Secure MLaaS with Temper: Trusted and Efficient Model Partitioning and Enclave Reuse

Fabing Li[1,2] , Xiang Li[3], Mingyu Gao[2,3,4]

[1] Xi’an Jiaotong University

[2] Institute for Interdisciplinary Information Core Technology

[3] Tsinghua University

[4] Shanghai Artificial Intelligence Lab Shanghai

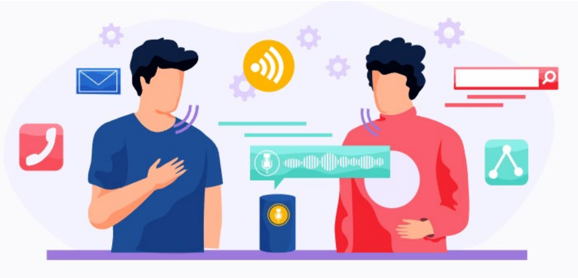
https://github.com/tsinghua-ideal/TEMPER-Secure-MLaaS

Background

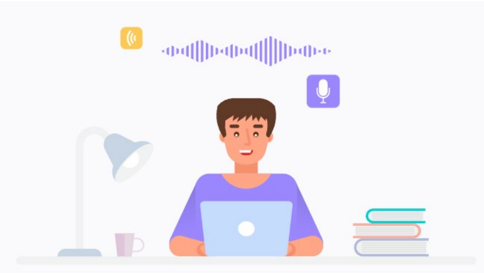
- DNN is widely-used in many applications

Face recognition

Autonomous driving



Voice recognition



Smart Assistant

- Machine Learning as a Service (MLaaS) becomes popular
  - Accessibility: simplified and user-friendly interfaces
  - Rapid Prototyping and Development
  - Scalability, Flexibility and Reduced Costs
  - Integration and Compatibility with Existing Workflows

Security Issues

Data Breaches

Cyber-attacks
Adversarial Attacks

↓

Secure Runtime

Data Privacy

Untrusted Service Provider
Malicious Tenants

↓

Data Isolation

Regulatory Compliance

GDPR
HIPAA

↓

Data Visibility

🤔**How to solve these issues?**

Solutions

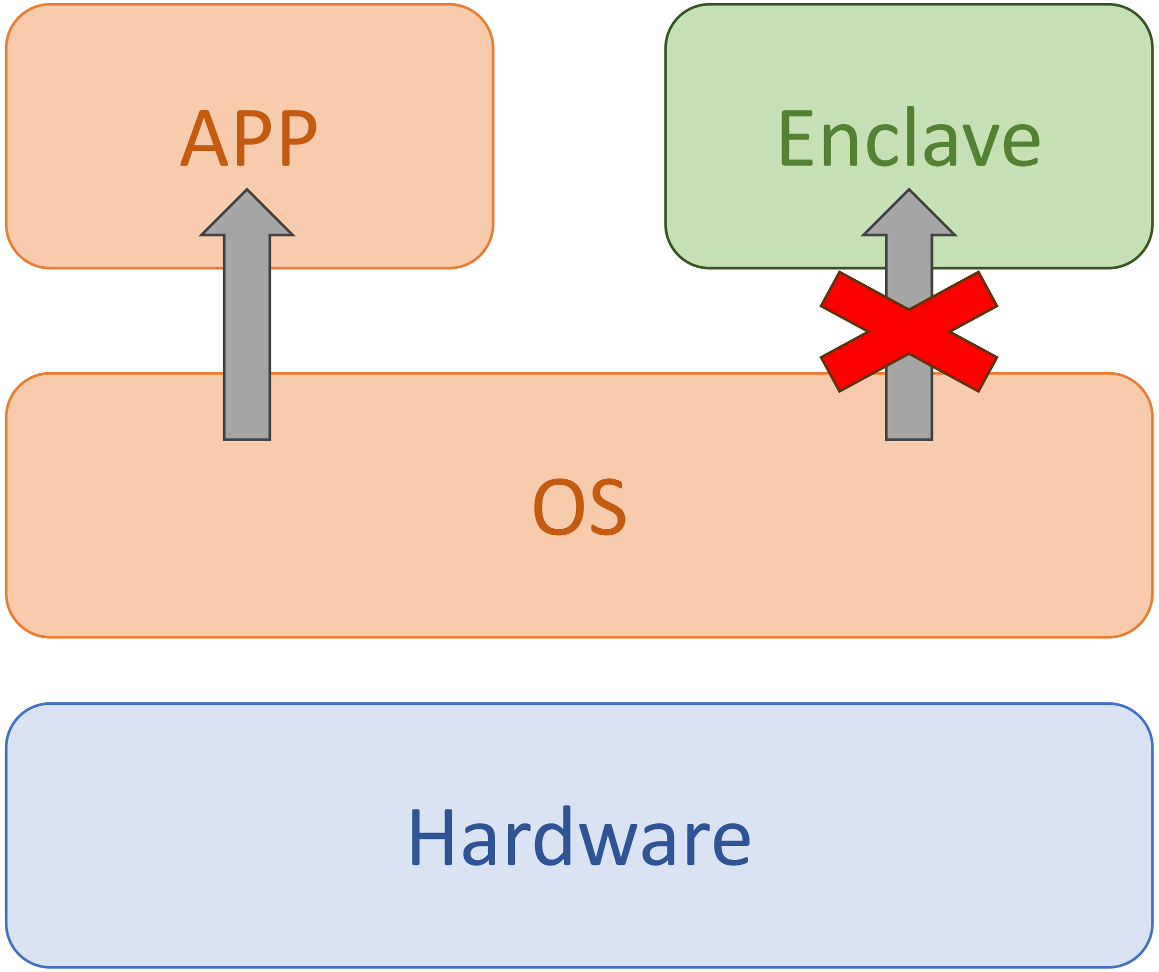
- Trusted Execution Environment (TEE) oriented approaches
- Cryptography oriented approaches

APPROACH	SECURITY LEVEL	EFFICIENCY	LIMITATIONS
TEE	System	Relatively High	Hardware Side-channels
Cryptography • HE • MPC • etc.	Cryptographic	Low	Accuracy Computation Communication

We focus on TEE-based MLaaS !

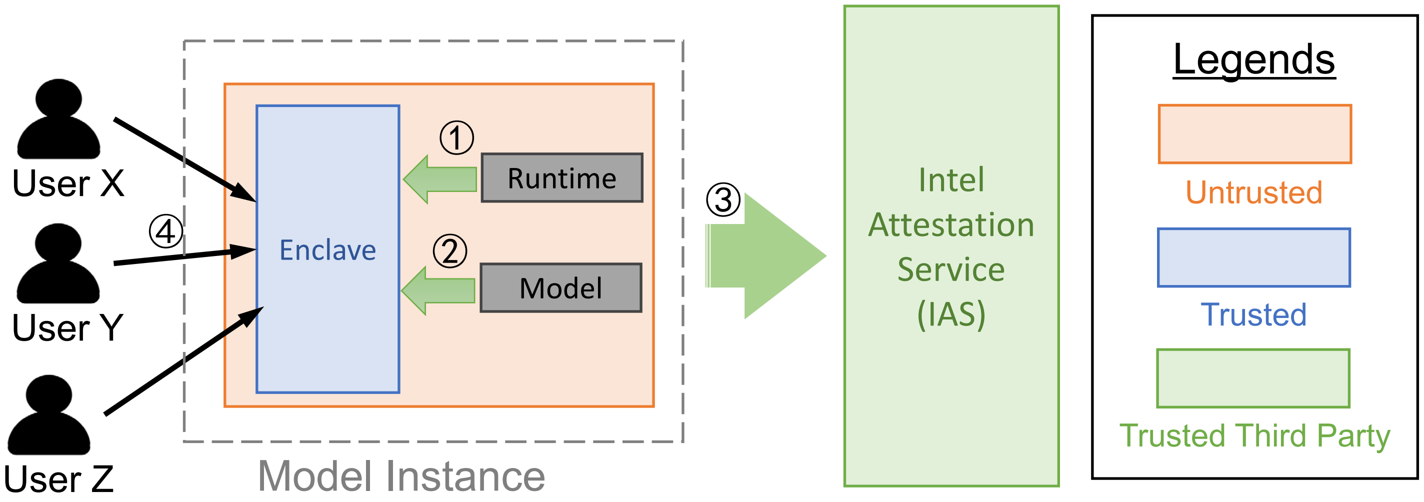
Intel SGX

- Enclave: a hardware-protected memory region
- Provide attestation for code and data inside the enclave
- Obstruct OS-level and physical attackers
- High accessibility: Azure, Alibaba Cloud, etc.



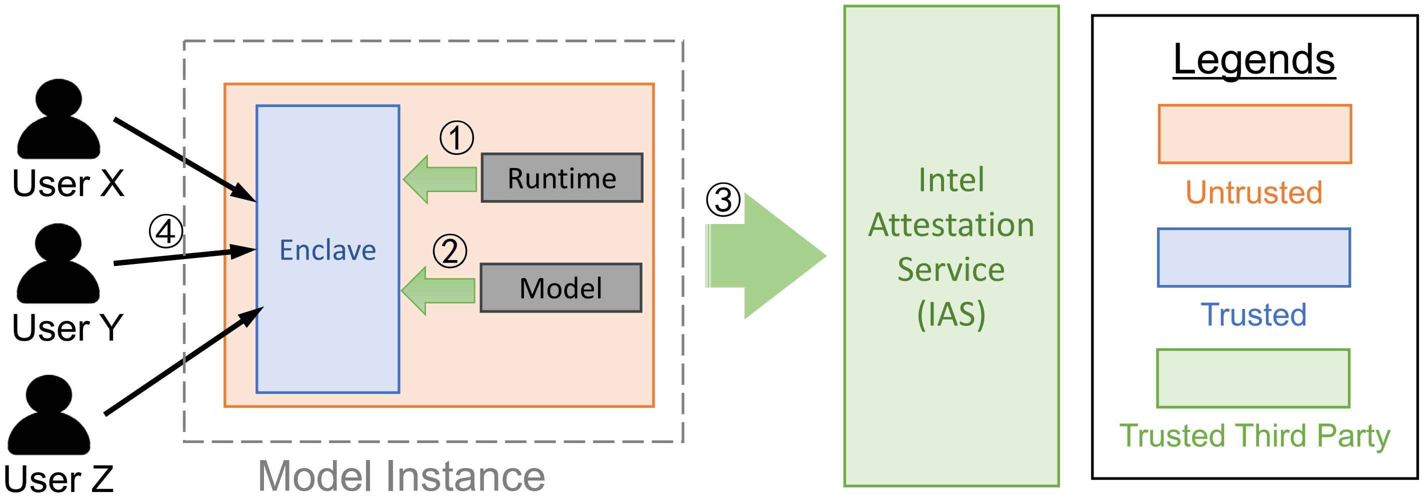
Secure MLaaS

- Integrate runtime with Intel SGX SDK
- Start and load the model weights into the enclave
- Verify the runtime by attestation after initialization
- Receive user requests and make inferences



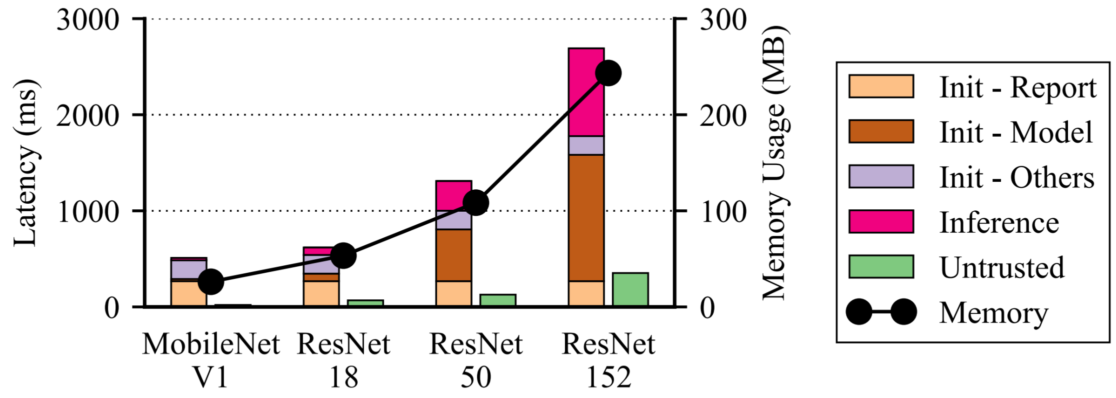
Goals

- Not sacrificing the security
- Unchanged accuracy
- Comparable efficiency and scalability



Challenges

😵‍💫 Why is TEE-based MLaaS slow?



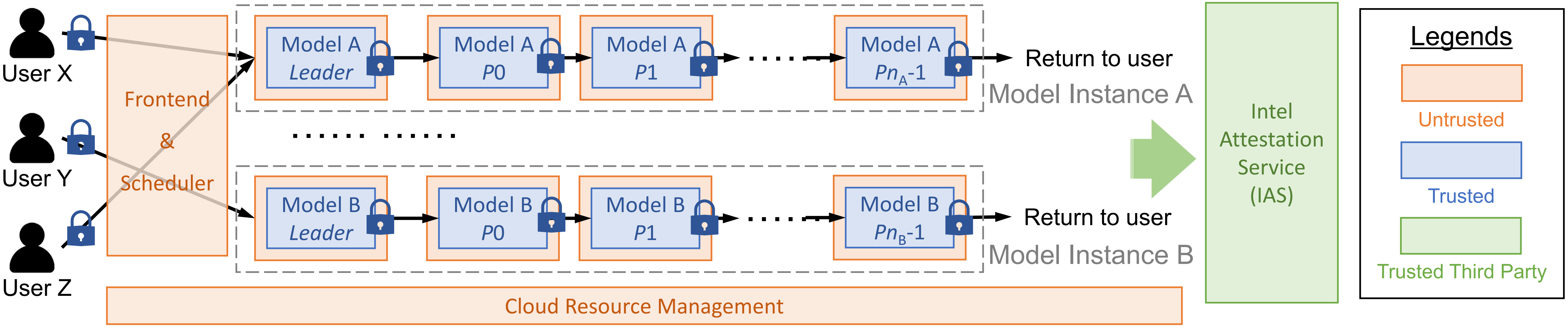
Performance breakdown of baseline secure MLaaS

- Enclave initialization
  - Attestation
- Model Loading
  - Load model weights
- Secure Paging
  - Performance degradation due to limited EPC size

Overview

☞ Approaches

- **Enclave reuse** for enclave initialization and model loading
- **Model partitioning** for secure paging

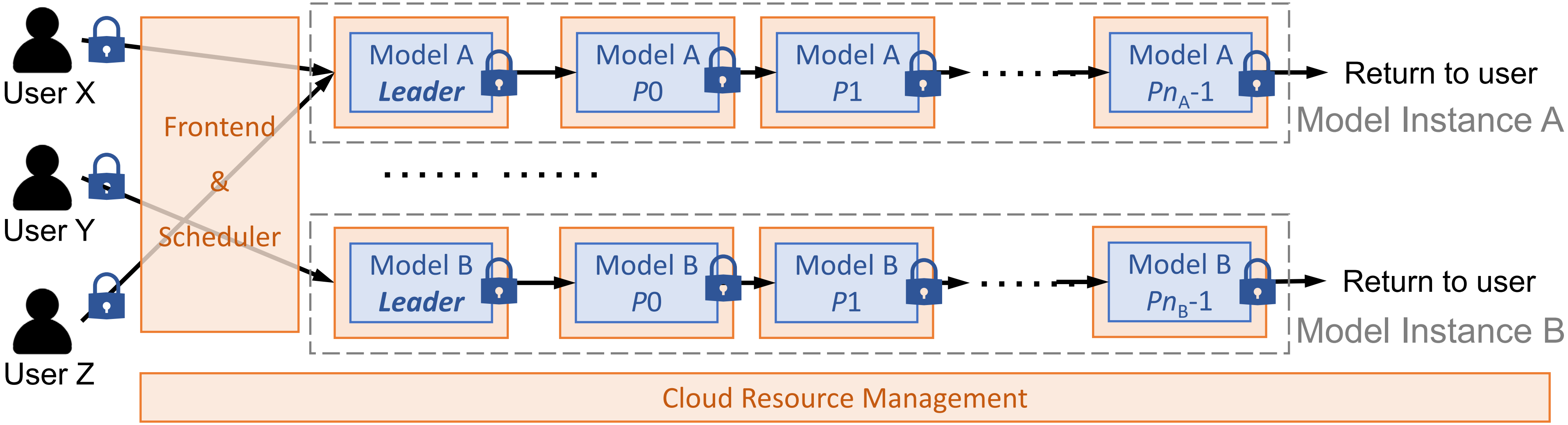


Enclave Reuse

- Long running enclave
  - Leader-Worker topology
  - Preloaded model weights

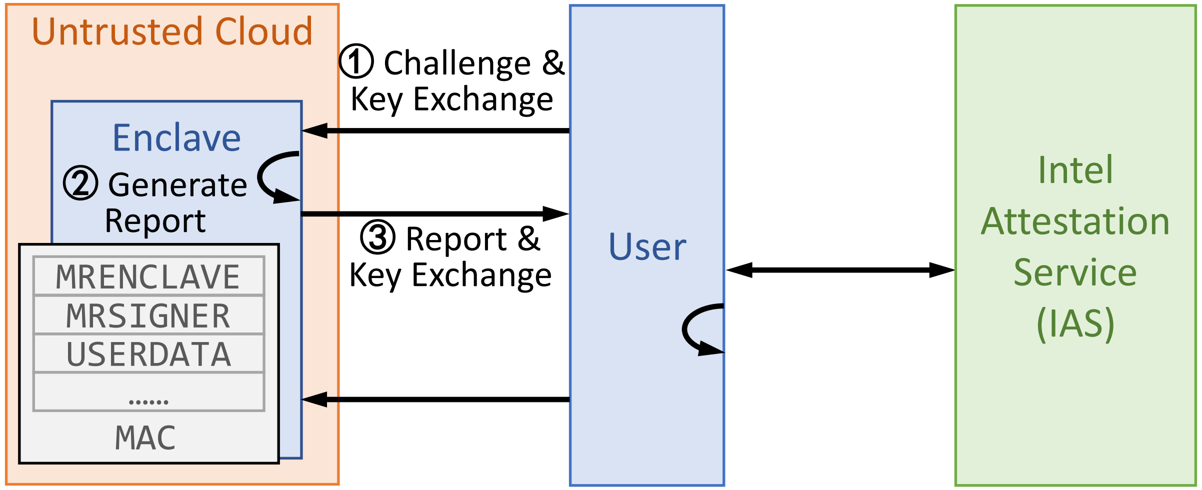
👍 No Model Loading and Scalability

🤔 How to solve the security issues?
e.g. attestation for enclaves, data interference, data residual



Enclave Reuse

- Traditional attestation
  - One report for one user
  - One report for one enclave

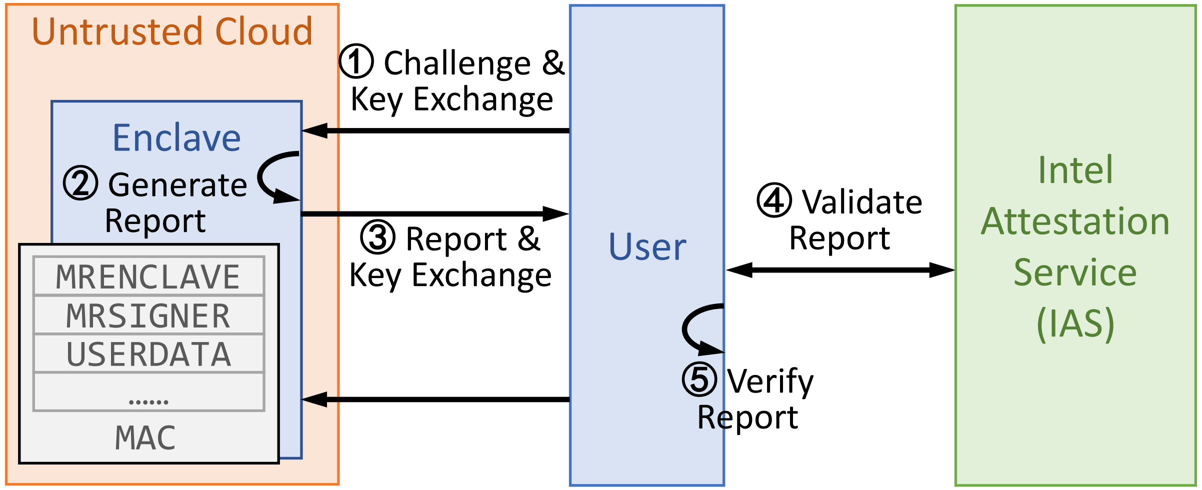


💡 Report validation happens without the enclave

😨 Can we verify a pre-generated report?

Enclave Reuse

- Traditional attestation
  - One report for one user
  - One report for one enclave

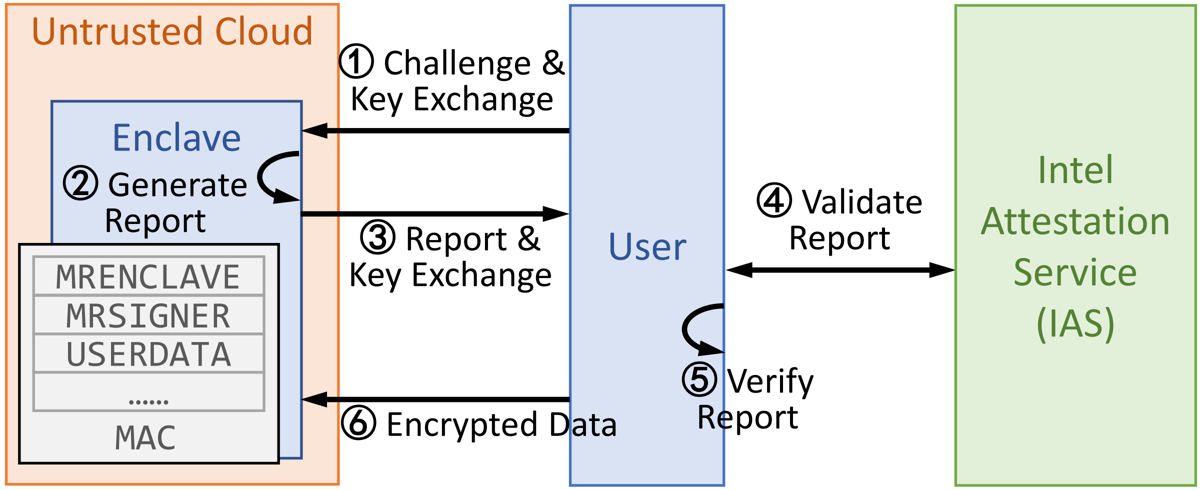


💡 Report validation happens without the enclave

🤔 Can we verify a pre-generated report?

Enclave Reuse

- Traditional attestation
  - One report for one user
  - One report for one enclave

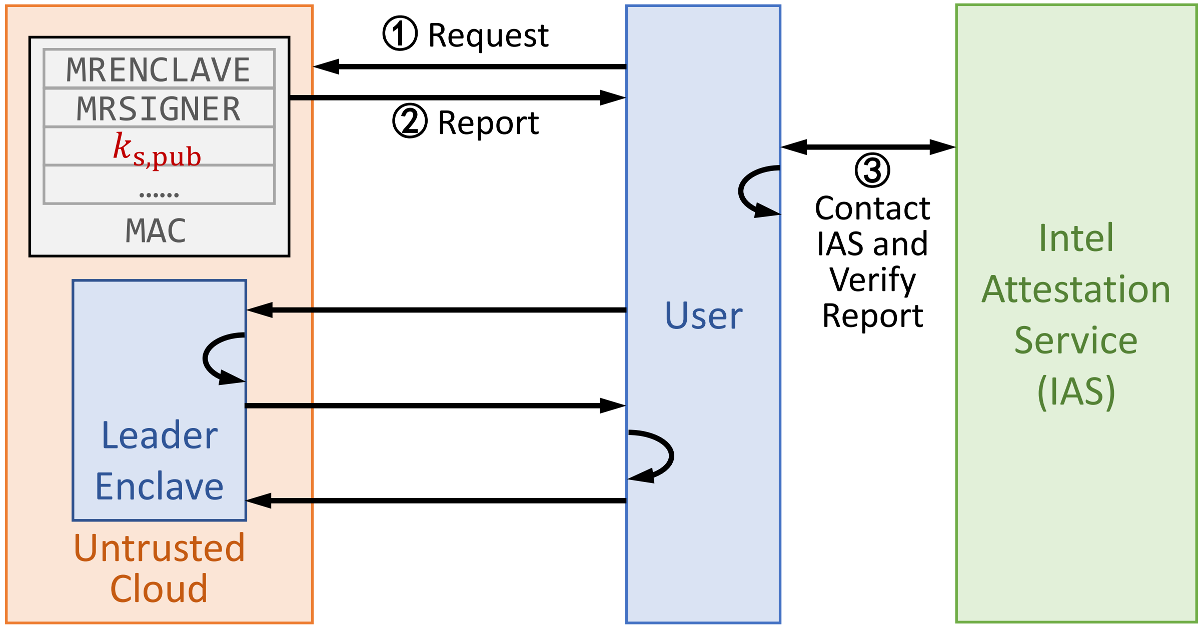


💡 Report validation happens without the enclave

😨 Can we verify a pre-generated report?

Enclave Reuse

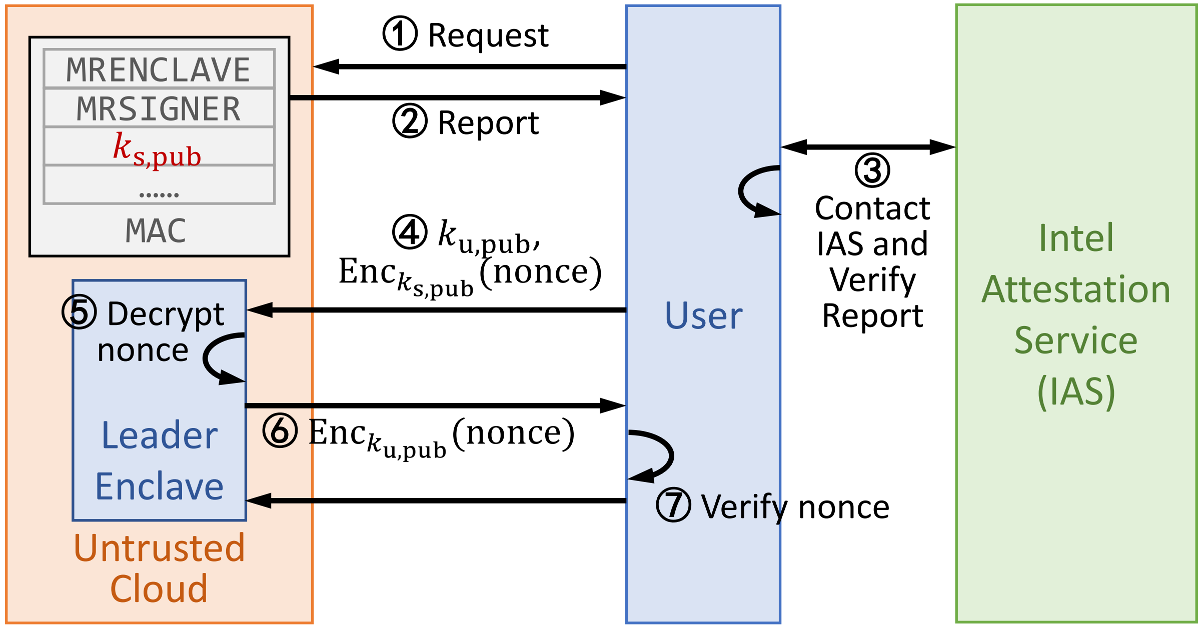
- Report reuse
  - One report for all users
  - One report for all enclaves



👍 No more enclave initialization

Enclave Reuse

- Report reuse
  - One report for all users
  - One report for all enclaves

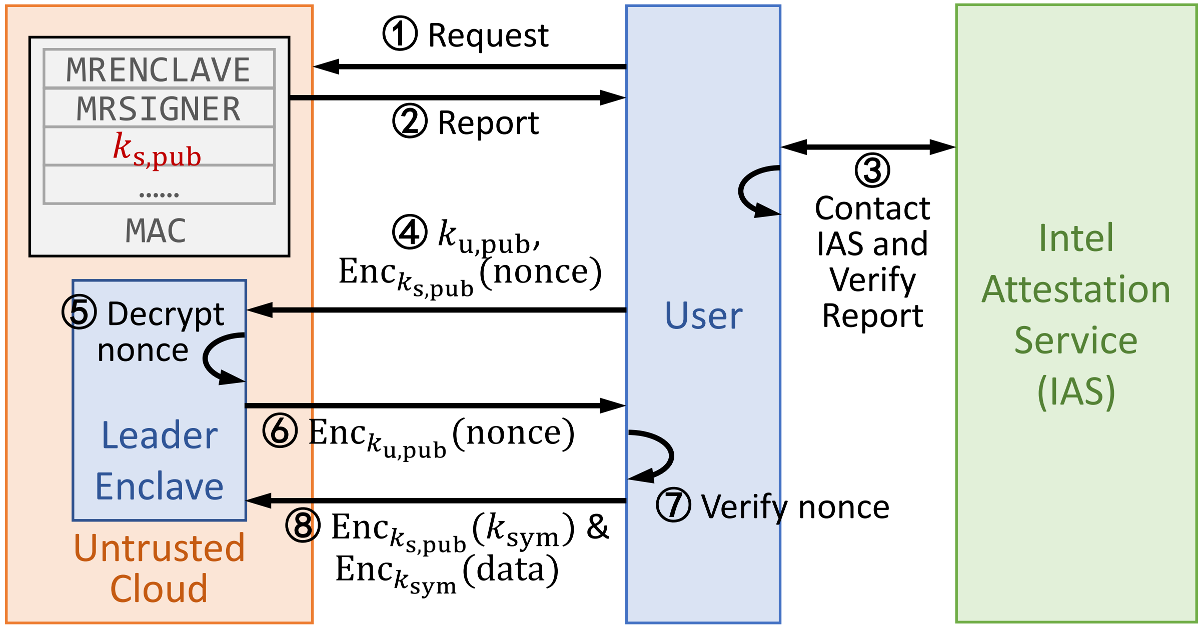


MRENCLAVE
MRSIGNER
$k_{\text{s,pub}}$
……
MAC
① Request
② Report
③ Contact IAS and Verify Report
④ $k_{\text{u,pub}}$, $\text{Enc}_{k_{\text{s,pub}}}(\text{nonce})$
⑤ Decrypt nonce
⑥ $\text{Enc}_{k_{\text{u,pub}}}(\text{nonce})$
⑦ Verify nonce
Leader Enclave
Untrusted Cloud
User
Intel Attestation Service (IAS)

👍 No more enclave initialization

Enclave Reuse

- Report reuse
  - One report for all users
  - One report for all enclaves



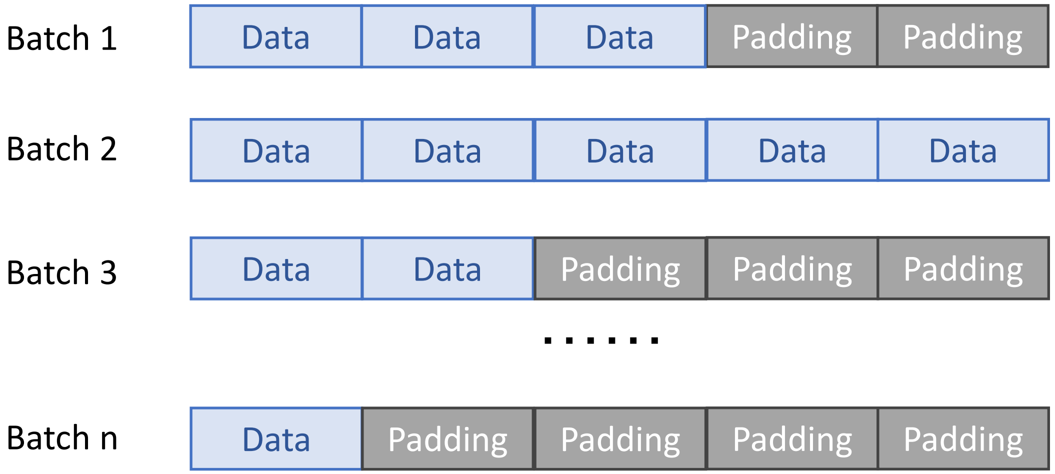
MRENCLAVE
MRSIGNER
$k_{s,pub}$
……
MAC
① Request
② Report
③ Contact IAS and Verify Report
④ $k_{u,pub}$, $\mathrm{Enc}_{k_{s,pub}}(\mathrm{nonce})$
⑤ Decrypt nonce
⑥ $\mathrm{Enc}_{k_{u,pub}}(\mathrm{nonce})$
⑦ Verify nonce
⑧ $\mathrm{Enc}_{k_{s,pub}}(k_{sym})$ & $\mathrm{Enc}_{k_{sym}}(\mathrm{data})$
Leader Enclave
Untrusted Cloud
User
Intel Attestation Service (IAS)

👍 No more enclave initialization

Enclave Reuse

- NN models are stateless
- Models are read-only for inferences
- Data streams flow with deterministic sizes and at fixed time

Security



Batch 1	Data	Data	Data	Padding	Padding
Batch 2	Data	Data	Data	Data	Data
Batch 3	Data	Data	Padding	Padding	Padding
			……		
Batch n	Data	Padding	Padding	Padding	Padding

Model Partitioning

How to get the optimal runtime efficiency?

Alleviate the secure paging

Partition the model to fit in the limited EPC size

Acceptable communication cost

Replace model loading with inter-enclave communication

Model Partitioning

Optimization Goals

😵Is it worth to have some secure paging cost to avoid large communication overheads?

$$\min\left\{n \times \max_{i}\{t_{\text{comp},i}, t_{\text{comm},i}\}\right\} \quad \leftarrow \text{high-throughput}$$

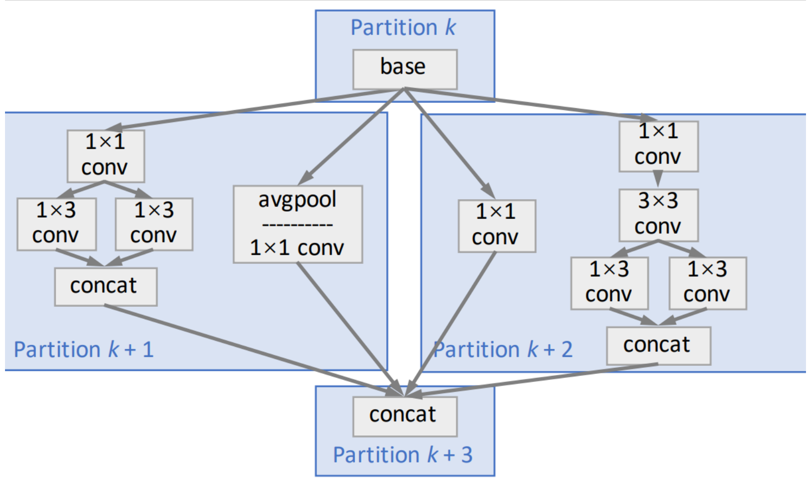
$$\min\left\{\sum_{i} t_{\text{comp},i} + t_{\text{comm},i}\right\} \quad \leftarrow \text{low-latency}$$

t_{comp}: Computation costs <-- Memory consumption
t_{comm}: Communication costs <-- Data Size

Basic units

- Fusing small operators
- Split large operators

😊Find the best partition, and do it faster



Model Partitioning

- Latency Estimation Model

👍 Get all possible costs

  - Computation cost
    - Memory ballooning for latency with secure paging
    - Normal execution for latency without secure paging
  - Communication cost

- Solving Optimized Partitioning

😔 Not so slow, compared to latency estimation

  - Partition (10 to 100) units to (1 to n) enclaves
  - Exhaustive search

Evaluation

Implementation

Intel SGX v2.9

Fortanix Rust enclave development platform (EDP)

TVM v0.7 for ML models

Platform

4 servers

Intel Core i7-9700 CPU

1 Gbps Ethernet

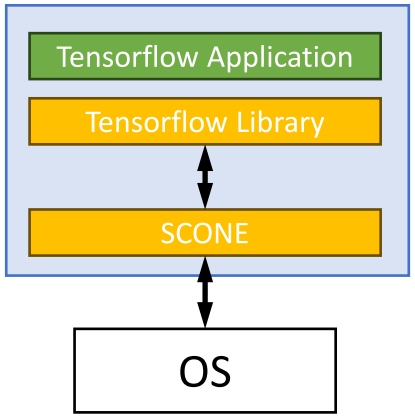
Experiment Setup

ImageNet

MobileNetV1, ResNet18/50/152, VGG19, InceptionV3, and DenseNet201

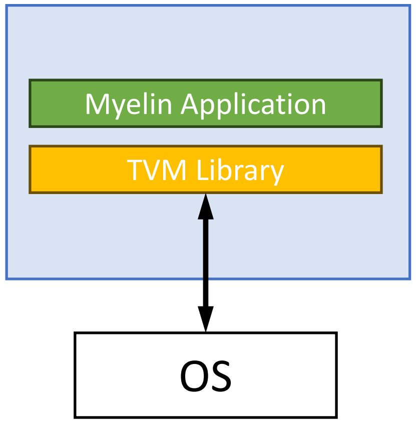
Evaluation

- **Baselines: runtime optimization and careful memory management**



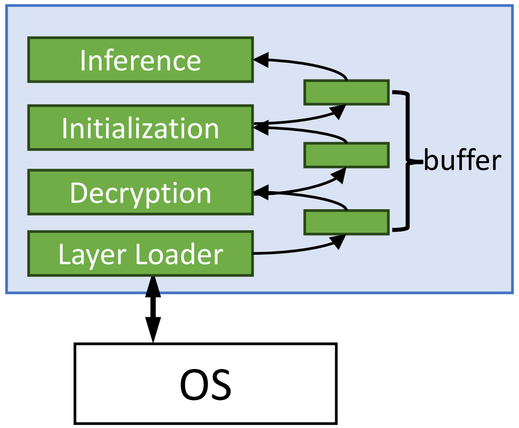
TensorSCONE

Large TCB without optimization
Enclave Initialization
Model Loading
Secure Paging



Myelin

Enclave Initialization
Model Loading
Secure Paging

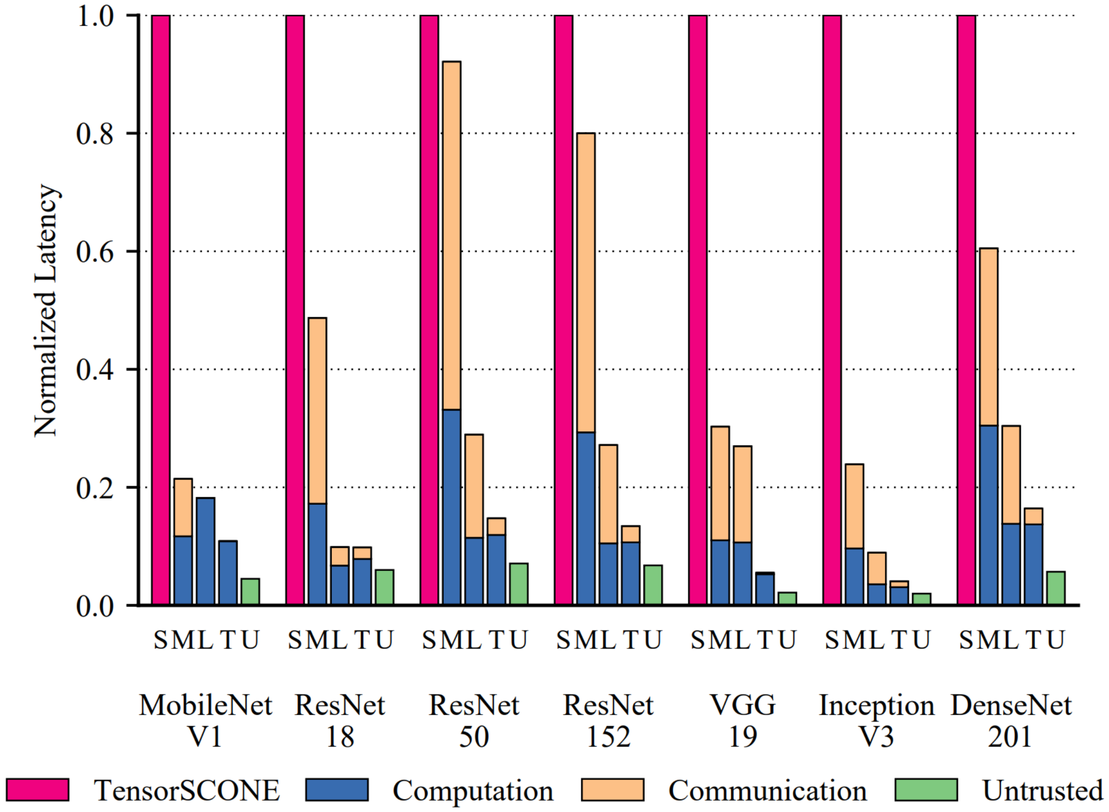


Lasagna

Enclave Initialization
Model Loading

Latency

- Comparison when optimizing for low-latency



😊10 × speedup against TenosrSCONE

😊4.9 × speedup against Myelin

😊2.2 × speedup against Lasagna

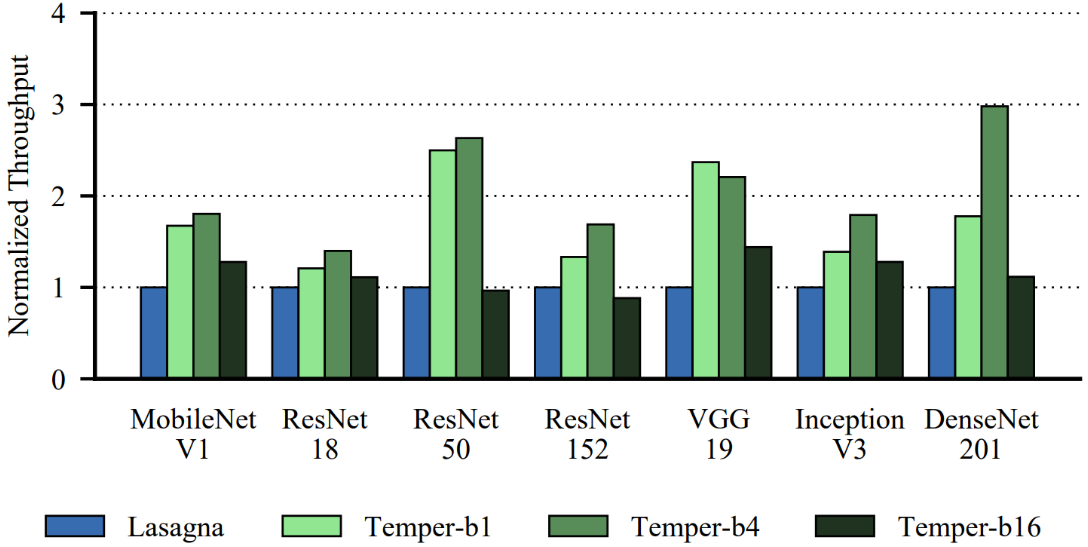
😊2.1 × slowdown for the untrusted

Throughput

- Throughput using batch sizes 1, 4, and 16

😊1.8 ×, 2.1 ×, and 1.2 × higher throughput over Lasagna with batch size 1, 4, and 16

🧐Increasing batch sizes will not always result in throughput improvements, because larger batch sizes incur more secure paging



Attestation and Communication

Attestation (msec)	Server	User	Total
Standard	462.43	111.25	573.68
TEMPER	30.48	112.97	143.45

😒Report generation inside the enclave takes close to half a second

😊4 × faster on overall performance

Partitioning Strategies

Model	TEMPER(img/sec/server)	DNN-Partition (img/sec/server)
MobileNetV1	41.57	41.53
ResNet18	23.96	15.37
ResNet50	9.41	2.47
ResNet152	1.89	2.67
VGG19	0.60	0.39
InceptionV3	4.39	1.06
DenseNet201	4.62	2.74

* DNN-Partition assumes a heterogeneous system with many accelerators and CPUs

😵The effect of appropriate secure paging instead of strict partition size

💡 It is necessary to allow secure paging sometimes

Conclusion

👍 In-depth analysis on TEE-based secure MLaaS designs and identify three key performance inefficiencies: enclave initialization, model loading, and limited trusted memory space.

👍 Propose a trusted and efficient MLaaS system, TEMPER, improving performance while not sacrificing security guarantees or inference accuracy.

👍 Outperform the SOTA baseline by over 2 × in terms of latency and throughput

🙌 Questions?