What does it take to catch a Chinchilla? Verifying Rules on Large-Scale Neural Network Training via Compute Monitoring

Yonadav Shavit
Harvard University
yonadav@g.harvard.edu

Presented by: Nandeeka Nayak

Question: How can we effectively regulate powerful ML models?

Assumptions

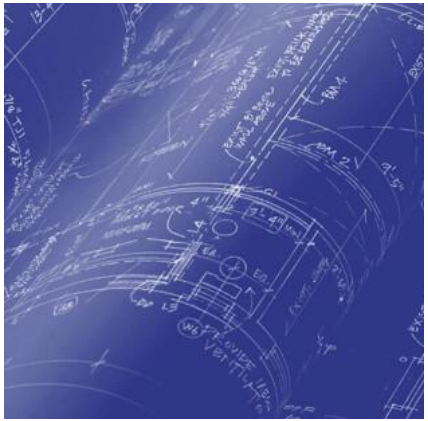
- Powerful models require large amounts of FLOPs to train
- Achieving this large number of FLOPs requires specialized accelerators with high interconnection bandwidth
- Monitoring these specialized accelerators will enable auditors to regulate these powerful models

Proposed Solution

1. Activity logging for machine learning training implemented in the firmware of ML chips
2. Inspection of the logs to ensure compliance with regulations
3. Monitoring of the supply chain to ensure that ML chips are compliant

Prover-Verifier Paradigm

Actually builds the building



Builder

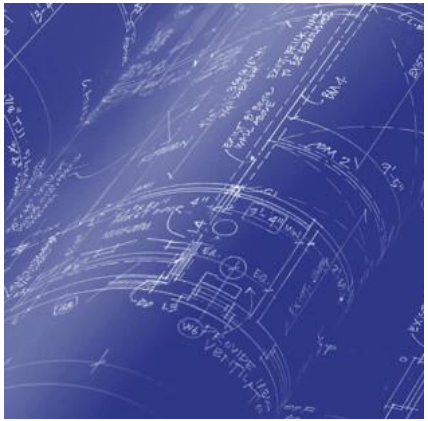
Provides proof of compliance

Checks the building is safe

Safety Inspector

Prover-Verifier Paradigm

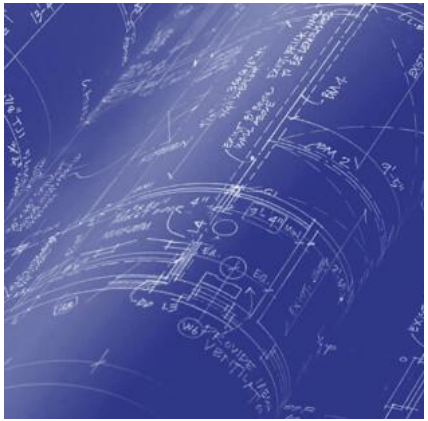
Prover-Verifier Paradigm



Builder

Prover-Verifier Paradigm

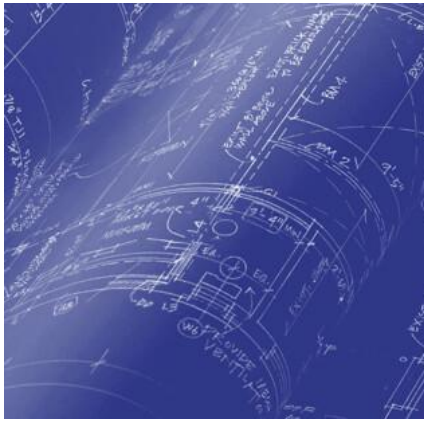
Actually builds the building



Builder

Prover-Verifier Paradigm

Actually builds the building

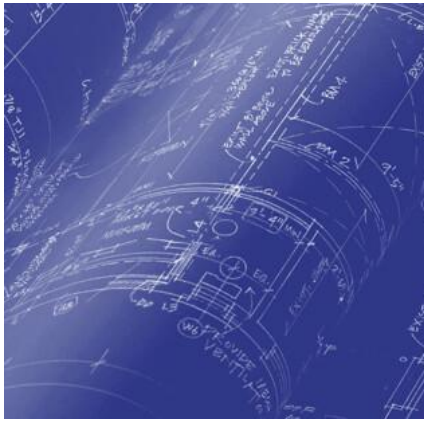


Builder

Safety Inspector

Prover-Verifier Paradigm

Actually builds the building



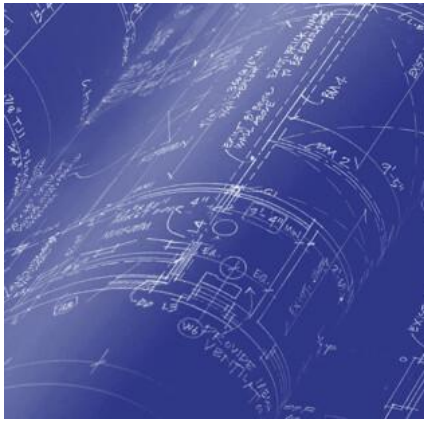
Builder

Checks the building is safe

Safety Inspector

Prover-Verifier Paradigm

Actually builds the building



Builder

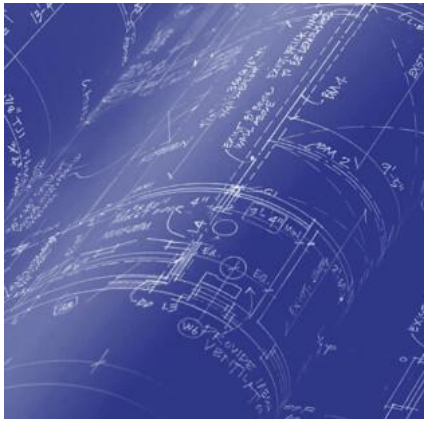
Provides proof of compliance

Checks the building is safe

Safety Inspector

Prover-Verifier Paradigm

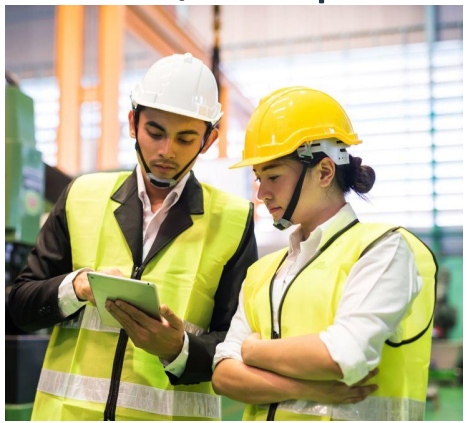
Trains the ML model



Prover

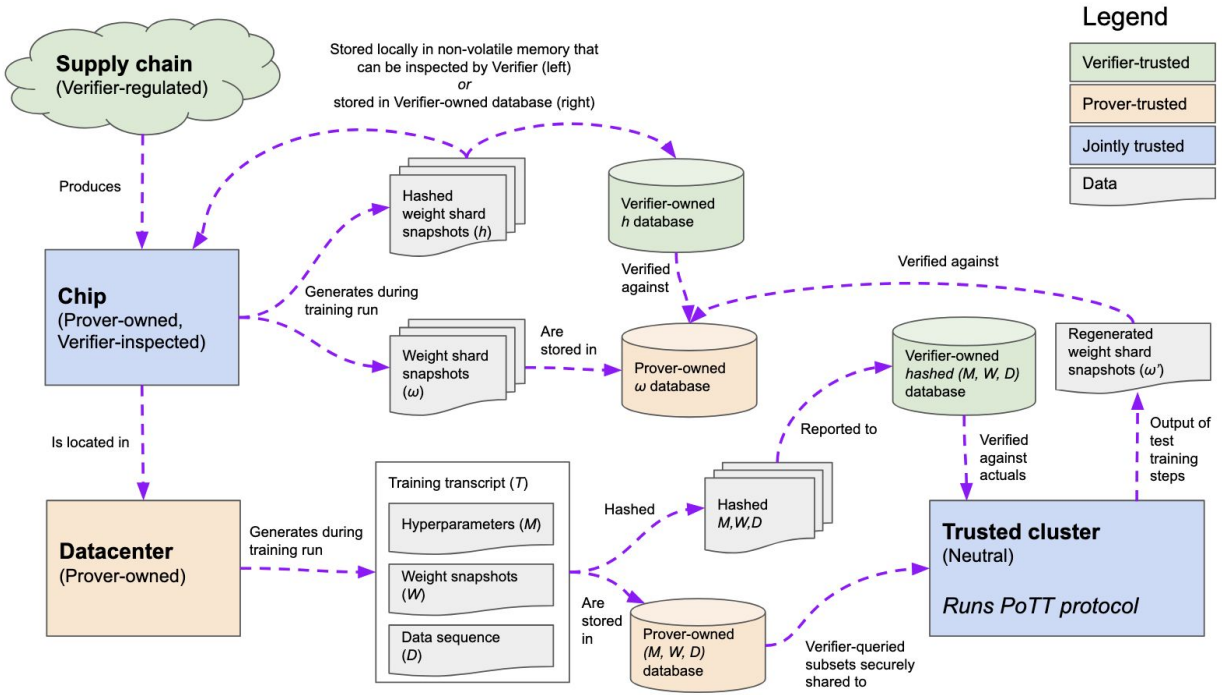
Provides proof of compliance

Checks the model is safe/compliant

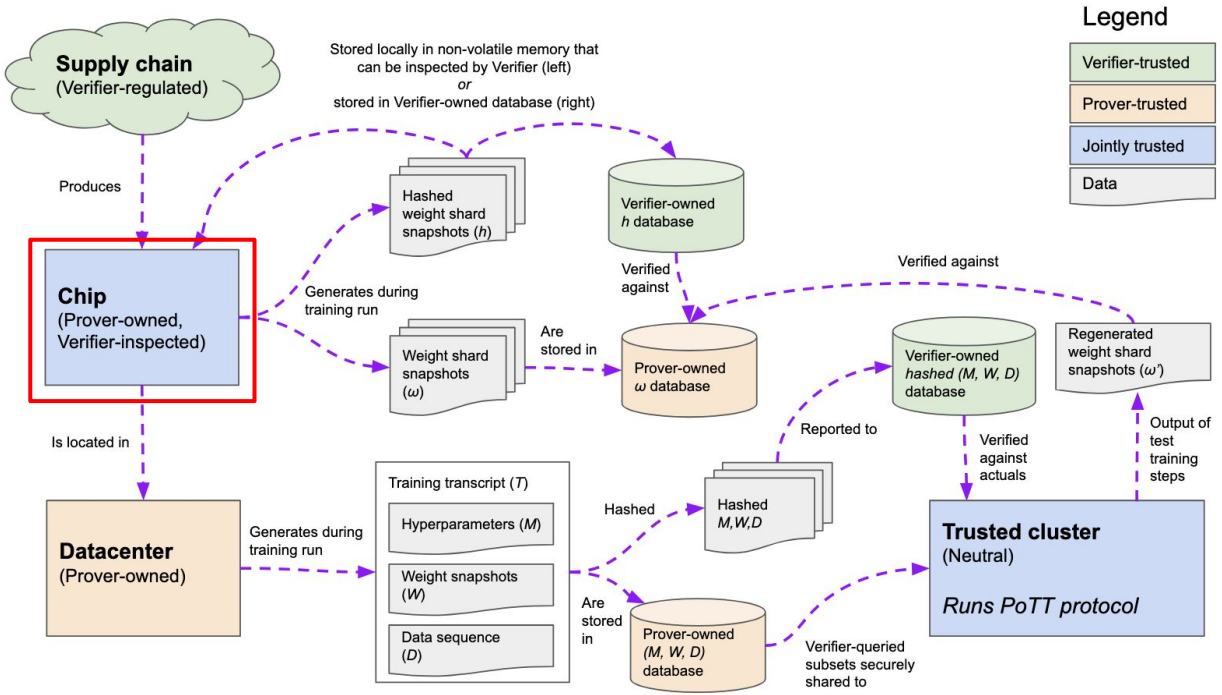


Verifier

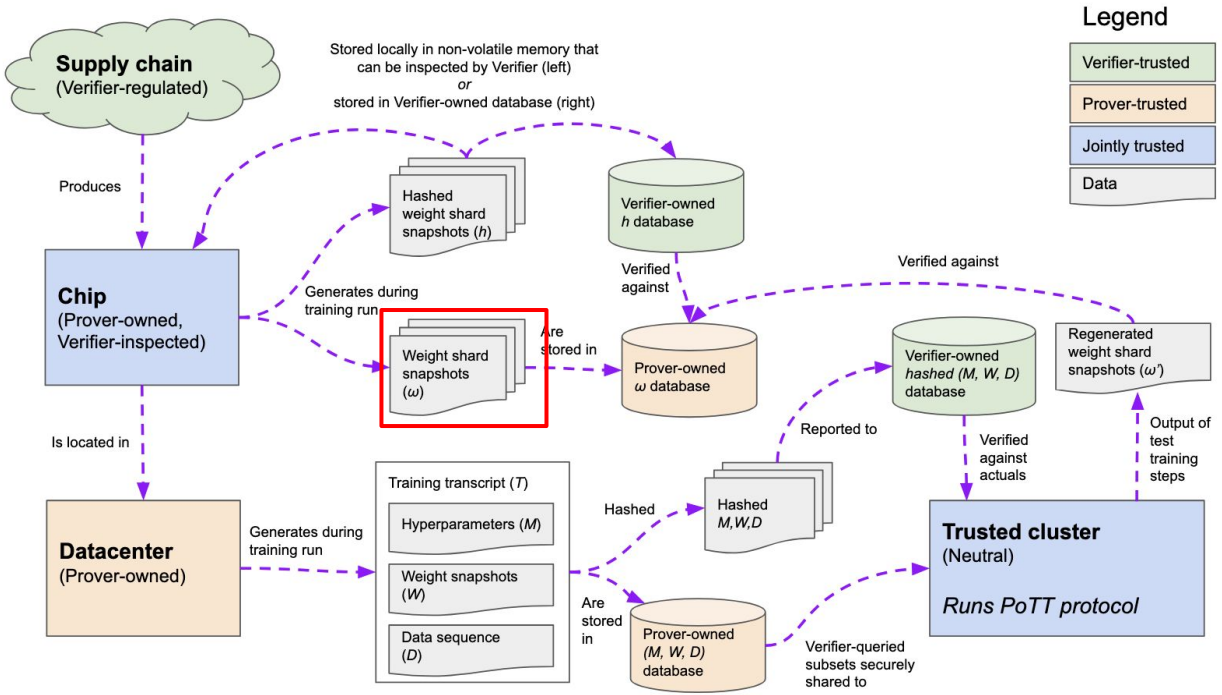
System Overview



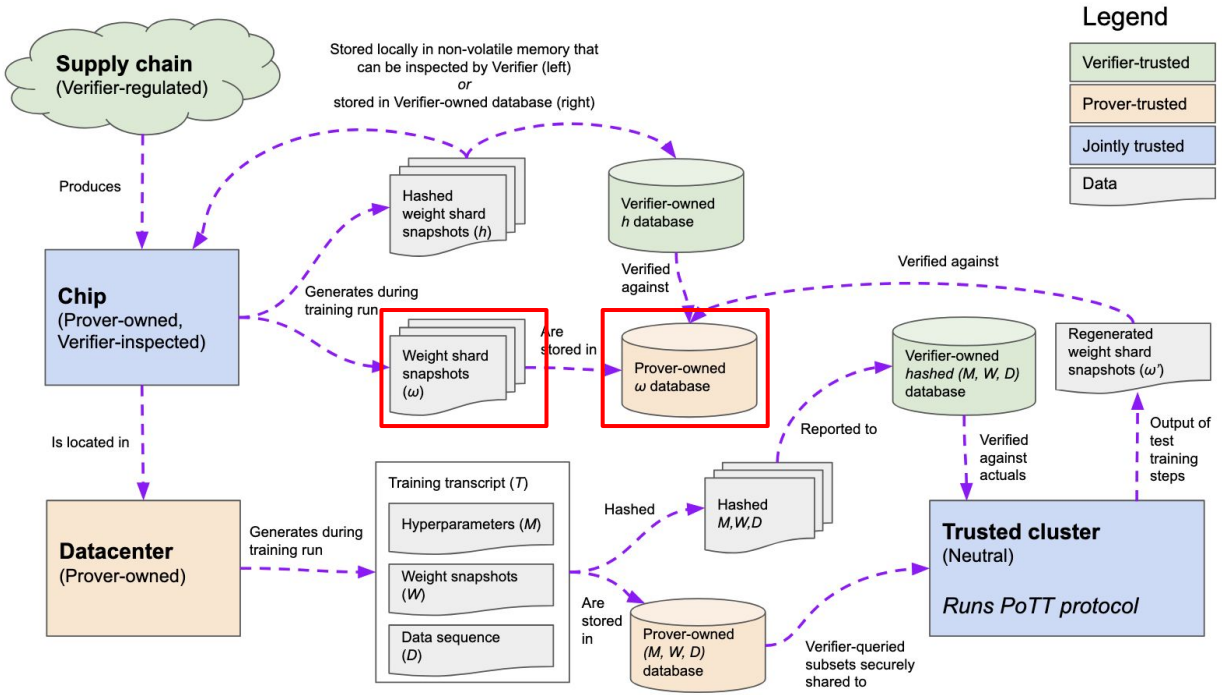
System Overview



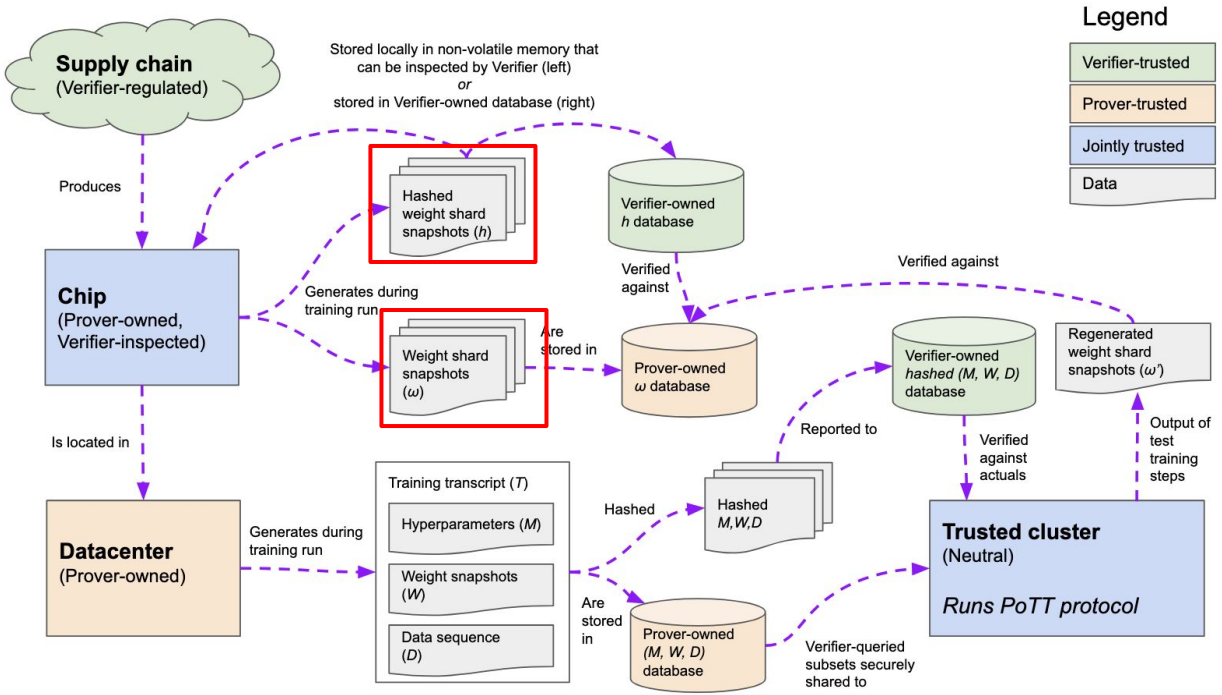
System Overview



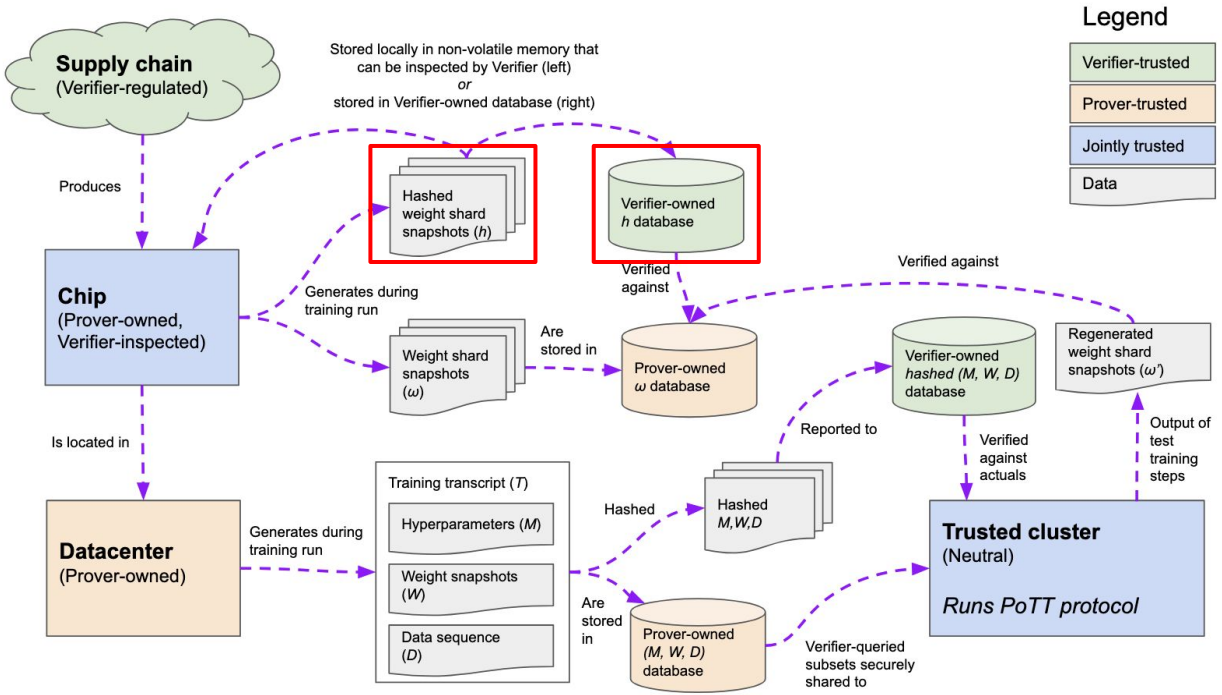
System Overview



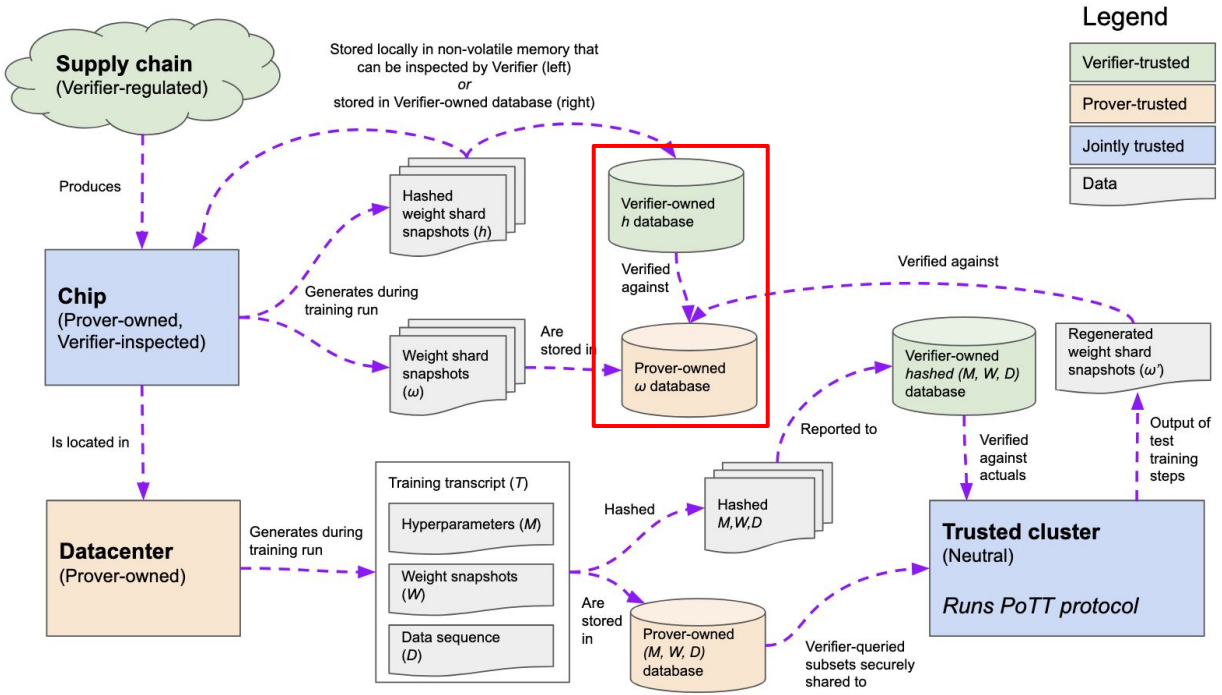
System Overview



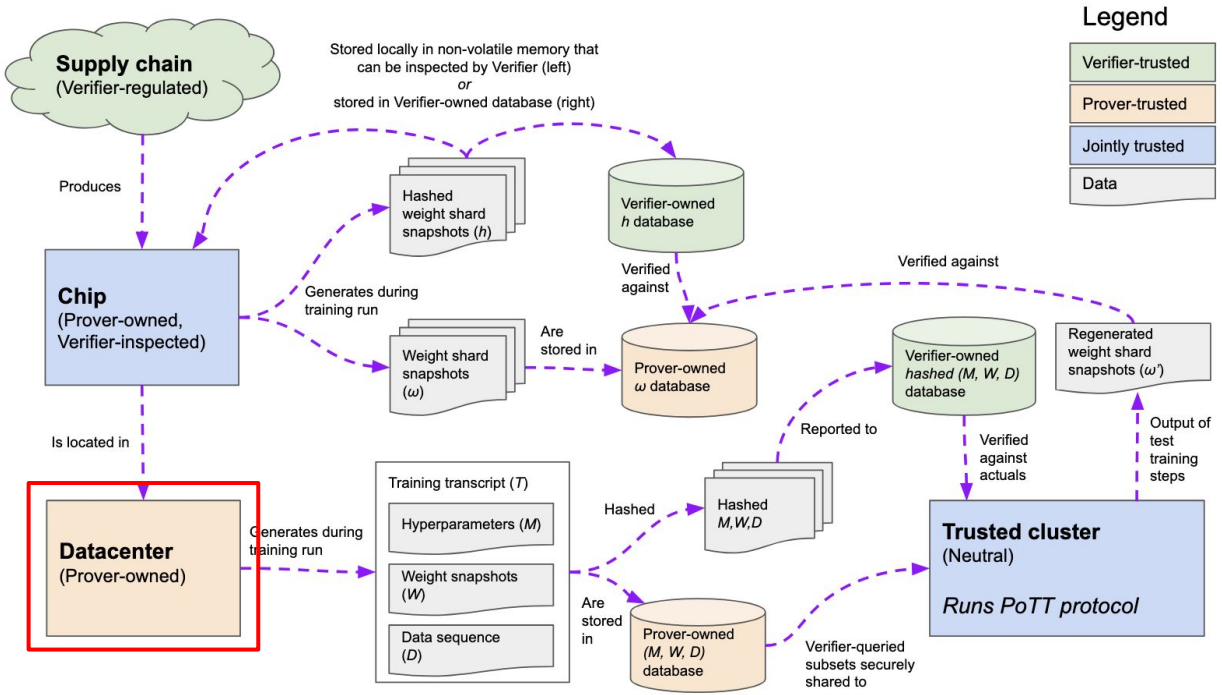
System Overview



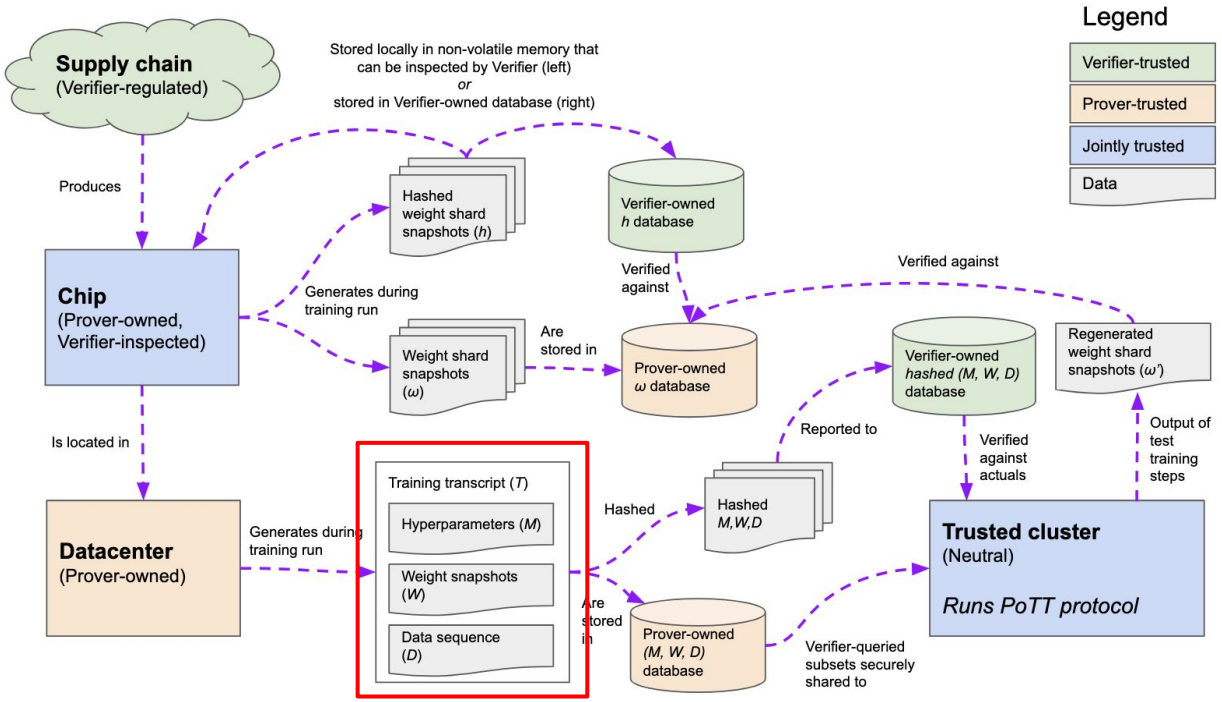
System Overview



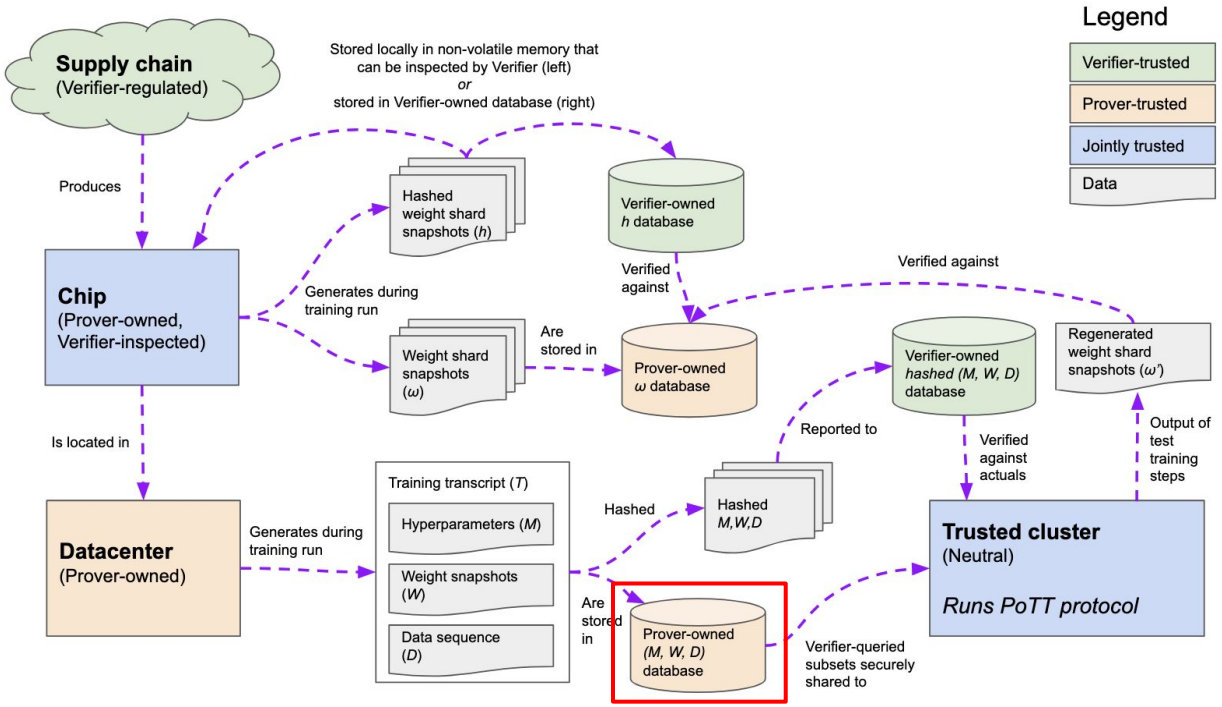
System Overview



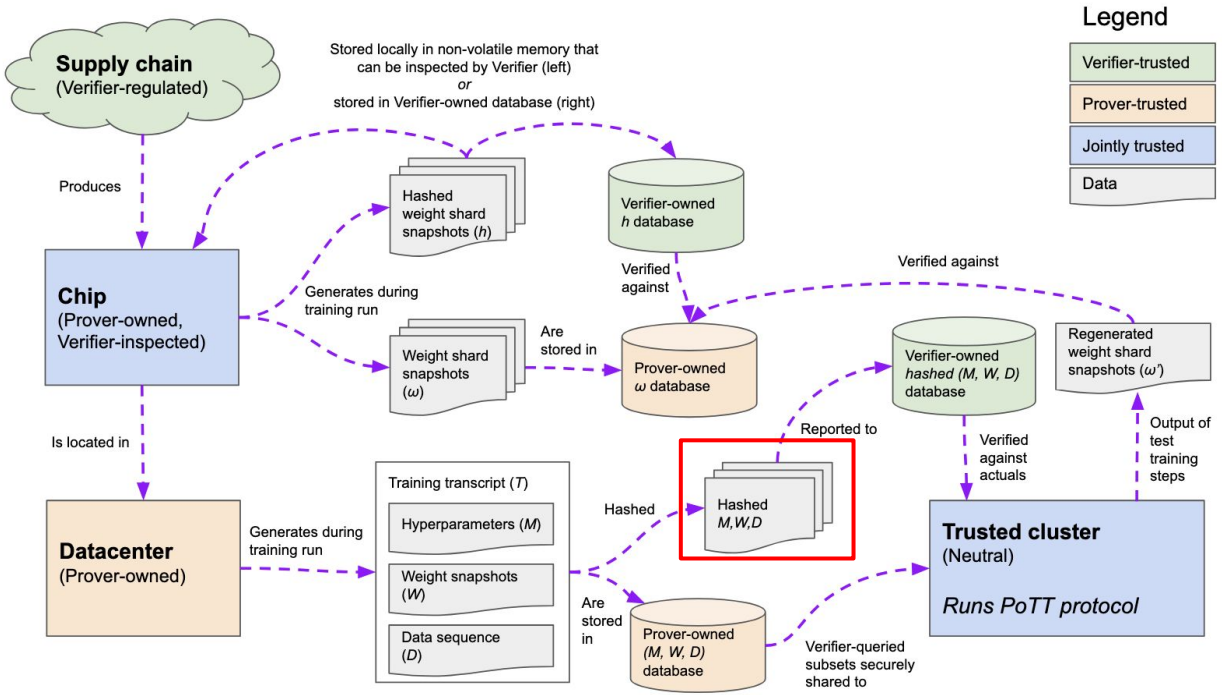
System Overview



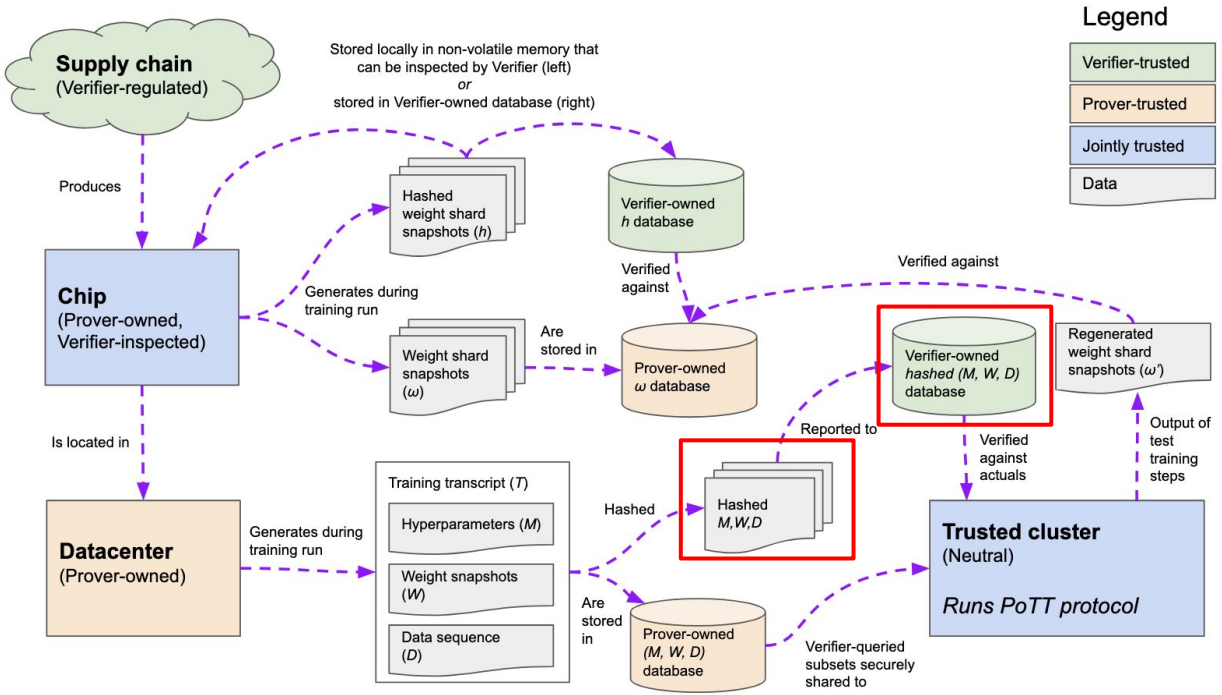
System Overview



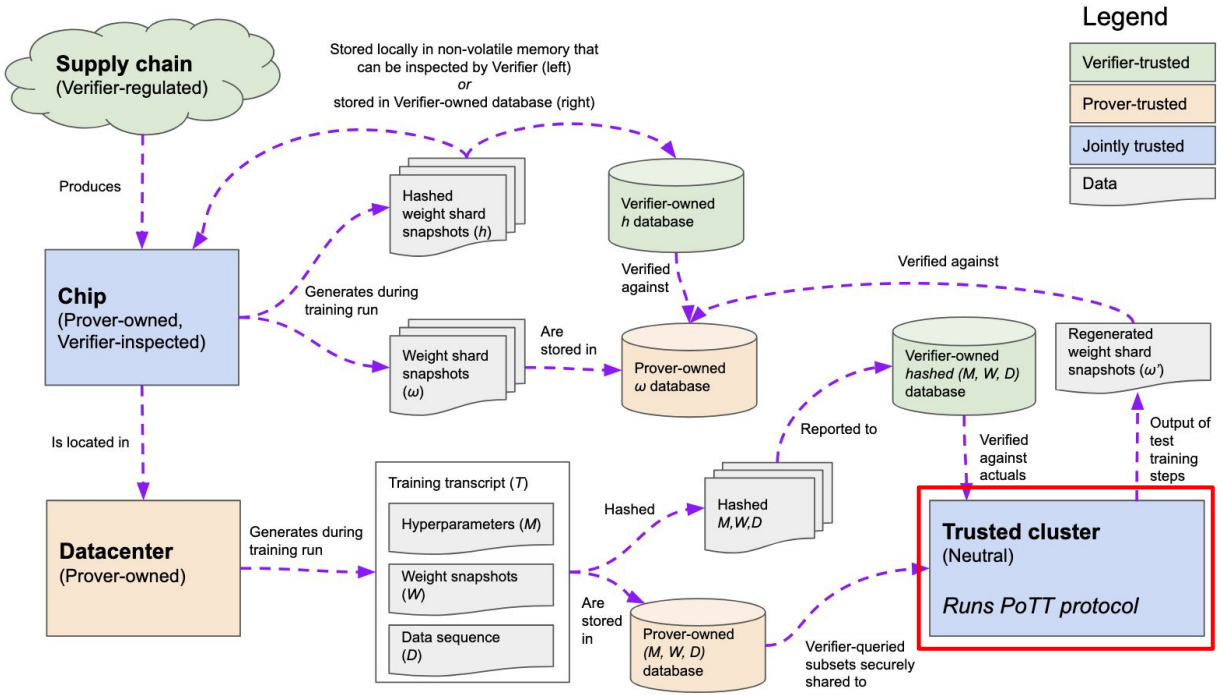
System Overview



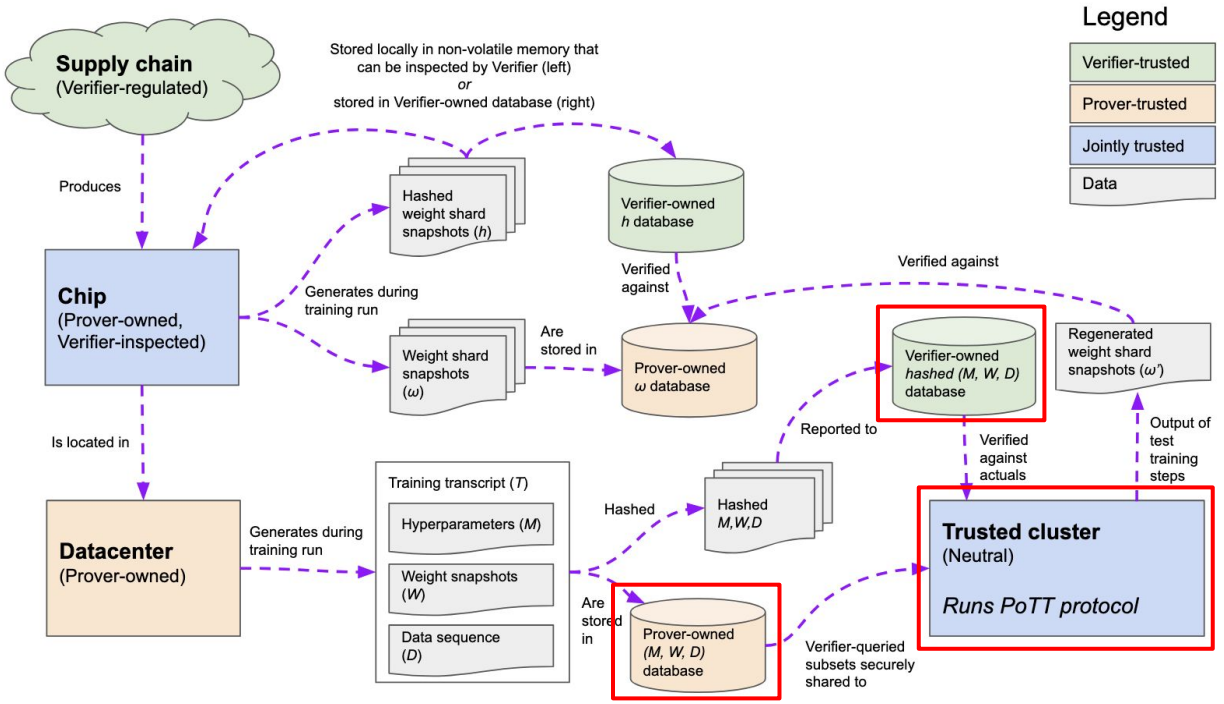
System Overview



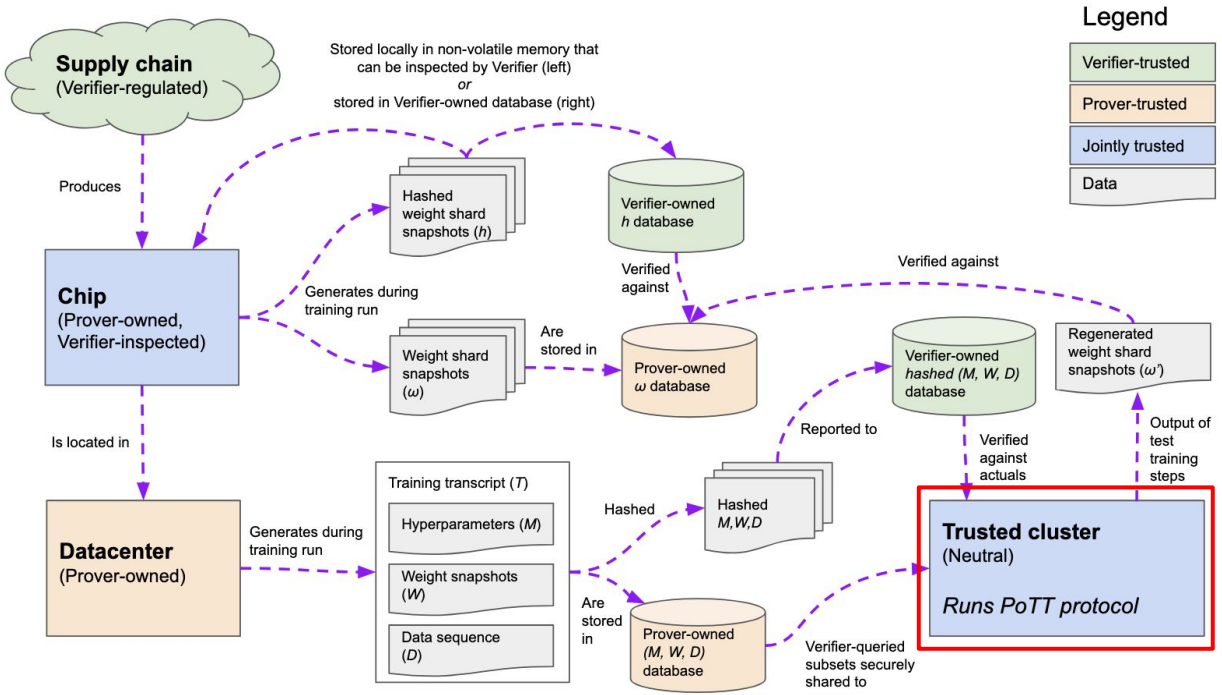
System Overview



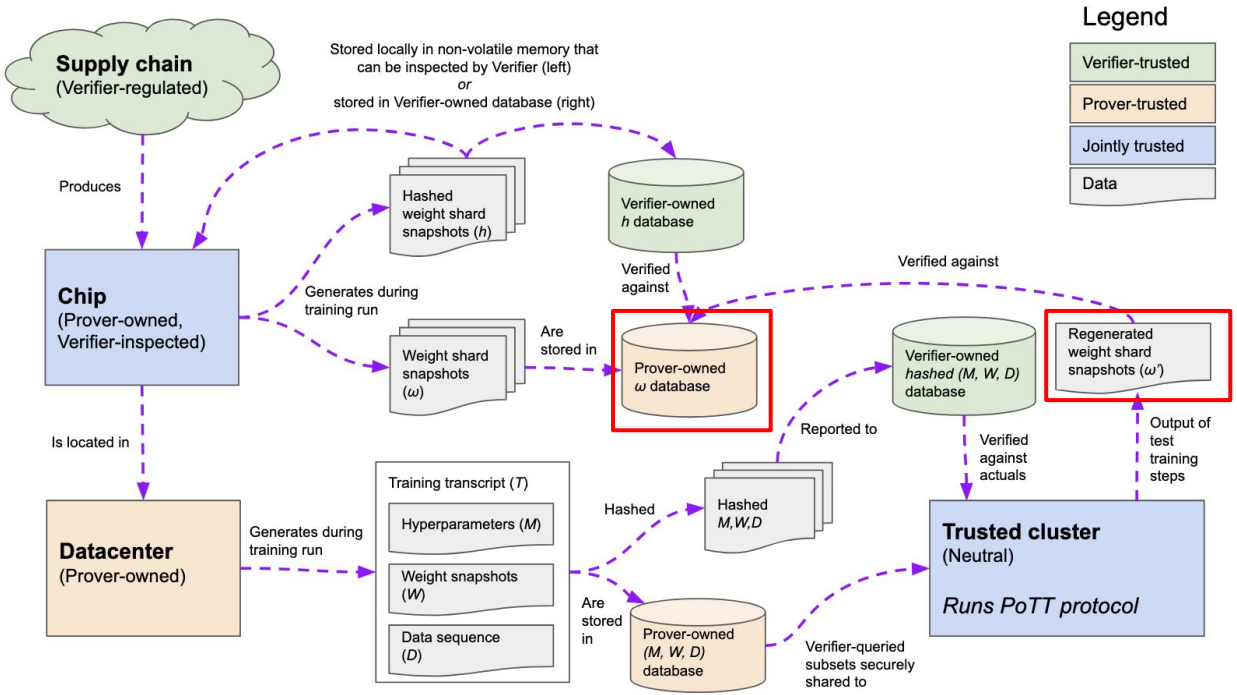
System Overview



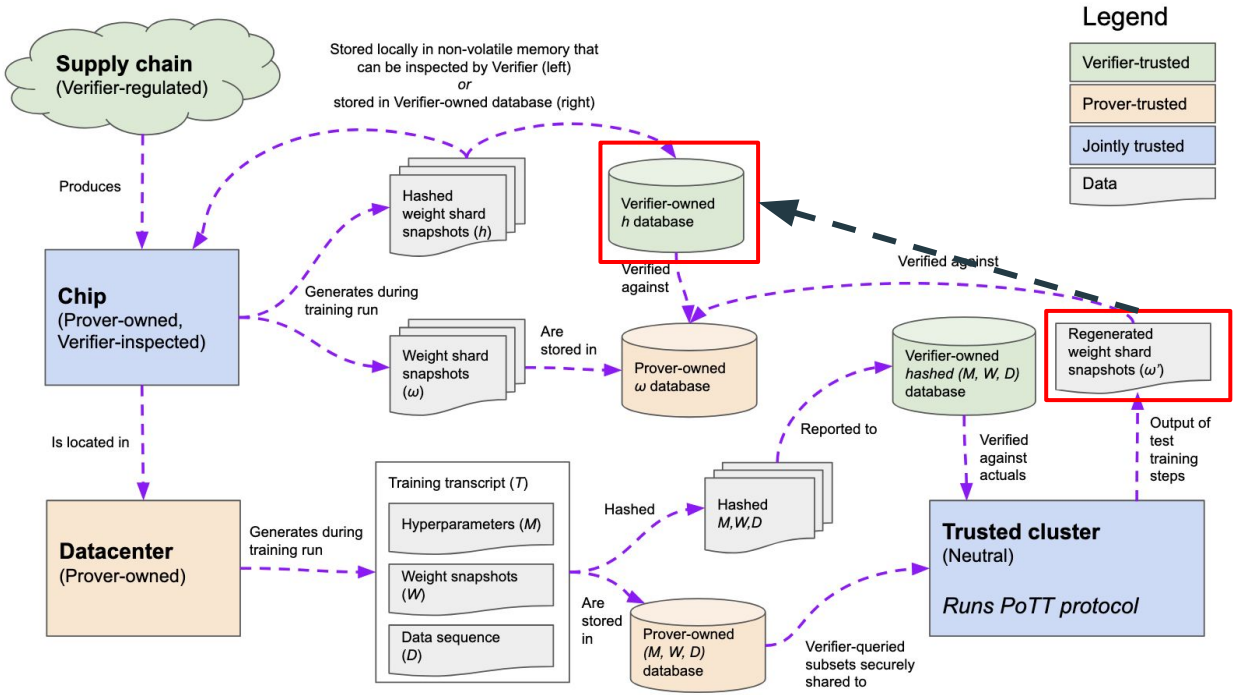
System Overview



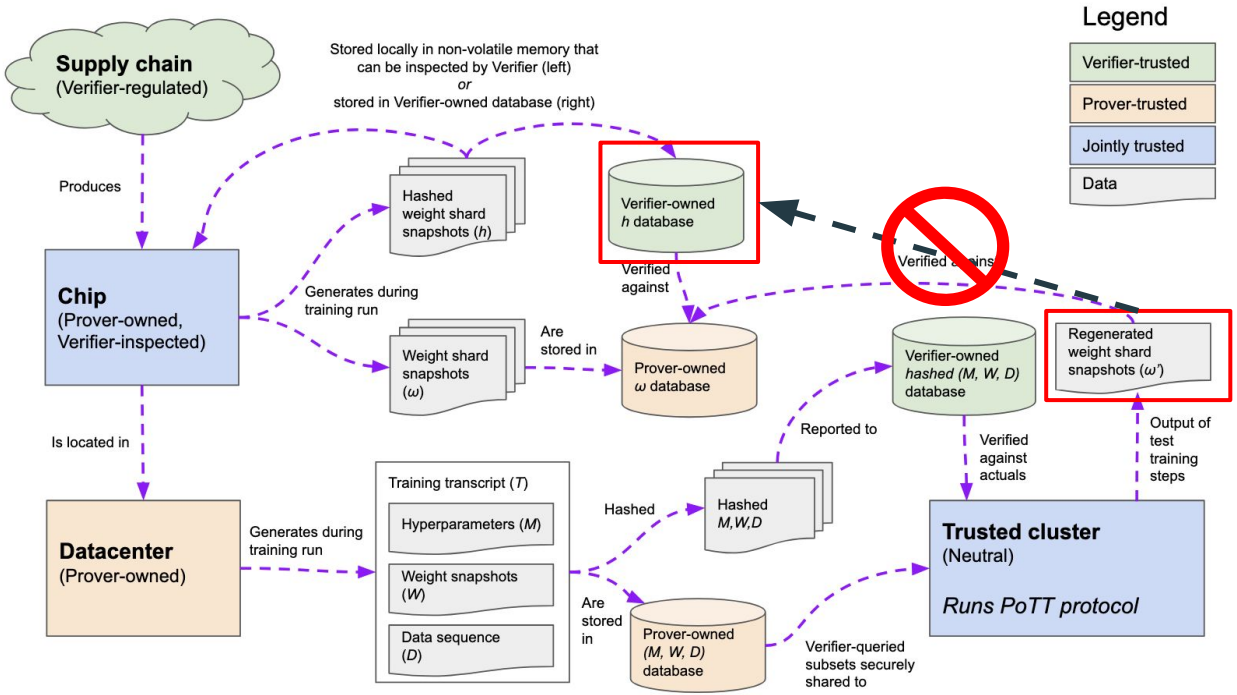
System Overview



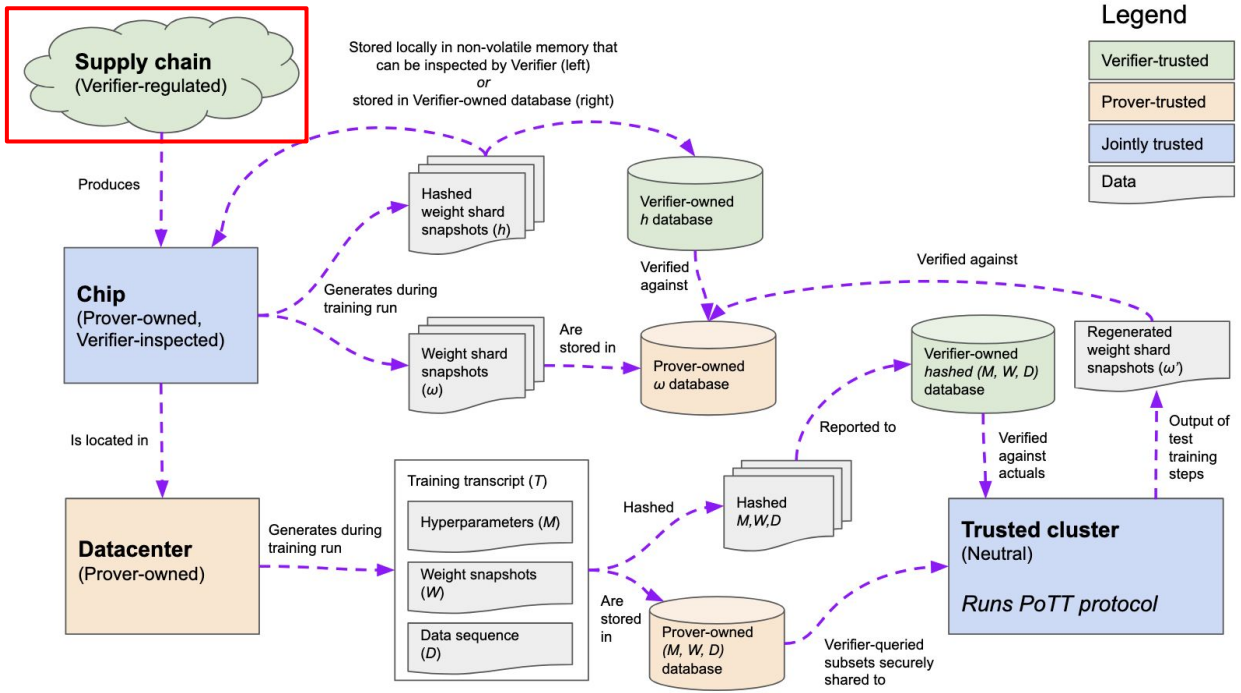
System Overview



System Overview



System Overview



How Many Chips to Audit?

Model	Training FLOPs H	H100-days H/a	H100s to train in 1 yr	Samples/yr if $C = 10^3$	Samples/yr if $C = 10^5$	Samples/yr if $C = 10^7$
GPT-3 [6]	3.14e+23	3.64e+3	10	243	2.43e+4	2.43e+6
Chinchilla [21]	5.76e+23	6.67e+3	19	132	1.33e+4	1.33e+6
PaLM [10]	2.56e+24	2.96e+4	82	29	2.98e+3	2.99e+5
Chinchilla-280B †	9.9e+24	1.15e+5	314	7	771	7.72e+4
Chinchilla-1T †	1.27e+26	1.47e+6	4.03e+3	—	60	6.02e+3
Chinchilla-10T †	1.3e+28	1.5e+8	4.12e+5	—	—	58

Table 1: Example numbers of required total *annual* samples $365 \cdot s/T_m$ to catch a chip from every large-scale training run within $T = 30$ days, given $a = 10^{15} \cdot 24 \cdot 3600$ (the daily 16-bit Tensor Core FLOPs of an NVIDIA H100 SXM GPU [42]), $f = 0.1$ weight snapshots per day (see Section 4), and the Verifier's desired probability of catching a rule-violating training run $p = 0.9$. Models marked with † are projections for future training requirements [21].

How Many Chips to Audit?

Model	Training FLOPs H	H100-days H/a	H100s to train in 1 yr	Samples/yr if $C = 10^3$	Samples/yr if $C = 10^5$	Samples/yr if $C = 10^7$
GPT-3 [6]	3.14e+23	3.64e+3	10	243	2.43e+4	2.43e+6
Chinchilla [21]	5.76e+23	6.67e+3	19	132	1.33e+4	1.33e+6
PaLM [10]	2.56e+24	2.96e+4	82	29	2.98e+3	2.99e+5
Chinchilla-280B †	9.9e+24	1.15e+5	314	7	771	7.72e+4
Chinchilla-1T †	1.27e+26	1.47e+6	4.03e+3	—	60	6.02e+3
Chinchilla-10T †	1.3e+28	1.5e+8	4.12e+5	—	—	58

Table 1: Example numbers of required total *annual* samples $365 \cdot s/T_m$ to catch a chip from every large-scale training run within $T = 30$ days, given $a = 10^{15} \cdot 24 \cdot 3600$ (the daily 16-bit Tensor Core FLOPs of an NVIDIA H100 SXM GPU [42]), $f = 0.1$ weight snapshots per day (see Section 4), and the Verifier's desired probability of catching a rule-violating training run $p = 0.9$. Models marked with † are projections for future training requirements [21].

How Many Chips to Audit?

Model	Training FLOPs H	H100-days H/a	H100s to train in 1 yr	Samples/yr if $C = 10^3$	Samples/yr if $C = 10^5$	Samples/yr if $C = 10^7$
GPT-3 [6]	3.14e+23	3.64e+3	10	243	2.43e+4	2.43e+6
Chinchilla [21]	5.76e+23	6.67e+3	19	132	1.33e+4	1.33e+6
PaLM [10]	2.56e+24	2.96e+4	82	29	2.98e+3	2.99e+5
Chinchilla-280B †	9.9e+24	1.15e+5	314	7	771	7.72e+4
Chinchilla-1T †	1.27e+26	1.47e+6	4.03e+3	—	60	6.02e+3
Chinchilla-10T †	1.3e+28	1.5e+8	4.12e+5	—	—	58

Table 1: Example numbers of required total *annual* samples $365 \cdot s/T_m$ to catch a chip from every large-scale training run within $T = 30$ days, given $a = 10^{15} \cdot 24 \cdot 3600$ (the daily 16-bit Tensor Core FLOPs of an NVIDIA H100 SXM GPU [42]), $f = 0.1$ weight snapshots per day (see Section 4), and the Verifier's desired probability of catching a rule-violating training run $p = 0.9$. Models marked with † are projections for future training requirements [21].

How Many Chips to Audit?

Model	Training FLOPs H	H100-days H/a	H100s to train in 1 yr	Samples/yr if $C = 10^3$	Samples/yr if $C = 10^5$	Samples/yr if $C = 10^7$
GPT-3 [6]	3.14e+23	3.64e+3	10	243	2.43e+4	2.43e+6
Chinchilla [21]	5.76e+23	6.67e+3	19	132	1.33e+4	1.33e+6
PaLM [10]	2.56e+24	2.96e+4	82	29	2.98e+3	2.99e+5
Chinchilla-280B †	9.9e+24	1.15e+5	314	7	771	7.72e+4
Chinchilla-1T †	1.27e+26	1.47e+6	4.03e+3	—	60	6.02e+3
Chinchilla-10T †	1.3e+28	1.5e+8	4.12e+5	—	—	58

Table 1: Example numbers of required total *annual* samples $365 \cdot s/T_m$ to catch a chip from every large-scale training run within $T = 30$ days, given $a = 10^{15} \cdot 24 \cdot 3600$ (the daily 16-bit Tensor Core FLOPs of an NVIDIA H100 SXM GPU [42]), $f = 0.1$ weight snapshots per day (see Section 4), and the Verifier's desired probability of catching a rule-violating training run $p = 0.9$. Models marked with † are projections for future training requirements [21].

Discussion: Privacy/Confidentiality Concerns

“I think the paper's idea is great but it's hard to execute all these rules. For instances, adding firmware to the GPU chip seems to add some "backdoor" to the hardware where the buyer might complain and potentially hurt the supplier's revenue.” - Patrick Wu

“Since many countries buy the same type of chips, this method would raise privacy concern between different countries. Specifically, how can the buyer countries ensure than the chip manufacturers are not installing backdoors to track the chip?” - Brandon Huang

Discussion: Role of Training Data

“In the same vein as Ritwik's paper, it seems like there is an excessive focus on model-focused rules like number of parameters, etc, but not on detecting dataset misuse. How would weight checkpointing/training logs give information on whether a (potentially proprietary) dataset used for training violates regulations?” - Sanjeev Raja

“How can a data-centric approach to AI governance be effectively implemented to mitigate risks associated with AI capabilities, considering the interdependence between dataset quality and model performance, and what frameworks or standards are necessary to support this approach across various AI applications?” - Junyi Zhang

Thank you!