Compute and the Governance of AI

Lennart Heim
lennart.heim@governance.ai
Centre for the Governance of AI

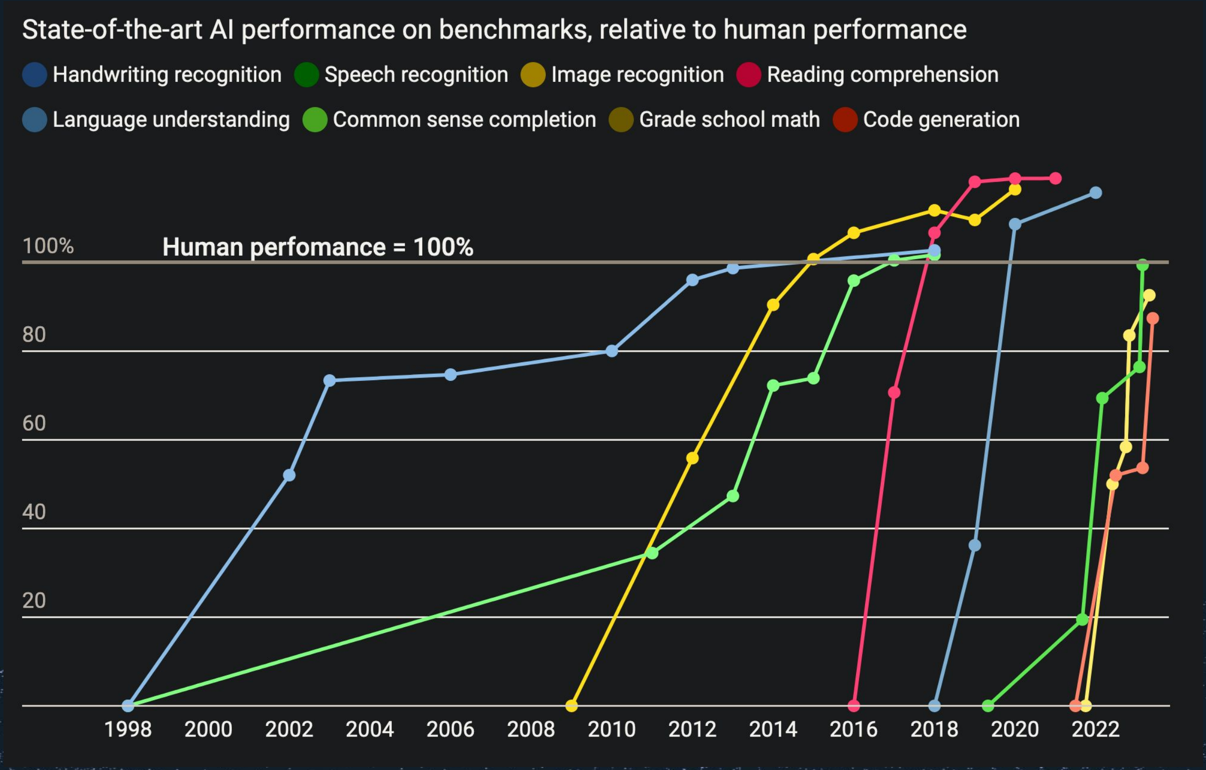
August 31st, 2023
Google Zürich

Outline

1. Risks from Advanced AI Systems
2. The Promise of Compute
3. Governance Capacities Enabled by Compute
4. Examples of Compute Governance
5. *Conclusion*: Compute and the Governance of AI

1. Risks from Advanced AI Systems

AI capabilities are advancing rapidly



Henshall, Time, 2023

Thinking about Risks from AI

Accident Risks

Misuse Risks

Structural Risks

Zwetsloot & Dafoe, 2019

Three Regulatory Challenges Posed by Frontier AI

Deployment Safety Problem

Unexpected Capabilities Problem

Proliferation Problem

The AI Governance Problem

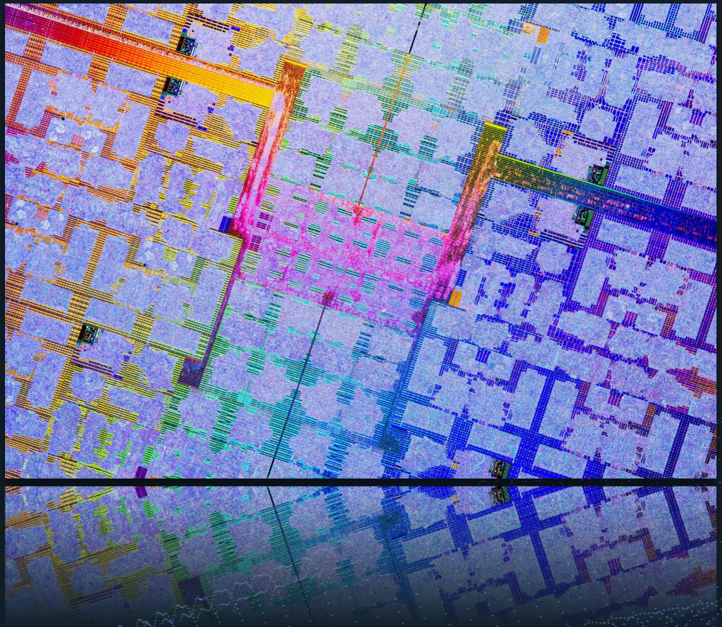
- AI has the potential to transform the economy, science, and security at a scale.
- Alongside the benefits, there are likely serious risks.
- Transformative AI systems might be developed in our lifetime, so they warrant more attention and caution.

AI Governance Definition

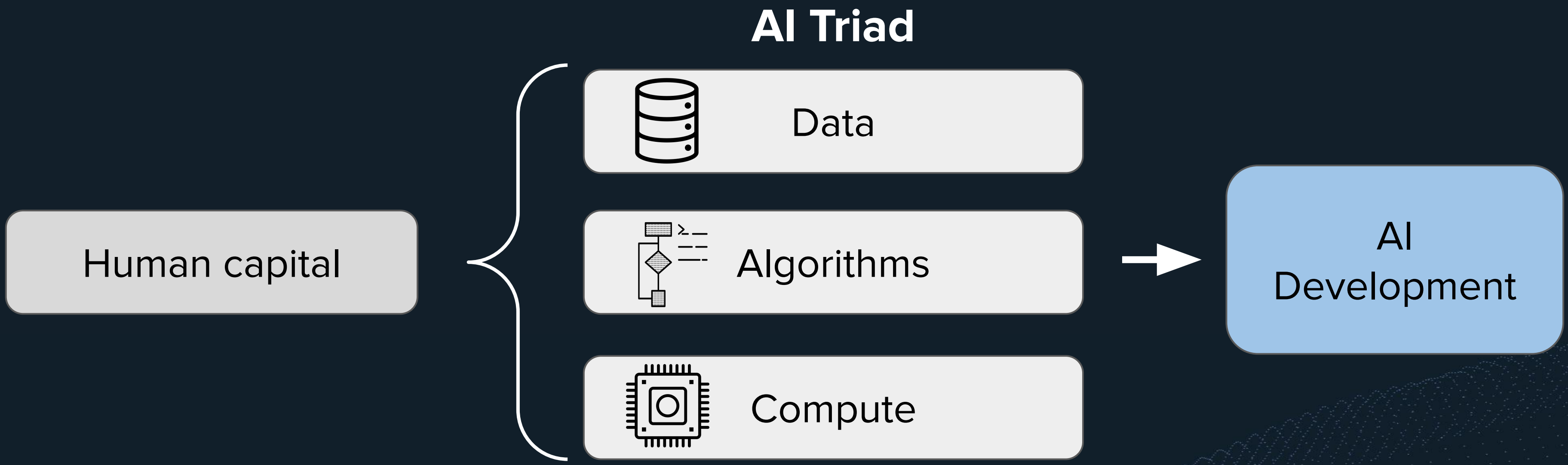
"The study and shaping of local and global governance systems — including norms, policies, laws, processes, politics, and institutions — that affect the research, development, deployment, and use of existing and future AI systems in ways that positively shape societal outcomes into the future."

Maas, 2022

2. The Promise of Compute



Compute in the AI Production Function



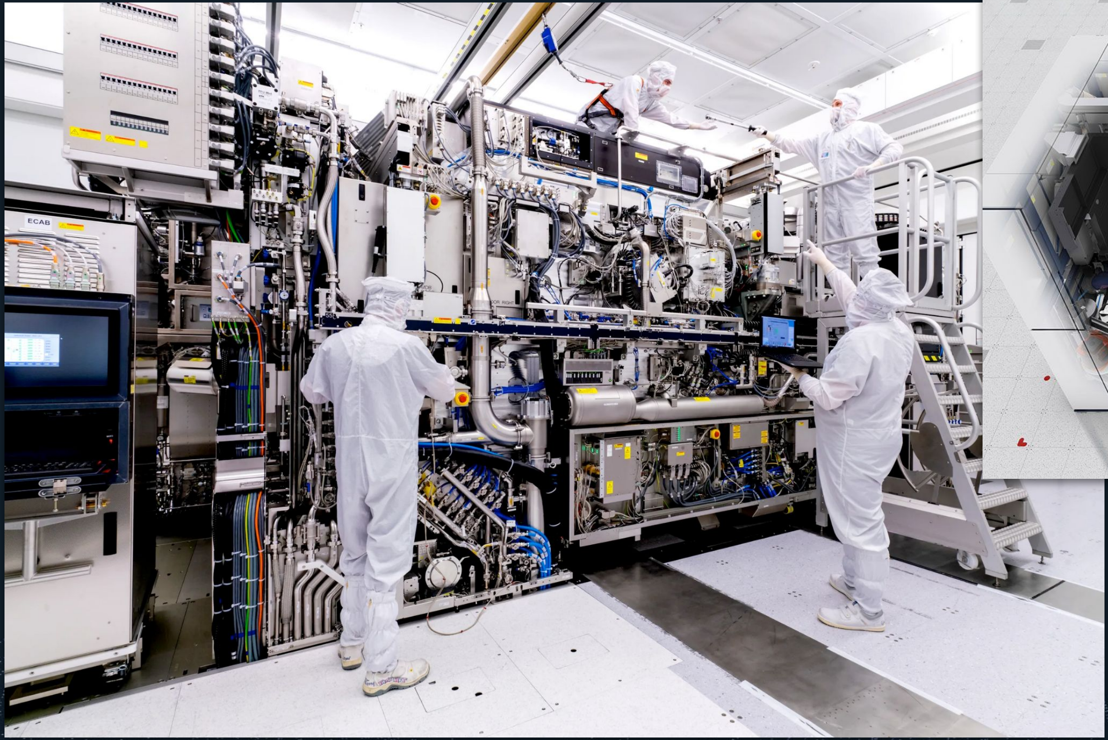
Feasibility: Compute is governable

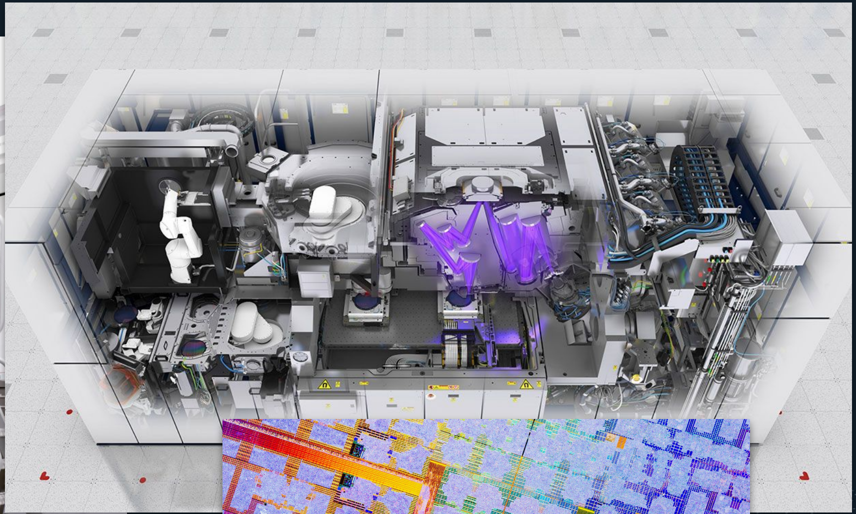
A. Feasibility: *Compute is governable*

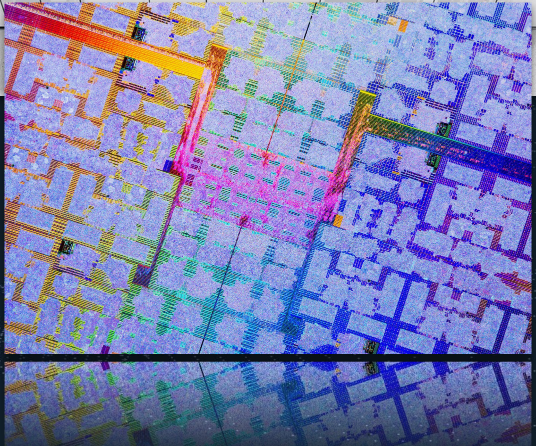
It is possible to monitor and shape who has access to computational resources and, to some extent, how they are used.

- Rivalry and Excludability
- Features of the Compute Supply Chain
- Quantifiability

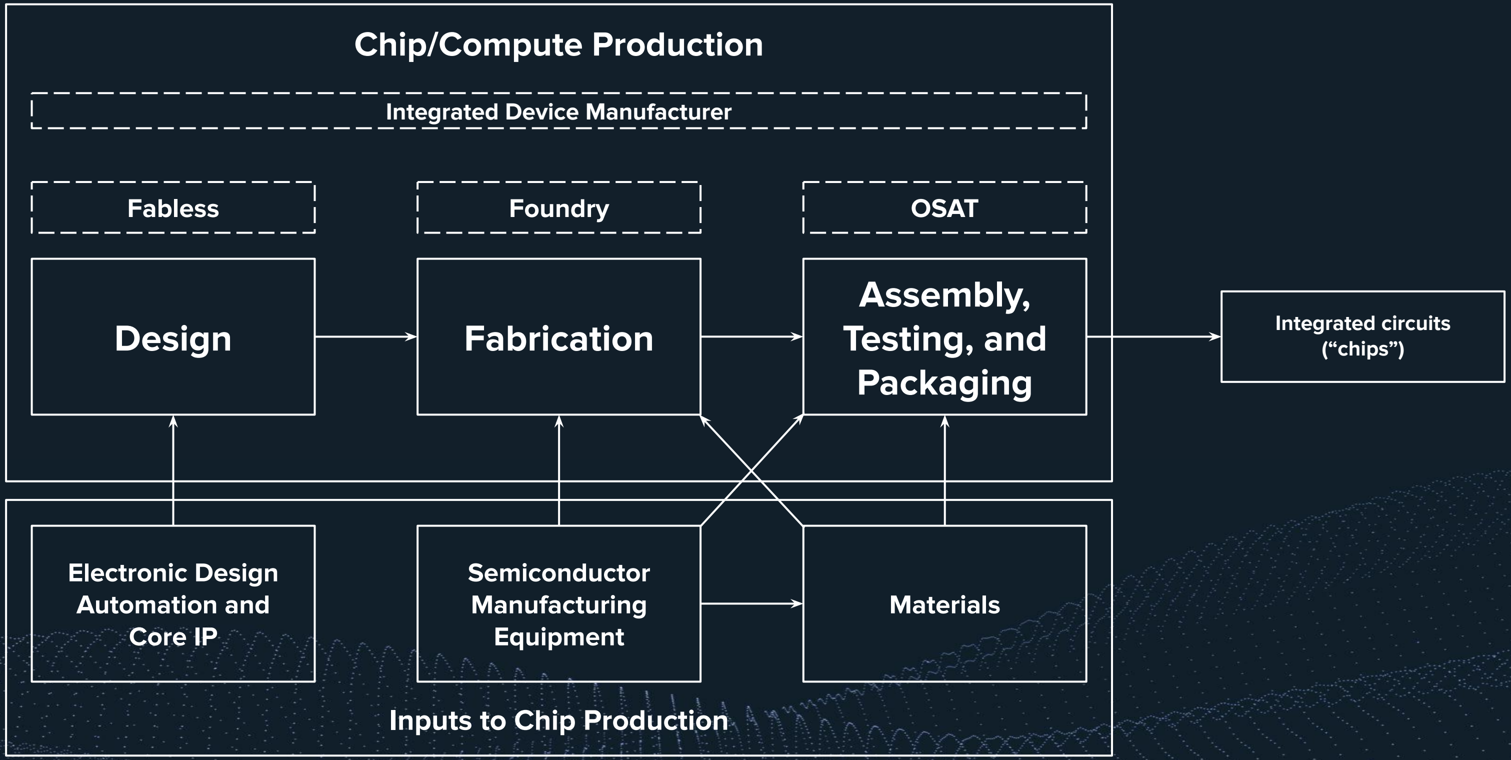
The World’s Most Complex Product: Chips



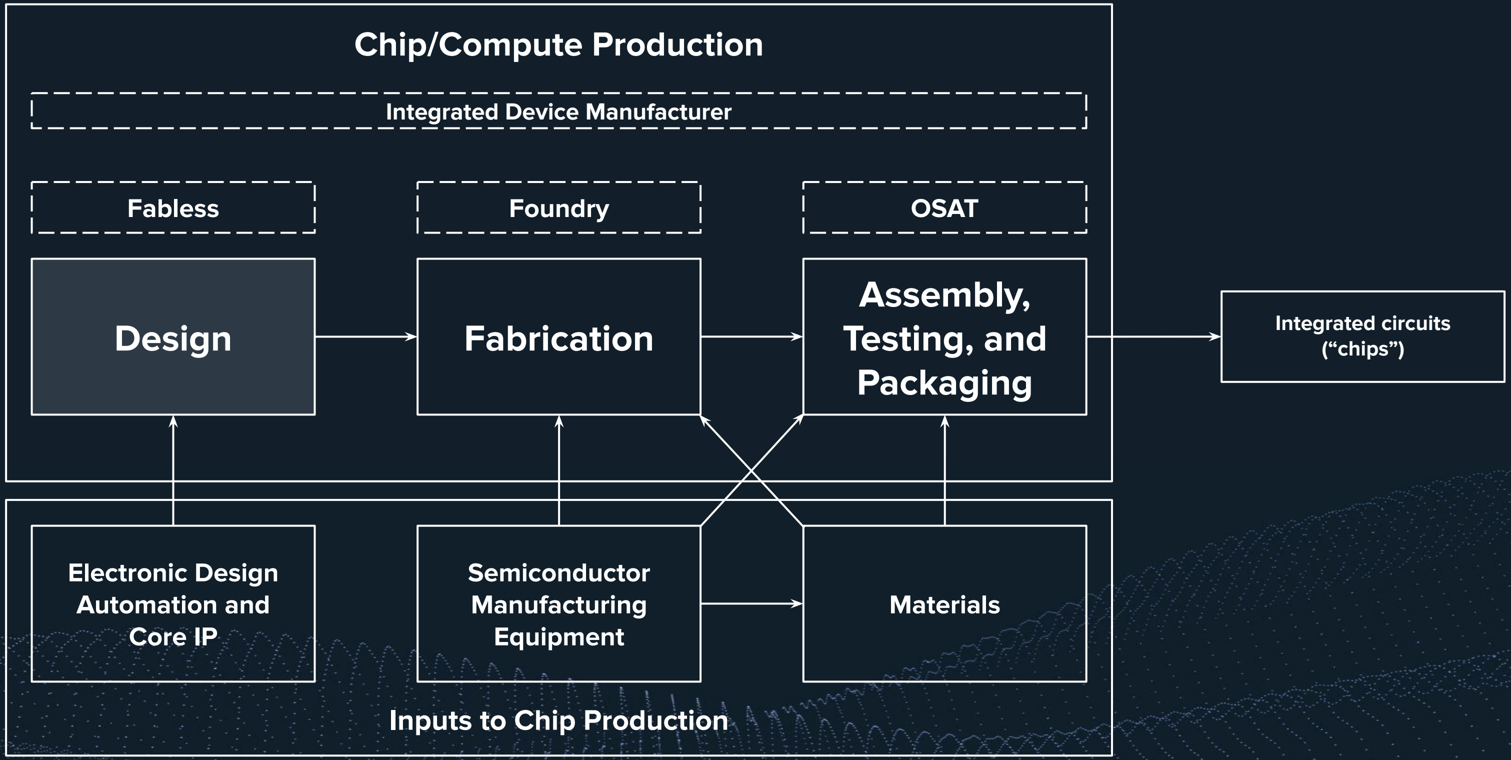




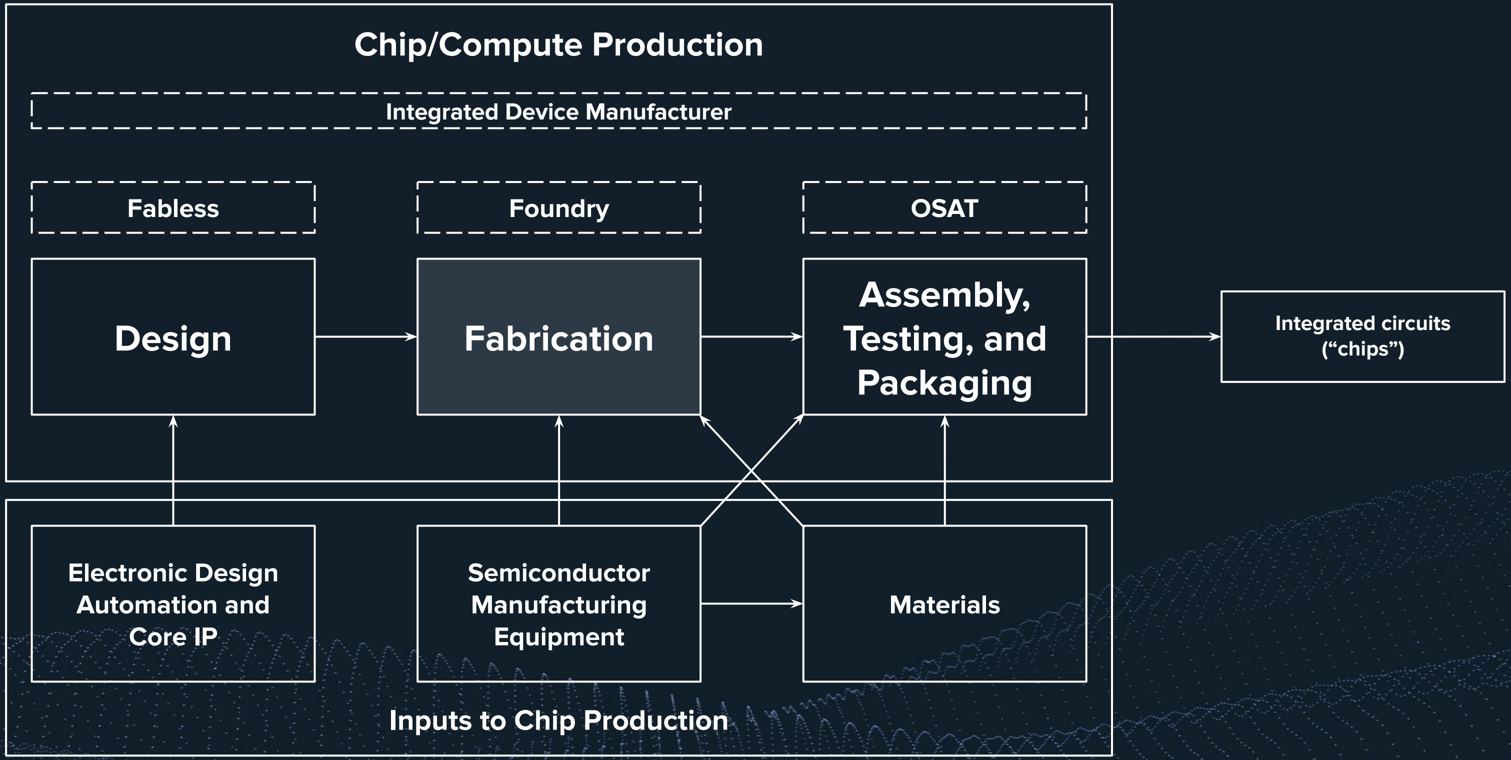
Compute Production



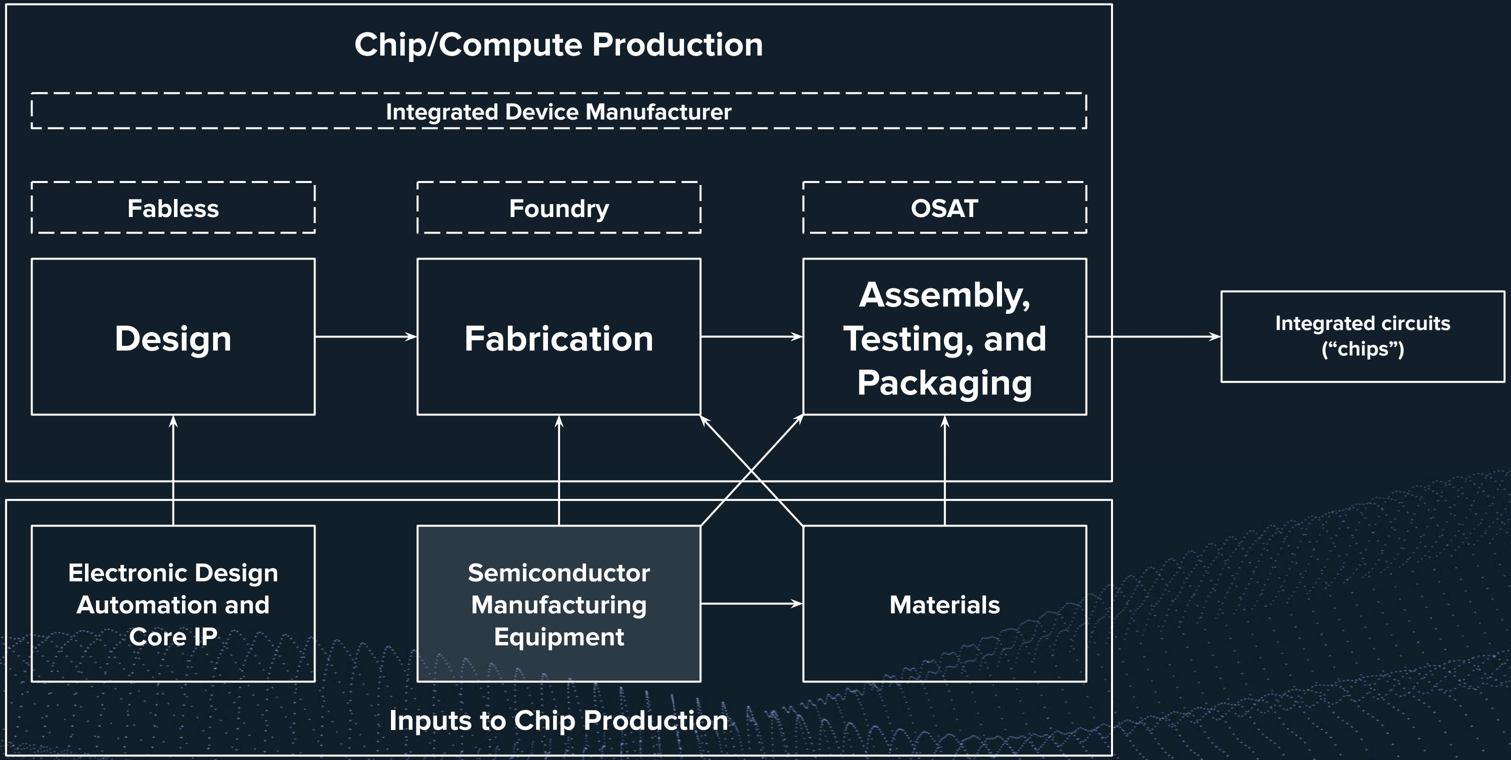
Compute Production



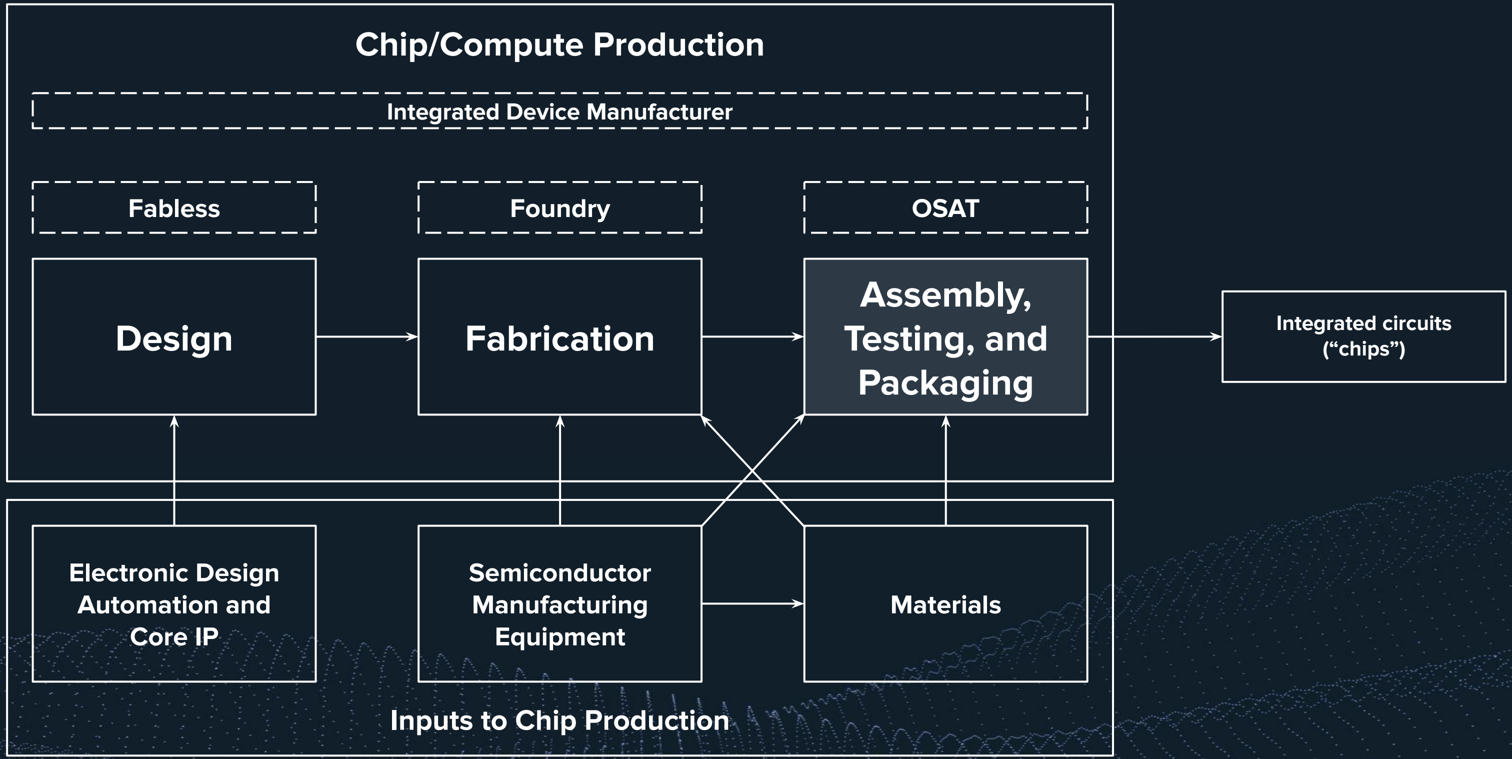
Compute Production



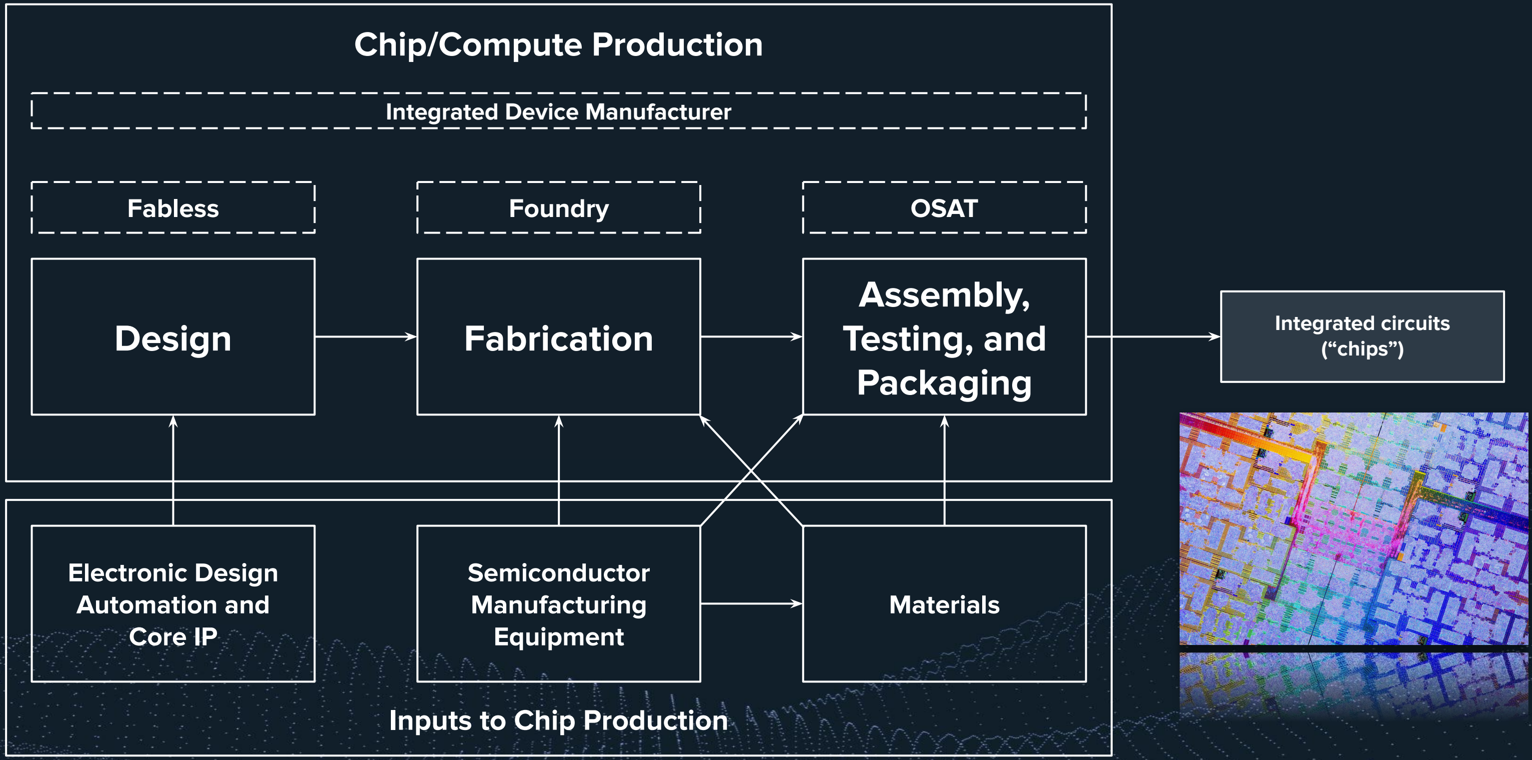
Compute Production



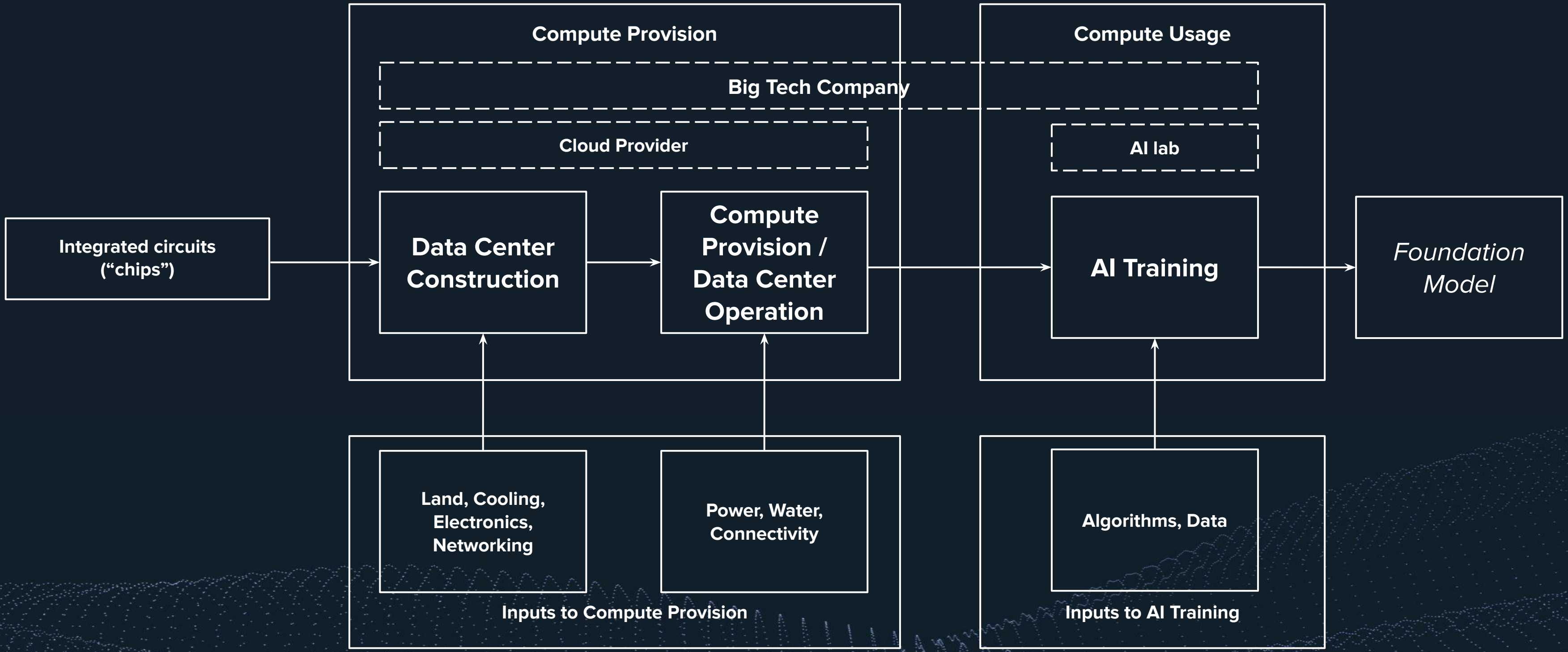
Compute Production



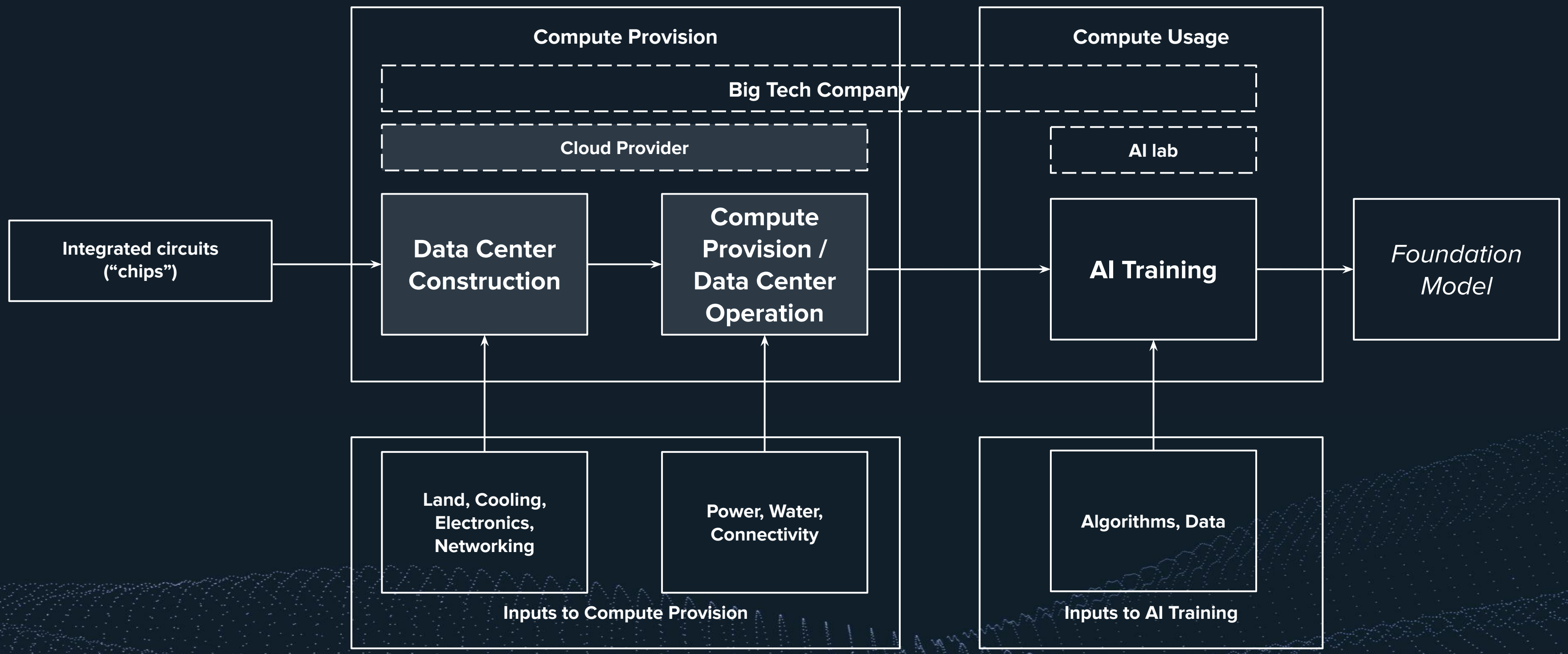
Compute Production



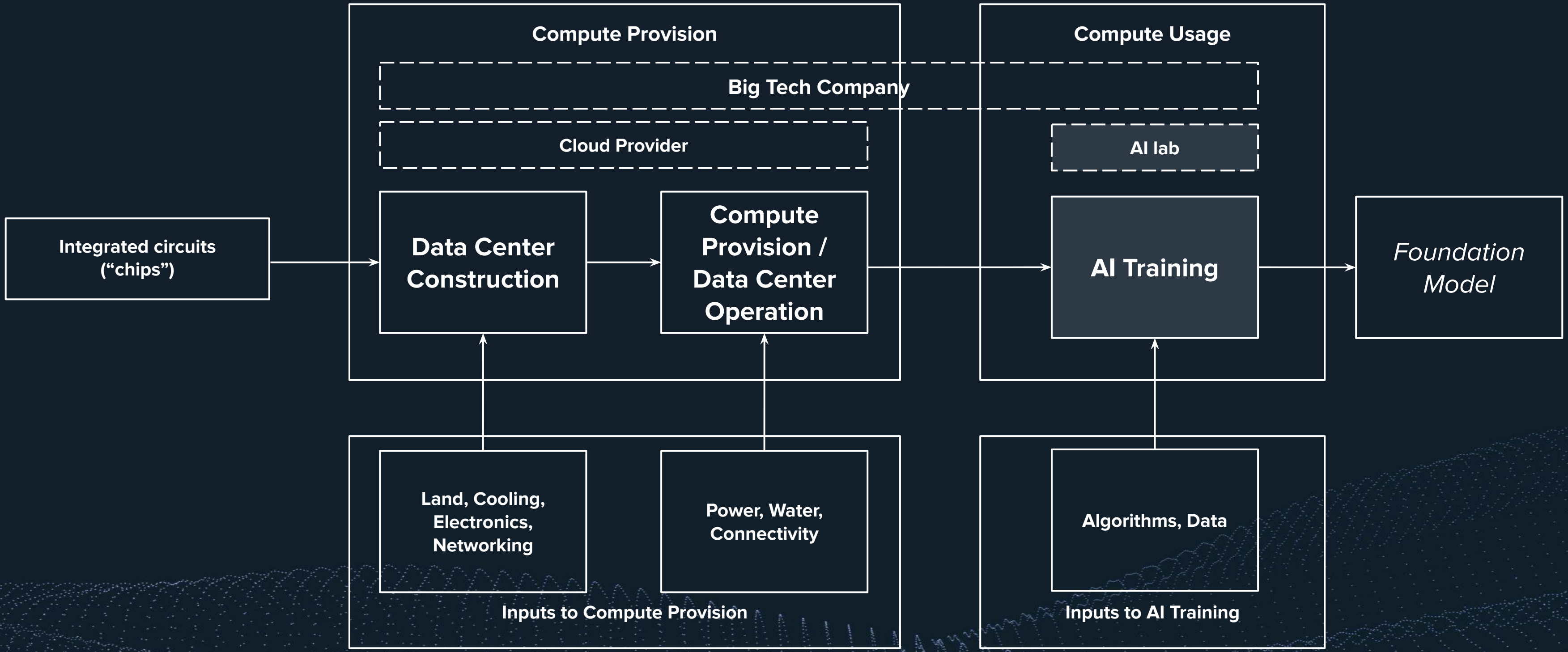
Compute Provision and Usage



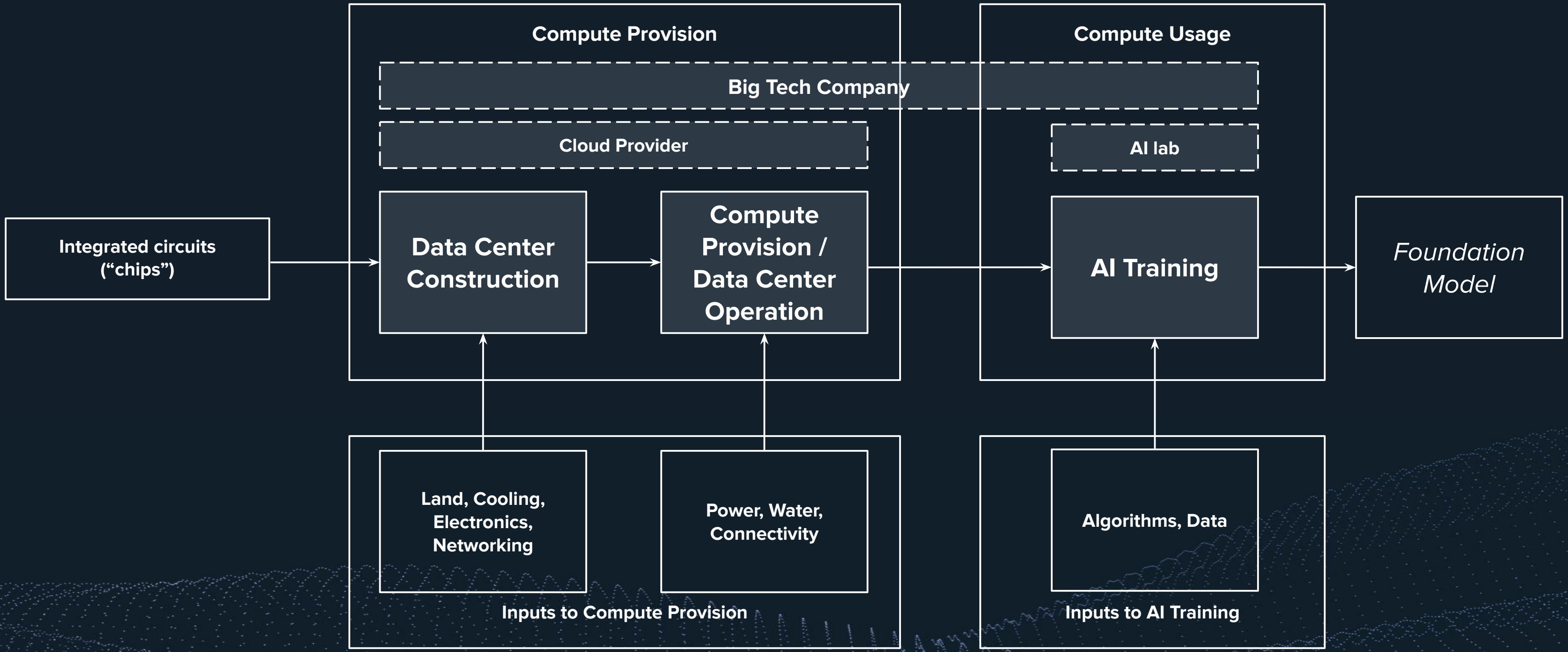
Compute Provision and Usage



Compute Provision and Usage



Compute Provision and Usage

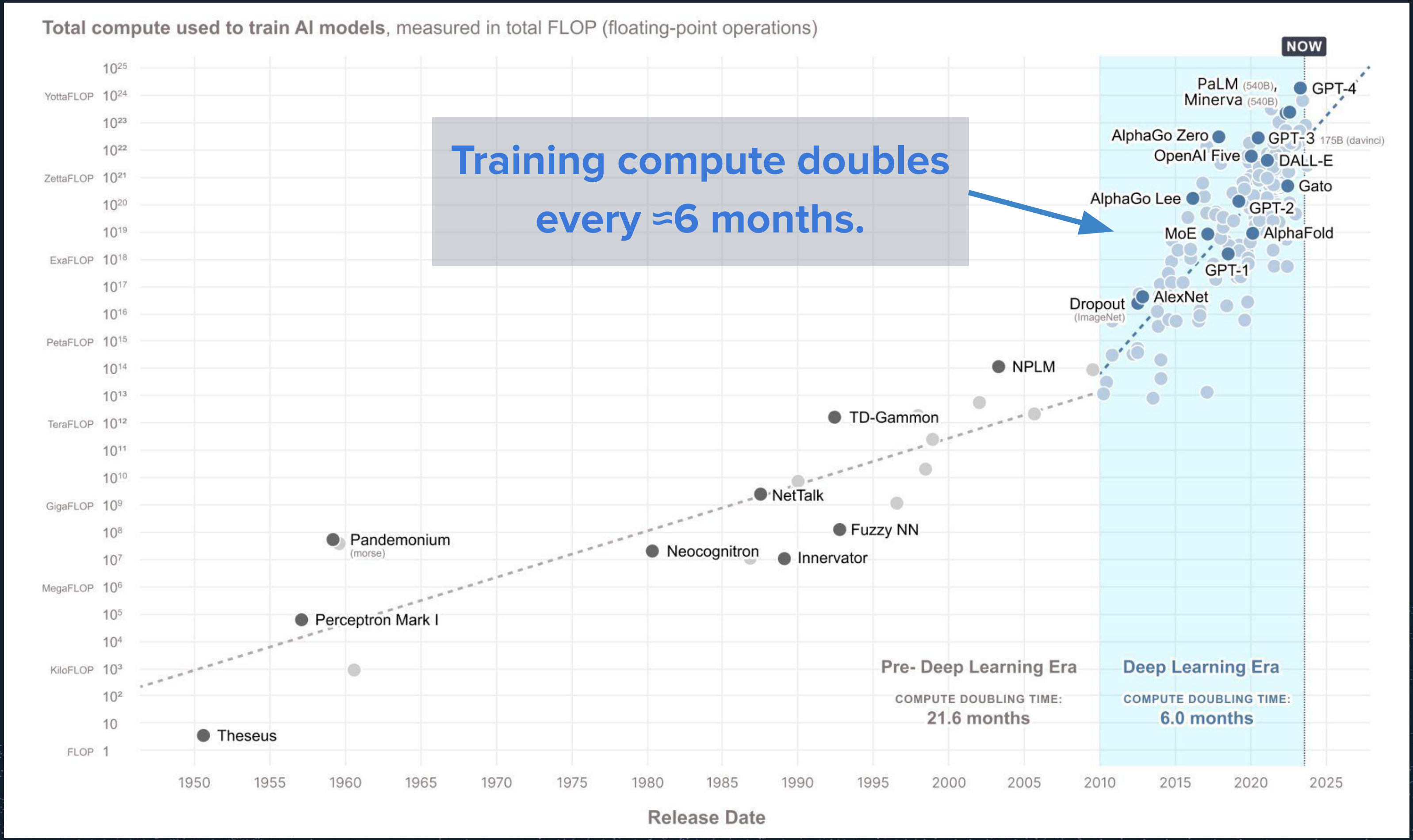


Feasibility: Compute is governable

A. Feasibility: ***Compute is governable***

It is possible to monitor and shape who has access to computational resources and, to some extent, how they are used.

- Rivalry and Excludability
- Features of the Compute Supply Chain
- Quantifiability



Sevilla et al., 2022

Efficacy: Compute is indicative of AI capabilities

A. Feasibility: *Compute is governable*

- Rivalry and Excludability
- Features of the Compute Supply Chain
- Quantifiability

B. Efficacy: *Compute is indicative of AI capabilities*

By observing, regulating, or influencing an entity's access to compute, one can predict and modulate actors' access to AI capabilities.

Why Governing Compute is Promising for Governing AI

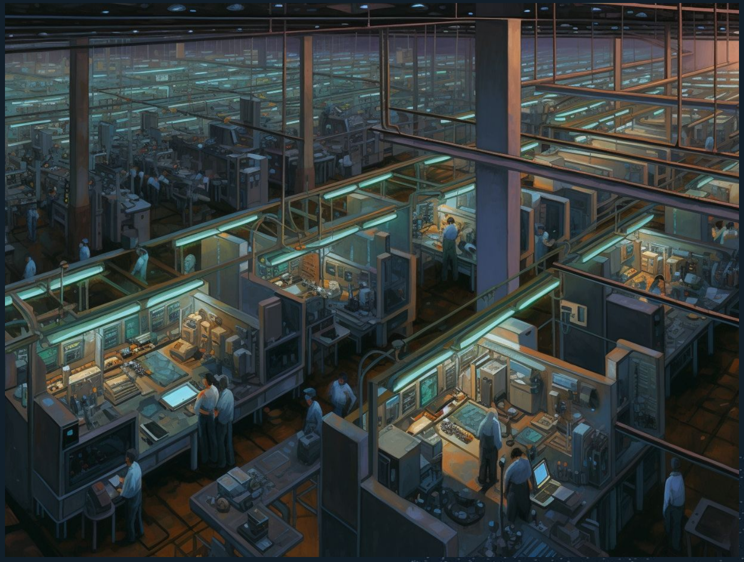
A. Feasibility: ***Compute is governable***

- Rivalry and Excludability
- Features of the Compute Supply Chain
- Quantifiability

B. Efficacy: ***Compute is indicative of AI capabilities***

By governing compute, you can govern AI capabilities.

3. Governance Capacities Enabled by Compute



1. Knowledge
2. Shaping
3. Enforcement

Brundage et al., forthcoming

1. **Knowledge**
2. Shaping
3. Enforcement

How actors use, develop, and deploy AI—and which actors are relevant.

1. Knowledge
2. **Shaping**
3. Enforcement

Direct and influence the trajectory of AI development and the distribution of AI capabilities among different actors.

1. Knowledge
2. Shaping
3. **Enforcement**

Respond to potential violations, such as an actor training an excessively risky AI system.

4. Examples of Compute Governance

US Semiconductor Export Restrictions

1. Block **access to high-end AI chips**
2. Block **designing AI chips domestically**
3. Block from **manufacturing advanced chips**
4. Block from **domestically producing semiconductor manufacturing equipment**
5. Block **“US persons” from supporting chip development**

The New York Times

Biden Administration Clamps Down on China’s Access to Chip Technology

The White House issued sweeping restrictions on selling semiconductors and chip-making equipment to China, an attempt to curb the country’s access to critical technologies.

Give this article 681

A semiconductor factory in Nantong, China. New limits on sales of semiconductor technology aim to slow the progress of Chinese military programs. Agence France-Presse — Getty Images

By Ana Swanson

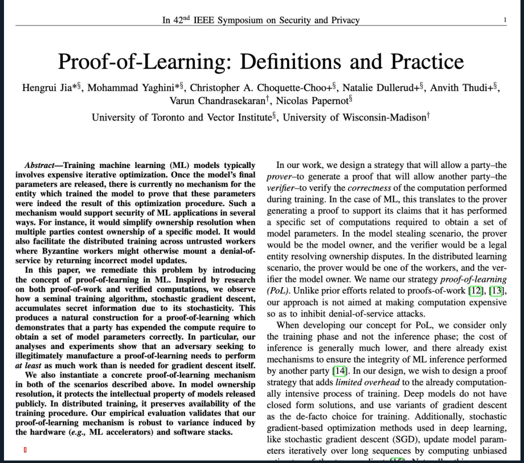
Oct. 7, 2022

Leverage Compute for Verification Mechanisms

- **Assurances & verifiable commitments** (across nations and actors)
- **Transparency**, e.g., transparent use of compute
- **Shared control**, e.g., on a joint AI project
- **Sanctions and restricted access**

Examples of Verification Mechanisms

- Proof-of-learning / training
- Proof-of-inference / deployment
- Proof-of-data
- (Verification of) properties of training runs
- Or proof-of-*non*-learning?



In 42nd IEEE Symposium on Security and Privacy 1

Proof-of-Learning: Definitions and Practice

Hengrui Jia*§, Mohammad Yaghini*§, Christopher A. Choquette-Choo+§, Natalie Dullerud+§, Anvith Thudi+§, Varun Chandrasekaran†, Nicolas Papernot§

University of Toronto and Vector Institute§, University of Wisconsin-Madison†

***Abstract*—Training machine learning (ML) models typically involves expensive iterative optimization. Once the model's final parameters are released, there is currently no mechanism for the entity which trained the model to prove that these parameters were indeed the result of this optimization procedure. Such a mechanism would support security of ML applications in several ways. For instance, it would simplify ownership resolution when multiple parties contest ownership of a specific model. It would also facilitate the distributed training across untrusted workers where Byzantine workers might otherwise mount a denial-of-service by returning incorrect model updates.**

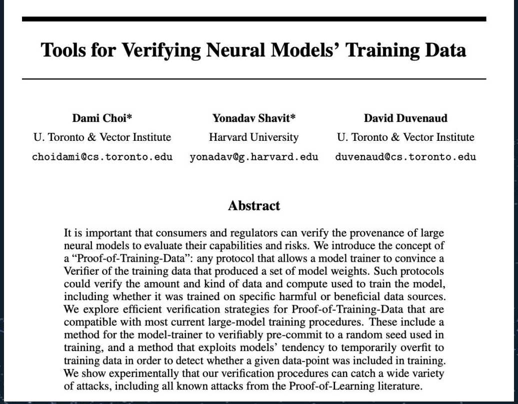
In this paper, we remediate this problem by introducing the concept of proof-of-learning in ML. Inspired by research on both proof-of-work and verified computations, we observe how a seminal training algorithm, stochastic gradient descent, accumulates secret information due to its stochasticity. This produces a natural construction for a proof-of-learning which demonstrates that a party has expended the compute require to obtain a set of model parameters correctly. In particular, our analyses and experiments show that an adversary seeking to illegitimately manufacture a proof-of-learning needs to perform *at least* as much work than is needed for gradient descent itself.

We also instantiate a concrete proof-of-learning mechanism in both of the scenarios described above. In model ownership resolution, it protects the intellectual property of models released publicly. In distributed training, it preserves availability of the training procedure. Our empirical evaluation validates that our proof-of-learning mechanism is robust to variance induced by the hardware (*e.g.,* ML accelerators) and software stacks.

In our work, we design a strategy that will allow a party–the *prover*–to generate a proof that will allow another party–the *verifier*–to verify the *correctness* of the computation performed during training. In the case of ML, this translates to the prover generating a proof to support its claims that it has performed a specific set of computations required to obtain a set of model parameters. In the model stealing scenario, the prover would be the model owner, and the verifier would be a legal entity resolving ownership disputes. In the distributed learning scenario, the prover would be one of the workers, and the verifier the model owner. We name our strategy *proof-of-learning (PoL)*. Unlike prior efforts related to proofs-of-work [12], [13], our approach is not aimed at making computation expensive so as to inhibit denial-of-service attacks.

When developing our concept for PoL, we consider only the training phase and not the inference phase; the cost of inference is generally much lower, and there already exist mechanisms to ensure the integrity of ML inference performed by another party [14]. In our design, we wish to design a proof strategy that adds *limited overhead* to the already computationally intensive process of training. Deep models do not have closed form solutions, and use variants of gradient descent as the de-facto choice for training. Additionally, stochastic gradient-based optimization methods used in deep learning, like stochastic gradient descent (SGD), update model parameters iteratively over long sequences by computing unbiased

Jia et al., 2021



Tools for Verifying Neural Models' Training Data

Dami Choi*	Yonadav Shavit*	David Duvenaud
U. Toronto & Vector Institute	Harvard University	U. Toronto & Vector Institute
choidami@cs.toronto.edu	yonadav@g.harvard.edu	duvenaud@cs.toronto.edu

Abstract

It is important that consumers and regulators can verify the provenance of large neural models to evaluate their capabilities and risks. We introduce the concept of a "Proof-of-Training-Data": any protocol that allows a model trainer to convince a Verifier of the training data that produced a set of model weights. Such protocols could verify the amount and kind of data and compute used to train the model, including whether it was trained on specific harmful or beneficial data sources. We explore efficient verification strategies for Proof-of-Training-Data that are compatible with most current large-model training procedures. These include a method for the model-trainer to verifiably pre-commit to a random seed used in training, and a method that exploits models' tendency to temporarily overfit to training data in order to detect whether a given data-point was included in training. We show experimentally that our verification procedures can catch a wide variety of attacks, including all known attacks from the Proof-of-Learning literature.

Choi & Shavit et al., 2023

5. Compute and the Governance of AI

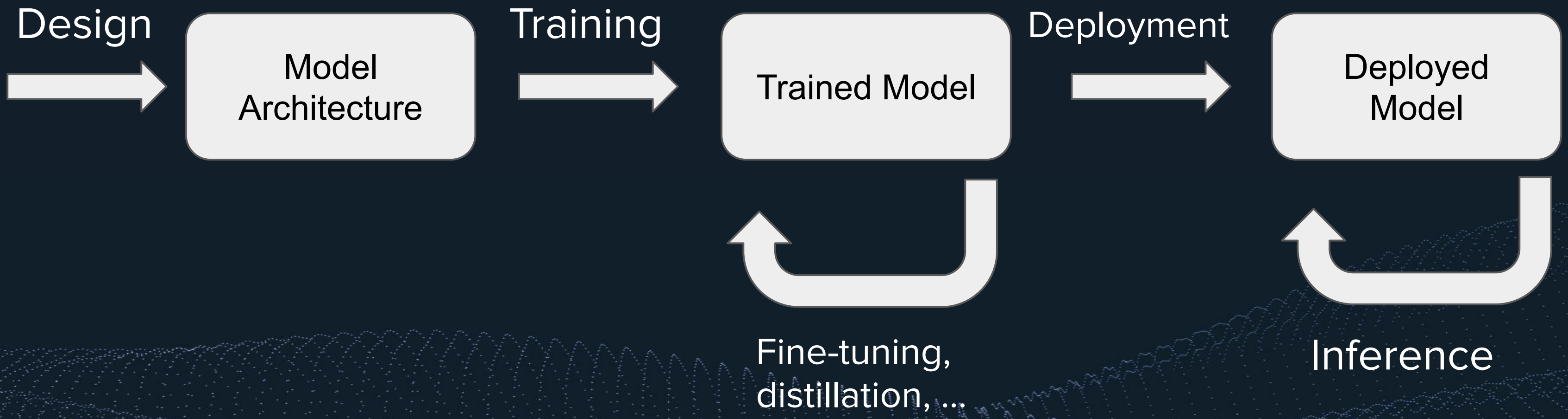
AI Governance Definition

"The study and shaping of local and global governance systems — including norms, policies, laws, processes, politics, and institutions — that affect the research, development, deployment, and use of existing and future AI systems in ways that positively shape societal outcomes into the future."

Maas, 2022

Governance throughout the AI Lifecycle

Development | **Deployment**



Governance throughout the AI Lifecycle

Development

Pre-emptive / pre-training authorization

Design → Model Architecture

Training → Trained Model

Training Compute Verification

Proof that model only used X FLOP

AI Chip Export Restrictions

Stop/hinder actors from training such systems

Training Compute Threshold

Which models are of concern?

Actor's Compute Capacity

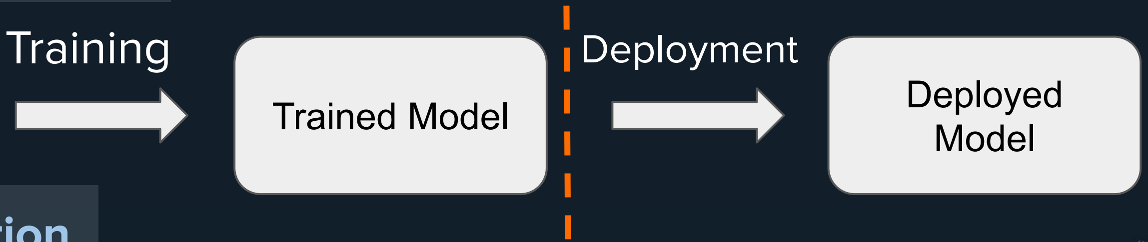
Who's able to train this model?

Governance throughout the AI Lifecycle

Licensing of AI Systems
Cloud providers verify the (electronic) license of an AI system before deploying

Deployment

Pre-deployment authorization



Pre-Deployment Notification
Governments and other entities receive a notification

Verify Model Authenticity during Evaluations (which is required for deployment)

Governance throughout the AI Lifecycle

Deployment

Compute Providers Shuts Down AI System to prevent continuous harm

Retrospective Deployment Correction

Compute Providers Identify Developer (enable post-incident attribution)

Training

Trained Model

Deployment

Deployed Model

Compute Providers Identify Deployer (enable post-incident attribution)

Conclusions

- Governing compute is ***feasible, effective and valuable*** but ***alone not sufficient***
- Enabling AI governance capacities that would otherwise be difficult to achieve: ***knowledge***, ***shaping***, ***enforcement***
- Mechanisms for verifiable claims that can enable more trust across actors
- Compute is already being used as a governance node — we should improve our understanding and build ***more nuanced instruments***

Lennart Heim
lennart.heim@governance.ai
@ohlennart
https://heim.xyz

Centre for the Governance of AI