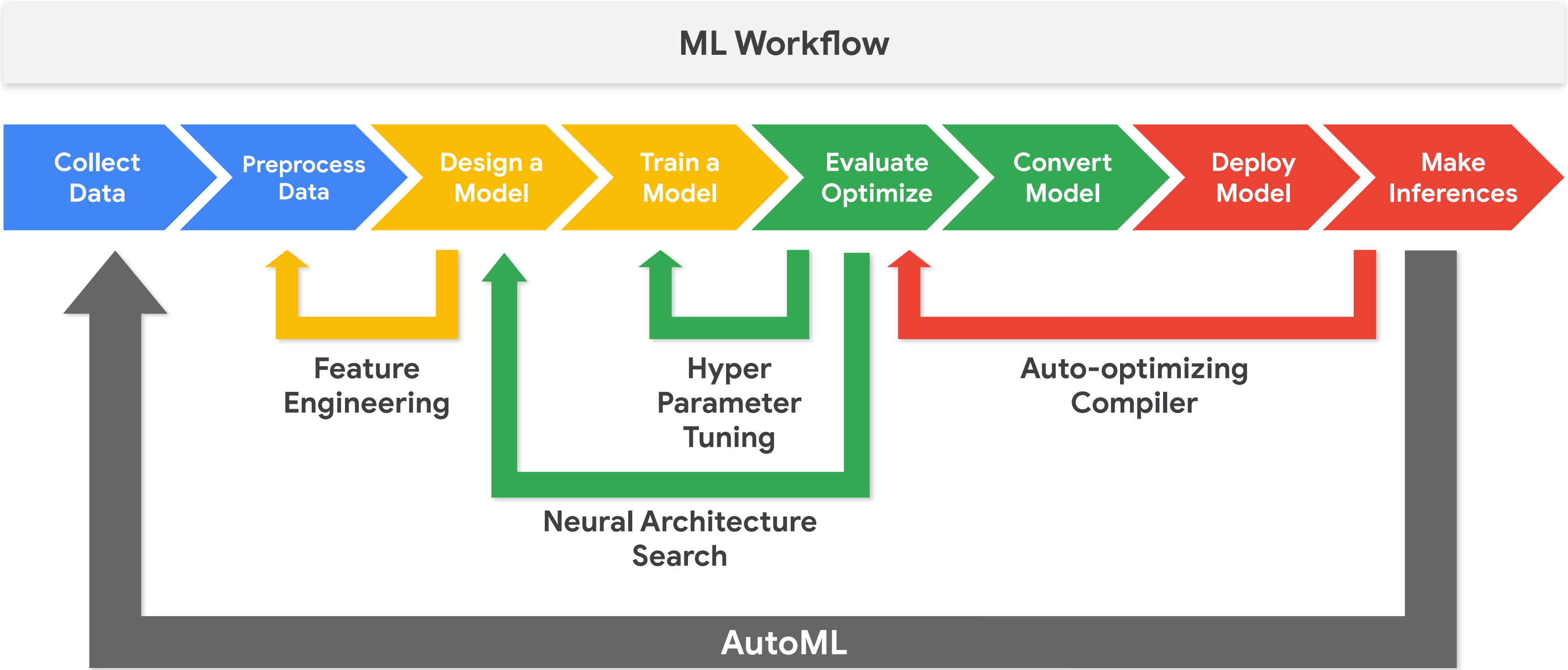
MLOps: Continuous Training



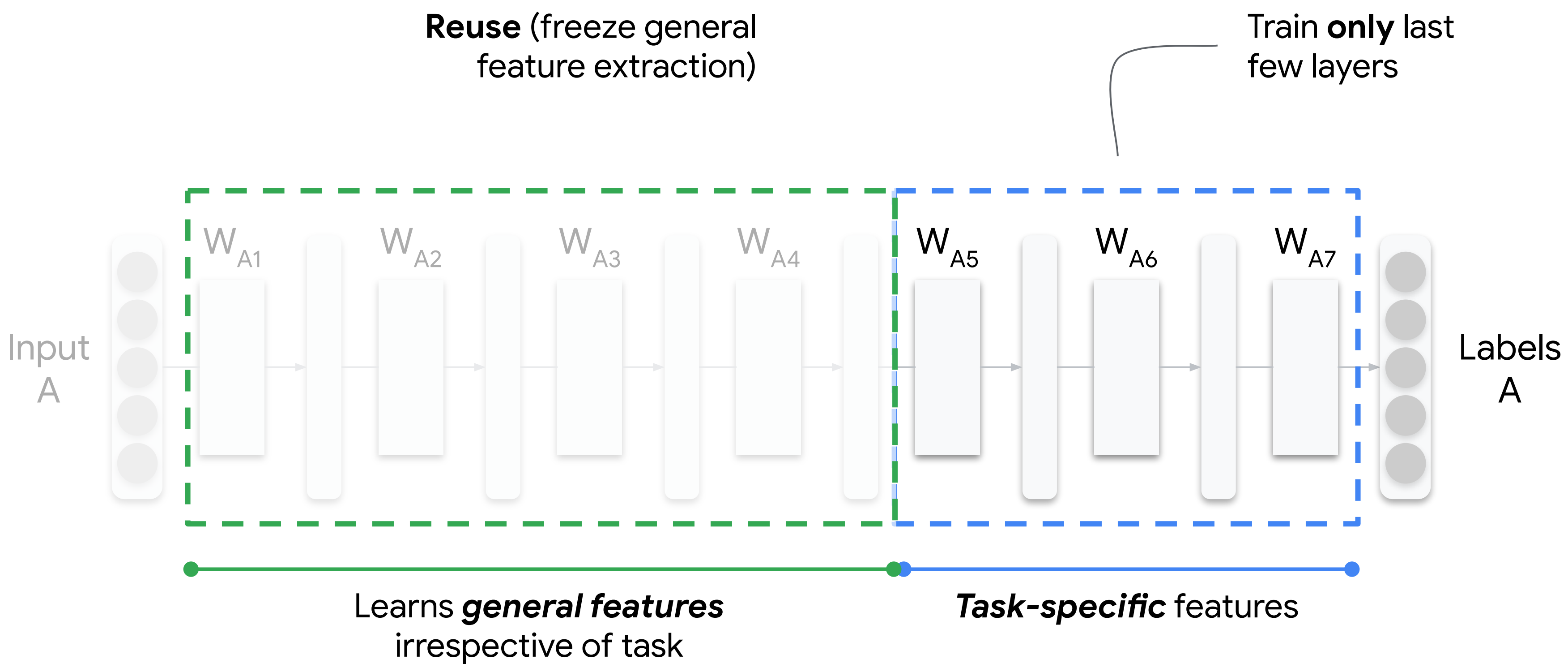
MLOps: Continuous Training



What are the **components** of AutoML?



Transfer Learning



Google Cloud
AutoML

Qeexo

Intelligent Agent
Neuton

Existing Solutions

EDGE IMPULSE

Eon
Tuner

"Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, Lora Aroyo
[nithyasamba,kapania,hhighfill,dakrong,pkp,loraa]@google.com
Google Research
Mountain View, CA

ABSTRACT

AI models are increasingly applied in high-stakes domains like health and conservation. Data quality carries an elevated significance in high-stakes AI due to its heightened downstream impact, impacting predictions like cancer detection, wildlife poaching, and loan allocations. Paradoxically, data is the most under-valued and de-glamorised aspect of AI. In this paper, we report on data practices in high-stakes AI, from interviews with 53 AI practitioners in India, East and West African countries, and USA. We define, identify, and present empirical evidence on *Data Cascades*—compounding events causing negative, downstream effects from data issues—triggered by conventional AI/ML practices that undervalue data quality. Data cascades are pervasive (92% prevalence), invisible, delayed, but often avoidable. We discuss HCI opportunities in designing and incentivizing data excellence as a first-class citizen of AI, resulting in safer and more robust systems for all.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in HCI.**

KEYWORDS

Data, AI, ML, high-stakes AI, data cascades, developers, raters, application-domain experts, data collectors, data quality, data politics, India, Nigeria, Kenya, Ghana, Uganda, USA

ACM Reference Format:

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, Lora Aroyo. 2021. *"Everyone wants to do the model work, not the data work"*: Data Cascades in High-Stakes AI . In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3411764.3445518

1 INTRODUCTION

Data is the critical infrastructure necessary to build Artificial Intelligence (AI) systems [44]. Data largely determines performance, fairness, robustness, safety, and scalability of AI systems [44, 81]. Paradoxically, for AI researchers and developers, data is often the least incentivized aspect, viewed as 'operational' relative to the lionized work of building novel models and algorithms [46, 125]. Intuitively, AI developers understand that data quality matters, often spending inordinate amounts of time on data tasks [60]. In practice, most organisations fail to create or meet any data quality standards [87], from under-valuing data work vis-a-vis model development.

Under-valuing of data work is common to all of AI development [125][1]. We pay particular attention to undervaluing of data in *high-stakes domains*[2] that have safety impacts on living beings, due to a few reasons. One, developers are increasingly deploying AI models in complex, humanitarian domains, *e.g.*, in maternal health, road safety, and climate change. Two, poor data quality in high-stakes domains can have outsized effects on vulnerable communities and contexts. As Hiatt *et al.* argue, high-stakes efforts are distinct from serving customers; these projects work with and for populations at risk of a litany of horrors [47]. As an example, poor data practices reduced accuracy in IBM's cancer treatment AI [115] and led to Google Flu Trends missing the flu peak by 140% [63, 73]). Three, high-stakes AI systems are typically deployed in low-resource contexts with a pronounced lack of readily available, high-quality datasets. Applications span into communities that live outside of a modern data infrastructure, or where everyday functions are not yet consistently tracked, *e.g.*, walking distances to gather water in rural areas—in contrast to, say, click data [26]. Finally, high-stakes AI is more often created at the combination of two or more disciplines; for example, AI and diabetic retinopathy, leading to greater collaboration challenges among stakeholders across organizations and domains [75, 121].

Considering the above factors, currently data quality issues in AI are addressed with the wrong tools created for, and fitted to other technology problems—they are approached as a database problem, legal compliance issue, or licensing deal. HCI and CSCW scholarship have long examined the practices of collaboration, problem formulation, and sensemaking, by humans behind the datasets, including data collectors and scientists, [69, 86, 127], and are designing computational artefacts for dataset development [53]. Our research extends this scholarship by empirically examining data practices and challenges of high-stakes AI practitioners impacting vulnerable groups.

We report our results from a qualitative study on practices and structural factors among 53 AI practitioners in India, the US, and

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05.
https://doi.org/10.1145/3411764.3445518

[1] Data work is broadly under-valued in many sociotechnical domains like [58, 85]

[2] We extend the vision of *AI for Social Good* (*i.e.*, using AI for social and environmental impact) and *Data for Good* (*i.e.*, providing data and education to benefit non-profit or government agencies) with *AI for high-stakes domains* involving safety, well-being and stakes (*e.g.*, road safety, credit assessment).

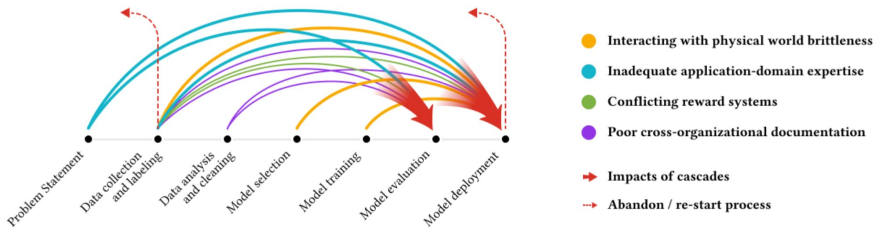
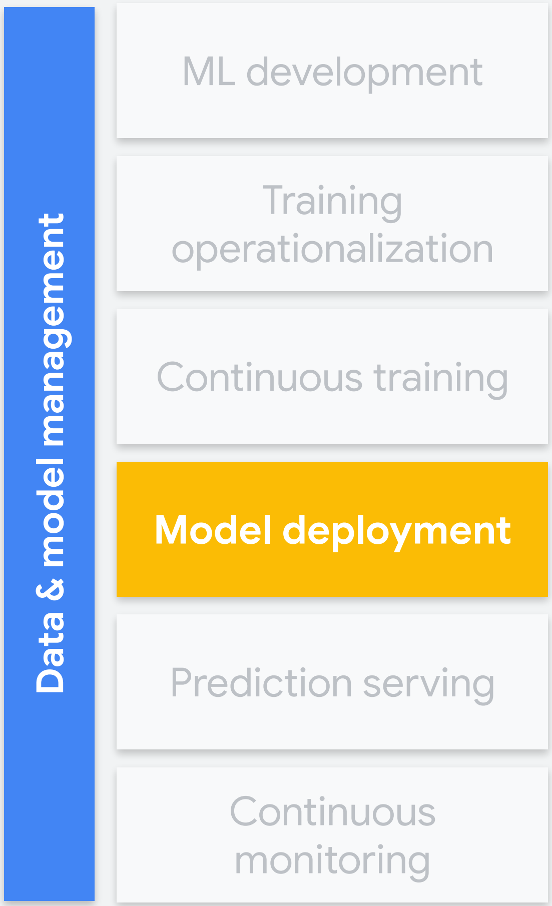


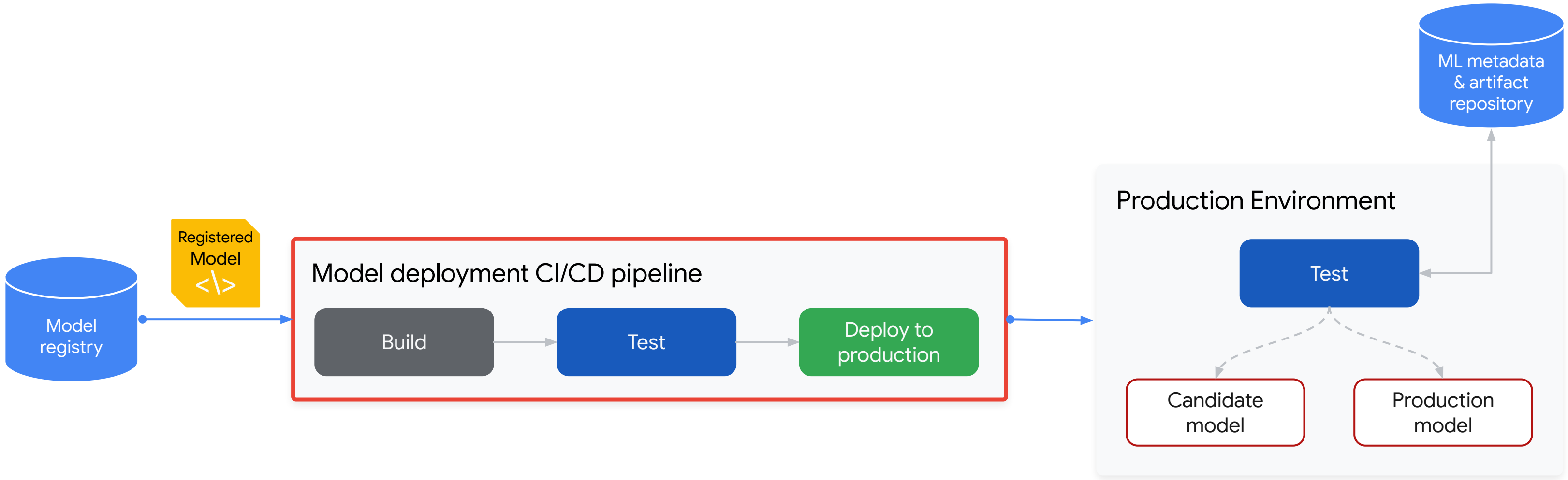
Figure 1: Data cascades in high-stakes AI. Cascades are opaque and protracted, with multiplied, negative impacts. Cascades are triggered in the upstream (*e.g.,* data collection) and have impacts on the downstream (*e.g.,* model deployment). Thick red arrows represent the compounding effects after data cascades start to become visible; dotted red arrows represent abandoning or re-starting of the ML data process. Indicators are mostly visible in model evaluation, as system metrics, and as malfunctioning or user feedback.

Model Deployment

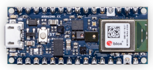
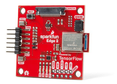
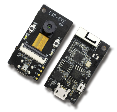
Packaging, testing, and deploying a model to a serving environment for online experimentation and production serving is what model deployment is all about.



MLOps: Model Deployment



Deployment Challenges

	Board	MCU / ASIC	Clock	Memory	Sensors	Radio
	Himax WE-I Plus EVB	HX6537-A 32-bit EM9D DSP	400 MHz	2MB flash 2MB RAM	Accelerometer, Mic, Camera	None
	Arduino Nano 33 BLE Sense	32-bit nRF52840	64 MHz	1MB flash 256kB RAM	Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color	BLE
	SparkFun Edge 2	32-bit ArtemisV1	48 MHz	1MB flash 384kB RAM	Accelerometer, Mic, Camera	BLE
	Espressif EYE	32-bit ESP32-D0WD	240 MHz	4MB flash 520kB RAM	Mic, Camera	WiFi, BLE

The Challenge

Software Stacks

Hardware Environments

The Challenge

Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend

Ruby + Rails + sass + Unicorn

API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Hardware Environments

The Challenge

Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB
postgresql + pgv8 + v8

Queue
Redis + redis-sentinel

Analytics DB
hadoop + hive + thrift + OpenJDK

Background workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend
Ruby + Rails + sass + Unicorn

API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Development VM

Public Cloud

QA server

Production Cluster

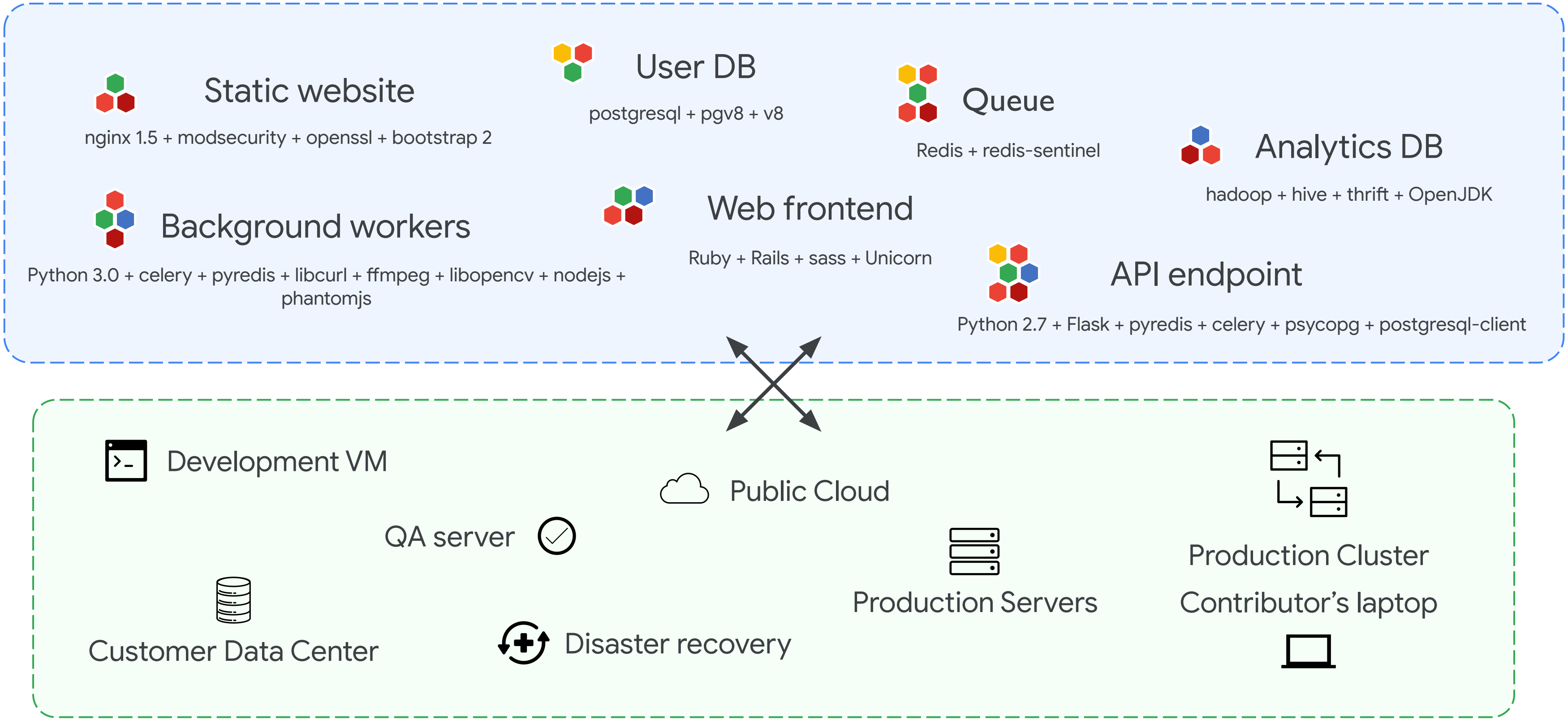
Production Servers

Contributor’s laptop

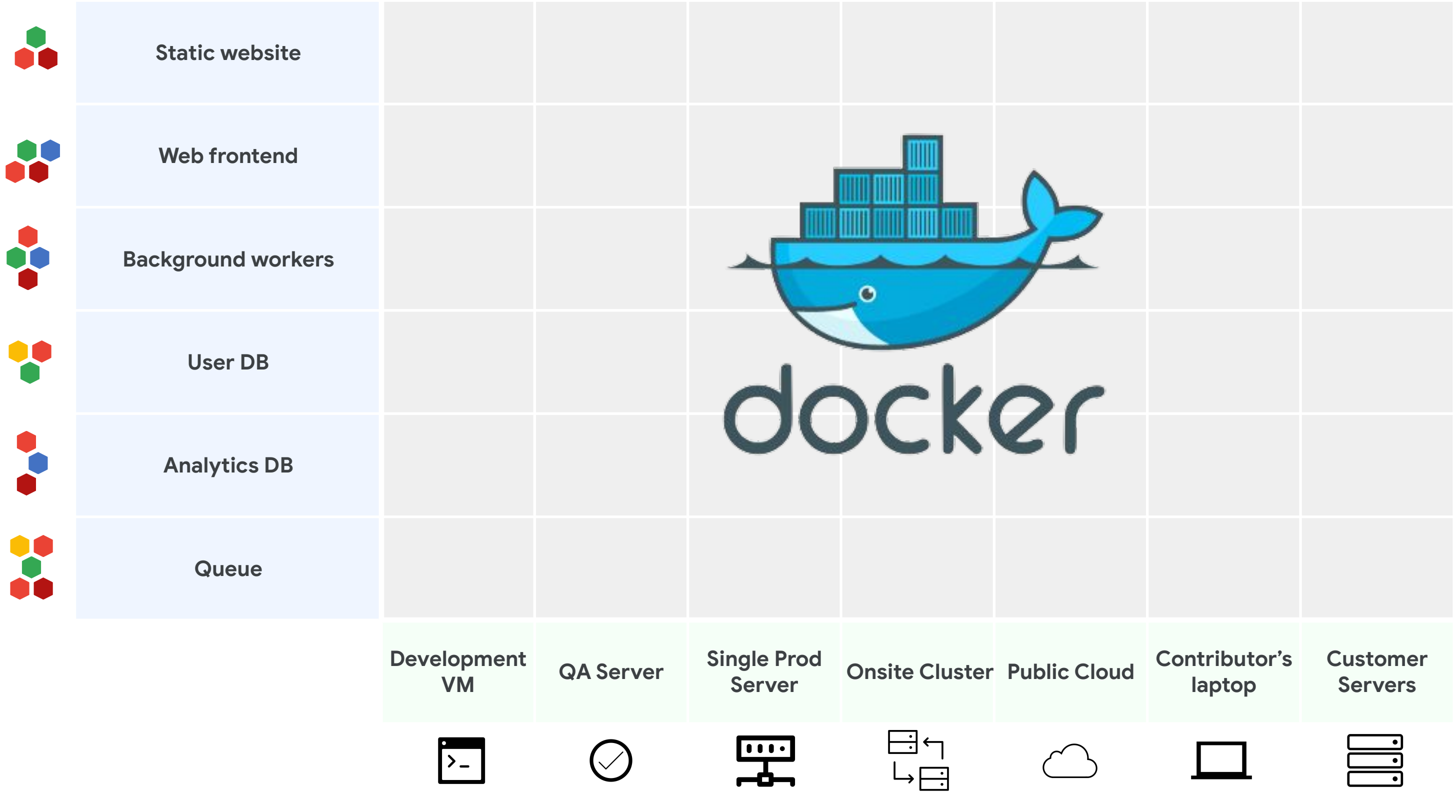
Customer Data Center

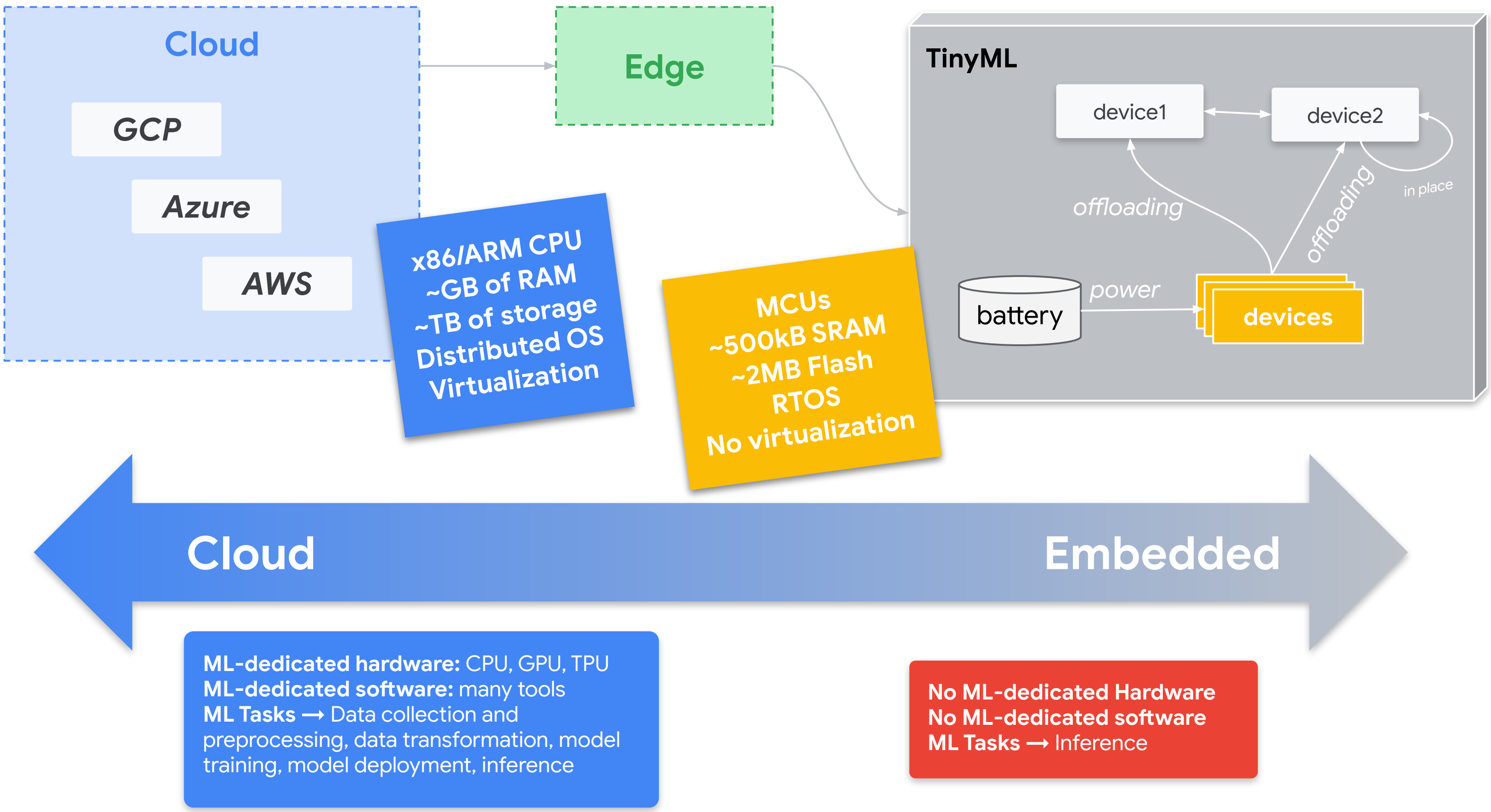
Disaster recovery

The Challenge

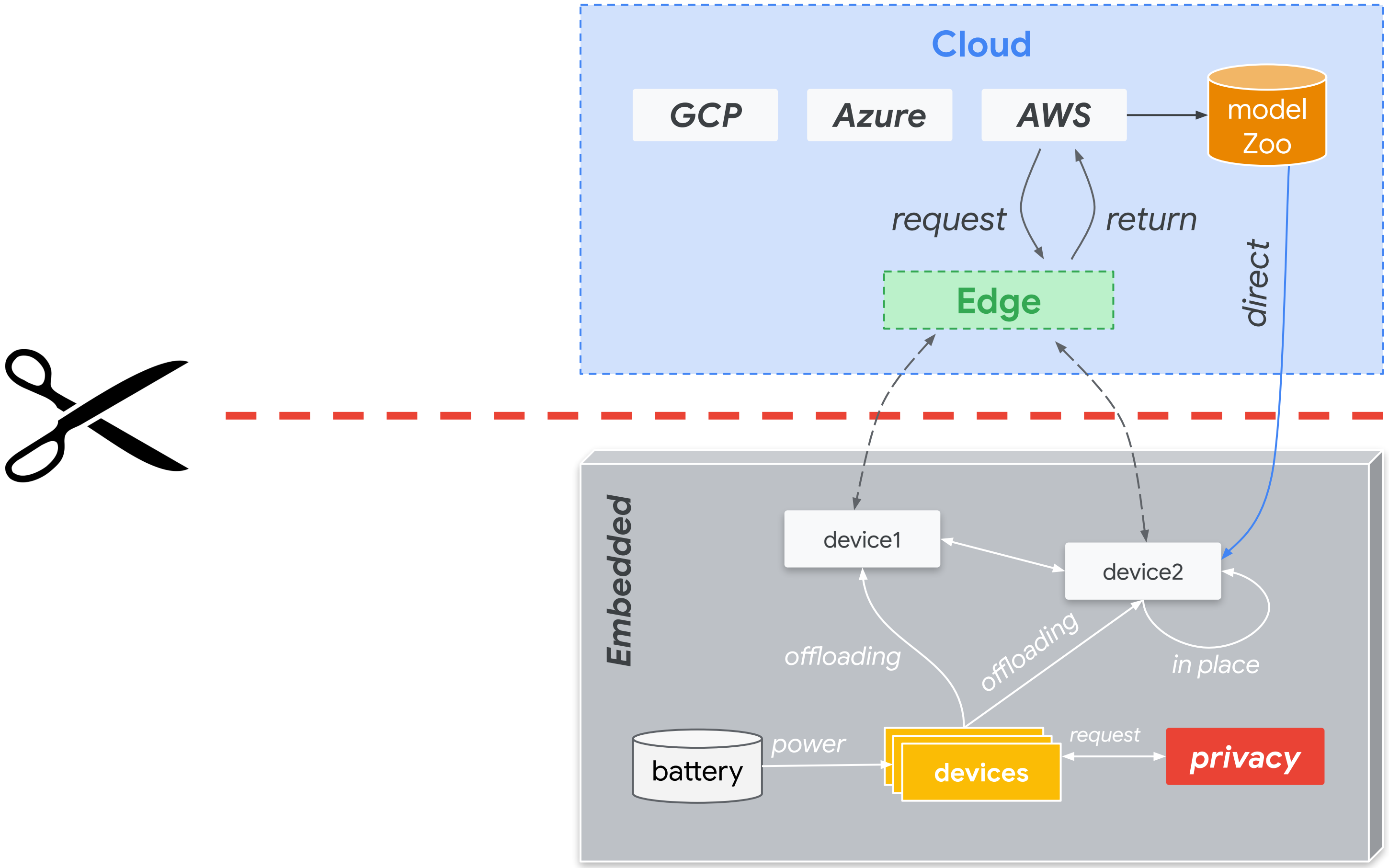


	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor’s laptop	Customer Servers
Static website	?	?	?	?	?	?	?`
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

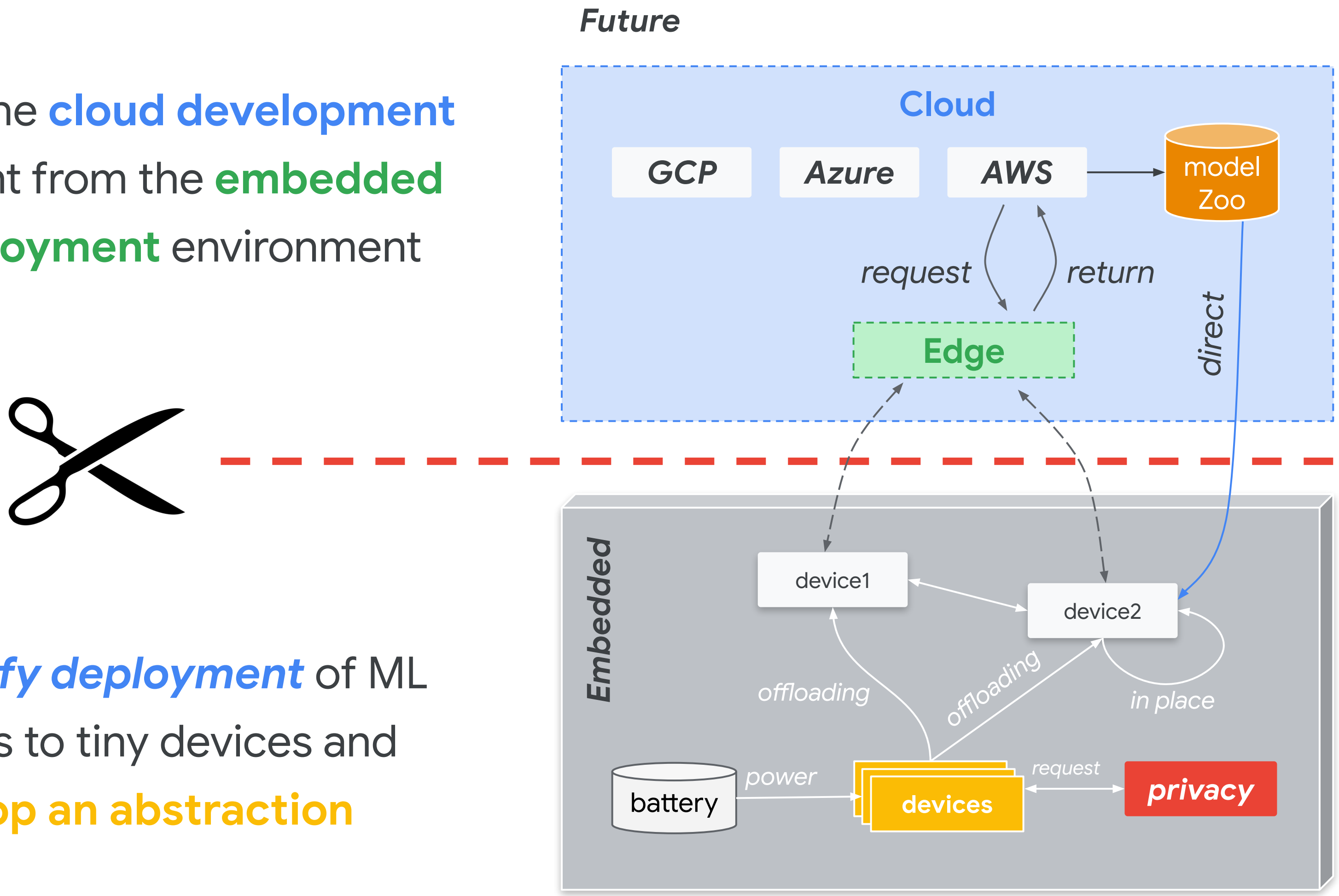

Static website
Web frontend
Background workers
User DB
Analytics DB
Queue
Development VM
QA Server
Single Prod Server
Onsite Cluster
Public Cloud
Contributor’s laptop
Customer Servers



Cloud
GCP
Azure
AWS
Edge
TinyML
device1
device2
offloading
offloading
in place
battery
power
devices
x86/ARM CPU
~GB of RAM
~TB of storage
Distributed OS
Virtualization
MCUs
~500kB SRAM
~2MB Flash
RTOS
No virtualization
Cloud
Embedded
ML-dedicated hardware: CPU, GPU, TPU
ML-dedicated software: many tools
ML Tasks ⟶ Data collection and preprocessing, data transformation, model training, model deployment, inference
No ML-dedicated Hardware
No ML-dedicated software
ML Tasks ⟶ Inference

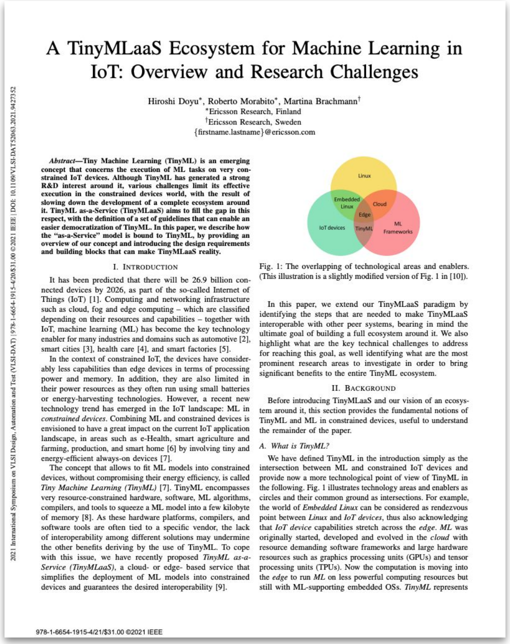

Cloud
GCP
Azure
AWS
model Zoo
request
return
direct
Edge
Embedded
device1
device2
offloading
offloading
in place
battery
power
devices
request
privacy

Decouple the **cloud development** environment from the **embedded** model **deployment** environment



Simplify deployment of ML models to tiny devices and **develop an abstraction**

TinyMLaaS



A TinyMLaaS Ecosystem for Machine Learning in IoT: Overview and Research Challenges

Hiroshi Doyu*, Roberto Morabito*, Martina Brachmann†
*Ericsson Research, Finland
†Ericsson Research, Sweden
{firstname.lastname}@ericsson.com

2021 International Symposium on VLSI Design, Automation and Test (VLSI-DAT) | 978-1-6654-1915-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/VLSI-DAT52063.2021.9427352

***Abstract*—Tiny Machine Learning (TinyML) is an emerging concept that concerns the execution of ML tasks on very constrained IoT devices. Although TinyML has generated a strong R&D interest around it, various challenges limit its effective execution in the constrained devices world, with the result of slowing down the development of a complete ecosystem around it. TinyML as-a-Service (TinyMLaaS) aims to fill the gap in this respect, with the definition of a set of guidelines that can enable an easier democratization of TinyML. In this paper, we describe how the "as-a-Service" model is bound to TinyML, by providing an overview of our concept and introducing the design requirements and building blocks that can make TinyMLaaS reality.**

I. INTRODUCTION

It has been predicted that there will be 26.9 billion connected devices by 2026, as part of the so-called Internet of Things (IoT) [1]. Computing and networking infrastructure such as cloud, fog and edge computing – which are classified depending on their resources and capabilities – together with IoT, machine learning (ML) has become the key technology enabler for many industries and domains such as automotive [2], smart cities [3], health care [4], and smart factories [5].

In the context of constrained IoT, the devices have considerably less capabilities than edge devices in terms of processing power and memory. In addition, they are also limited in their power resources as they often run using small batteries or energy-harvesting technologies. However, a recent new technology trend has emerged in the IoT landscape: ML in *constrained devices*. Combining ML and constrained devices is envisioned to have a great impact on the current IoT application landscape, in areas such as e-Health, smart agriculture and farming, production, and smart home [6] by involving tiny and energy-efficient always-on devices [7].

The concept that allows to fit ML models into constrained devices, without compromising their energy efficiency, is called *Tiny Machine Learning (TinyML)* [7]. TinyML encompasses very resource-constrained hardware, software, ML algorithms, compilers, and tools to squeeze a ML model into a few kilobyte of memory [8]. As these hardware platforms, compilers, and software tools are often tied to a specific vendor, the lack of interoperability among different solutions may undermine the other benefits deriving by the use of TinyML. To cope with this issue, we have recently proposed *TinyML as-a-Service (TinyMLaaS)*, a cloud- or edge- based service that simplifies the deployment of ML models into constrained devices and guarantees the desired interoperability [9].

Linux
Embedded Linux
Cloud
Edge
IoT devices
TinyML
ML Frameworks

Fig. 1: The overlapping of technological areas and enablers. (This illustration is a slightly modified version of Fig. 1 in [10]).

In this paper, we extend our TinyMLaaS paradigm by identifying the steps that are needed to make TinyMLaaS interoperable with other peer systems, bearing in mind the ultimate goal of building a full ecosystem around it. We also highlight what are the key technical challenges to address for reaching this goal, as well identifying what are the most prominent research areas to investigate in order to bring significant benefits to the entire TinyML ecosystem.

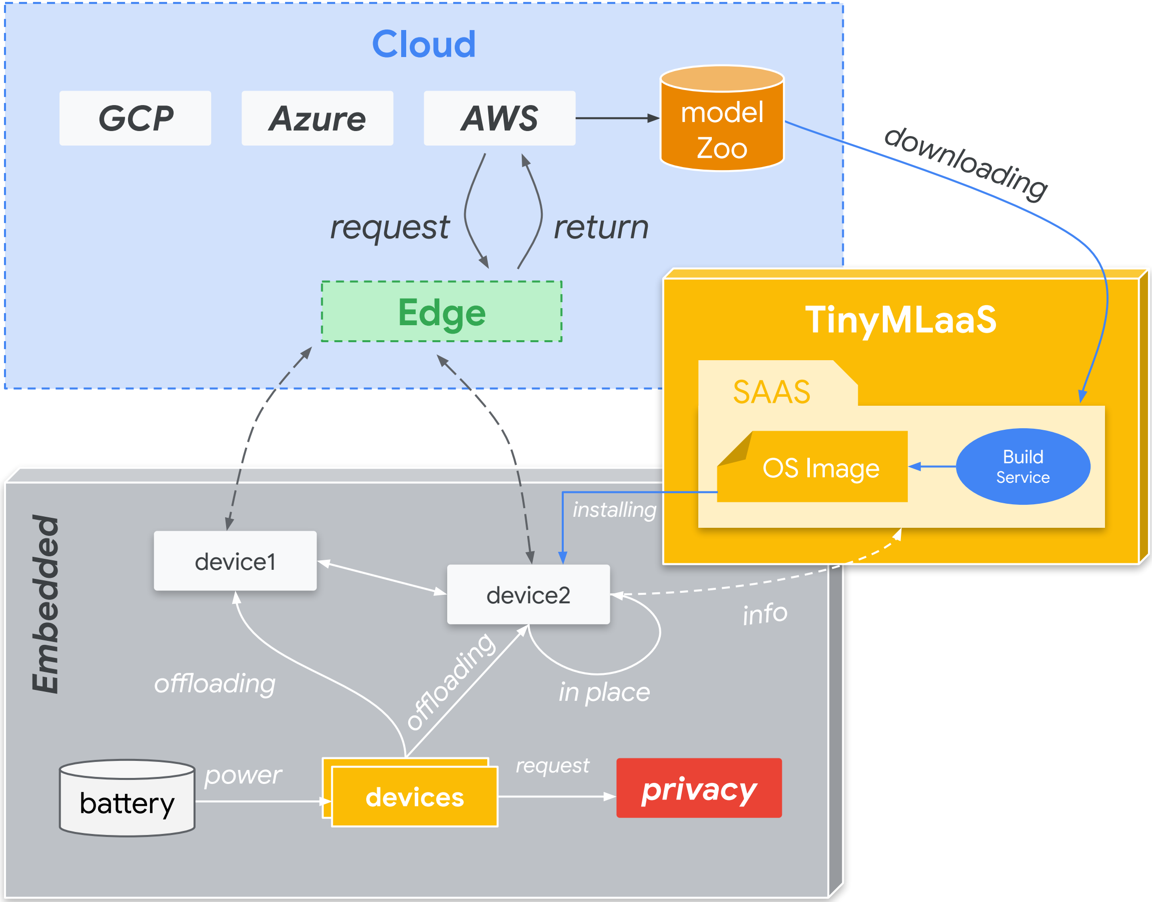
II. BACKGROUND

Before introducing TinyMLaaS and our vision of an ecosystem around it, this section provides the fundamental notions of TinyML and ML in constrained devices, useful to understand the remainder of the paper.

A. What is TinyML?

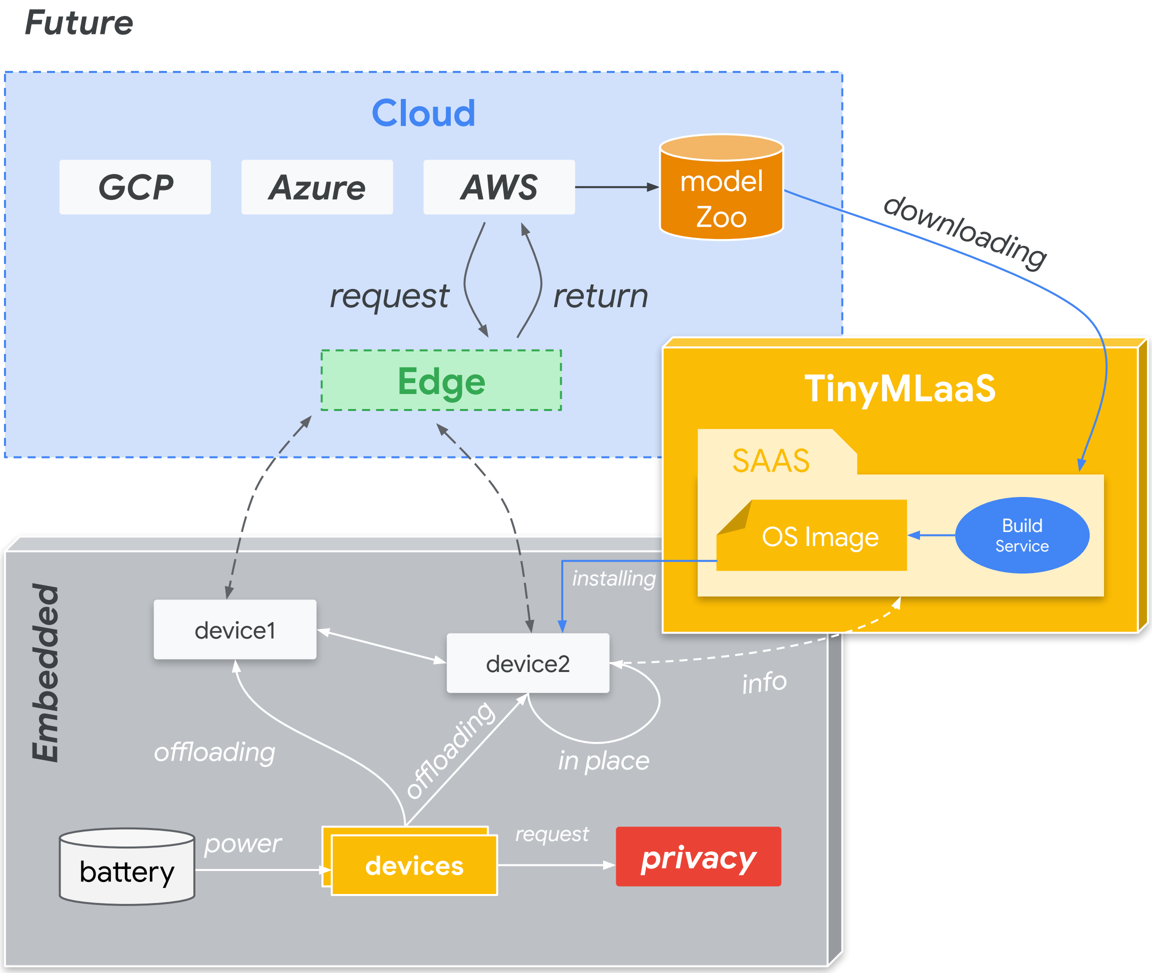
We have defined TinyML in the introduction simply as the intersection between ML and constrained IoT devices and provide now a more technological point of view of TinyML in the following. Fig. 1 illustrates technology areas and enablers as circles and their common ground as intersections. For example, the world of *Embedded Linux* can be considered as rendezvous point between *Linux* and *IoT devices*, thus also acknowledging that *IoT device* capabilities stretch across the *edge*. *ML* was originally started, developed and evolved in the *cloud* with resource demanding software frameworks and large hardware resources such as graphics processing units (GPUs) and tensor processing units (TPUs). Now the computation is moving into the *edge* to run *ML* on less powerful computing resources but still with ML-supporting embedded OSs. *TinyML* represents

Future



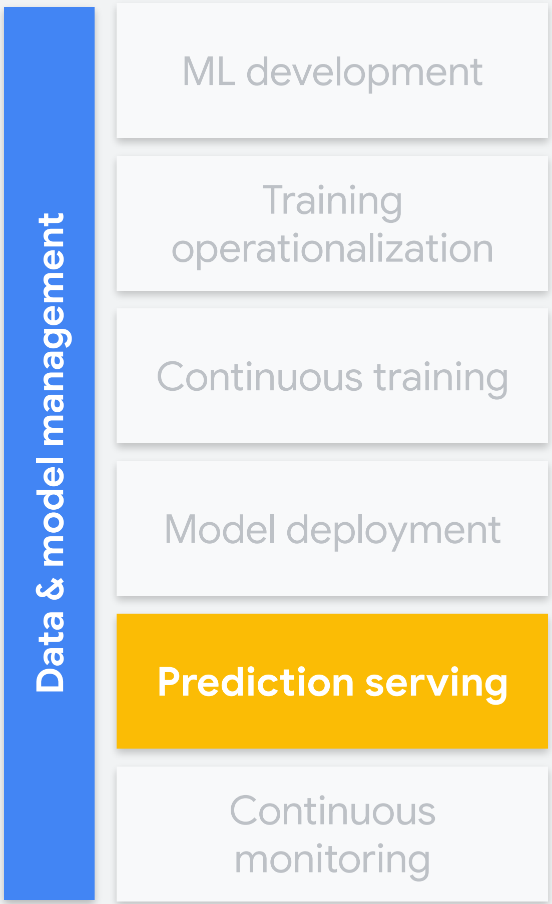
TinyMLaaS

- TinyML as a Service is a cloud or edge-based machine learning as a service
- Simplifies the deployment of ML models ⟶ abstraction

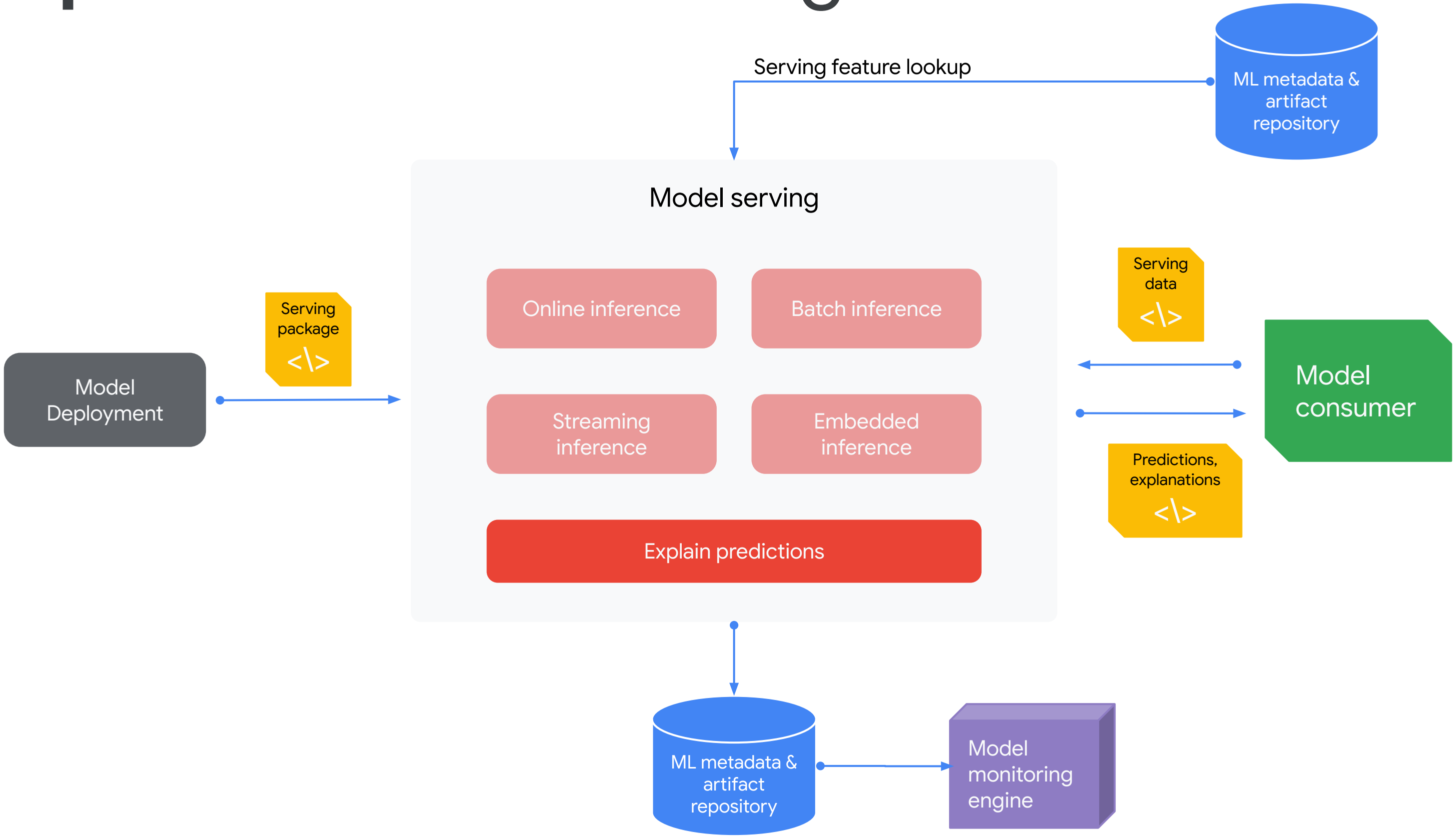


Prediction Serving

Serving the model that is deployed in production for inference is known as prediction serving.



MLOps: Prediction Serving



	Scenario	Metric
	**Batch inference** (e.g. photo sorting app)	**Throughput**
	**Online inference** (e.g. translation app)	**QPS** subject to latency bound
	**Streaming inference** (e.g. multiple camera driving assistance)	**Number streams** subject to latency bound
	**Embedded inference** (e.g. cell phone augmented vision)	**Latency**

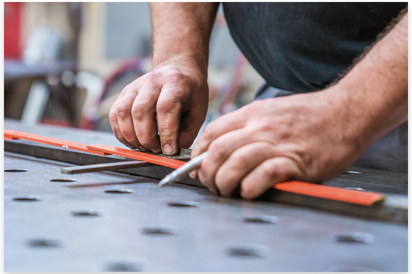
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus

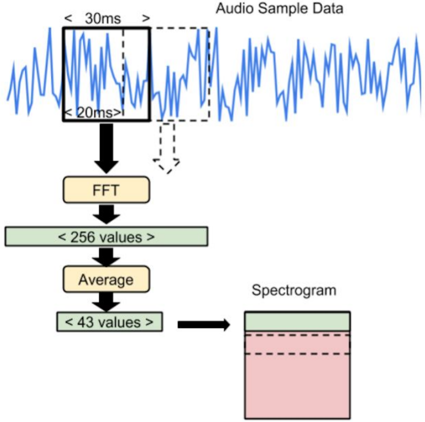


Provides

- **Standardization** of use cases and workloads
- **Comparability** across heterogeneous HW/SW systems
- **Complex characterization** of system compromises
- **Verifiable and Reproducible** results

MLPerf "Tiny" Tasks

Keyword Spotting



Warden, Pete. "Speech commands: A dataset for limited-vocabulary speech recognition." *arXiv preprint arXiv:1804.03209* (2018).

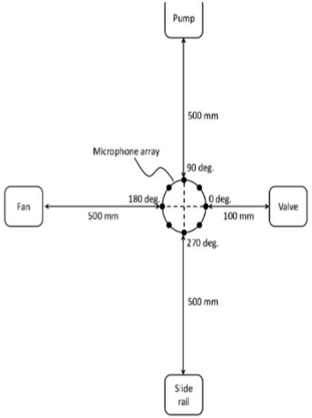
Visual Wake Words

(a) 'Person'

(b) 'Not-person'

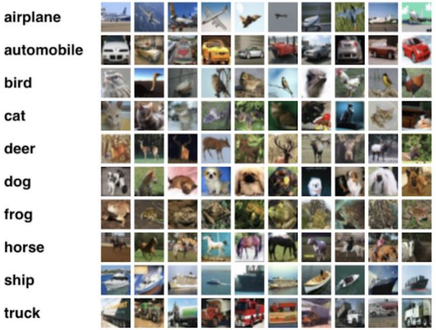
Chowdhery, Aakanksha, et al. "Visual wake words dataset." *arXiv preprint arXiv:1906.05721* (2019).

Anomaly Detection



Purohit, Harsh, et al. "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection." *arXiv preprint arXiv:1909.09347* (2019).

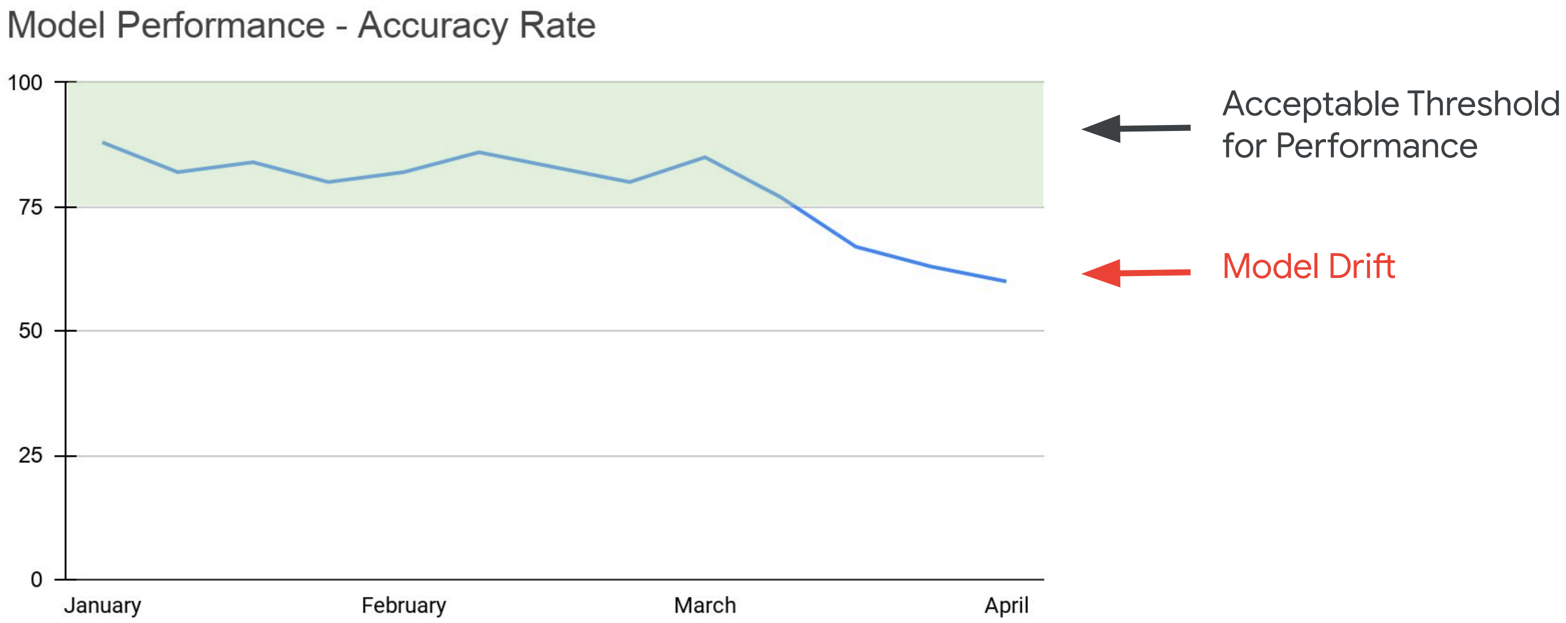
Tiny Image Classification



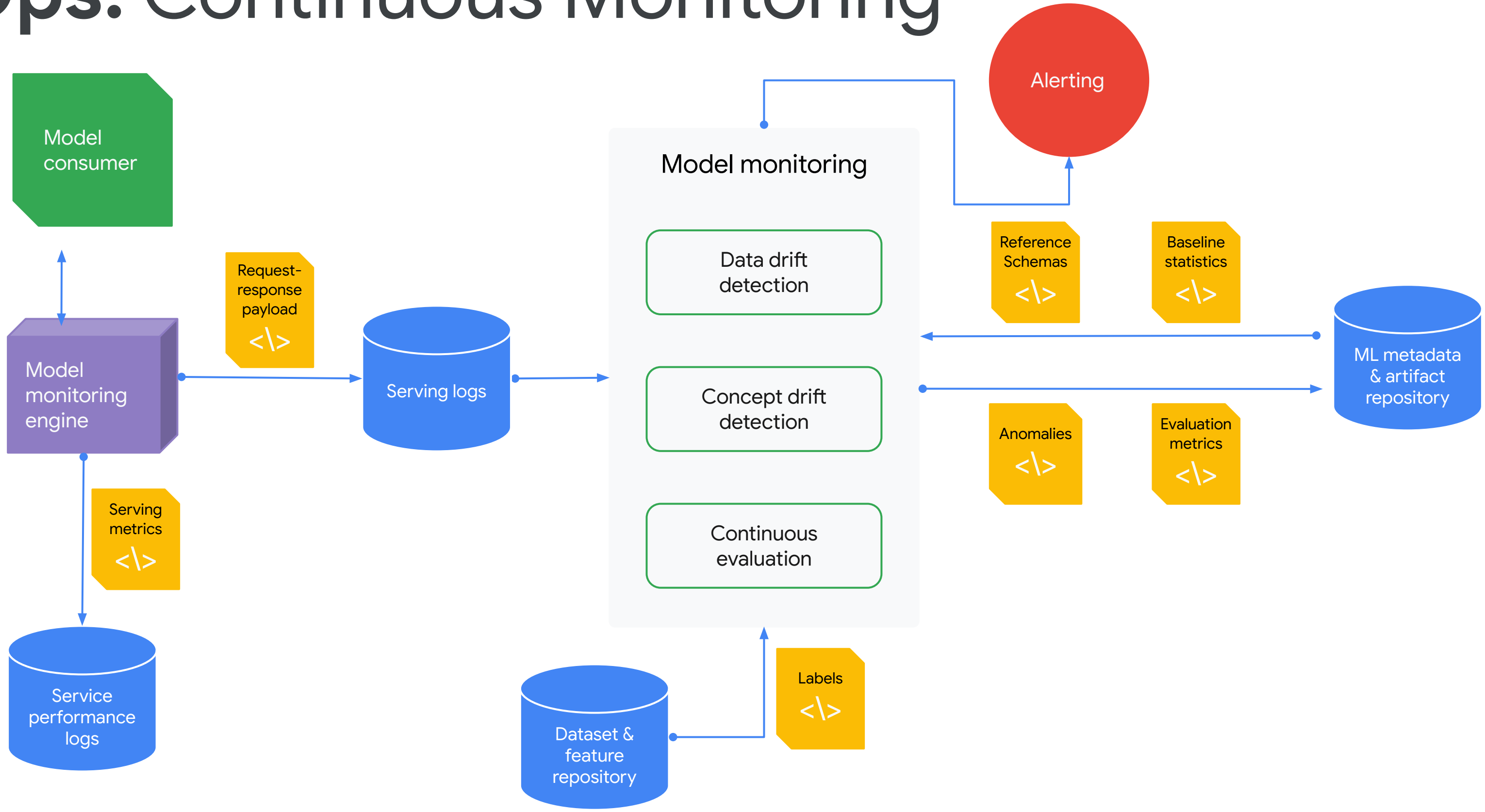
Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

Continuous Monitoring

Continuous monitoring refers to keeping track of a deployed model's effectiveness and efficiency.


Model Performance - Accuracy Rate
100
75
50
25
0
January
February
March
April
Acceptable Threshold
for Performance
Model Drift

MLOps: Continuous Monitoring



Drift Types

Concept Drift

the affected *old data* needs to be relabeled

Data Drift

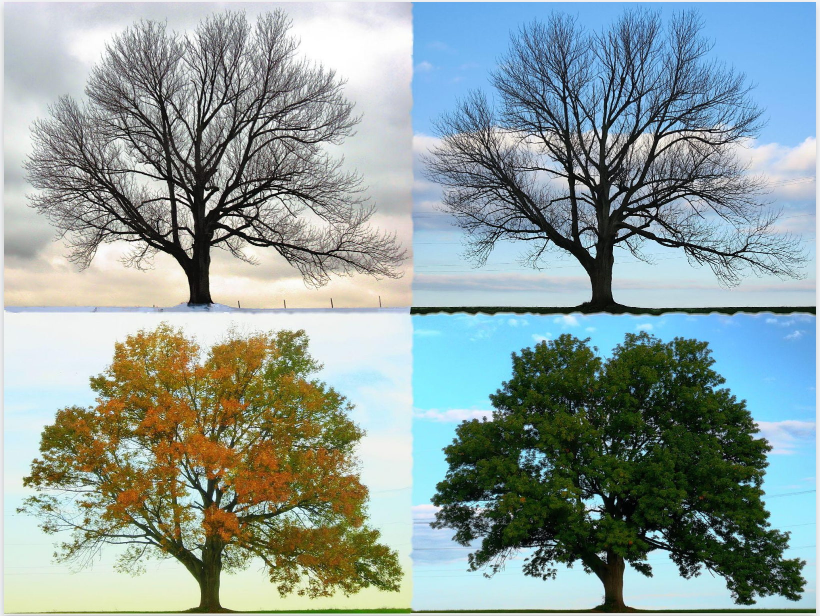
enough new data needs to be labeled

Concept Drift

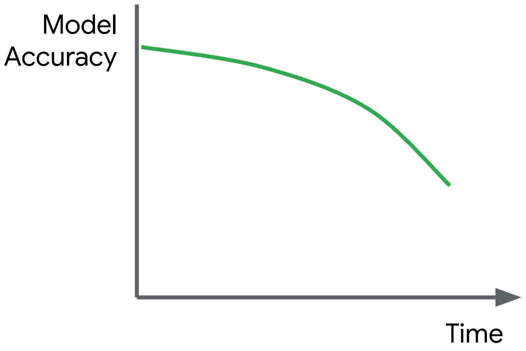
Concept drift in machine learning is when the **relationship** between the input and target changes over time.

Data Drift

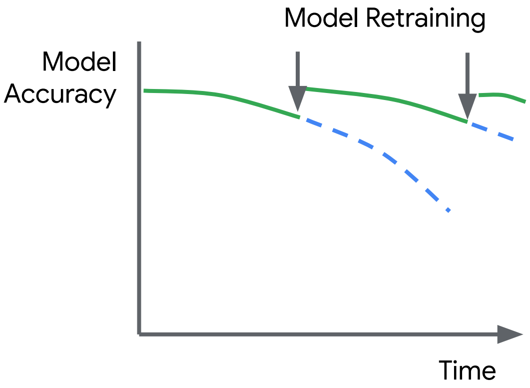
Data drift is a change in the **distribution** of data over time.



Goal of Continuous Training



Model Decay over time



Regularly updated model

Continuous Monitoring for **TinyML**

- Monitoring may **not always** be a **feasible** option
  - Low power communication protocol
  - Device isn’t wifi-enabled

- Monitoring opens up **security and privacy risks**

Continuous Monitoring for TinyML

- Monitoring may **not always** be a **feasible** option
  - Low power communication protocol
  - Device isn't wifi-enabled

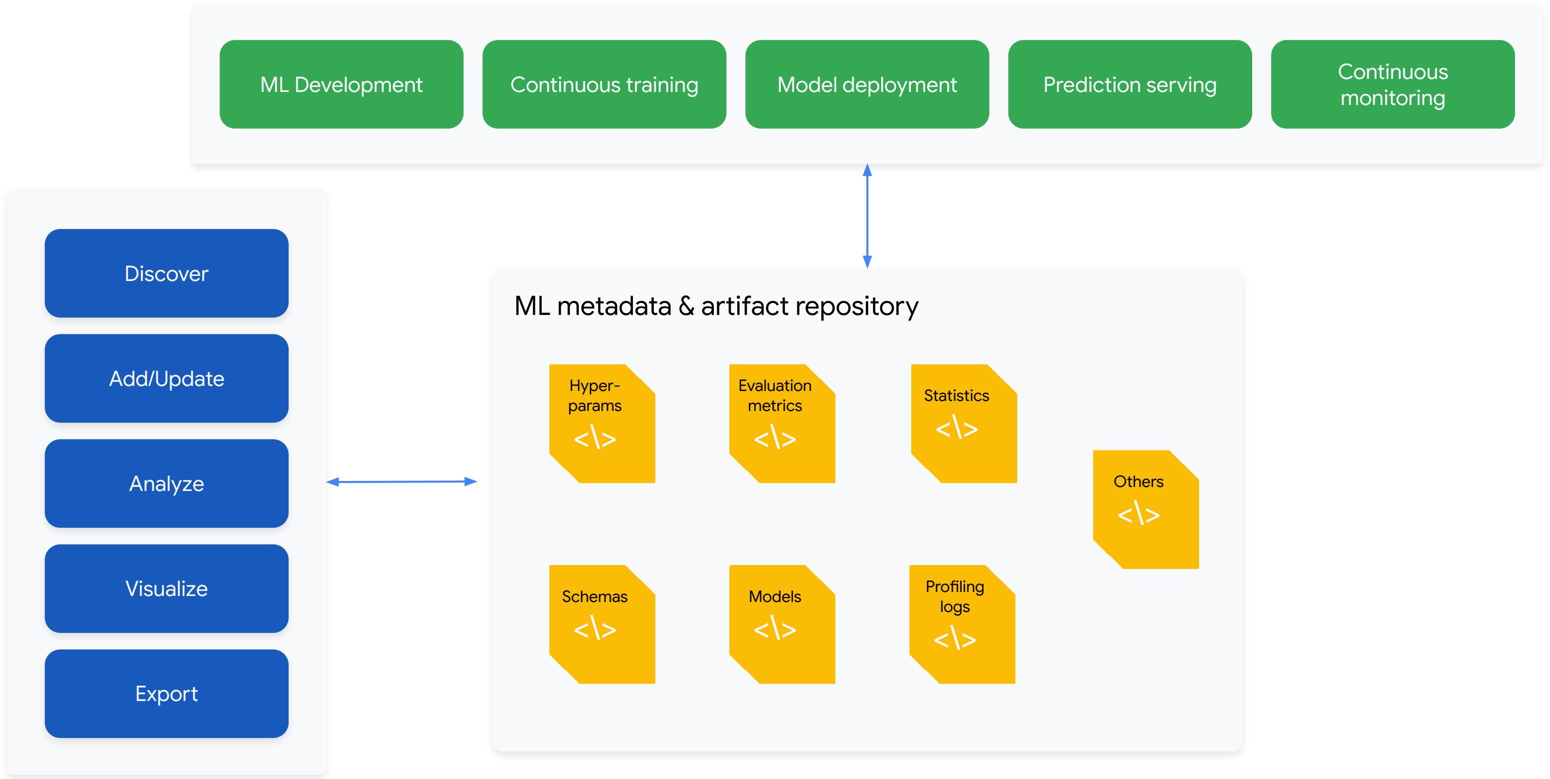
- Monitoring opens up **security and privacy risks**

- How can we enable **Continuous Monitoring** to enable **Continuous Training** without moving the data off the endpoint tiny ML device?

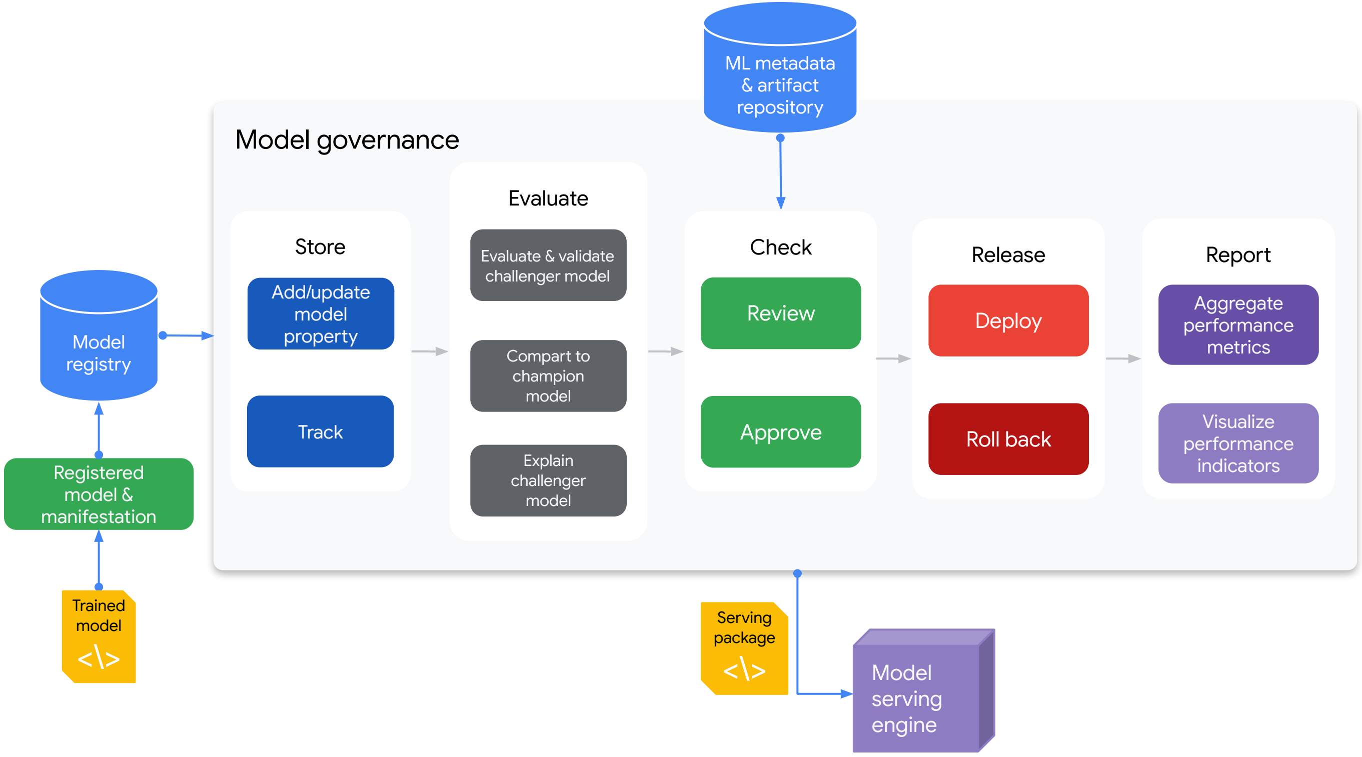
Data & Model Management

Data and model management is a central, cross-cutting function for governing ML artifacts to support ability, traceability, and compliance. Data and model management can also promote shareability, reusability, and discoverability of ML assets.

MLOps: Data and Model Management

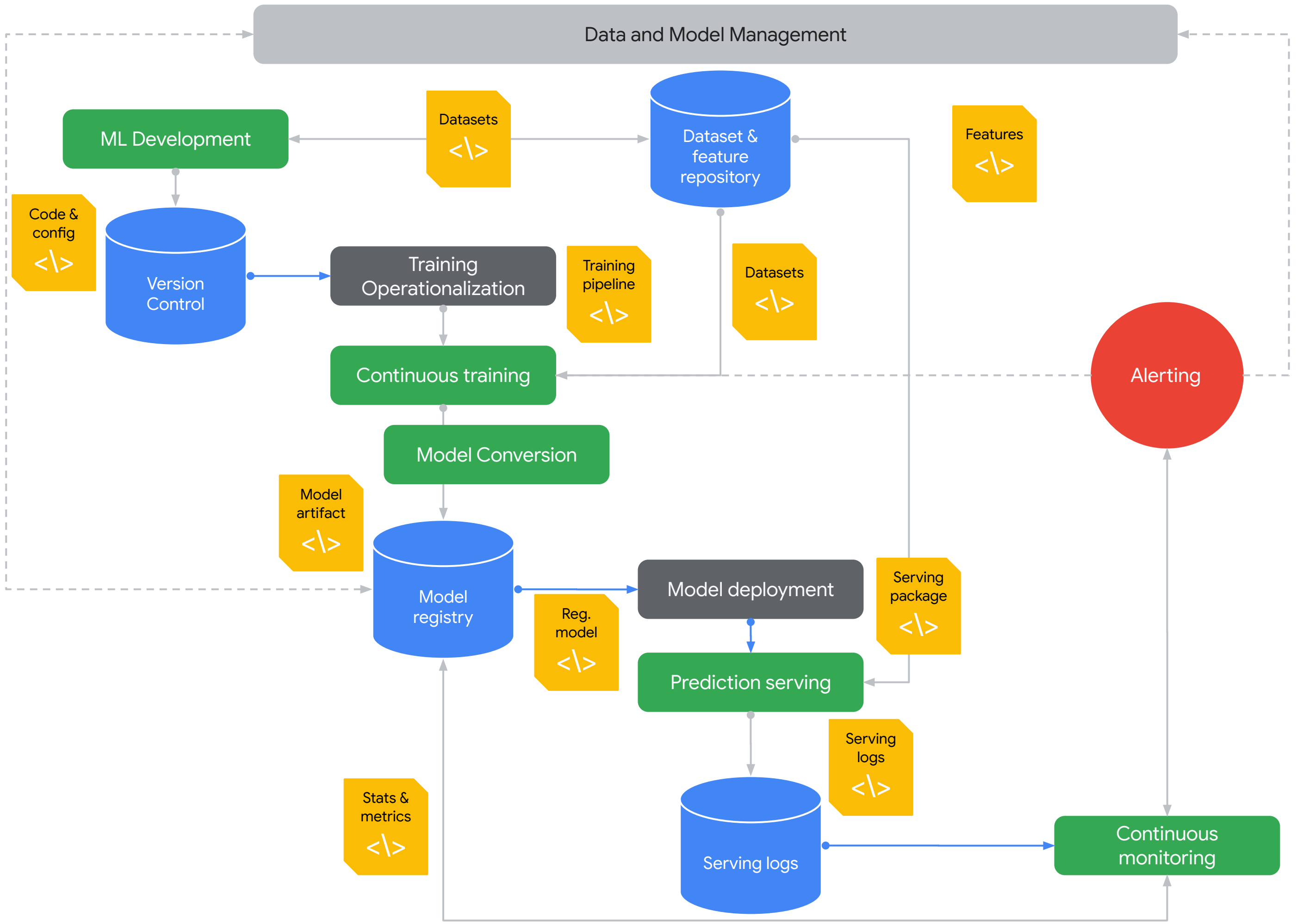


MLOps: Data and Model Management



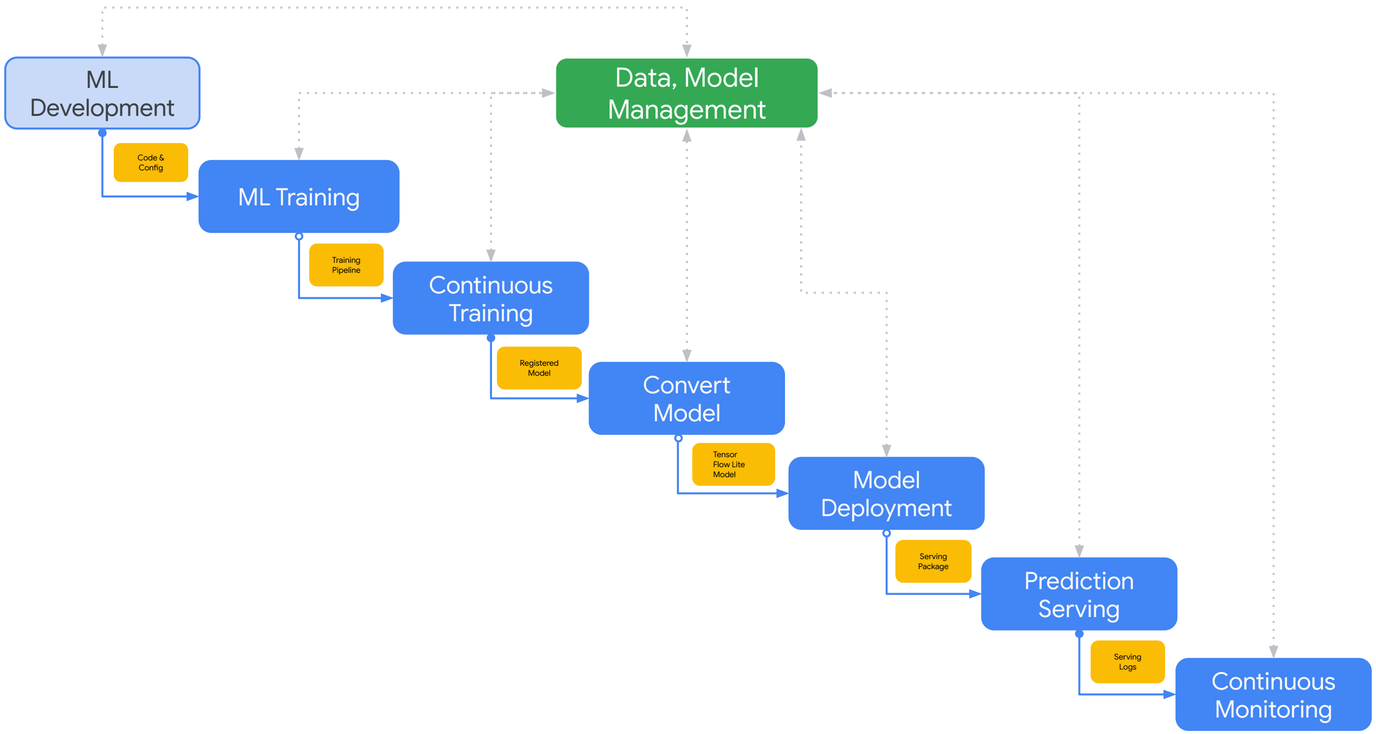
Advantages of **strong** MLOps

- Manage overwhelming complexity
- Reduce knowledge burden
- Be more scientific
- Ease long term maintenance
- Improve model performance

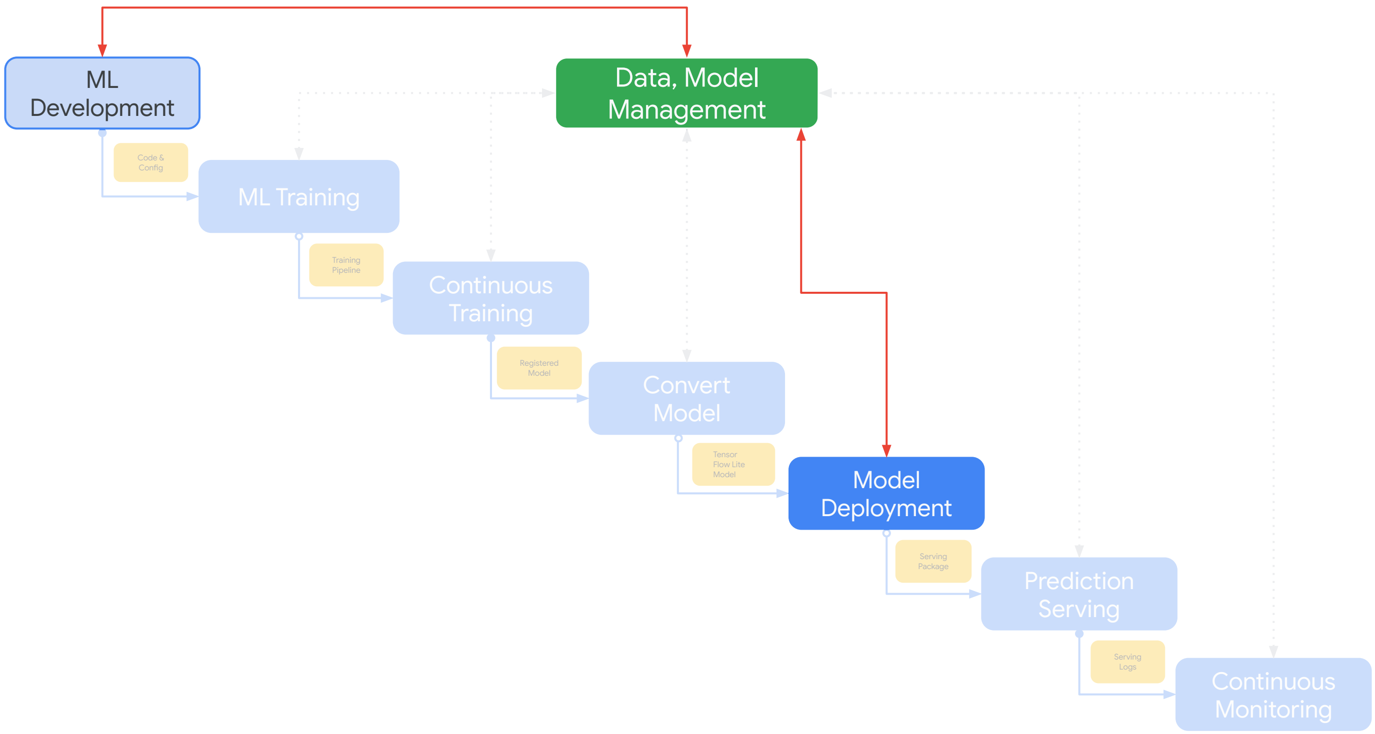


Data and Model Management
ML Development
Datasets
Dataset & feature repository
Features
Code & config
Version Control
Training Operationalization
Training pipeline
Datasets
Continuous training
Alerting
Model Conversion
Model artifact
Model registry
Model deployment
Serving package
Reg. model
Prediction serving
Serving logs
Stats & metrics
Serving logs
Continuous monitoring

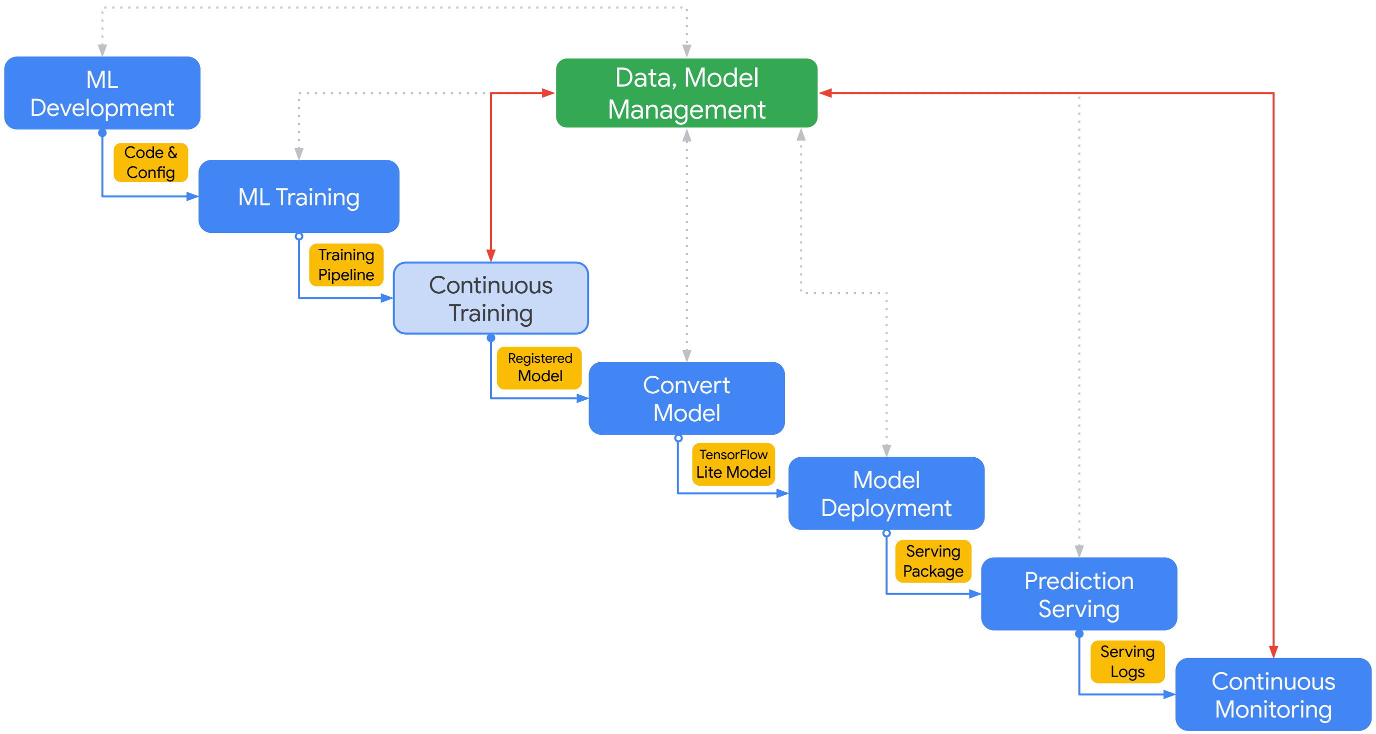
Downstream & Upstream Impact



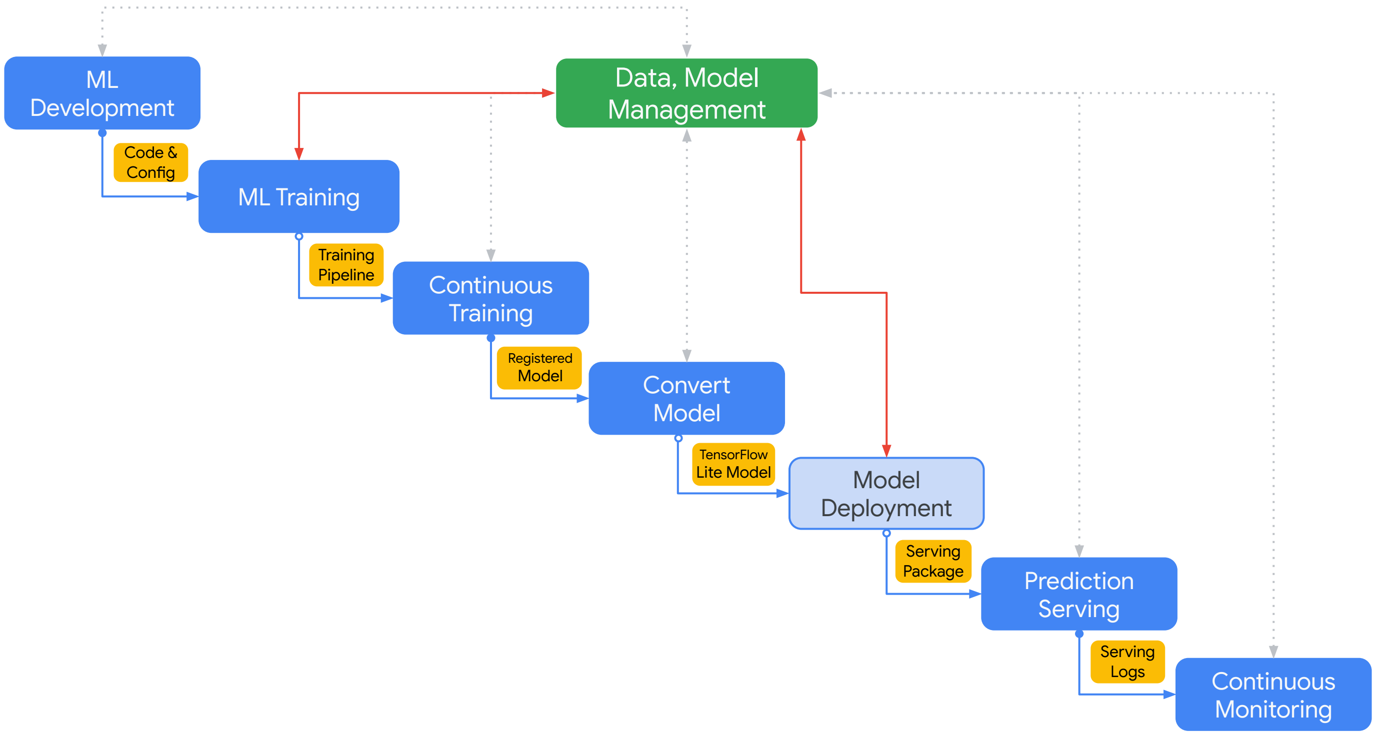
ML Development
Code & Config
ML Training
Training Pipeline
Continuous Training
Registered Model
Convert Model
Tensor Flow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring
Data, Model Management



ML Development
Code & Config
ML Training
Training Pipeline
Continuous Training
Registered Model
Data, Model Management
Convert Model
Tensor Flow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring

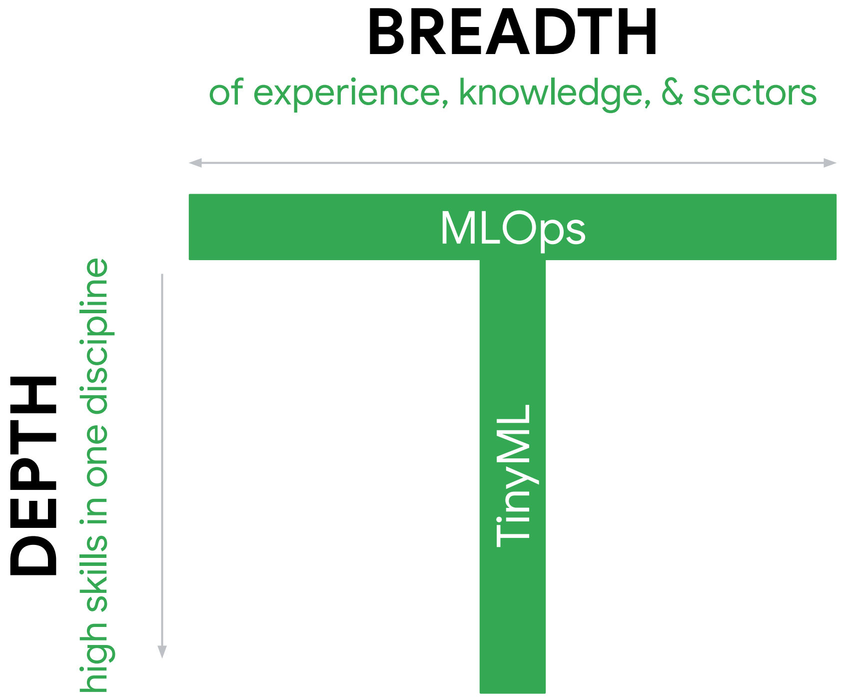


ML Development
Code & Config
ML Training
Data, Model Management
Training Pipeline
Continuous Training
Registered Model
Convert Model
TensorFlow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring



ML Development
Code & Config
ML Training
Data, Model Management
Training Pipeline
Continuous Training
Registered Model
Convert Model
TensorFlow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring

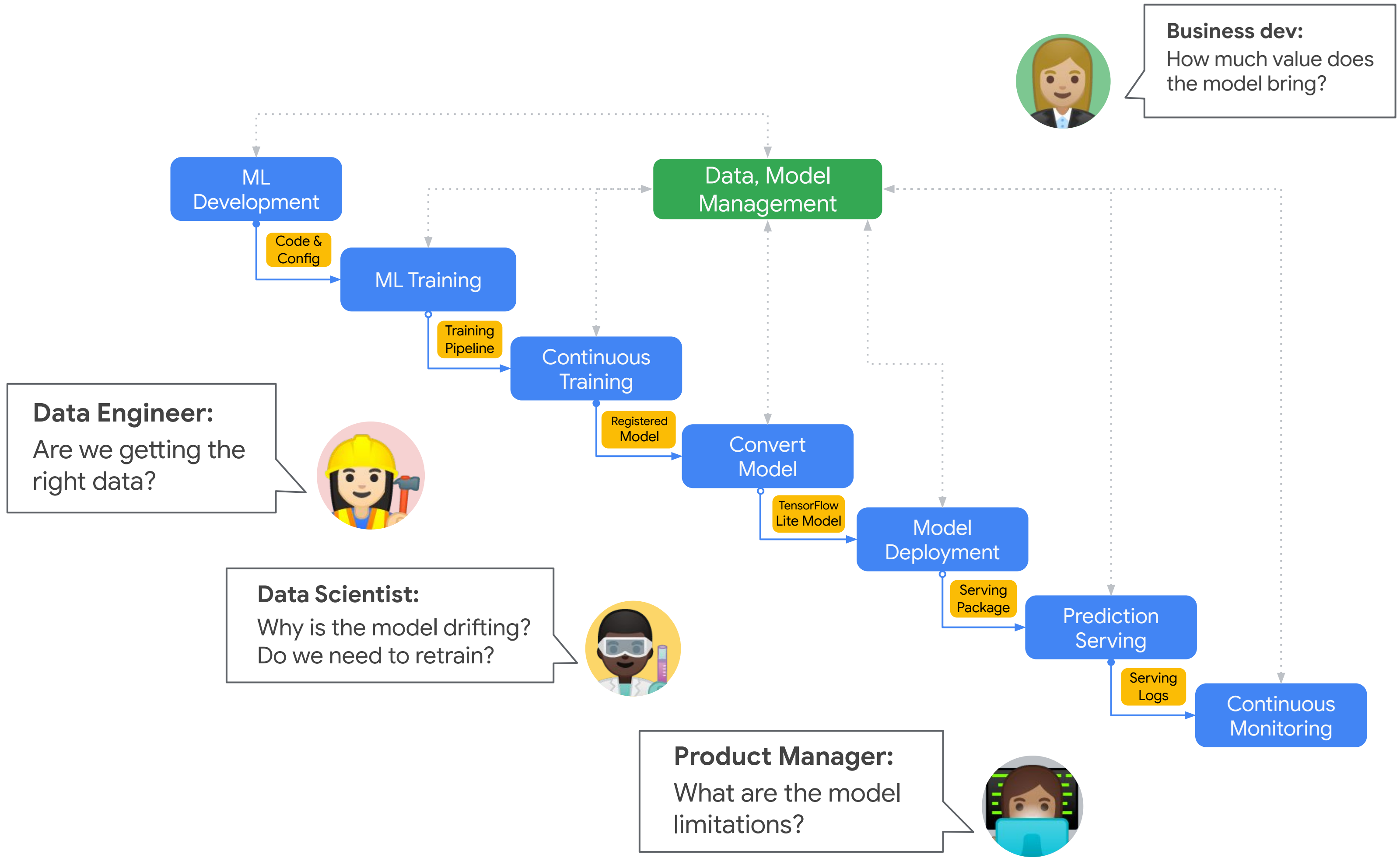
Keeping the Big Picture in Mind



Note:

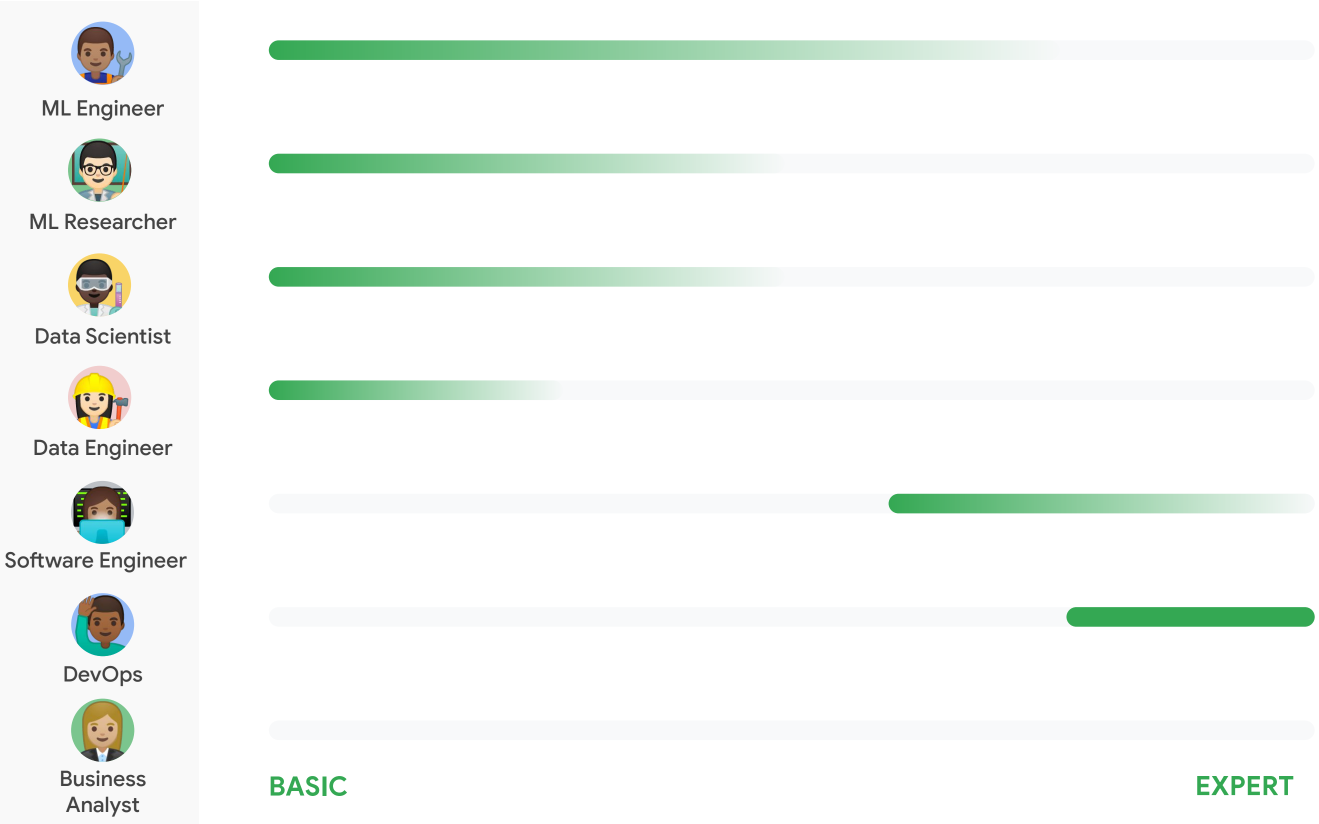
Companies like T-shaped people

ML Engineer | ML Researcher | Data Scientist | Data Engineer | Software Engineer | DevOps | Business Analyst


Business dev:
How much value does the model bring?
ML Development
Data, Model Management
Code & Config
ML Training
Training Pipeline
Continuous Training
Data Engineer:
Are we getting the right data?
Registered Model
Convert Model
TensorFlow Lite Model
Model Deployment
Data Scientist:
Why is the model drifting?
Do we need to retrain?
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring
Product Manager:
What are the model limitations?

ML Expertise

Deployment Expertise



ML Engineer

ML Researcher

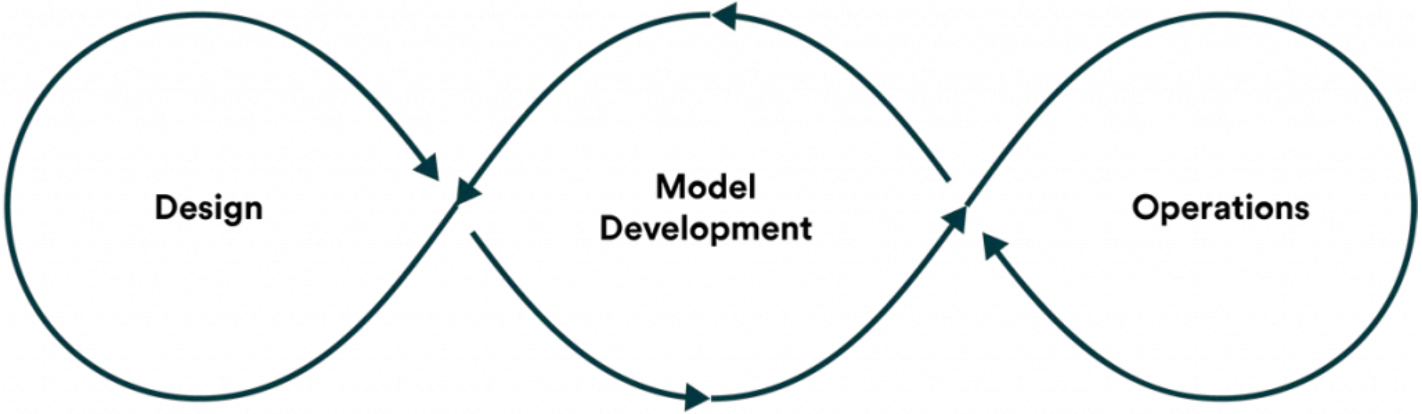
Data Scientist

Data Engineer

Software Engineer

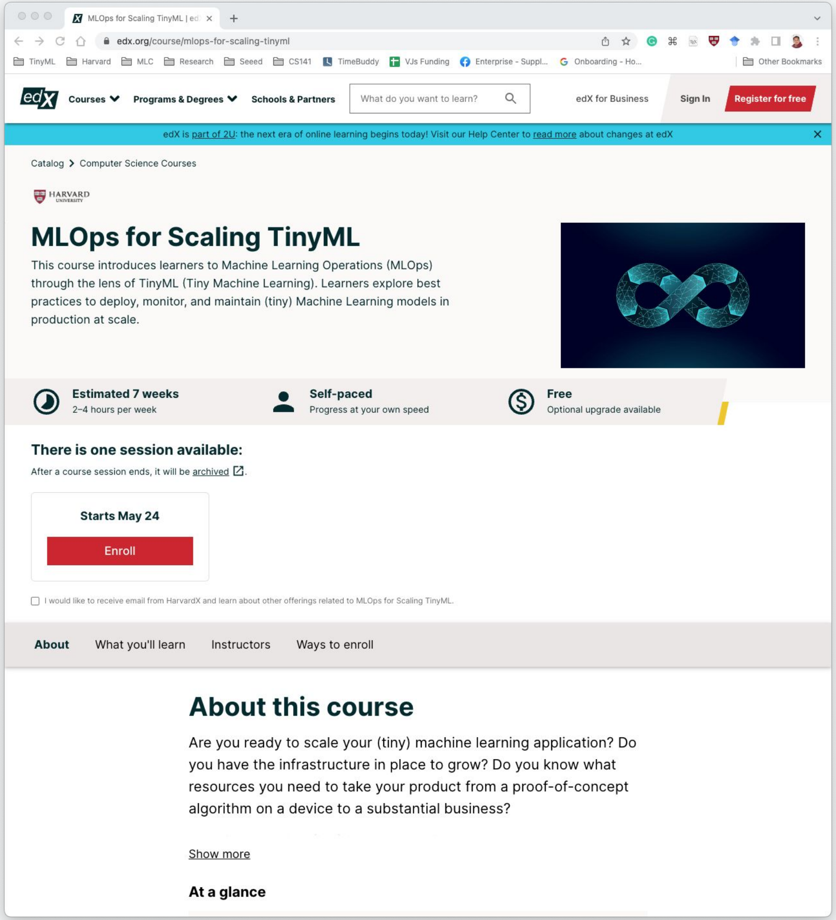
DevOps

Business Analyst



MLOps Course for Scaling TinyML

- Know why and when deploying MLOps can help your (tiny) product or business
- Key MLOps platform features that you can deploy for your data science project
- How can you automate the MLOps life cycle for robust development and deployment
- Real-world examples and case studies of MLOps Platforms targeting tiny devices



Conclusion

The Future of ML is Tiny and Bright

Course Topics

1. Overview and Introduction to Embedded Machine Learning
2. Data Engineering
3. Embedded Machine Learning Frameworks
4. Efficient Model Representation and Compression
5. ~~Performance Metrics and Benchmarking of ML Systems~~
6. Learning on the Edge
7. Hardware Acceleration for Edge ML: GPUs, TPUs and FPGAs
8. **Embedded MLOps**
9. Secure and Privacy-Preserving On-Device ML
10. Responsible AI
11. Sustainability at the Edge
12. Generative AI at the Edge

Guest Speaker

Daniel Situnayake

Daniel Situnayake is Head of Machine Learning at Edge Impulse, and a technologist, entrepreneur, and author. Daniel is co-author of two popular books on embedded artificial intelligence: *AI at the Edge*, which provides practical insights for engineers, PMs, and engineering leaders, and *TinyML*, which has become the standard introductory textbook for teaching embedded machine learning. He was al

Website