DATA+AI SUMMIT 2022

# Privacy Preserving Machine Learning and Big Data Analytics Using Apache Spark

## Ehance Security & Privacy in Big Data with Intel SGX

ORGANIZED BY databricks


Qiyuan Gong
Software Arch, Intel

Chunyang Hui
Senior Engineer, Ant Group

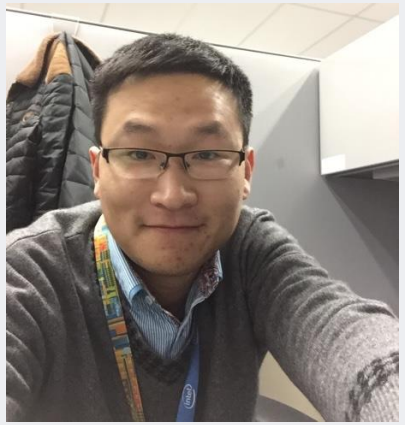
# About

## Qiyuan Gong

PhD at data anonymization & privacy.

Key contributor of BigDL and SSM.

Focus on Privacy Preserving Machine learning, Federated Learning and Cluster Serving etc.



## Chunyang hui

Occlum team in Ant Group.

Key contributor of Occlum.

Focus on operating system and system security.

# Acknowledge


## Results of teams works

Intel

Dongjie Shi, Xin Qiu, Wesley Du, Jason Dai

Yuan Wu, Ligang Wang, Zongmin Gu

Ant Group

Hongliang Tian, Zehuan Li, Ran Duan,

Qing Li, Qi Zheng, Shoumeng Yan

# Part 1
# Brief overview

# Privacy & Security is no more optional

## Security & Privacy

### Privacy & Security Requirements

- Fundamental requirement for individual
- Regulation requirement
  - GDPR, HIPPA, CCPA, CyberSec, PRC Personal Data Protection Law etc

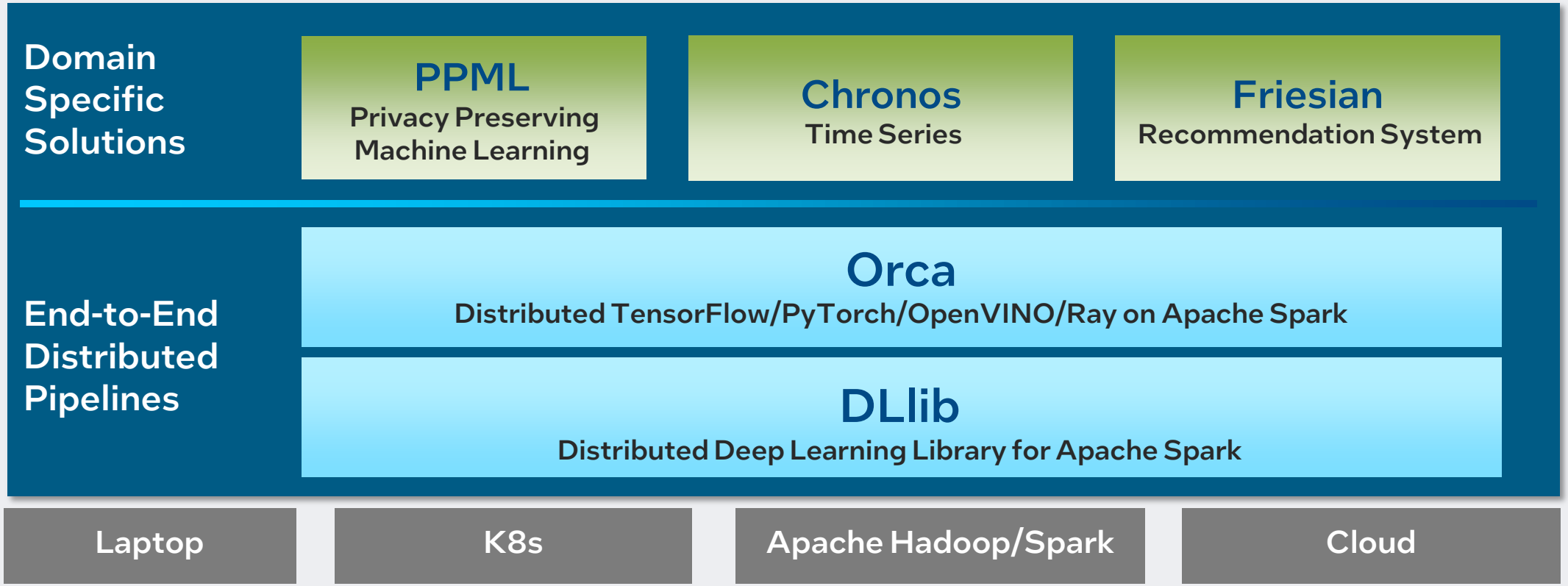
### If not (Violate Regulation)

- Huge penalty (at most 10% Gain)
- Impact to reputation/business
- ...


https://www.privacyaffairs.com/gdpr-fines/

# BigDL: Open Source Big Data AI Project

*Making it easy for building end-to-end, distributed AI applications*

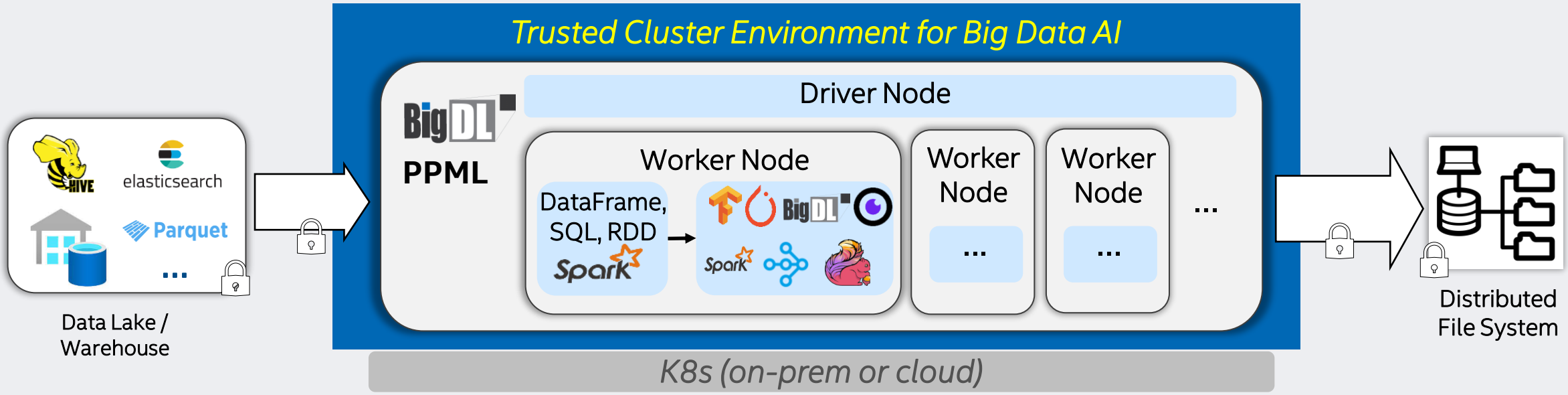



**BigDL 2.0** (https://github.com/intel-analytics/BigDL/) combines the *original BigDL* and *Analytics Zoo* projects

* *"End-to-End Big Data AI Pipeline Using Ray and Apache Spark", 2021 Conference on Computer Vision and Pattern Recognition (CVPR 2021) Tutorial*
* *"BigDL: A Distributed Deep Learning Framework for Big Data", in Proceedings of ACM Symposium on Cloud Computing 2019 (SOCC'19)*
* *"Building Deep Learning Applications for Big Data Platforms", 33rd AAAI Conference on Artificial Intelligence (AAAI-19) Tutorial*
* *BigDL 2.0: Seamless Scaling of AI Pipelines from Laptops to Distributed Cluster", 2022 Conference on Computer Vision and Pattern Recognition (CVPR 2022)*

# ML & Big Data Analytics in Privacy Way

## Secure & Trusted Big Data and AI, even on Untrusted env

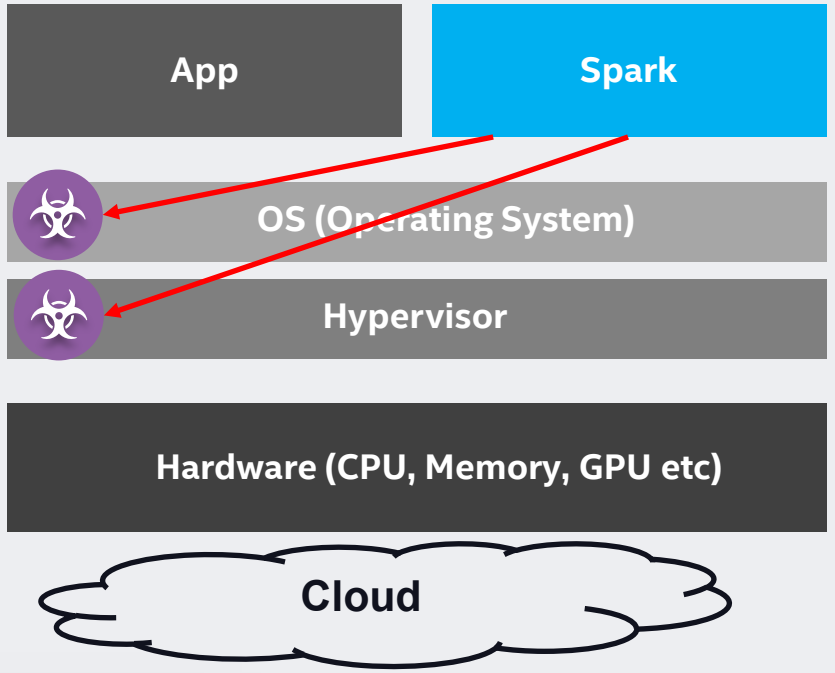



- Standard, distributed AI applications on encrypted data
- Hardware (Intel SGX/TDX) protected computation (and memory)
- End-to-end security enabled for the entire workflow

# A Quick look at Security in Apache Spark

## Security in Apache Spark

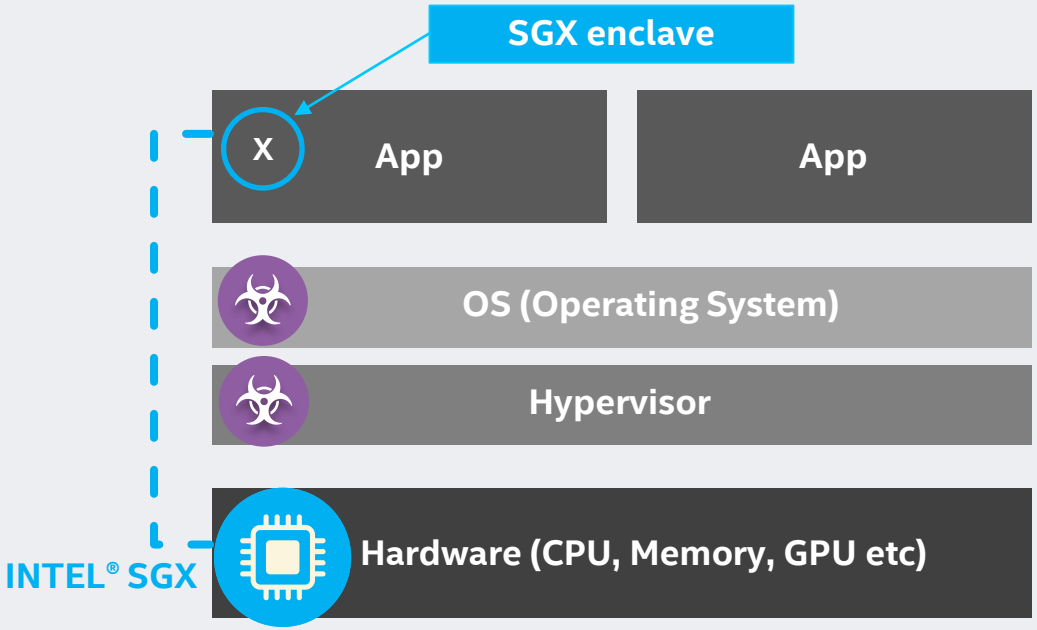
- Spark Security
  - Network (Protected) :
    - PRC: AES
    - Http: TLS
  - Storage (Protected) :
    - AES for local storage & shuffle
  - Computation: (Not Protected)




If OS/VM/Hypervisor/BIOS is hacked by adversaries, then they can dump sensitive data (input, temp, output etc) from Spark.

# Intel SGX (Software Guard Extensions)

## Trusted Execution Environment




### Features

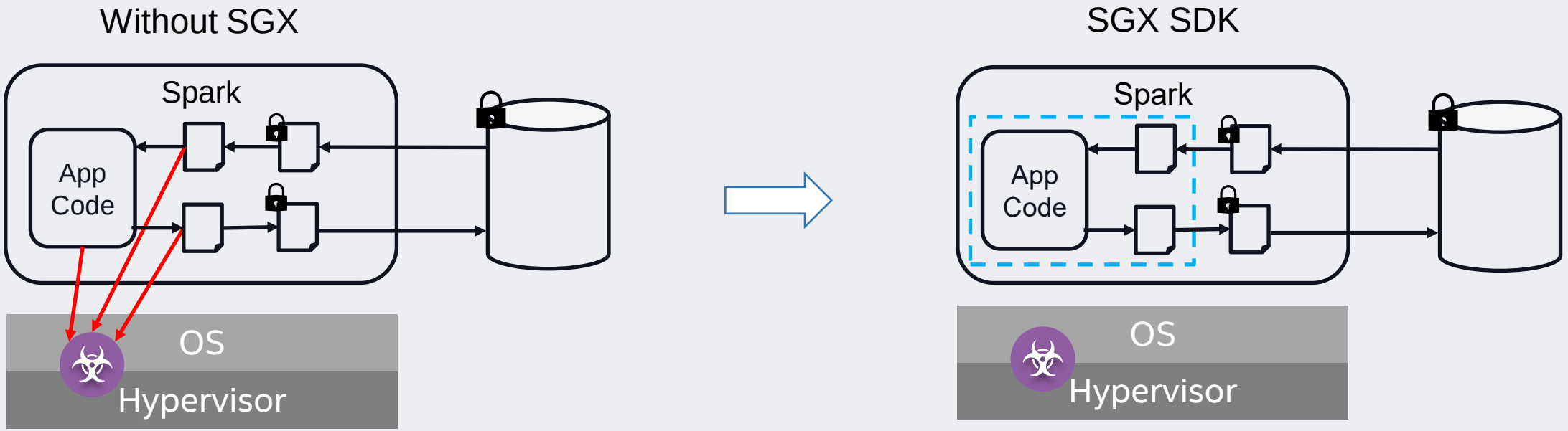
- Hardware based secured env
  - Protected by CPU, still safe even BIOS/OS VM is hacked
  - Reduce Attack Surface (limited API & access control)
- Low impact on performance

### Update

- SGX comes to 3rd Xeon Platform (New!)
  - **Much Large**r SGX EPC (64/512GB/socket)
  - Dynamic memory allocation

# SGX Secure Computation

## Running unchanged Spark Applications in SGX




Adversary can get App code, input data, output

Protect sensitive modules & plain text

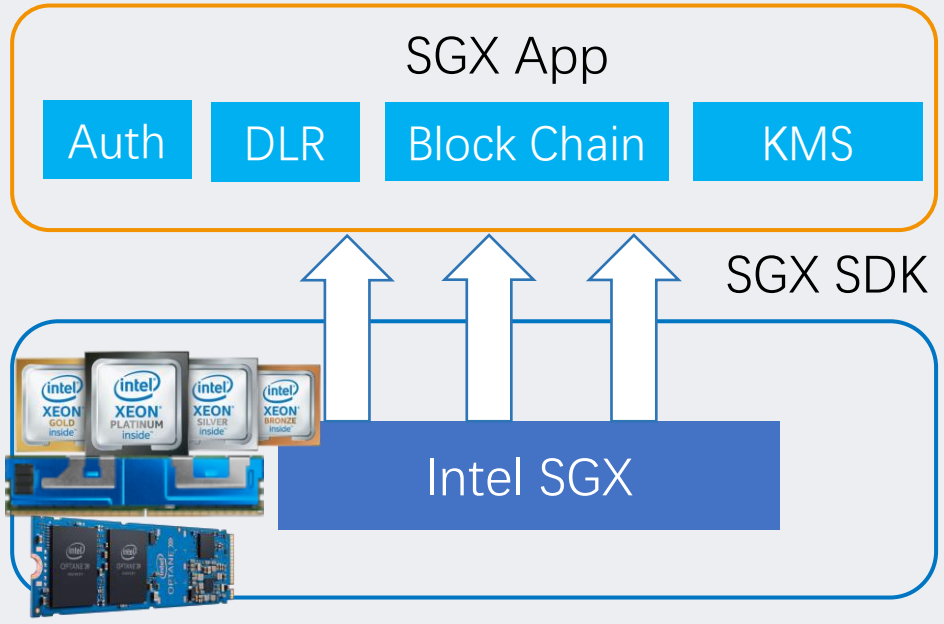
Adversary get nothing (only encrypted data)

Running in SGX

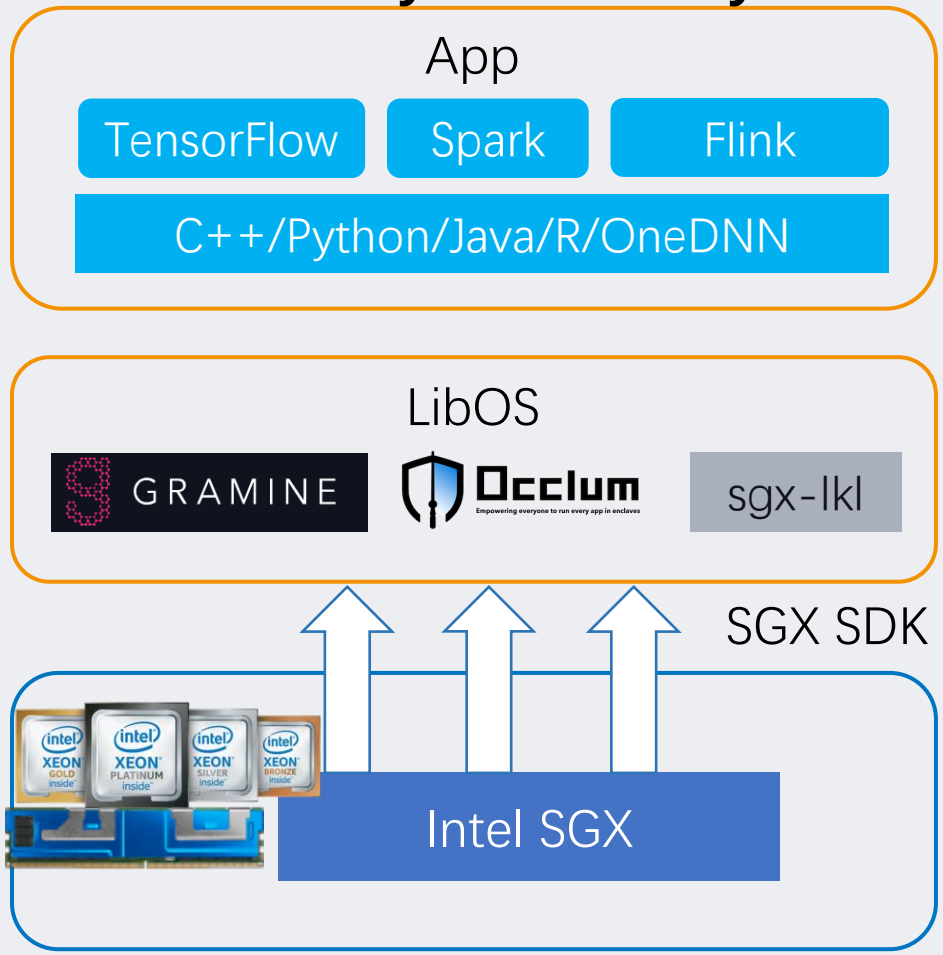
# SGX LibOS (Details in Part 2)

Seemly transfer existing apps into SGX ecosystem

**Security**

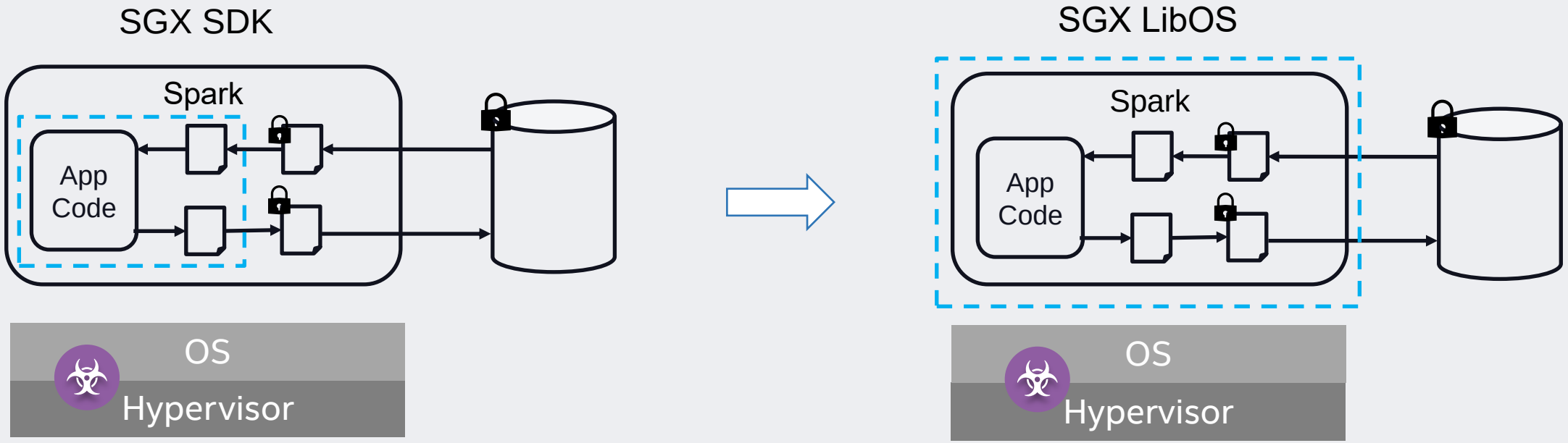



**Security + Usability**



# SGX LibOS Secure Computation

## Running unchanged Spark Applications in SGX




Protect sensitive modules & plain text

Less SGX EPC requirement
Need to change Spark design

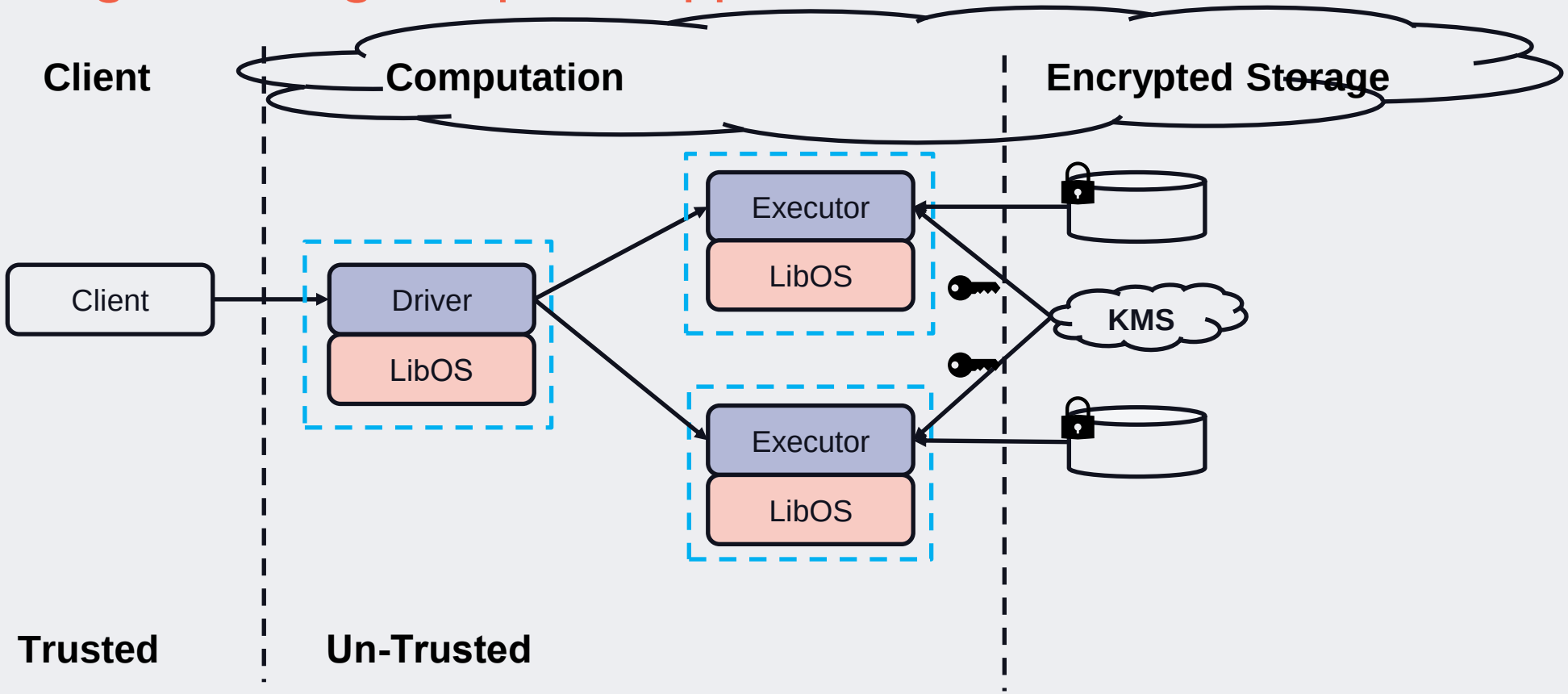
Protect entire Spark & apps

More SGX EPC requirement
Don't have to change Spark design

Running in SGX

# Apache Spark in SGX with LibOS (Overview)

## Running unchanged Spark Applications in SGX




Running in SGX

# Apache Spark Solutions on SGX

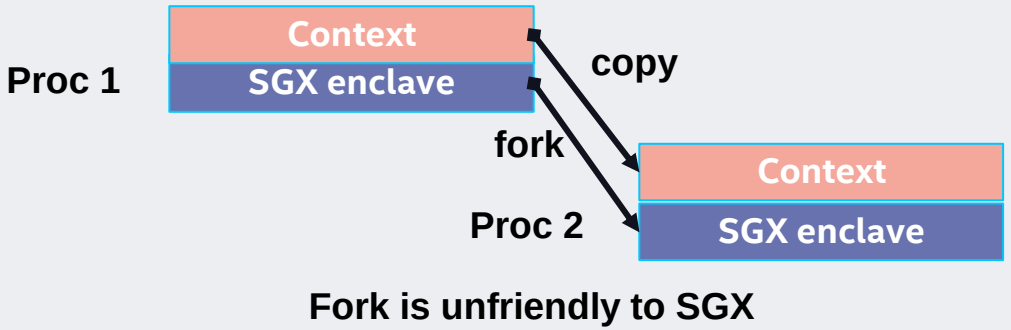
## State of the arts

| | Path | Year | Scope | Deployment |
|---|---|---|---|---|
| **Opaue** | SGX SDK | 2017 | SparkSQL | K8S, Cloud |
| **SGX-Spark** | LibOS sgx-lkl | 2017 | Spark, SparkMLlib | Standalone |
| **PySpark SGX** | LibOS Scone | 2019 | PySpark | Standalone/K8S/Cloud |
| **SoTeRIA** | LibOS Gramine | 2021 | Spark, SparkMLlib | Standalone |
| **BigDL PPML** | LibOS Occlum & Gramine | Since 2021 | Spark/SparkSQL/PySpark/SparkMLib/Deep Learning etc | Standalone/K8S/Cloud |

# What makes the difference? – (1)

## How can we add so many scopes?

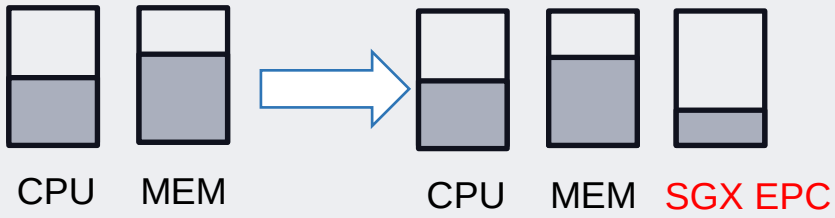
- Hardware: Much larger SGX EPC provided by Xeon Platform
- Mid-Layer: LibOS becomes robust (Part 2)
- Software: Min changes to Apache Spark (upstream in future!)
  - Roles we followed: Only small & necessary changes
  - Remove unnecessary forks from Spark
    - File related
      - rm /tmp/xxxxx
      - chmod /tmp/xxx..
    - Python related..



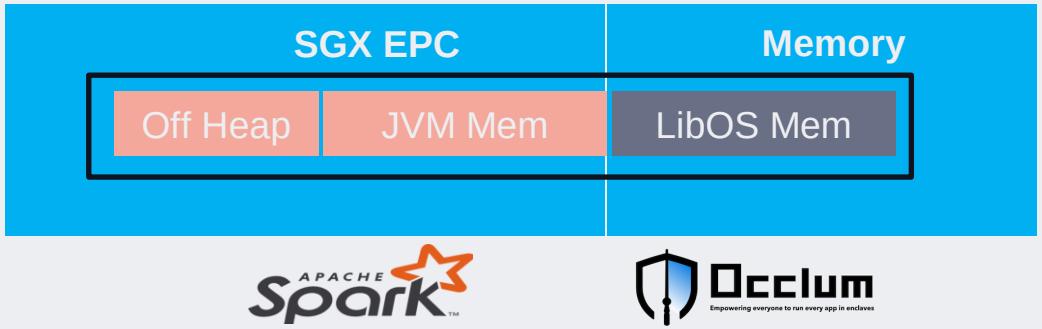
# What makes the difference? – (2)

How can we add so many scopes?

- SGX Resource management
  - Kubernetes SGX device plugins

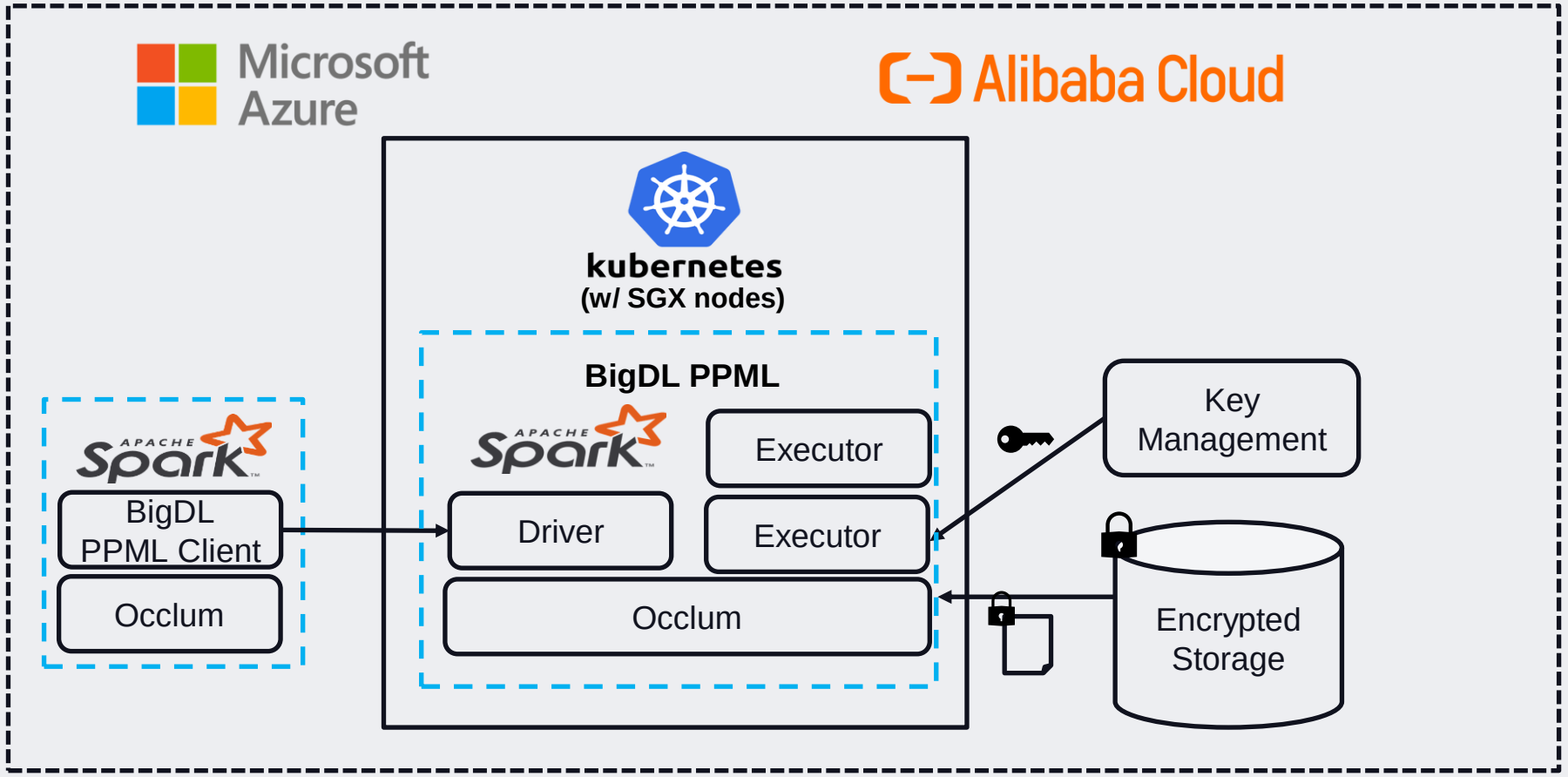



  - Both SGX EPC and Memory are required for SGX LibOS applications



# Full Pipeline in Cloud Environment

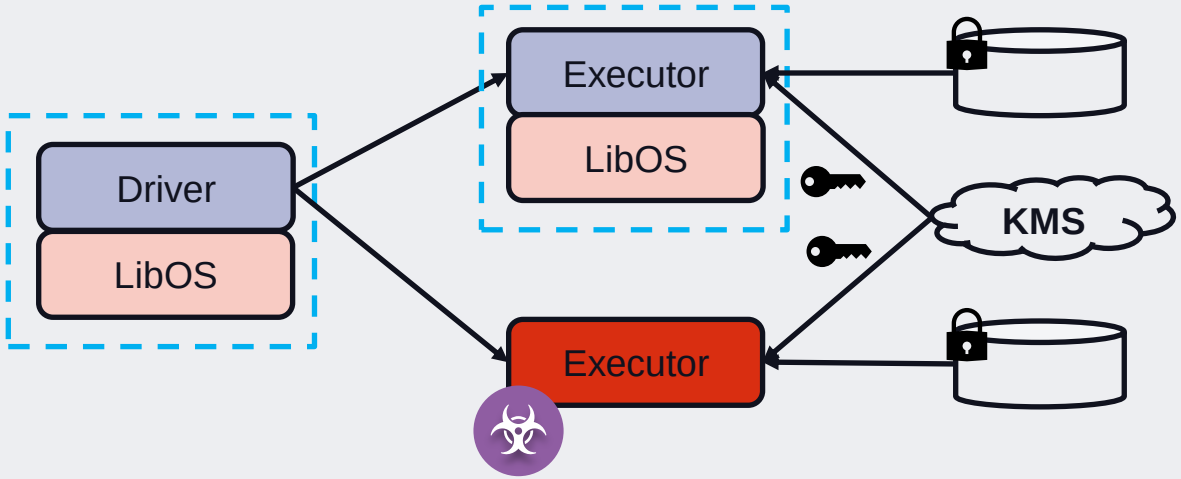
## Running unchanged Spark Applications in SGX



# A step before Privacy & Security

**Security & Privacy in E2E is never an easy job**

- Now Spark is secured by SGX. Do it mean
  - my workloads are secured?
  - my workloads are privacy preserved?
- Answer: No yet!

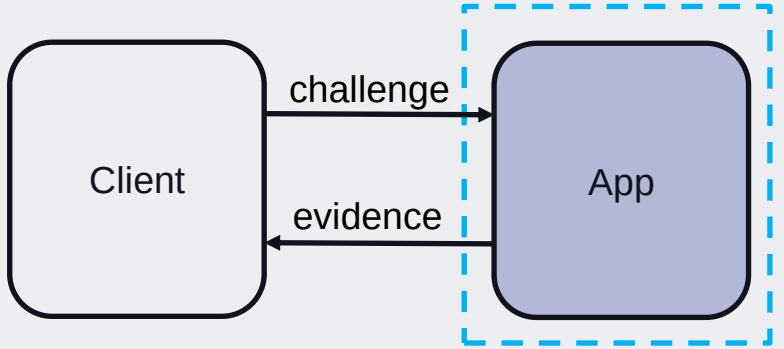



Client/Driver: All executors are running in SGX. Let's go (key & data)!
Attacker: I don't think so. Your data & key are mine! ☺

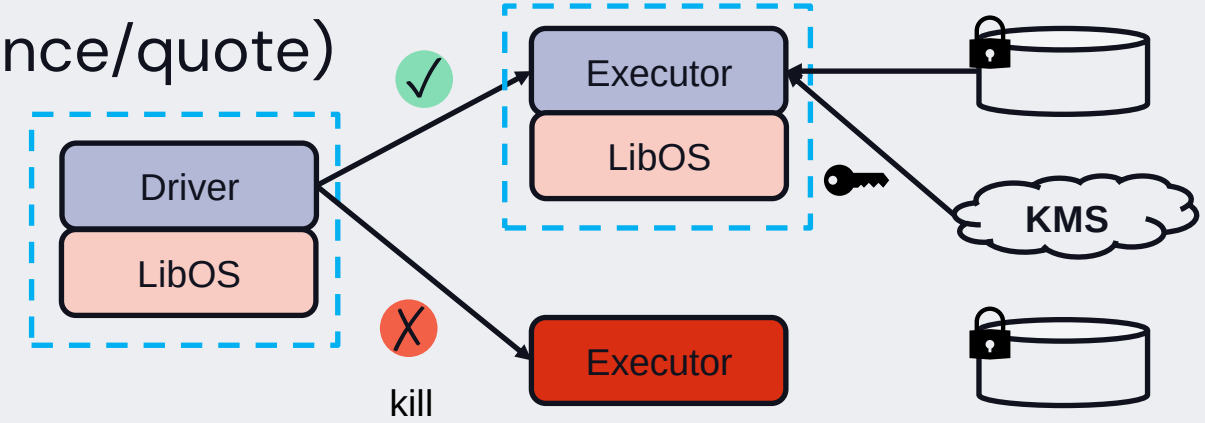
# Ensure Integrity with SGX Attestation

**Security & Privacy in E2E is never an easy job**

- Attestation in short: Verify if an application is running in SGX
  - Application is expected
  - Within SGX
  - Running env is secured
  - …



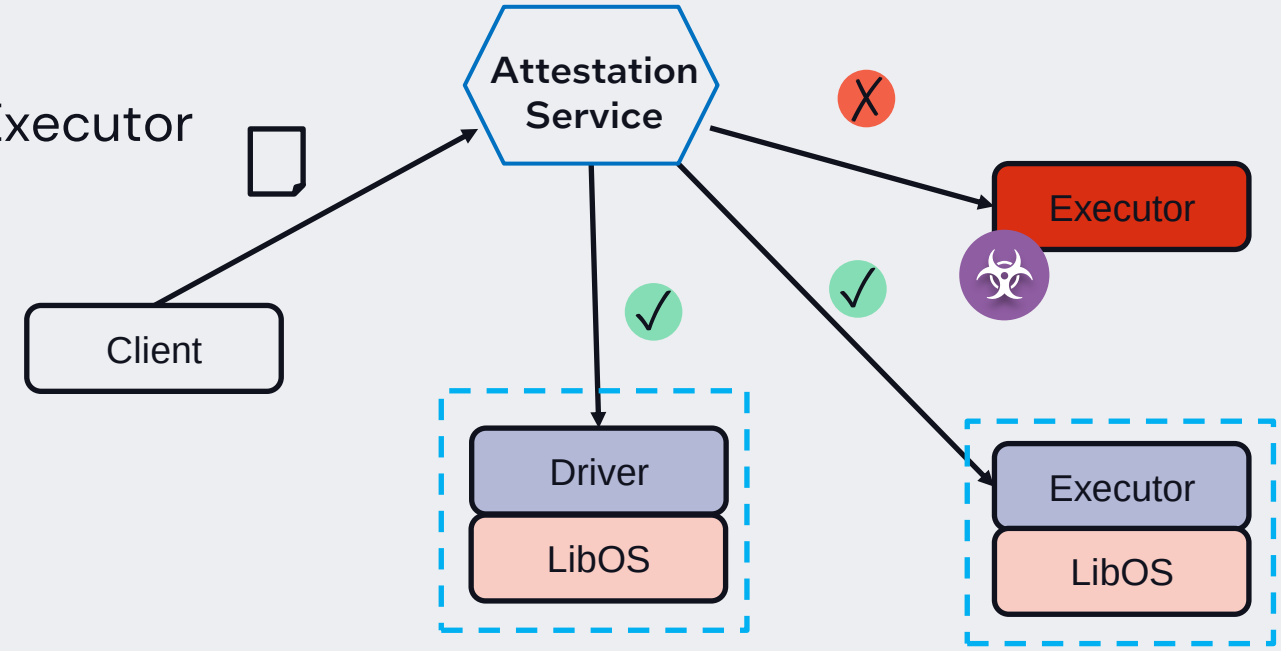

- Attestation result (verify evidence/quote)
  - Look good ✓
  - Not good ✗



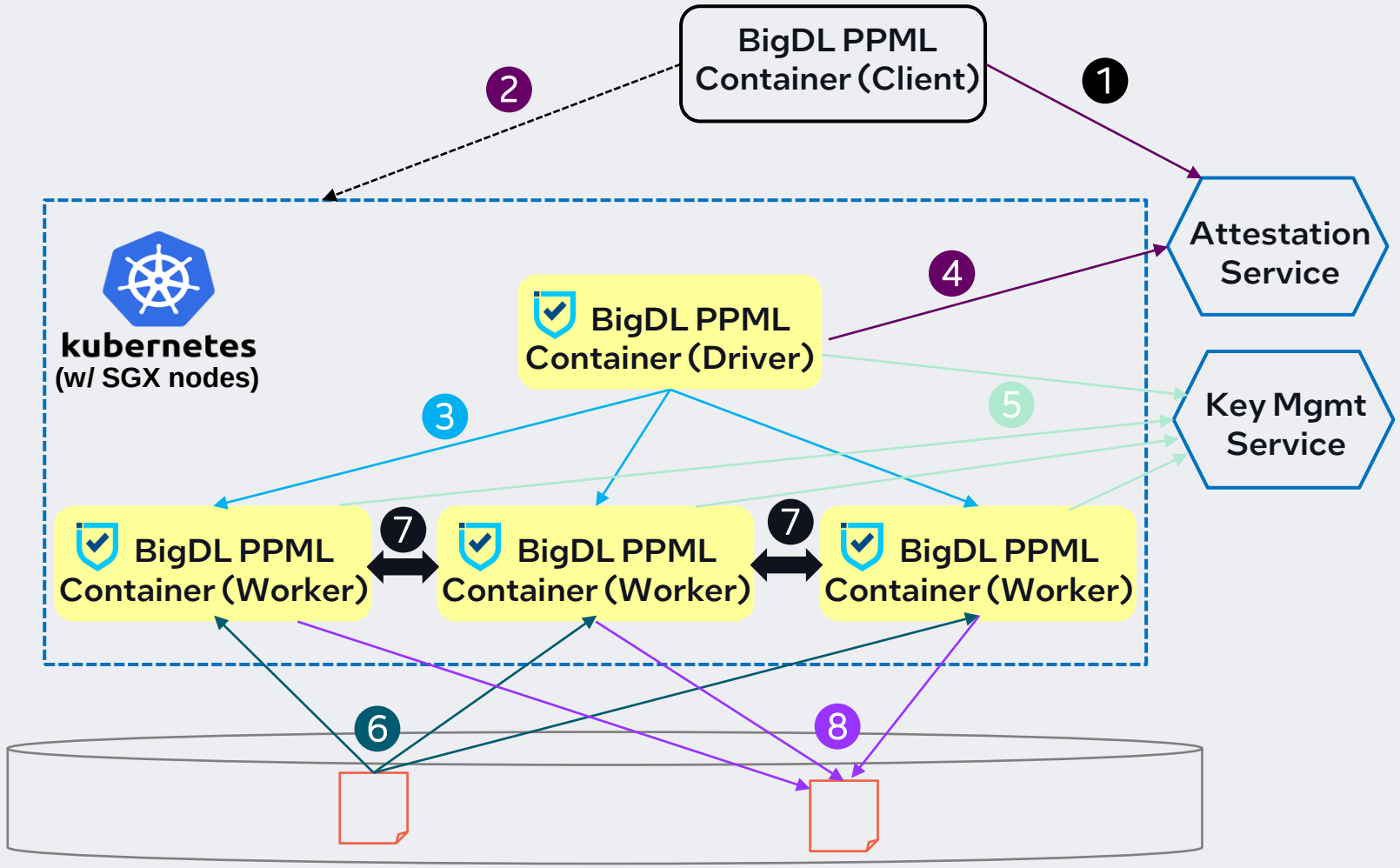
# Ensure Integrity with SGX Attestation

**Security & Privacy in E2E is never an easy job**

- Attestation Service (provided by Cloud/Intel)
  - Client set Policy/Requirement
  - Attestation Service attests Driver/Executor
- Min Impact to Spark
  - Change Context/Entrypoints
  - Attestation dependency in app

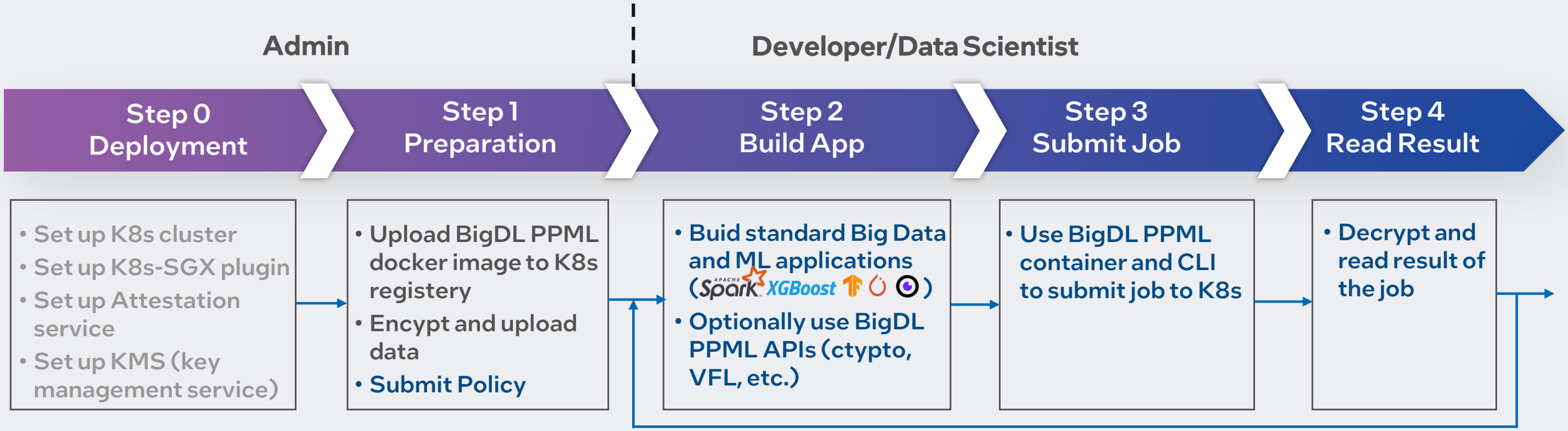


# End-to-End Architecture of BigDL PPML




1. User submits Policy
2. User submits job to K8s (using BigDL PPML CLI), which creates the driver node
3. Driver creates more worker nodes
4. AS attests Driver/executor
5. Driver and workers request keys from KMS
6. Workers read and decrypt input data
7. Workers run distributed Big Data, ML and DL programs
8. Workers encrypt and write output data

# End-to-End PPML Workflow for the User



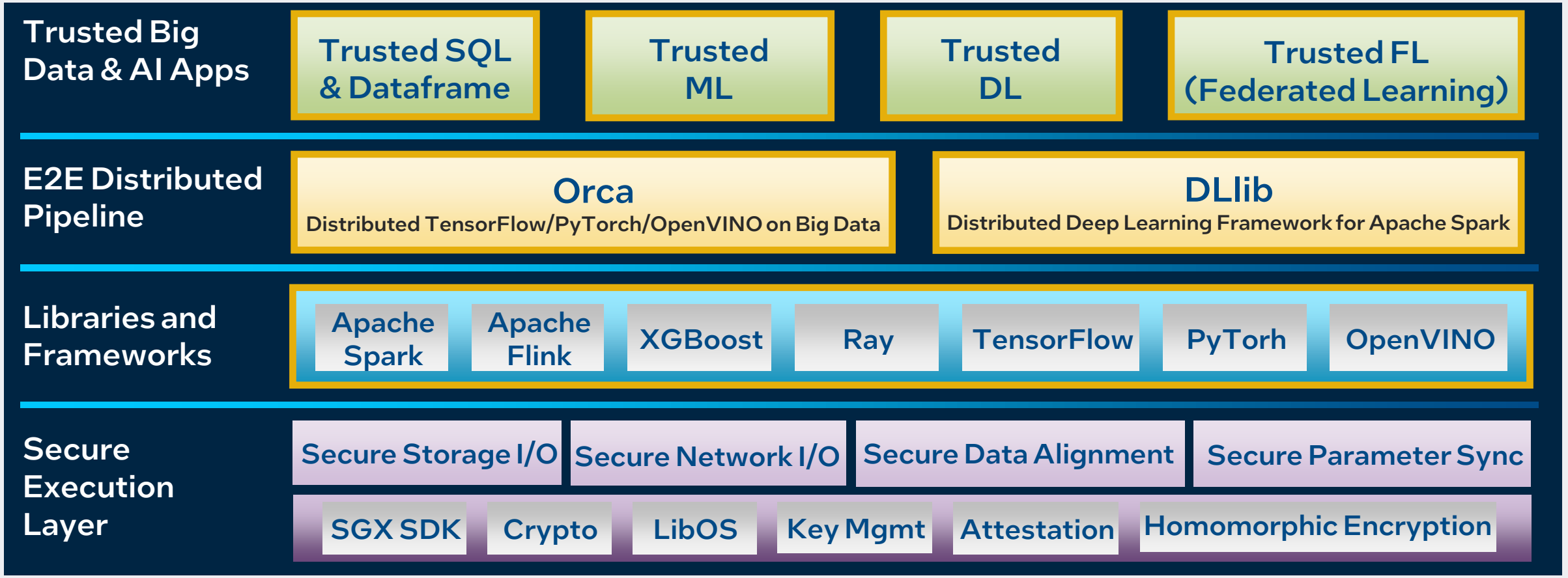

SGX related deployment/setup

Seems like normal Spark with different image

https://hub.docker.com/r/intelanalytics/bigdl-ppml-trusted-big-data-ml-scala-occlum

# BigDL PPML (Privacy Preserving ML)

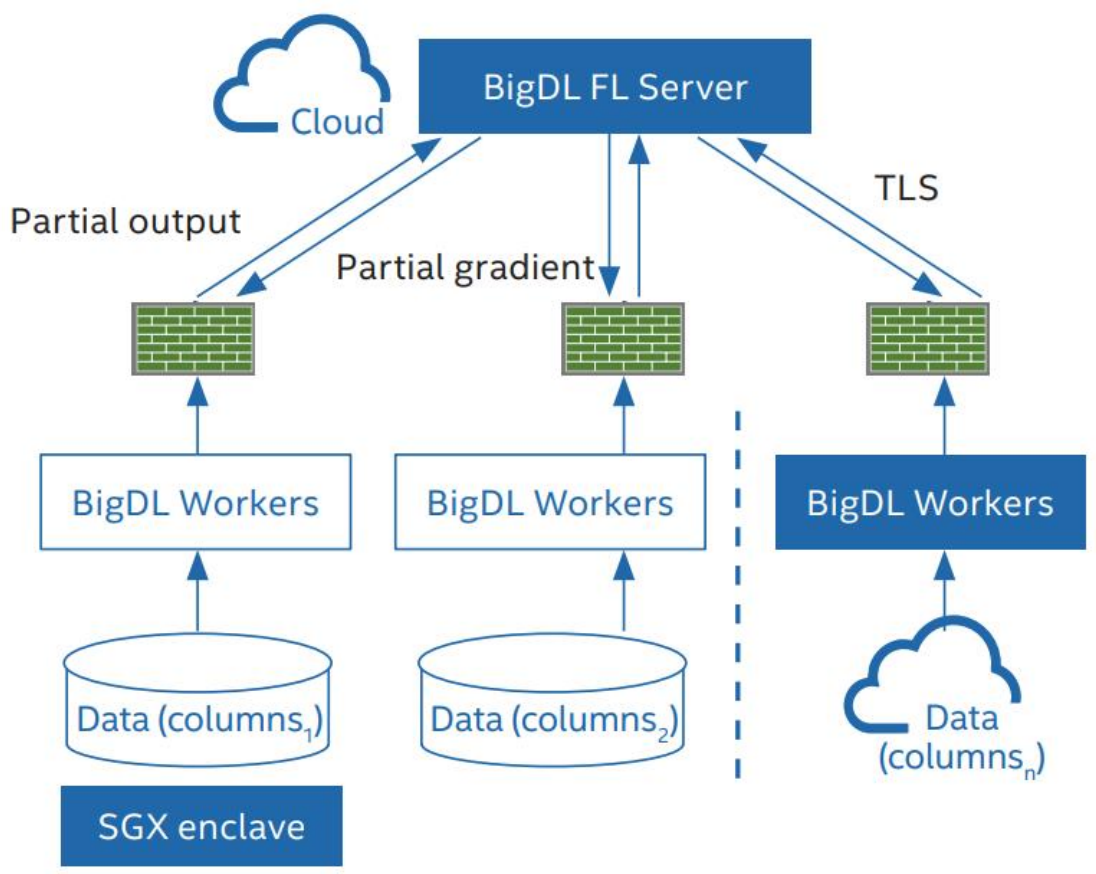
*Secure, Trusted Big Data and AI, even on Untrusted Cloud (using SGX)*




Intel SGX on kubernetes

# Trusted Federated Learning in Finance

## Distributed & Secured Big Data and ML/DL Pipelines




**Trusted Federated Learning**

- Build united model across different parities
  - Training data remain local
  - Aggregation temp/partial results
- Secured computation environment with SGX

Win-Win for all parties

- End users
- Enterprises
- Cloud Service providers

# Examples available on Github


## Privacy Preserving ML & Data Analytics

- Spark Local/Kubernetes Examples with SGX
  - Spark PI
  - TPC-H
  - SparkMLlib: GBDT, Logistic Regression
  - Spark XGBoost
  - Resnet50 Training
  - Federated Learning

https://bigdl.readthedocs.io/en/latest/doc/PPML/Overview/ppml.html
https://github.com/intel-analytics/BigDL/tree/main/ppml/trusted-big-data-ml

# Future works

## Security & Privacy is a long journey

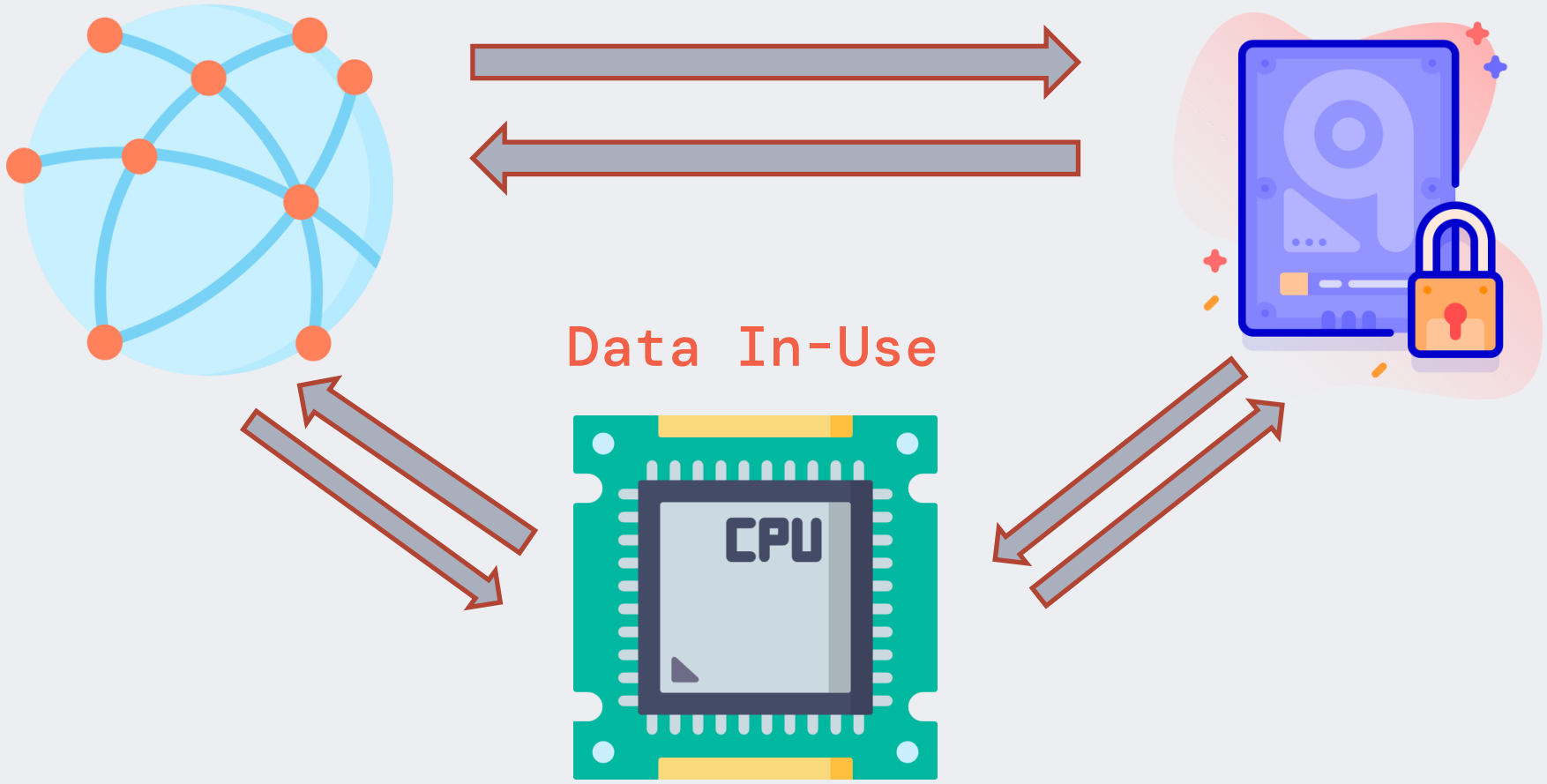
- Towards production
  - Build Ecosystem (Attestation, SGX container)
  - Usability
  - Scalability
- New features
  - From LibOS to RichOS (Normal OS) : Intel TDX (Incoming! This year)
  - Homomorphic Encryption (Computation on encrypted data)

# Part 2
# Security Details with Occlum

# States of Digital Data

Data In-Transmit

Data At-Rest



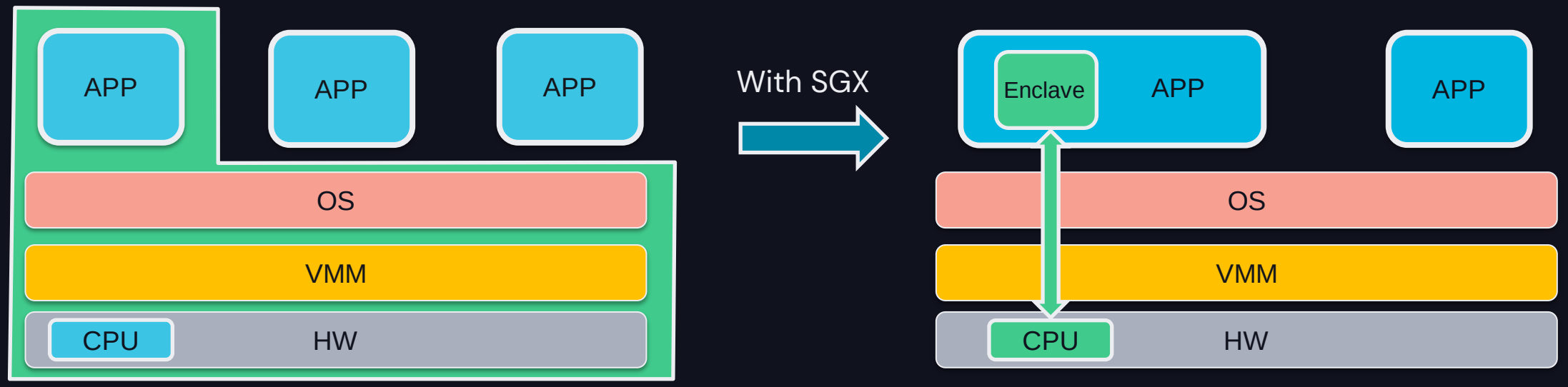
# Confidential Computing

- Using hardware-based Trusted Execution Environments (TEE)
- Protect data in-use for data integrity, data confidentiality
- Only need to trust the hardware, small trusted computing base (TCB)
- Verifiable with Attestation

# Intel® Software Guard Extension

## An implementation of TEE technology

- mature, widely-used
- A set of CPU instructions that manage the hardware-protected memory (Enclave)
- Reduce the TCB to CPU + Enclave



# Use Cases

## Multi-Party Computing

Keep private data protected
while collaboration

## Public Cloud Deployment

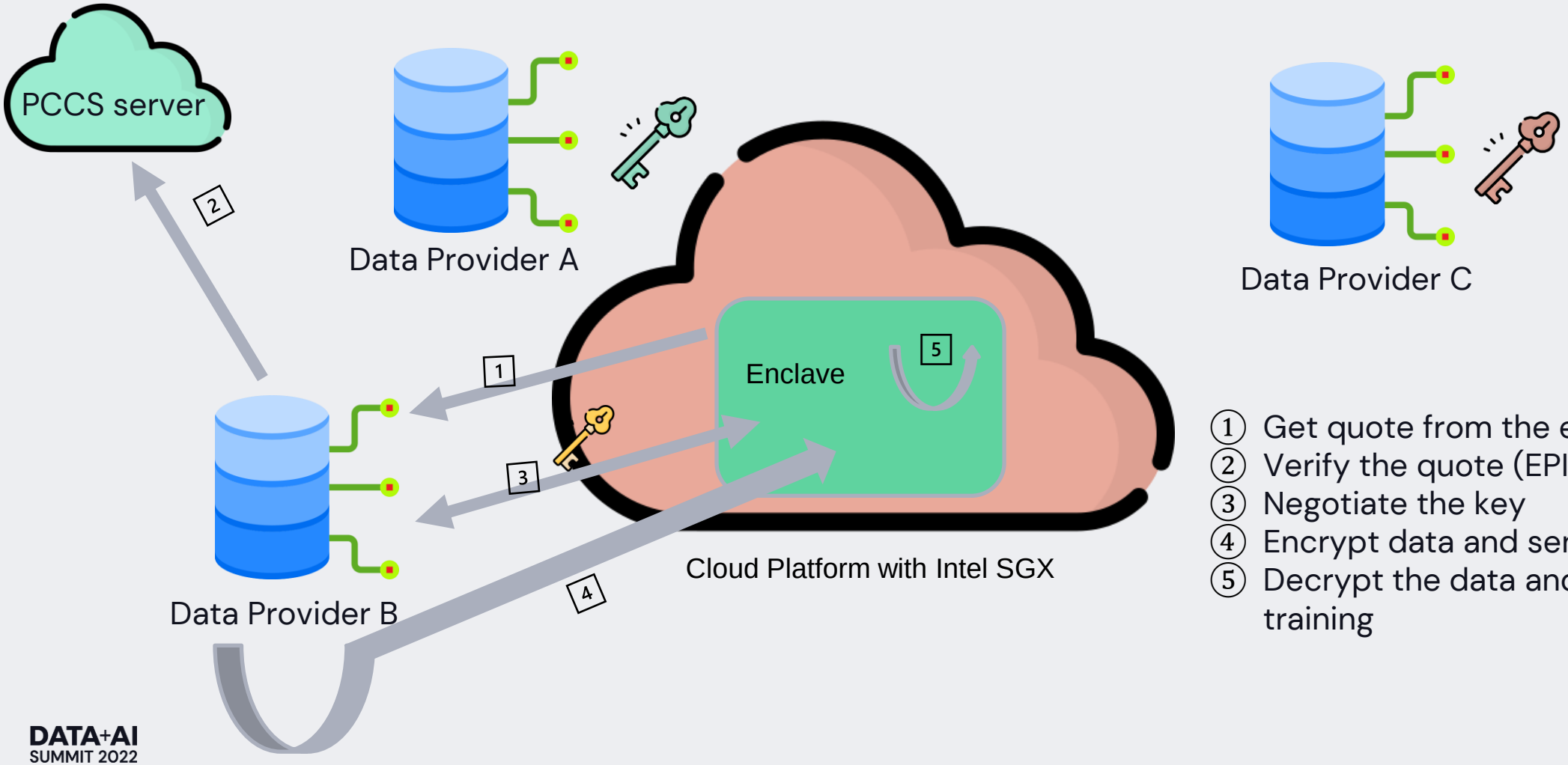
Keep critical app secure

## Key Management Service

Keep private key protected in the Enclave

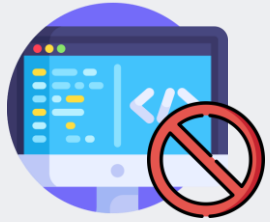
# Example

Multiple data holders train model on public cloud with TEE capabilities




① Get quote from the enclave
② Verify the quote (EPID, DCAP)
③ Negotiate the key
④ Encrypt data and send to enclave
⑤ Decrypt the data and start training

# SGX SDK vs. Library OS

| SGX SDK | Library OS |
| --- | --- |
|  Re-Design<br>Ring 3, no OS access<br>Trusted, untrusted |  Almost Full OS Accessibility |
|  Re-Engineering<br>Code change |  No Code Change |
|  Re-Compilation<br>Extra SGX dependencies |  No recompilation |

# The LibOS Solution for SGX

Occlum

Empowering everyone to run every app in enclaves

Occlum

Empowering everyone to run every app in enclaves

# Empowering Everyone to run every app in enclaves

- *Occlum: Secure and Efficient Multitasking Inside a Single Enclave of Intel SGX (ASPLOS' 20)*
- Created by Ant Group in 2019
- Donated to CCC (Confidential Computing Consortium of Linux Foundation) in 2021
- https://github.com/occlum/occlum

ANT GROUP

THE LINUX FOUNDATION PROJECTS

CONFIDENTIAL COMPUTING CONSORTIUM

# Key Features

## Efficient Multi-tasking

- Single-address-space architecture
- Multiple processes share the same enclave
- Super fast process startup and IPC

## Memory Safety

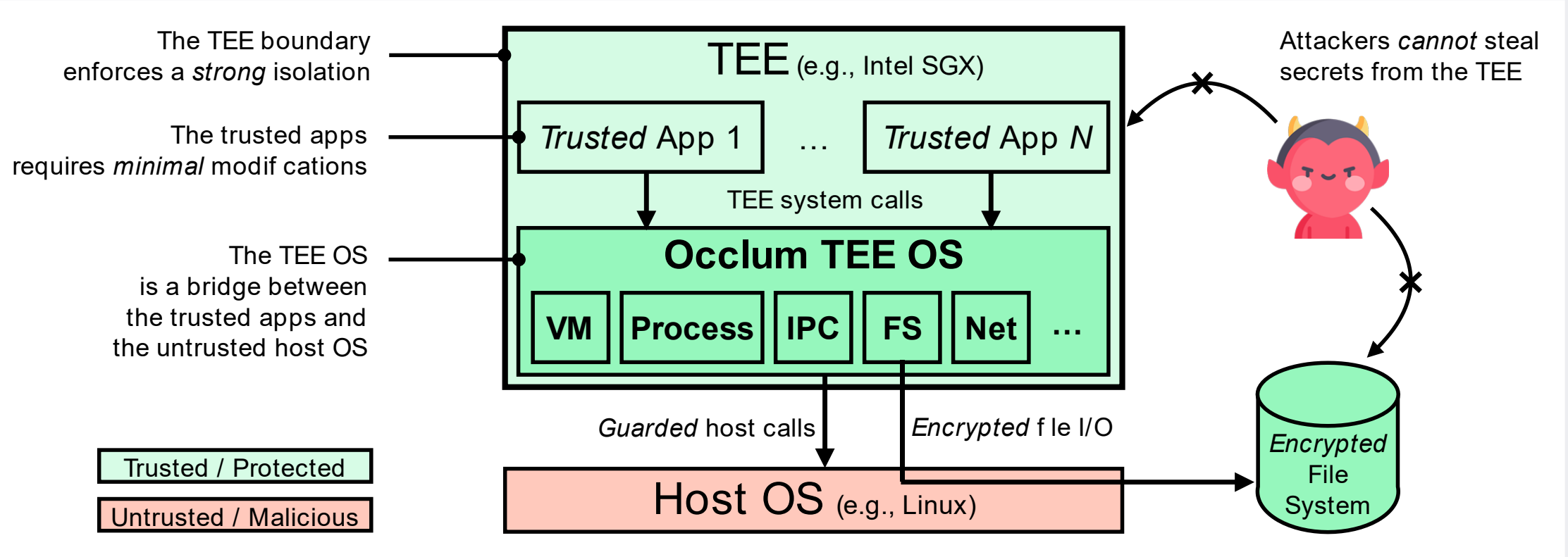
- First SGX LibOS written in Rust
- Rust is designed to be memory safe. It does not permit null pointers, dangling pointers, or data races

## Ease of Use

- Empowering everyone to run apps in Enclave
- Similar user commands with Docker

# Architecture

Occlum
Empowering everyone to run every app in enclaves

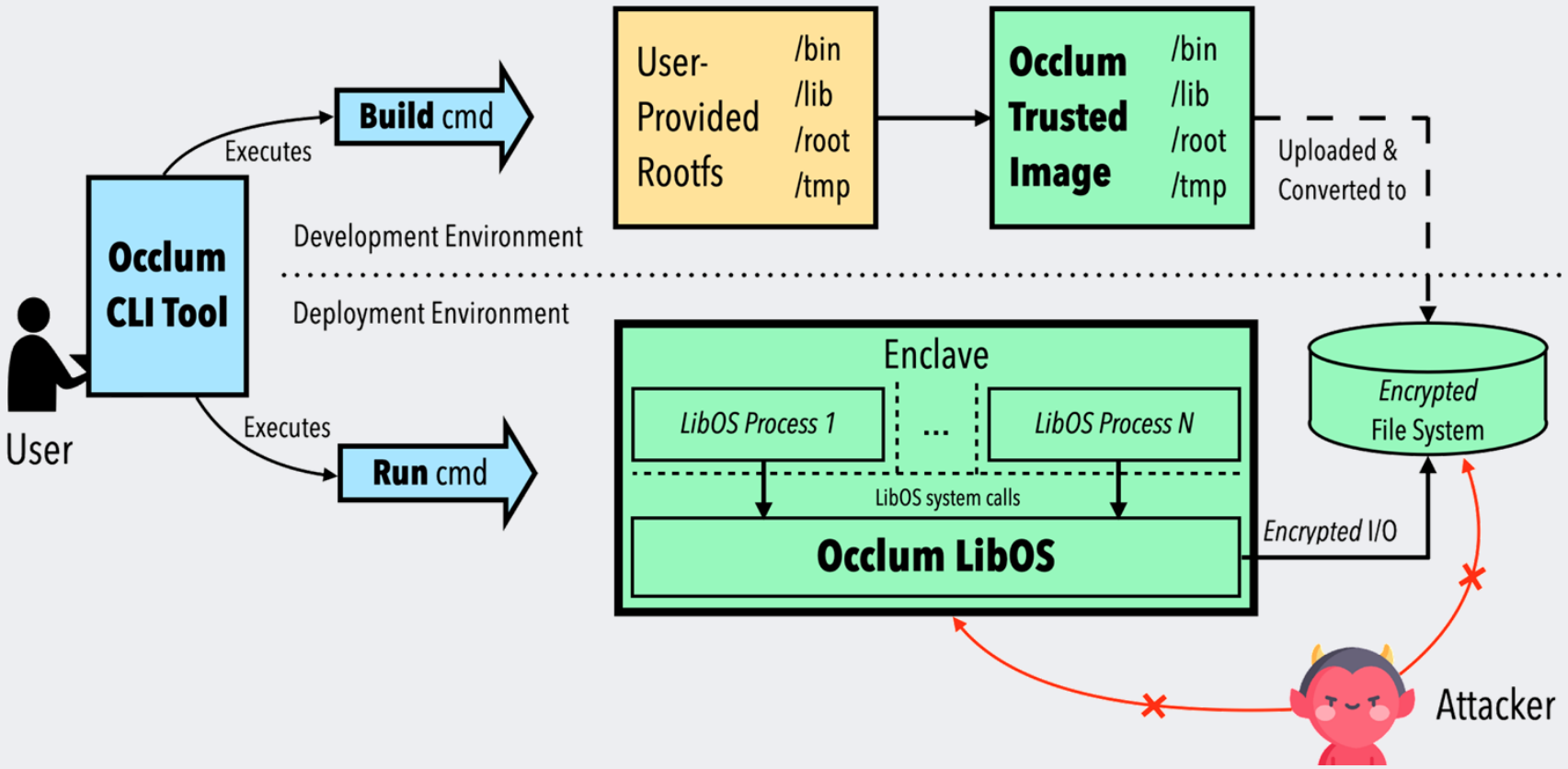



https://github.com/occlum/occlum

# Occlum Commands

## Ease of Use

- occlum new/init
- occlum build
- occlum run
- occlum start/exec



# Use Cases

https://github.com/occlum/occlum/tree/master/demos

| Programming Language |
| --- |
| C/C++ |
| JAVA |
| Python |
| Go |
| Rust |
| Shell Script (Bash, Fish) |
| … |

| Popular Applications |
| --- |
| Redis |
| SQLite |
| Vault |
| PyTorch |
| Flink |
| Xgboost |
| … |

# Collaboration

## Who is using Occlum



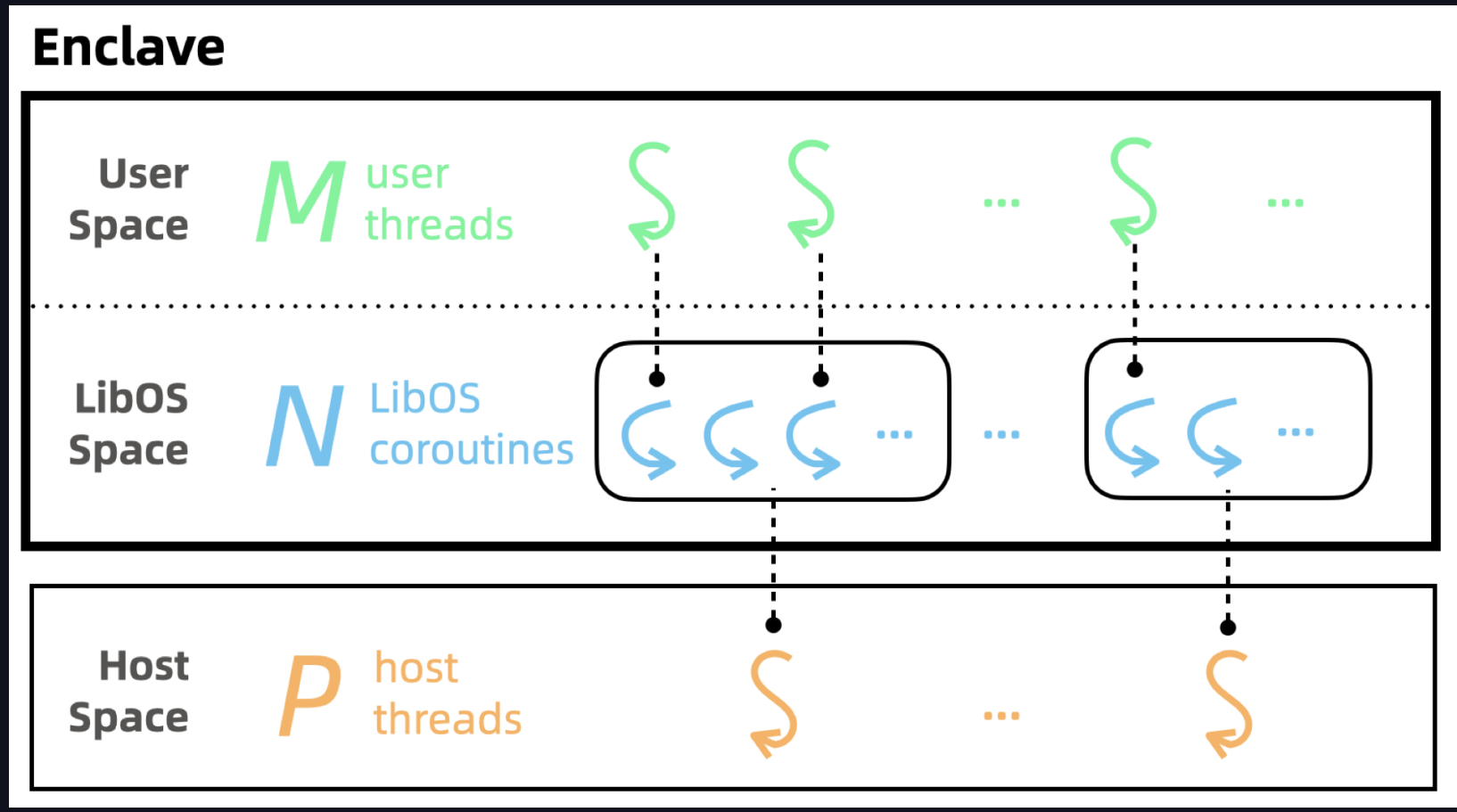



[1] Azure: https://docs.microsoft.com/en-us/azure/confidential-computing/confidential-containers#occlum
[2] Alibaba Cloud: https://www.alibabacloud.com/blog/inclavare-confidential-computing-container-technology-for-cloud-native_596708
[3] Edgeless System: https://blog.edgeless.systems/marblerun-now-supports-occlum-even-more-confidential-computing-at-scale-2f6dd17e00c0
[4] Intel: https://community.intel.com/t5/Blogs/Tech-Innovation/Artificial-Intelligence-AI/Better-Together-Privacy-Preserving-Machine-Learning-Powered-by/post/1335716
[5] Ant: https://www.mo4tech.com/sofaenclave-the-next-generation-trusted-programming-environment-of-ant-financial-enables-confidential-computing-to-protect-financial-business-for-102-years.html

# Next Generation Occlum

## In-Enclave Scheduling

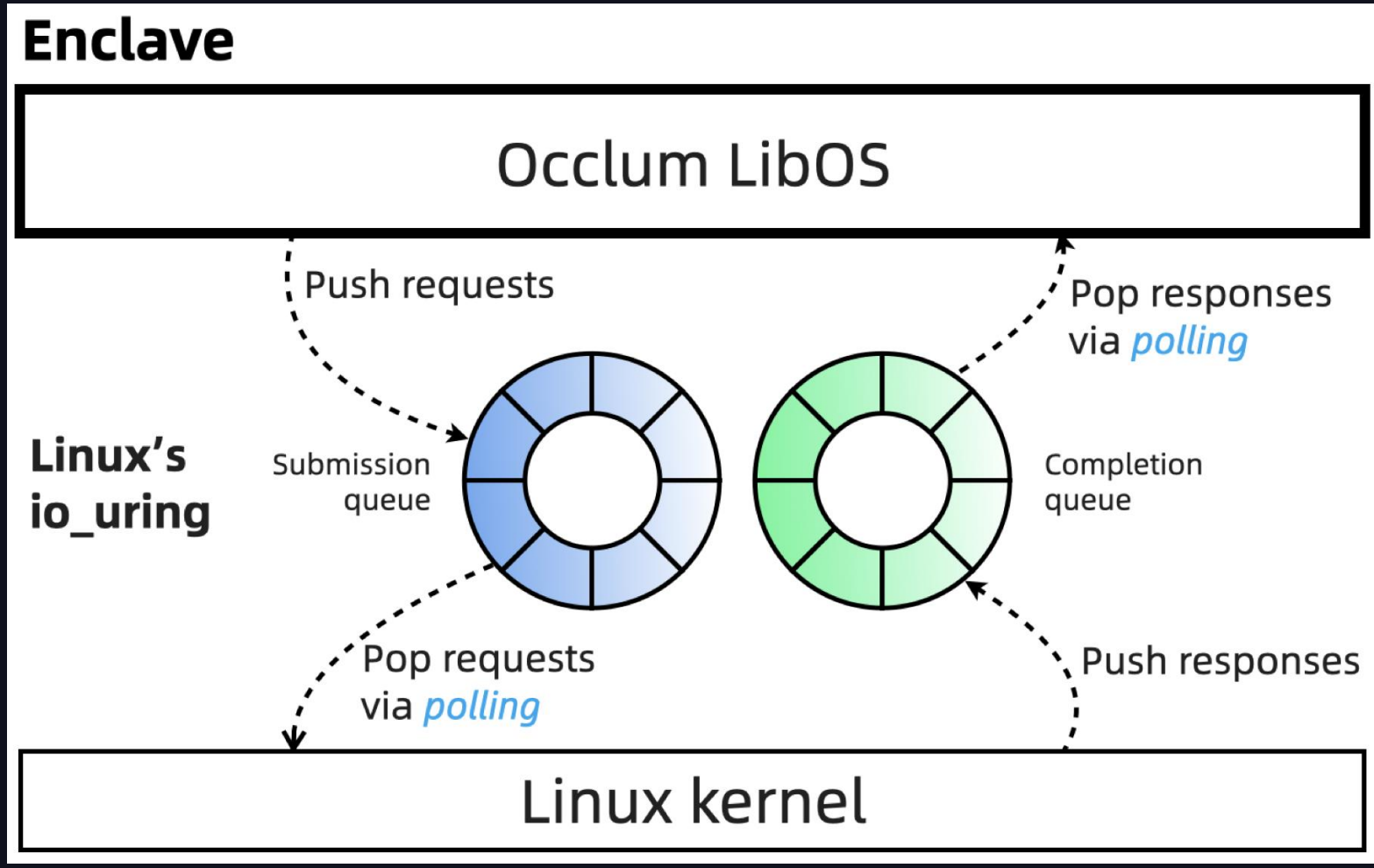
- Coroutine based
- Supports tons of user threads



# Next Generation Occlum

## Switchless Async IO

- Based on Linux io_uring
- Two ring buffers shared by the kernel and applications
- Very efficient for large IO throughput



# Future Work

- Add SGX EDMM support for higher memory performance
- Polish Next-Gen Occlum (NGO: https://github.com/occlum/ngo) for best performance and stability
- Support a long list of frequently-used applications

DATA+AI SUMMIT 2022

# Thank you

Qiyuan Gong
Software Arch, Intel

Hui, Chunyang
Senior Engineer, Ant Group