Gramine:

Securing unmodified Linux Applications with Confidential Computing

Prof Don Porter, UNC, Chapel Hill

Mona Vij, Principal Engineer, Intel Labs

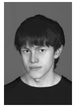
CONFIDENTIAL COMPUTING CONSORTIUM

Meet the team

intel

Dmitrii Kuvaiskii
Mona Vij
Isaku Yamahata

ITL
INVISIBLE THINGS LAB



Michal Kowalczyk
Paweł Marczewski
Borys Popławski
Rafał Wojdyła

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Don Porter

TEXAS A&M
UNIVERSITY

Chia-Che Tsai

IBM

Alibaba Cloud

golem

Several other contributors from several companies and academic partners

CONFIDENTIAL COMPUTING
CONSORTIUM

Lift and Shift Unmodified Application

- In un-trusted cloud and edge deployments, there is a strong desire to shield the whole application from rest of the infrastructure

- Developers want end-to-end secure solutions with “push-button” approach

- Gramine supports lift and shift paradigm for unmodified application for CC with Intel SGX

Gramine Project Summary

- Gramine project (formerly Graphene) joined Confidential Compute Consortium in Sept ‘21 with initial TAC approval in APR’20
- Gramine runs unmodified Linux Applications on several platforms
  - Current focus on Intel® SGX
- Community maintained Open-Source (LGPL) project hosted on Github
- Well defined testing and validation criteria with CI/CD (Jenkins)
- Project maintenance is governed via a well-defined governance criteria
- Cloud deployment with Azure Kubernetes Service
- Production ready Gramine 1.0 released in Oct’21 with active development towards future releases

CONFIDENTIAL COMPUTING CONSORTIUM

Growing Community

intel

Alibaba Cloud

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

TEXAS A&M UNIVERSITY

IBM

golem

ITL INVISIBLE THINGS LAB

AI/ML

TensorFlow

OpenVINO

PyTorch

Databases

MEMCACHED

redis

Web Servers

LIGHTTPD fly light.

NGINX

APACHE HTTP SERVER

Languages

python™

GCC

R

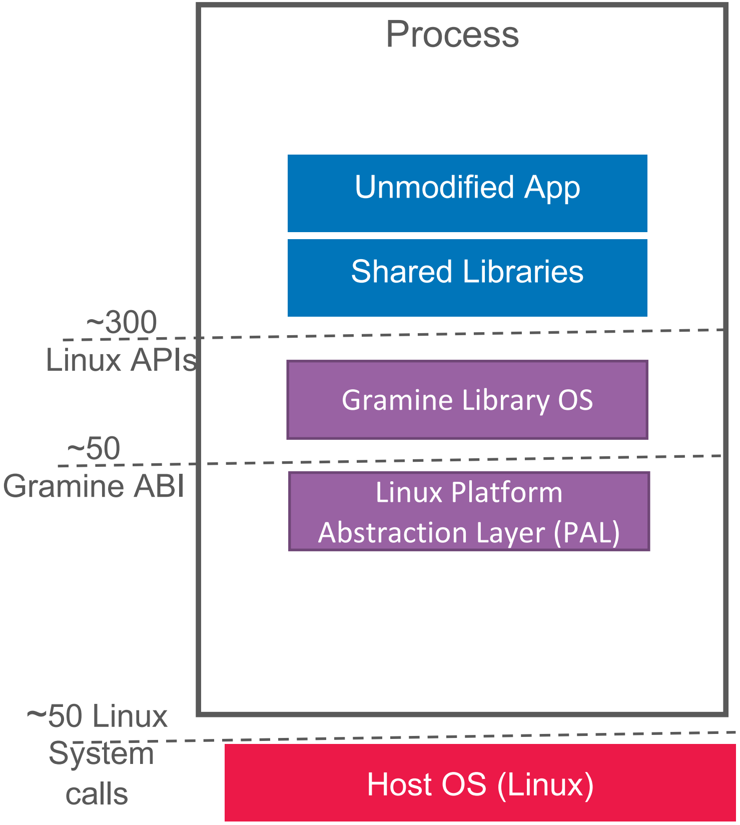
R

Misc

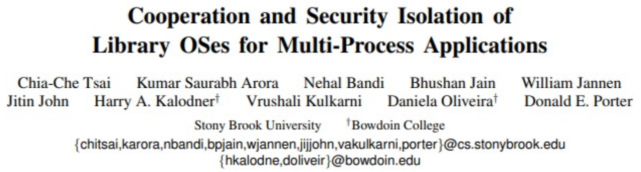
blender®

node JS®

Gramine Library OS Architecture



SUNY Stony Brook
Graphene [EuroSys'14]

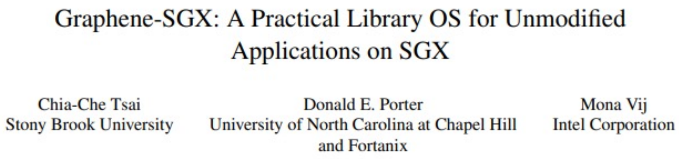


Cooperation and Security Isolation of Library OSes for Multi-Process Applications

Chia-Che Tsai Kumar Saurabh Arora Nehal Bandi Bhushan Jain William Jannen
Jitin John Harry A. Kalodner[†] Vrushali Kulkarni Daniela Oliveira[†] Donald E. Porter

Stony Brook University [†]Bowdoin College
{chitsai,karora,nbandi,bpjain,wjannen,jijjohn,vakulkarni,porter}@cs.stonybrook.edu
{hkalodne,doliveir}@bowdoin.edu

Intel Labs and SUNY Stony Brook
Graphene-SGX [ATC'17]



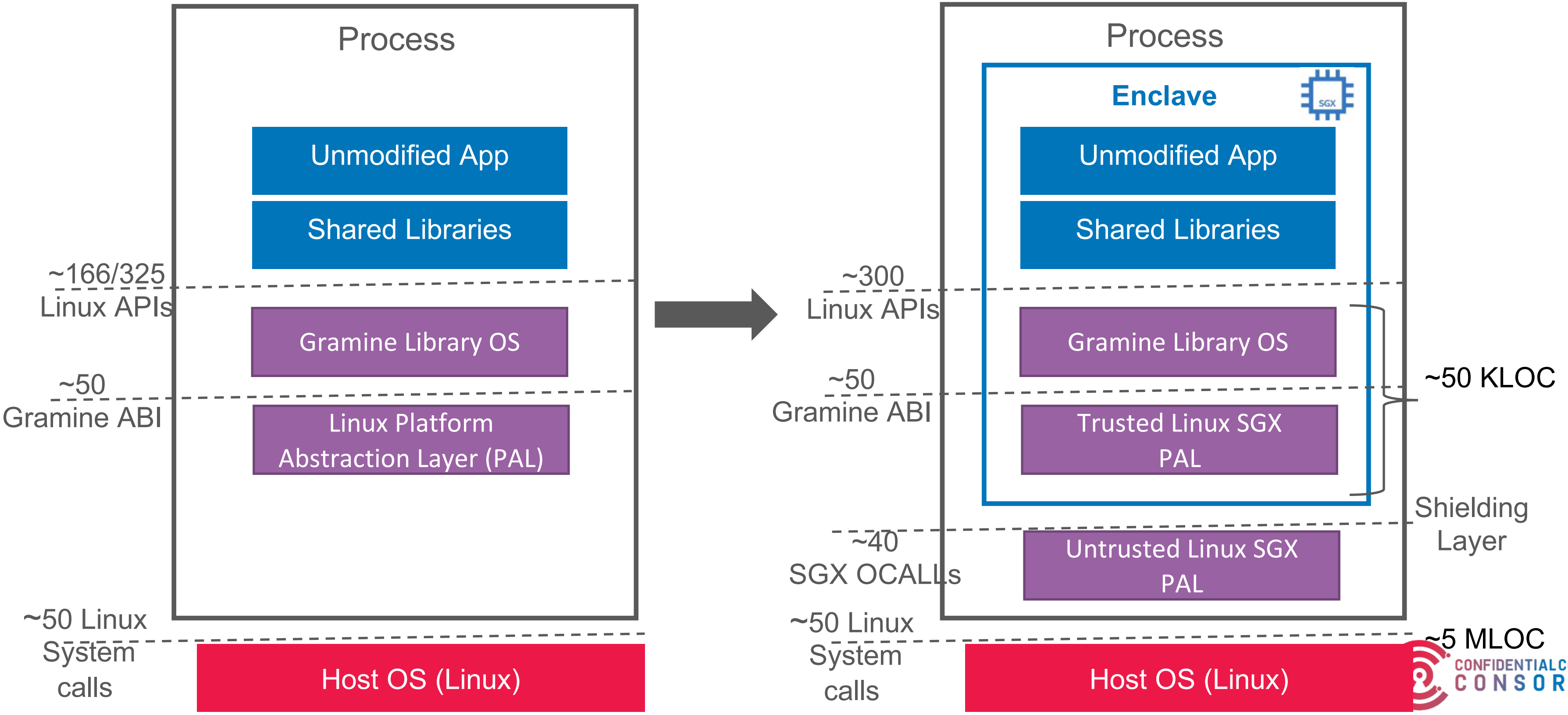
Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX

Chia-Che Tsai
Stony Brook University

Donald E. Porter
University of North Carolina at Chapel Hill and Fortanix

Mona Vij
Intel Corporation

Library OS architecture is very suitable for Intel® SGX



CONFIDENTIAL COMPUTING CONSORTIUM

Gramine Shielding Layer

- Enabling applications in Gramine requires a manifest defining the security policies enforced by Gramine

- All security-critical paths are hardened against eavesdropping/attacks

- Gramine supports dynamic loading and Integrity of the loadable libraries is verified via checking against valid hash values as specified in the application manifest

- All network communication is assumed to be SSL/TLS-protected by the app itself

Gramine Features for SGX Deployments

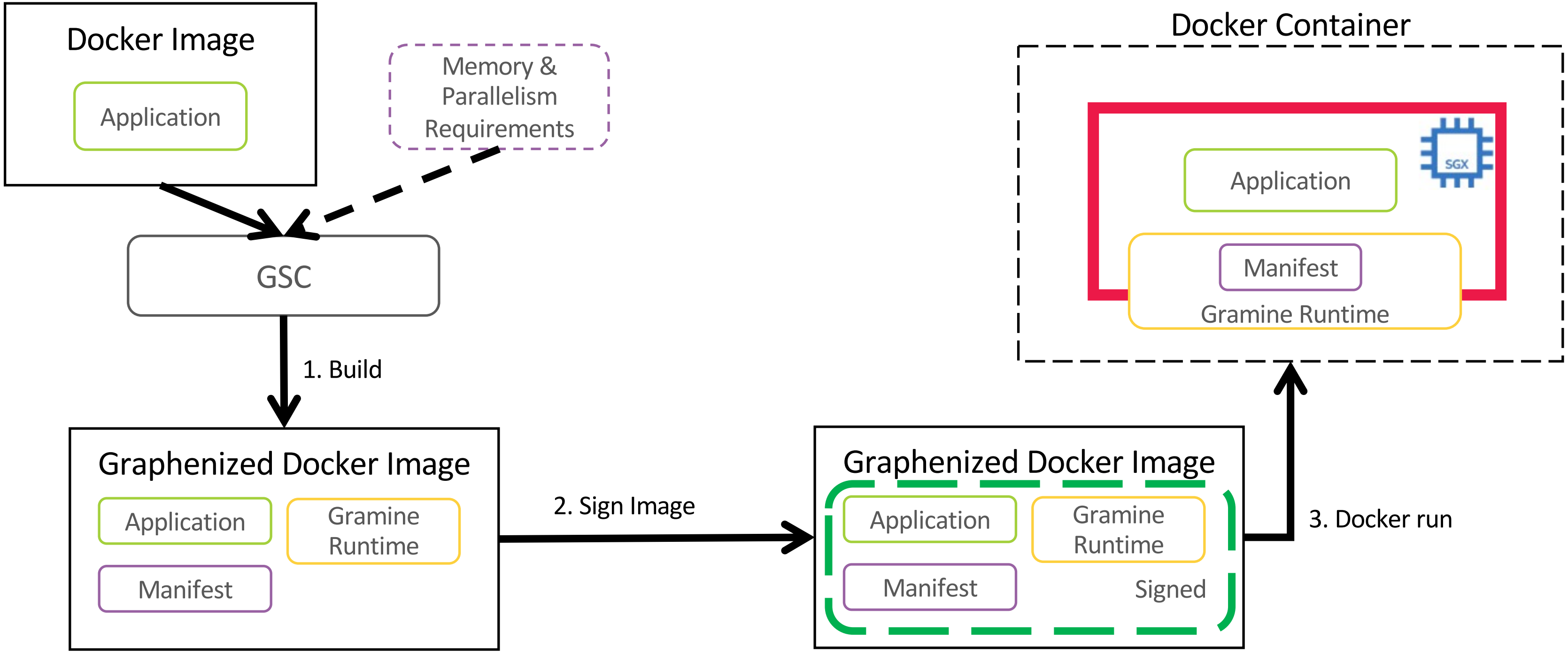
- SGX Attestation
  - Supports both EPID and DCAP/ECDSA SGX attestations
- Protected Files
  - Automatically encrypt/decrypt specified files in the manifest
- Asynchronous System Calls
  - Exit-less support as a performance enhancement feature
- Multi-process support
  - Fork and secure comm between parent and child process via encrypted IPC
- Docker Integration
  - Automatically convert Docker images to Gramine images

Gramine Remote Attestation

There level approach to Attestation

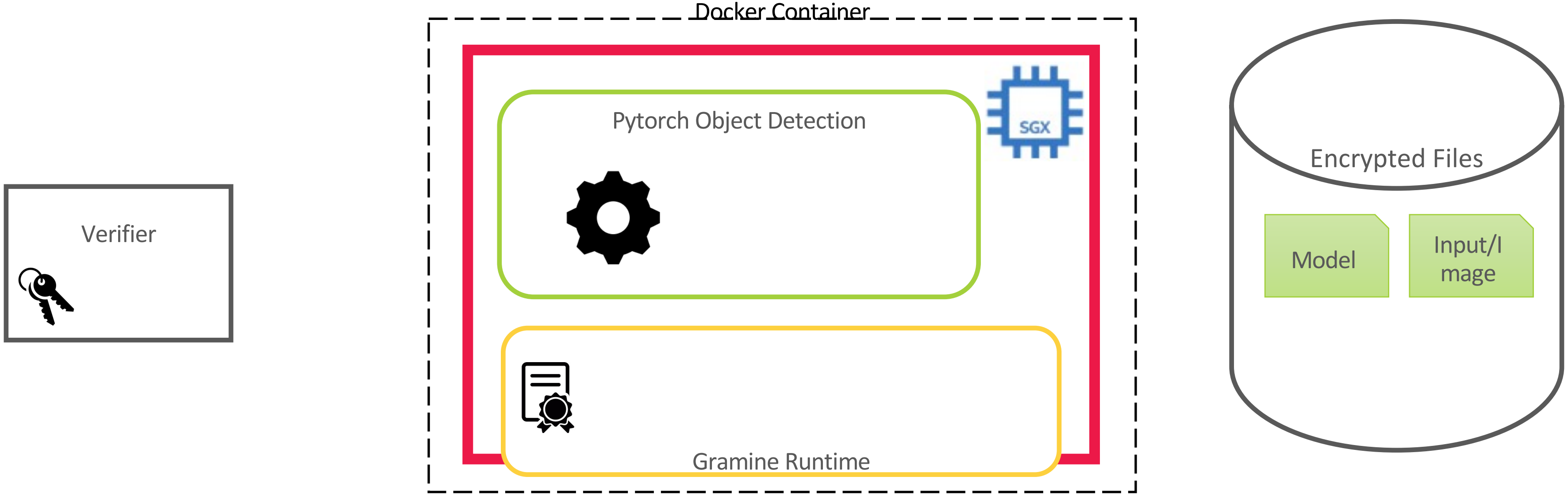
- Remote/Local Attestation Support:
  - Exposed via `/dev/attestation` pseudo-filesystem
  - Integrates with multiple backends under the hood including Intel DCAP
- Protected Channel Establishment
  - Constructed using RA-TLS (Remote Attestation integrated with Transport Layer Security)
- Secret Provisioning
  - Built using secret provisioning libraries

Gramine Shielded Containers (GSC)

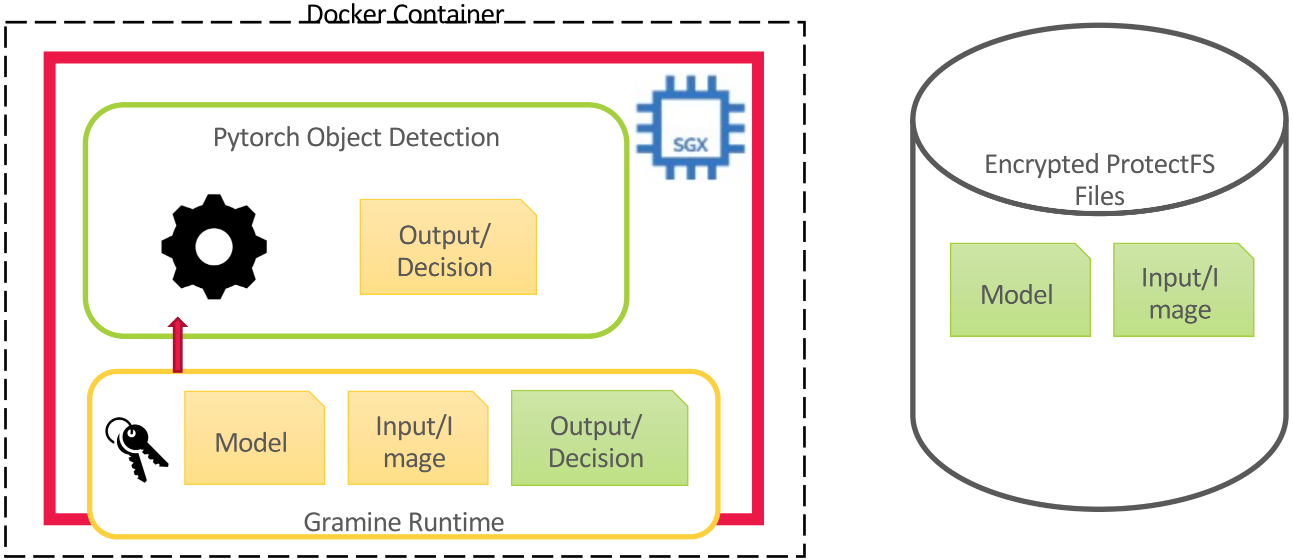


CONFIDENTIAL COMPUTING CONSORTIUM

End-to-End Secure Machine Learning with Pytorch



End-to-End Use Case using Pytorch



[PyTorch PPML Framework Tutorial](#)

Gramine is actively evolving

- Initial SGX port released in 2017
- Open-source community established in Dec 2018
- First major release was in September 2019
- First production release 1.0 in Oct 2021
  - All known security issues were fixed
  - Huge difference between this release and the first release (~2 years)
- Continue to do future releases at a quarterly cadence

Sample Open-source Project Integration

- Edgeless systems/MarbleRun - Service mesh for confidential Computing
  - Supports Gramine for deployment with K8 environments
  - Stand alone backend for Gramine attestation and secret provisioning

Use Cases

- Several use cases under development – expect to see deployments in upcoming months
  - Trusted Federated learning
  - Trusted model training
  - Trusted analytics
  - Privacy Preserving machine learning

- Several startups building their use cases with Gramine Confidential Containers

Gramine Project Future Plans

- Continue development to support additional runtimes and workloads
- Integration with industry confidential container deployments
- Support additional TEE backends e,g TDX
- Support for communication with hardware accelerators
- Explore coarse grain partitioning for certain I/O bound applications

Gramine Project

- Technical Charter
  - Gramine charter is slightly modified from the CCC template
    - Minor changes on requiring majority votes
- Project Code of Conduct
  - We started with Contributor Covenant
  - Discussion ongoing and working on finalizing something that works for our project.
- Gramine Project - https://github.com/gramineproject
  - Core gramine - https://github.com/gramineproject/gramine
  - Examples - https://github.com/gramineproject/examples
  - Gramine Shielded Containers - https://github.com/gramineproject/gsc
  - Third party code related to Gramine - https://github.com/gramineproject/contrib
  - Archived Graphene - https://github.com/gramineproject/graphene
- Issue Tracker
  - https://github.com/gramineproject/gramine/issues
- Documentation
  - Gramine: https://gramine.readthedocs.io/en/latest/
  - GSC: https://gramine.readthedocs.io/projects/gsc/en/latest/

Current Mode of Operation

- UNC Zoom for team meetings
- Gitter chat service
  - Moved from Slack
- Google group mailing list
  - Open to moving to confidential computing mailing list
- Website hosted by Golem
  - Would like help from LF to maintain and update the website
- Jenkins infrastructure hosted at UNC
  - Would like help from getting latest hardware
- LF License Scanning
  - Would like to learn more and potentially use

Vulnerability Management Coordination

- Provide a way to easily communicate and exchange security information between the projects

GRAMINE

Gramine project:
http://www.Gramineproject.io

GitHub repo:
https://github.com/Gramineproject

Gramine Documentation:
https://Gramine.readthedocs.io

CONFIDENTIAL COMPUTING CONSORTIUM

@ConfidentialC2
confidential-computing
confidentialcomputing.io

CONFIDENTIAL COMPUTING
CONSORTIUM