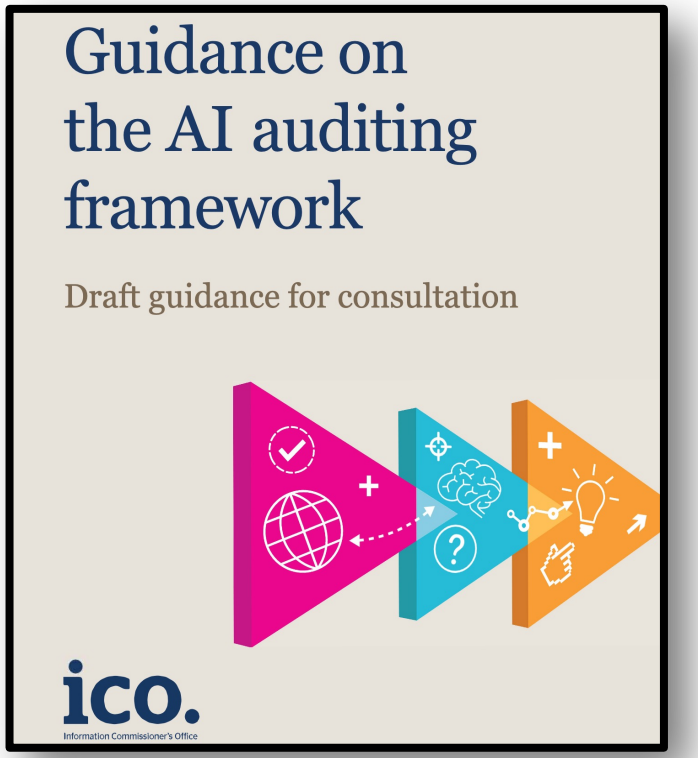
UNIVERSITY OF WATERLOO

Towards Verifiable ML Properties using Trusted Hardware

Vasisht Duddu *vasisht.duddu@uwaterloo.ca* *vasishtduddu.github.io*

(Joint work with Adam Caulfield, Prach Chantasanthitam, N. Asokan, Anudeep Das, Lachlan Gunn)

Regulations in Machine Learning (ML)

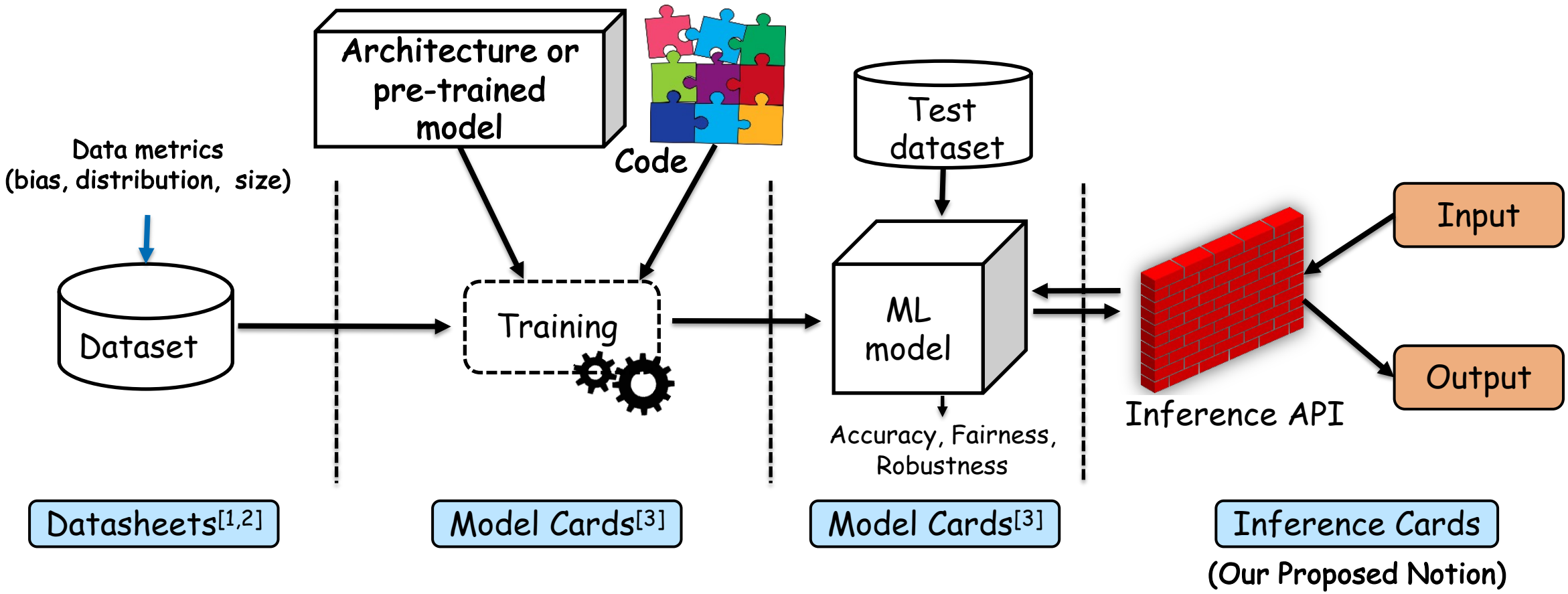


"Establish a risk management system"
"Conduct data governance"
"An Accountability framework"
"Appropriate accuracy, robustness, fairness"

Practitioners must demonstrate properties about models, training, and datasets

- Need mechanisms for transparency, accountability, and show compliance

Advertising ML Properties for Transparency



Collectively, refer to them as "ML property cards"

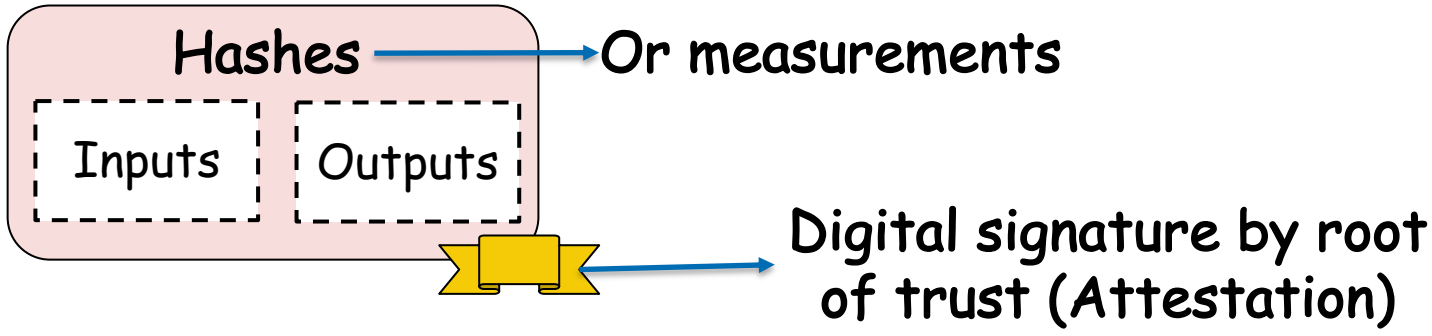
[1] Gebru et al. *Datasheets for datasets*. Communications of ACM. 2021.
[2] Pushkarna et al. *Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI*. FaccT. 2022.
[3] Mitchell et al. *Model Cards for Model Reporting*. FaccT. 2019.

Need Verifiable ML Property Cards

Malicious practitioner can make false claims about model/data (e.g., HuggingFace[1])

ML property attestation[2]

- Prover (e.g., model trainer) demonstrates ML properties to Verifier (e.g., regulator)
- Without revealing proprietary model and training data → Confidentiality
- Attestation = hashes of inputs and outputs signed by root of trust to certify properties



[1] Mithril-Security. *PoisonGPT: How to poison LLM supply chain on HuggingFace*. 2023.
[2] Duddu et al. *Attesting Distributional Properties of Machine Learning Training Data*. ESORICS. 2024.

Can we use Software-based Mechanisms?

ML-based Mechanisms (proof of learning[1], re-purposing privacy attacks[2])

- Neither effective nor robust[3,4]
- Ensuring robustness further degrades effectiveness

Cryptographic Mechanisms (multi-party computation[2], zero-knowledge proofs[5,6])

- Neither efficient[5,6] nor versatile (modify standard ML operations)

Neither are practical for current ecosystem of ML models

[1] Jia et al. *Proof of Learning: Definitions and Practice*. IEEE S&P. 2021.
[2] Duddu et al. *Attesting Distributional Properties of Machine Learning Training Data*. ESORICS. 2024.
[3] Zhang et al. *"Adversarial Examples" for Proof- of-Learning*. IEEE S&P. 2022.
[4] Fang et al. *Proof of Learning is more Broken than You Think*. IEEE EuroS&P. 2023.
[5] Sun et al. *zkLLMs: Zero Knowledge Proofs for Large Language Models*. ACM CCS. 2024.
[6] Abbaszadeh et al. *Zero-Knowledge Proofs of Training for Deep Neural networks*. ACM CCS. 2024.

Can we use Hardware-based Mechanisms?

Use trusted execution environments (TEEs)

- Confidentiality and integrity guarantees of TEEs + Remote attestation

Need to efficiently run ML models in TEEs → Intel AMX[1], NVIDIA's H100 GPUs[2]

Assumptions

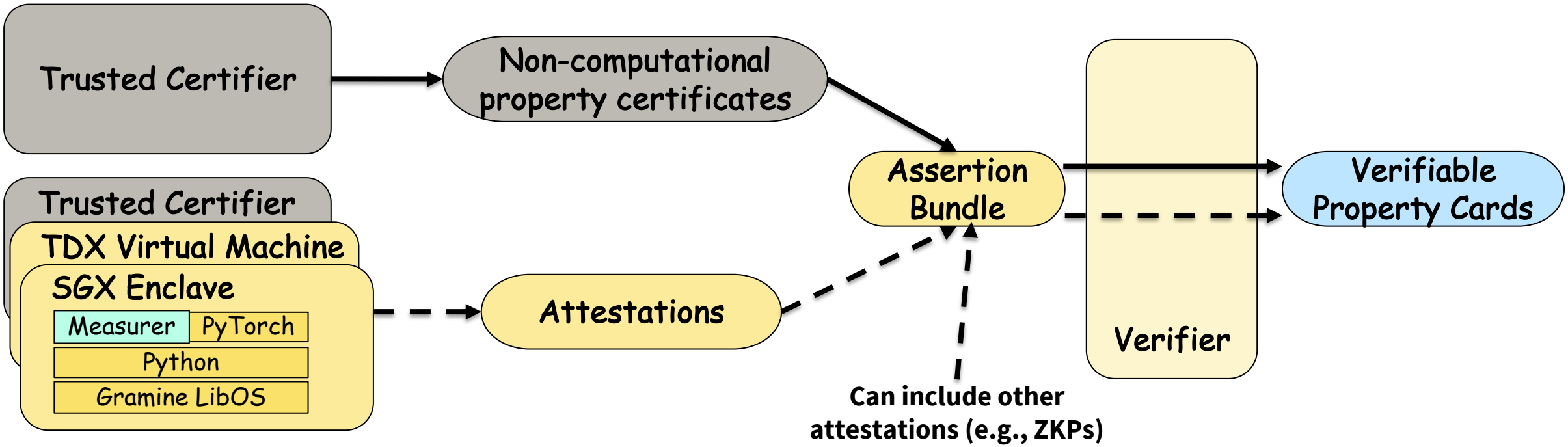
- Verifier does not trust Prover's software outside TEE (e.g., OS/hypervisor, apps)
- Root of trust
  → Hardware manufacturer (e.g., Intel, Nvidia)
  → Trusted certifiers (e.g., CIFAR) provide non-computational certificates
- Attestation keys are not leaked

[1] Google Cloud Team. *We tested Intel's AMX CPU accelerator for AI and here's what we learned.* 2024
[2] Zhu et al. *Confidential Computing on NVIDIA's H100 GPU: A Performance Benchmark Study*. 2024

Laminator and PAL*M Framework

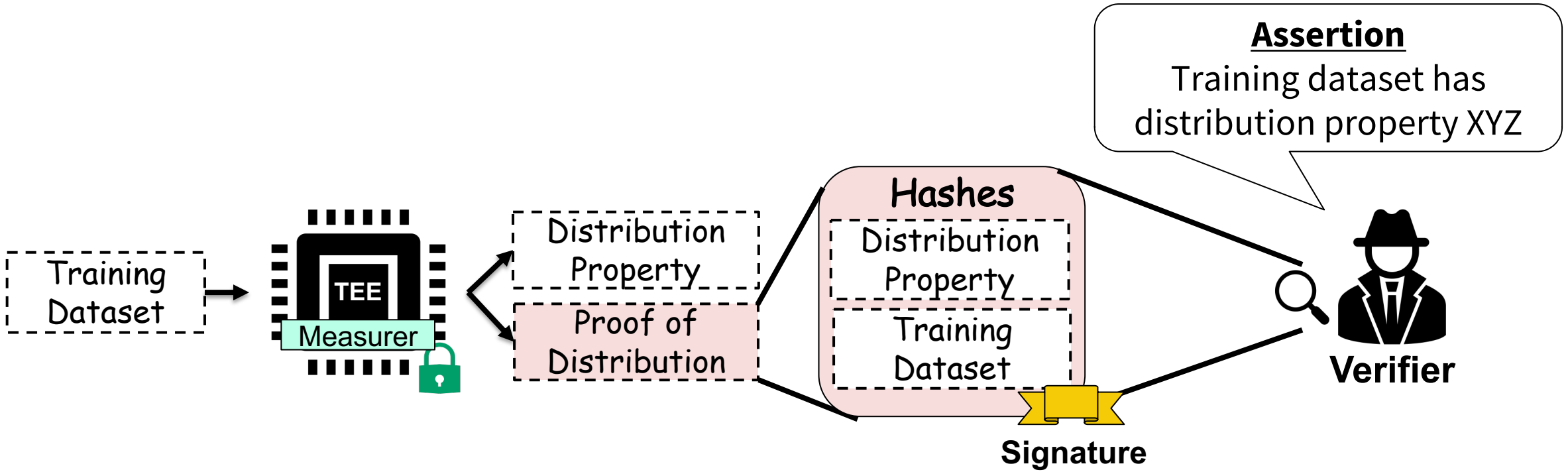
Property attestations

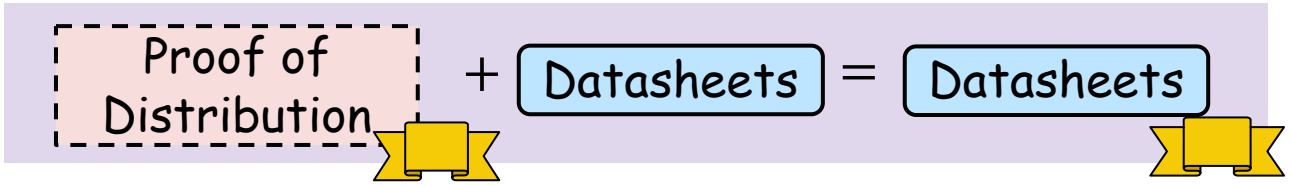
- Laminator[1]: SGX-based for classifiers
- PAL*M[2]: TDX-based for large generative models



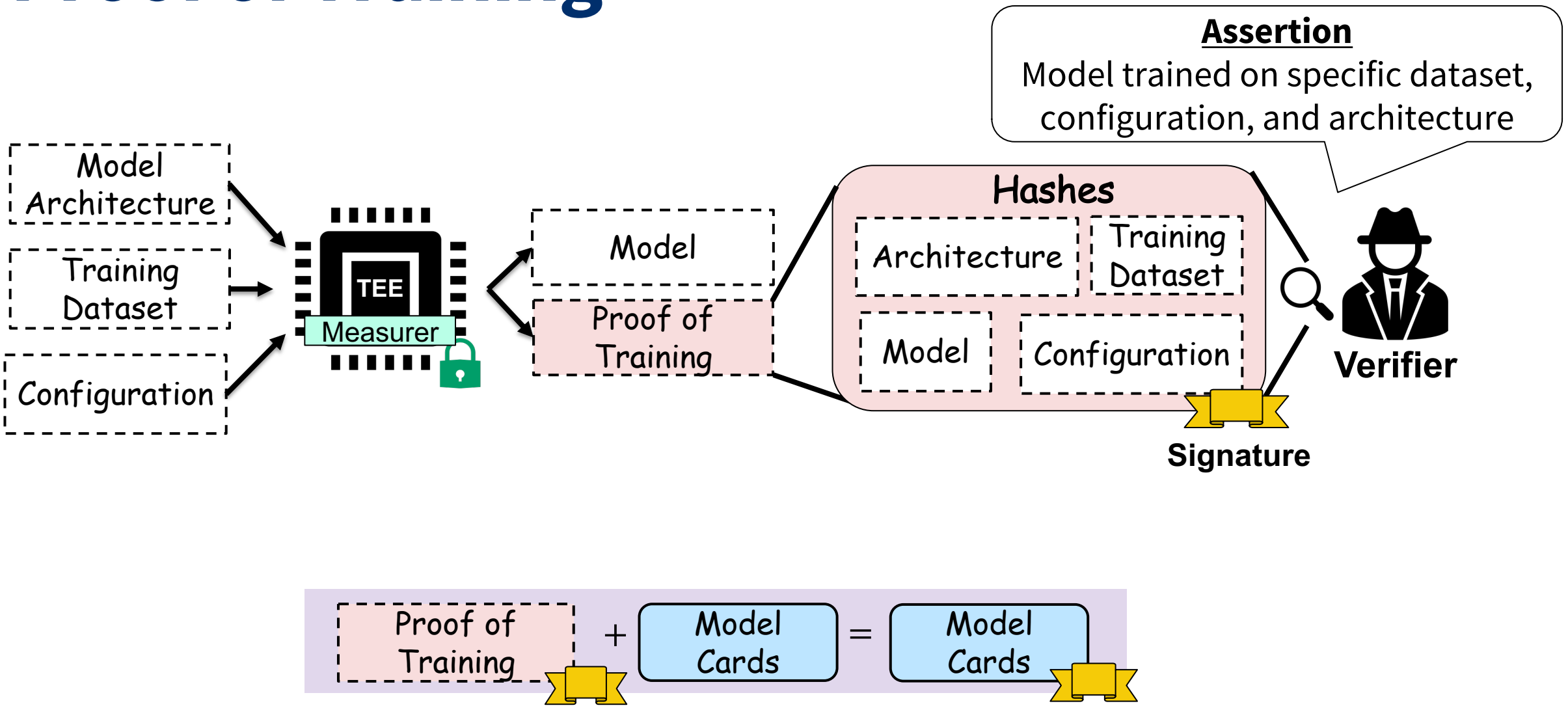
[1] Duddu et al. *Laminator: Verifiable ML property cards using hardware-assisted attestations*. CODASPY 2025
[2] Chantasantitam et al. *PAL*M: Property Attestation for Large Generative Models*. arXiv 2026

Proof of Attribute Distribution

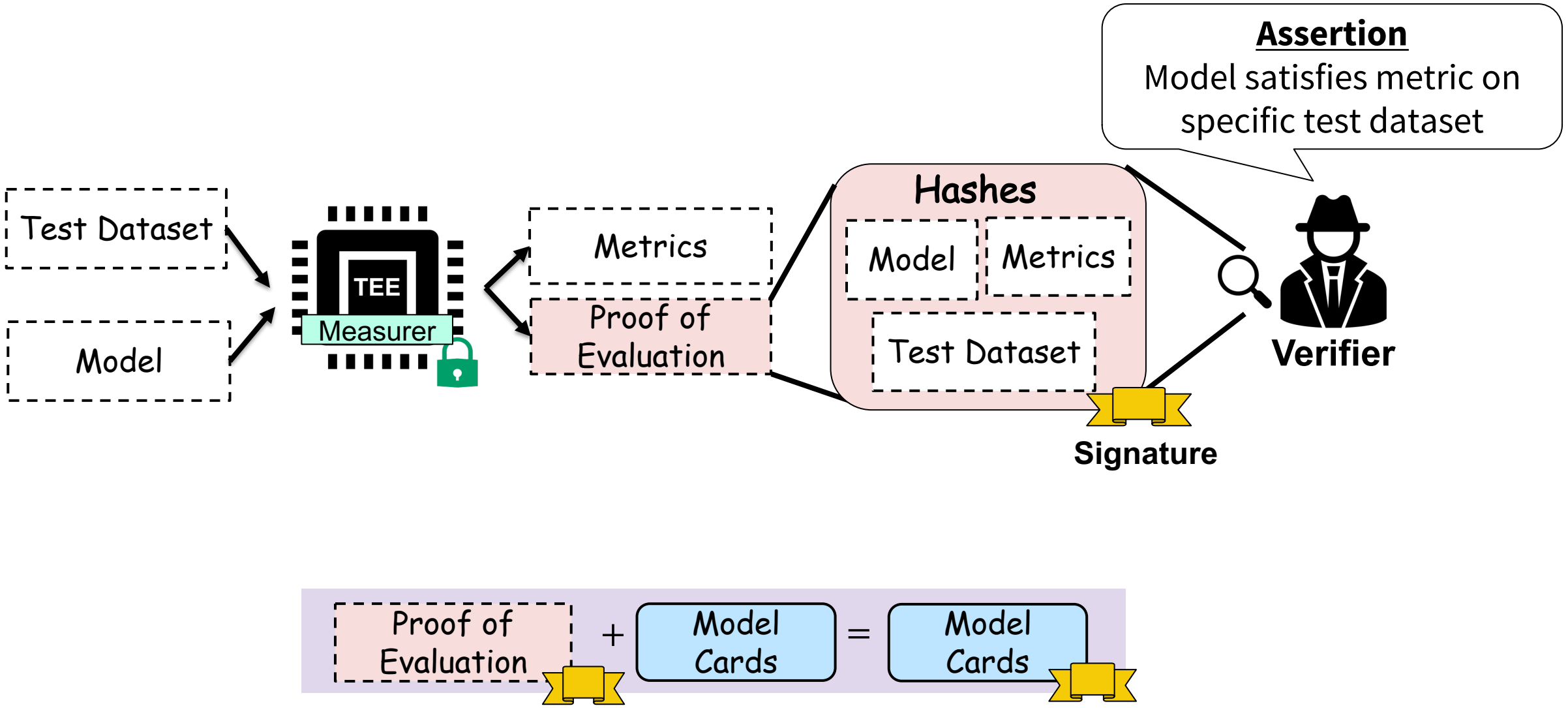




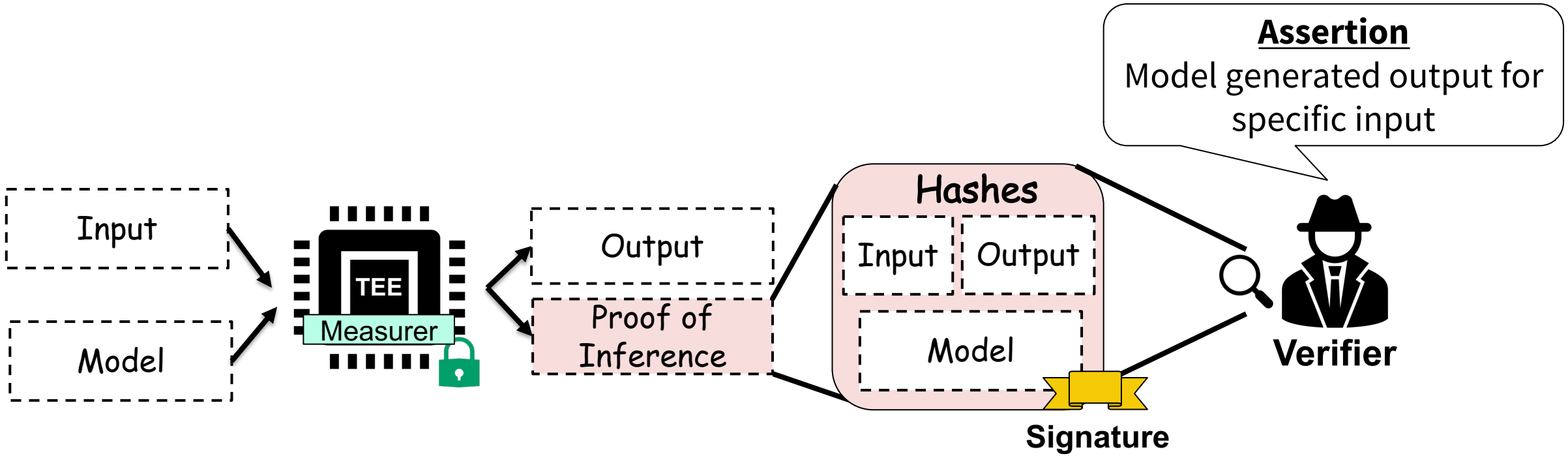
Proof of Training

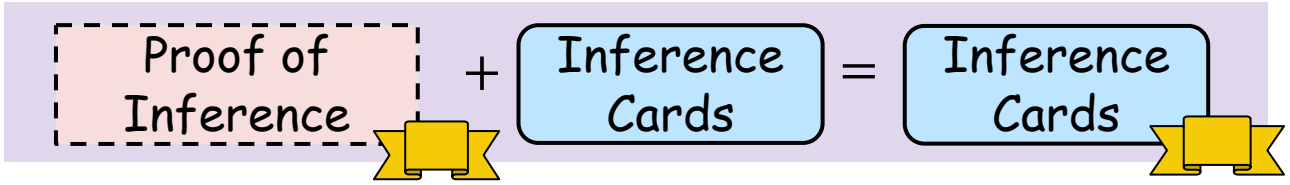


Proof of Evaluation



Proof of Inference





Efficiency of Generating Attestations

All attestations have reasonable overhead → low overhead (<5%)

- Both in “in-memory” and “memory-mapped” setting

Proof of Inference: High overhead between 39% and 3955%

- **Solution 1:** Amortizing overhead over several inferences
- **Solution 2**: Generate a signing keypair, attest it, and use for each inference
  - Overhead between 0.17% and 1.17%

40% overhead for “Memory-mapped” proof of distribution → One-time cost

Summary

Proactive demonstration of compliance without manual inspection by agencies

Looking forward

- Verifiable ecosystem of models and datasets across applications
- Runtime attestations for verifiable agents to monitor and check tool usage

https://ssg-research.github.io/mlsec/mlattestation

ESORICS'24 **CODASPY'25**

ArXiv'26
(In submission)