Intel® Software Guard Extensions(Intel® SGX), Instructions and Programming Model

Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos Rozas, Vedvyas Shanbhogue, Uday Savagaonkar

June 24, 2013

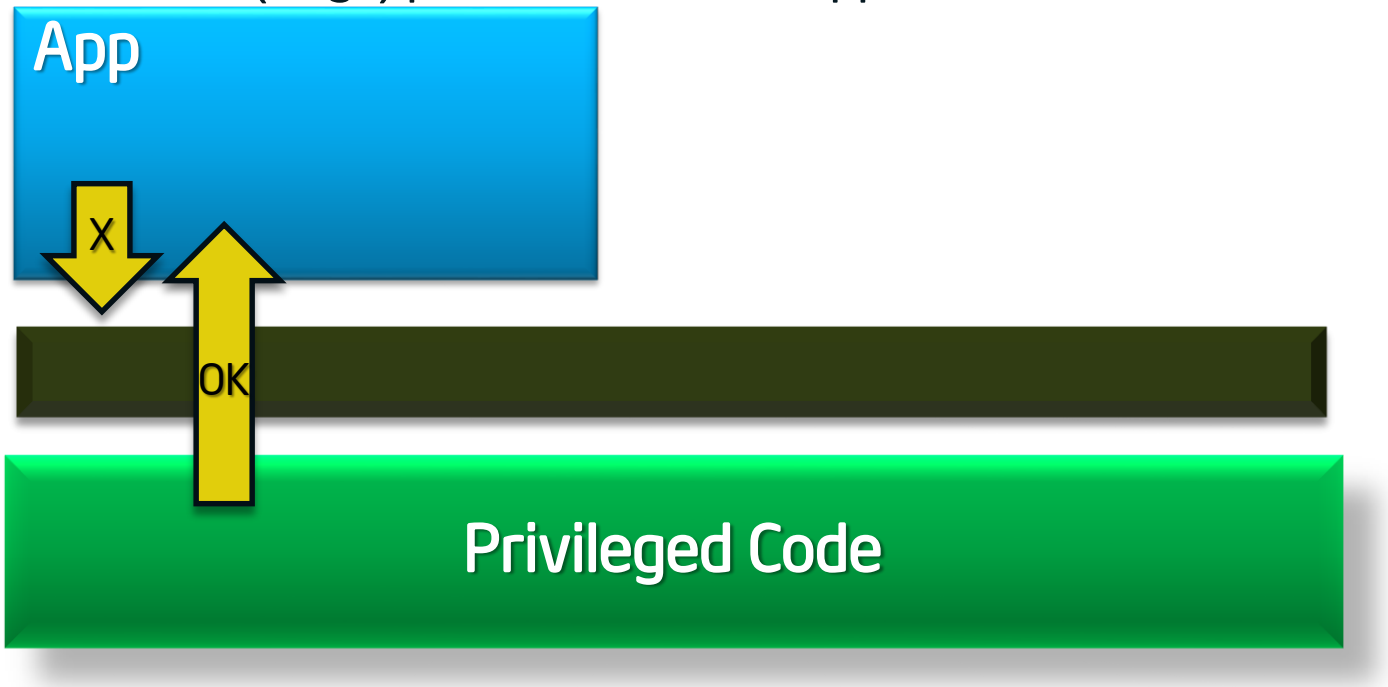
Legal Disclaimers

Outline

- Problem Statement
- Attack Surface and Overview
- Programming environment
  - System programming view
  - Day in the life of an enclave
- SGX protected memory paging
  - Evictions
  - Loads
- Off Chip protections
- Summary

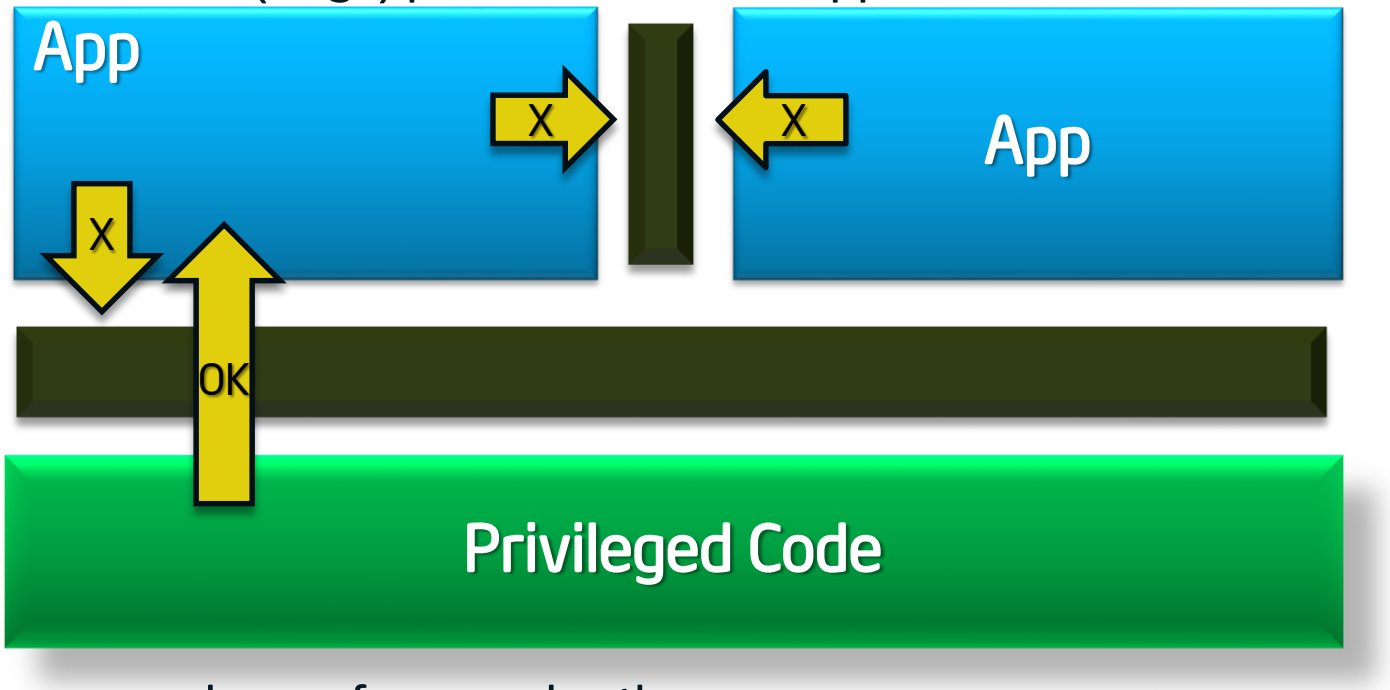
The Basic Issue: Why Aren't Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps …



The Basic Issue: Why Aren't Compute Devices Trustworthy?

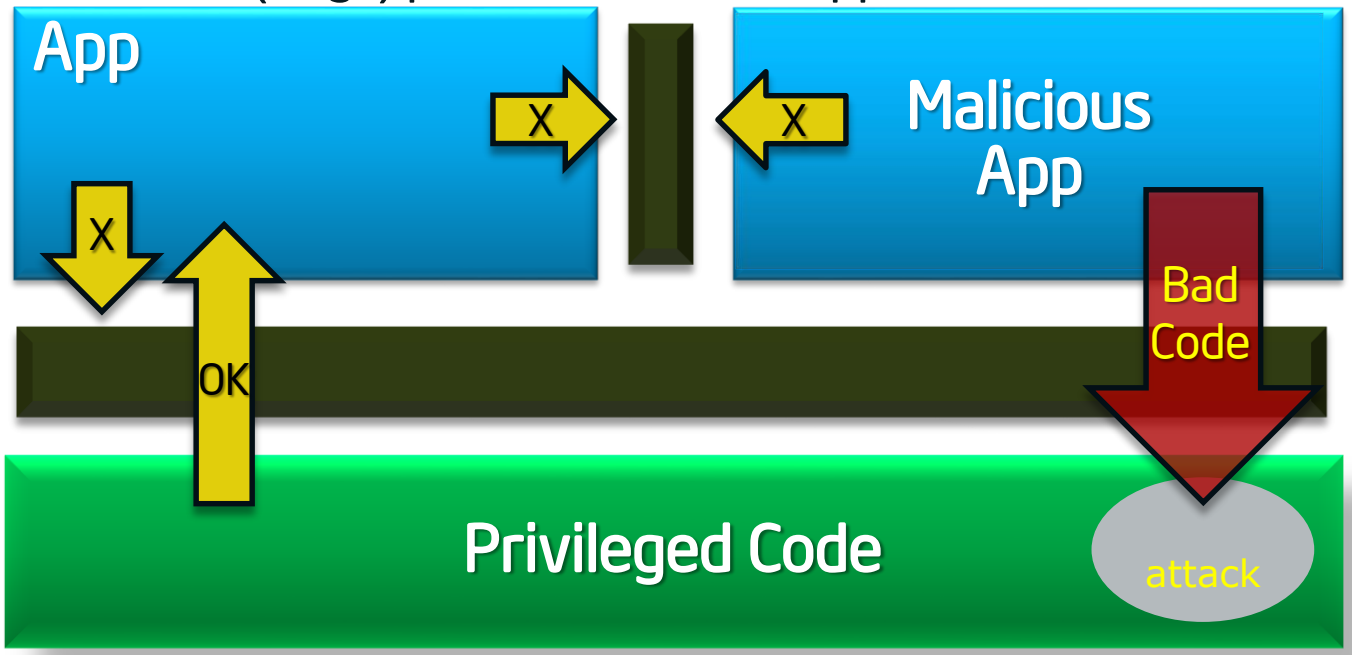
Protected Mode (rings) protects OS from apps …



… and apps from each other …

The Basic Issue: Why Aren't Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps …

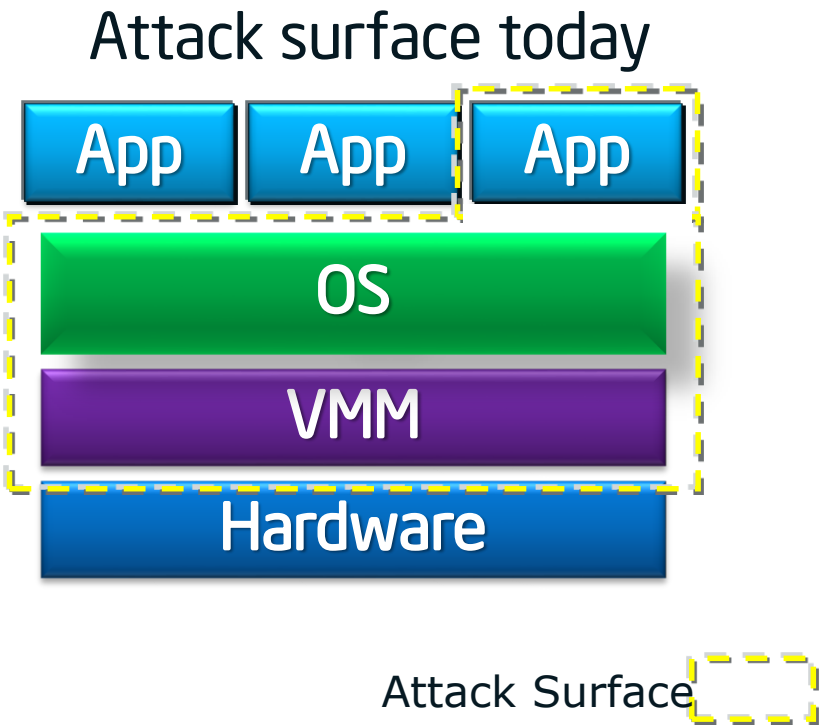


… and apps from each other …

… UNTIL a malicious app exploits a flaw to gain full privileges and then tampers with the OS or other apps

Apps not protected from privileged code attacks

Reduced attack surface with Intel® SGX

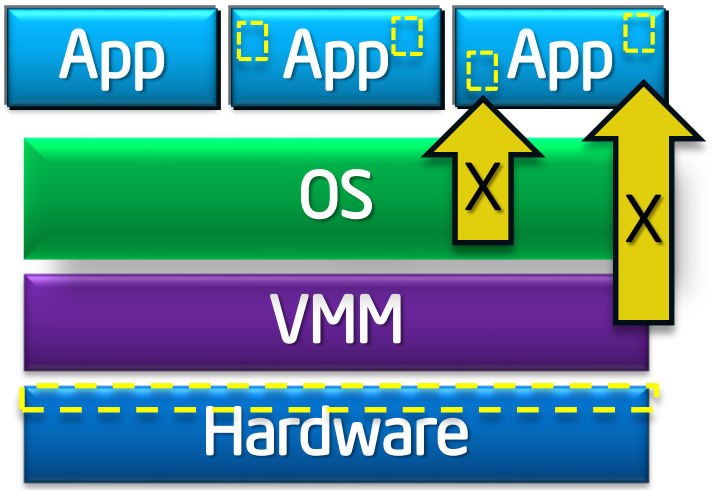


Reduced attack surface with Intel® SGX

Application gains ability to defend its own secrets

- Smallest attack surface (App + processor)
- Malware that subverts OS/VMM, BIOS, Drivers etc. cannot steal app secrets

Attack surface with Intel® SGX



Attack Surface

Reduced attack surface with Intel® SGX

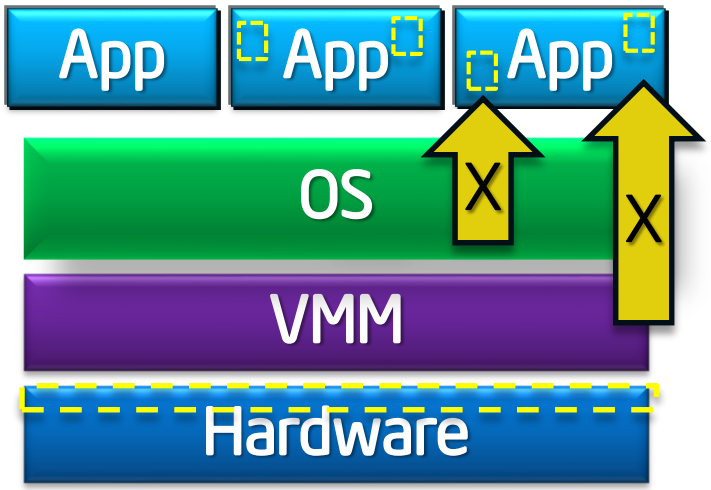
Application gains ability to defend its own secrets

- Smallest attack surface (App + processor)
- Malware that subverts OS/VMM, BIOS, Drivers etc. cannot steal app secrets

Familiar development/debug

- Single application environment
- Build on existing ecosystem expertise

Attack surface with Intel® SGX



Attack Surface

Reduced attack surface with Intel® SGX

Application gains ability to defend its own secrets

- Smallest attack surface (App + processor)
- Malware that subverts OS/VMM, BIOS, Drivers etc. cannot steal app secrets

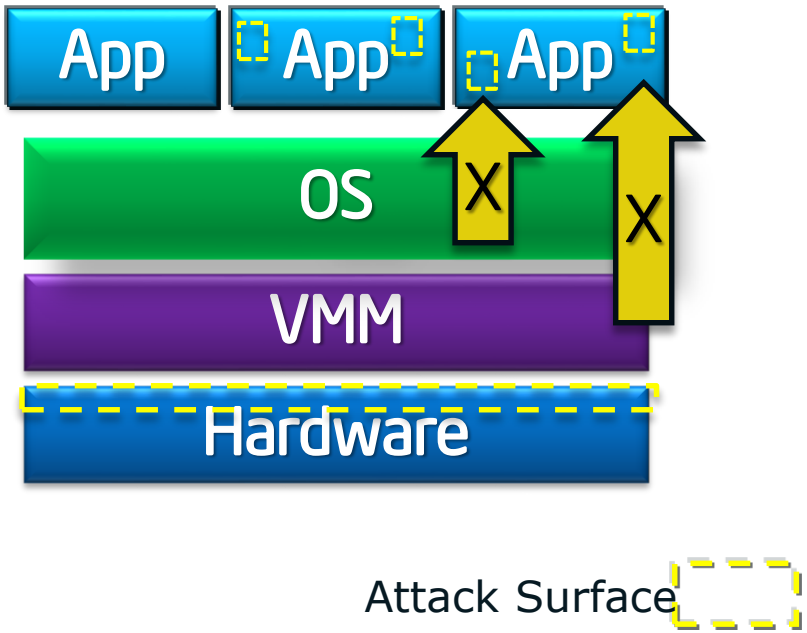
Familiar development/debug

- Single application environment
- Build on existing ecosystem expertise

Familiar deployment model

- Trusted applications can be distributed and updated by app developers as needed

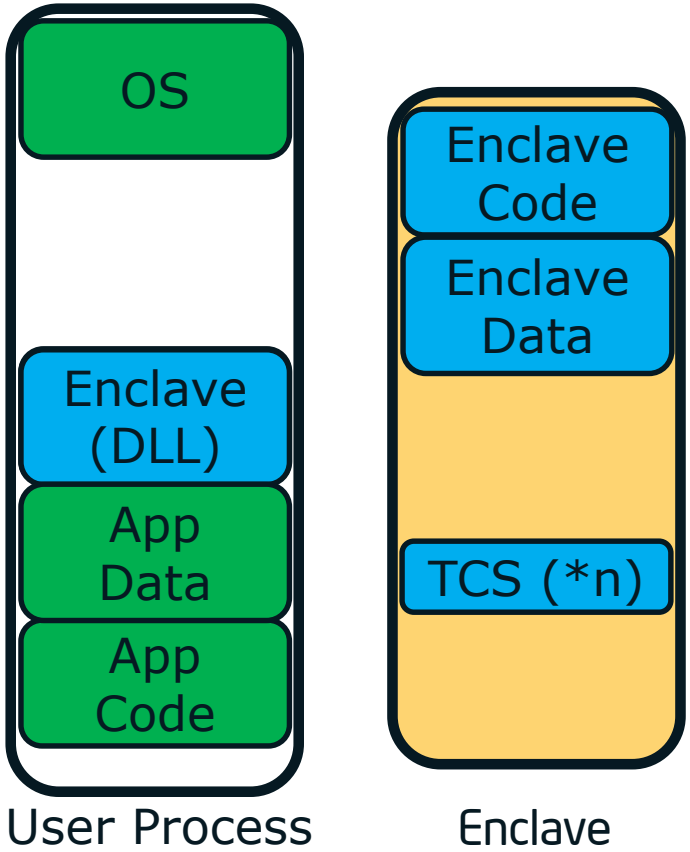
Attack surface with Intel® SGX



Scalable security within mainstream environment

SGX Programming Environment

Protected execution environment embedded in a process



With its own code and data

Provide Confidentiality

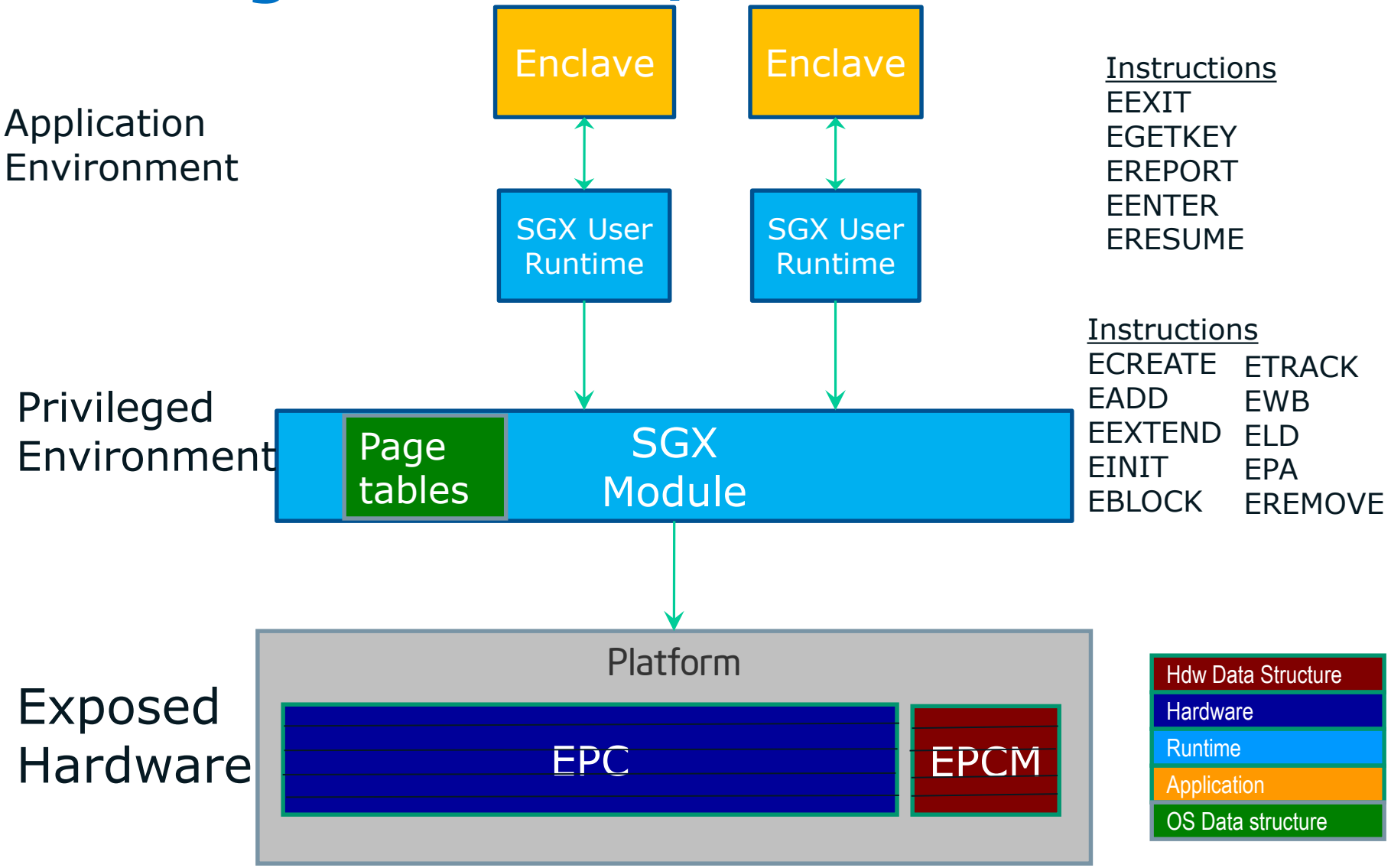
Provide integrity

With controlled entry points

Supporting multiple threads

With full access to app memory

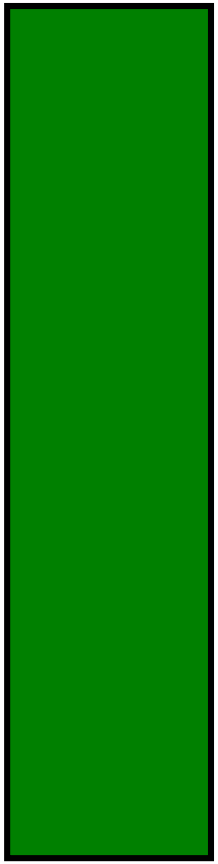
SGX High-level HW/SW Picture



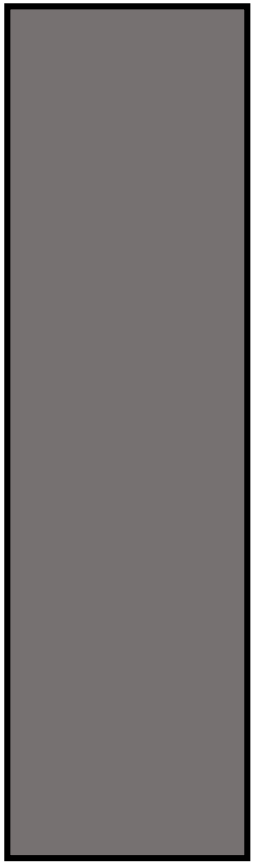
Life Cycle of An Enclave

Build

Virtual Addr Space

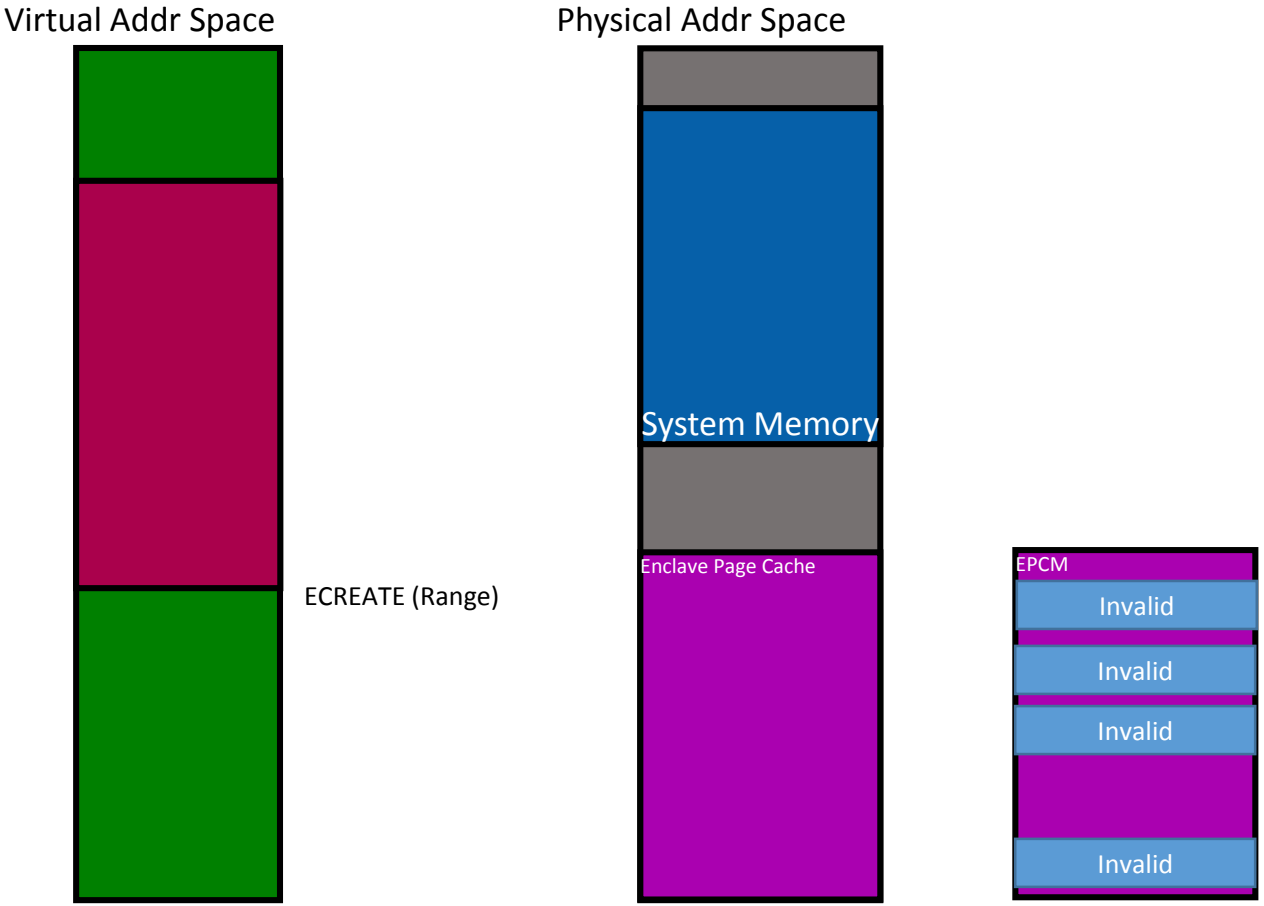


Physical Addr Space

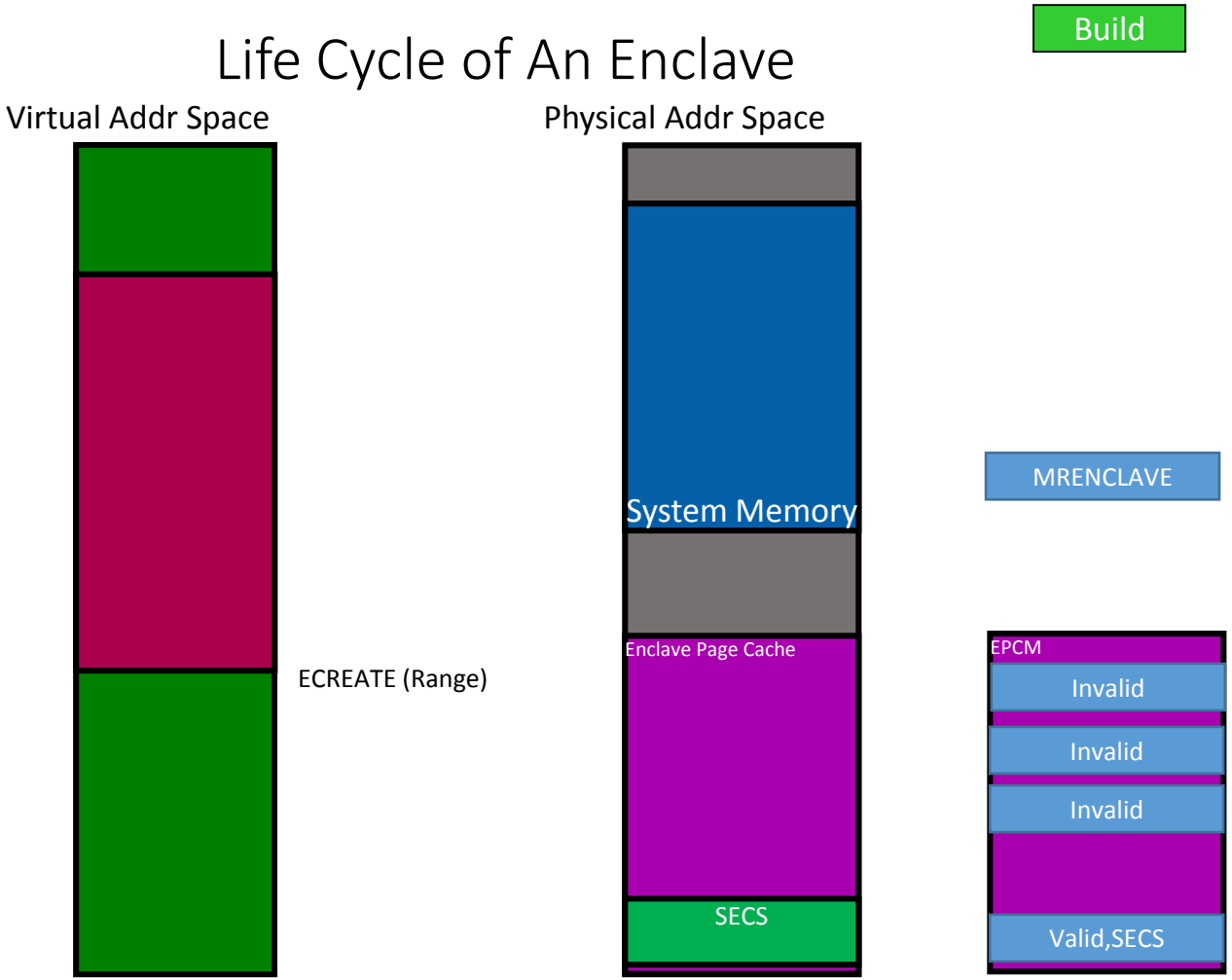


Life Cycle of An Enclave

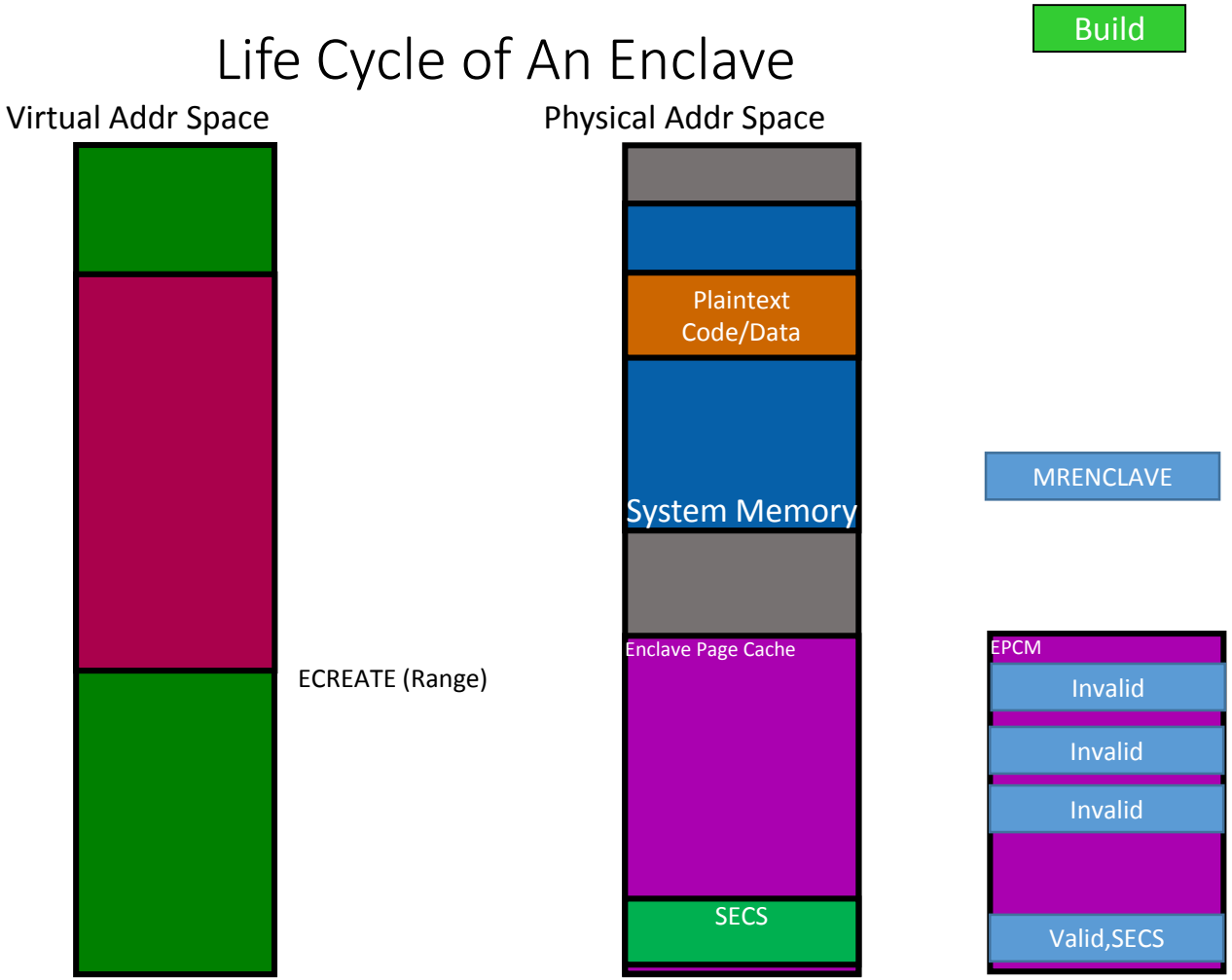
Build



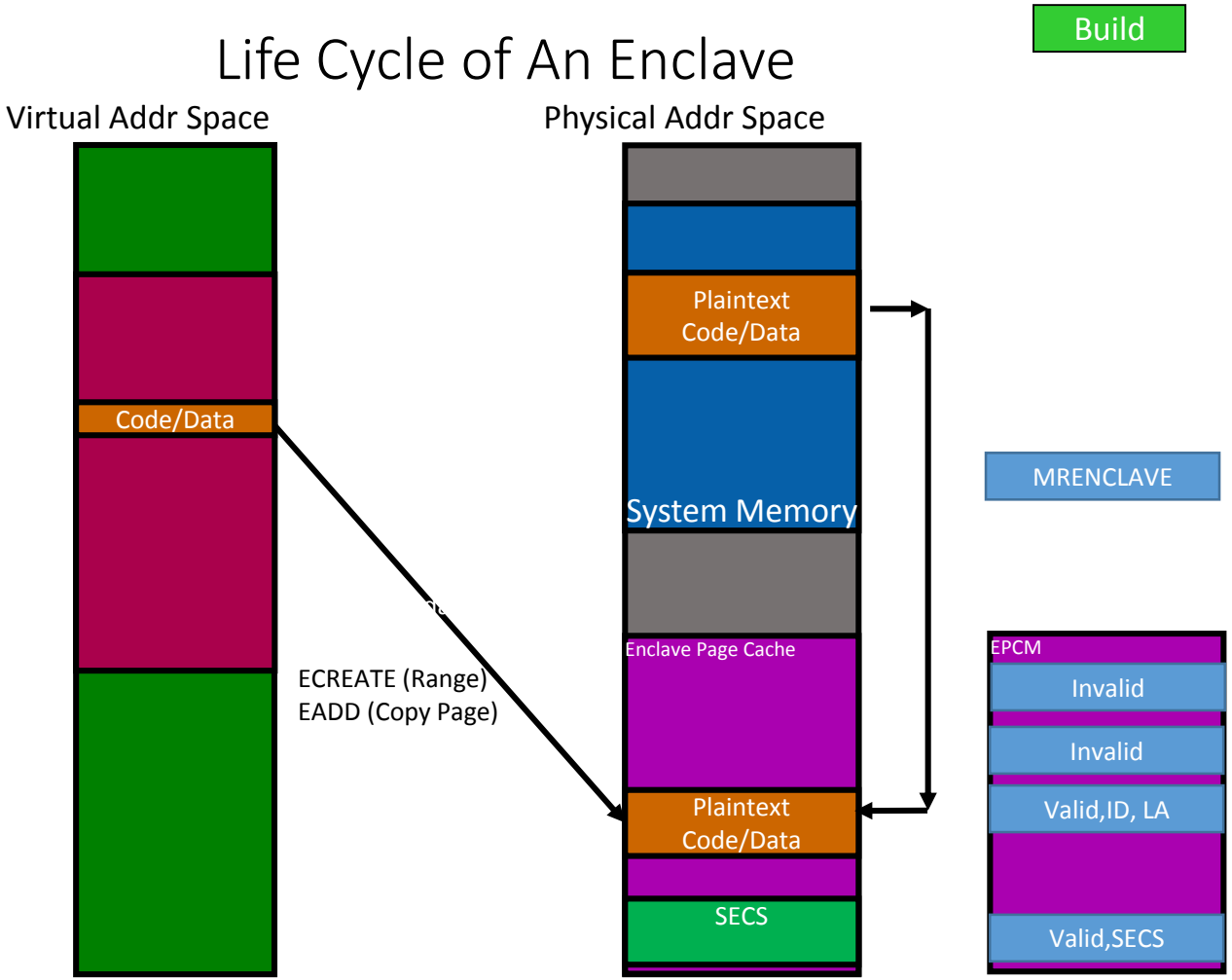
Life Cycle of An Enclave



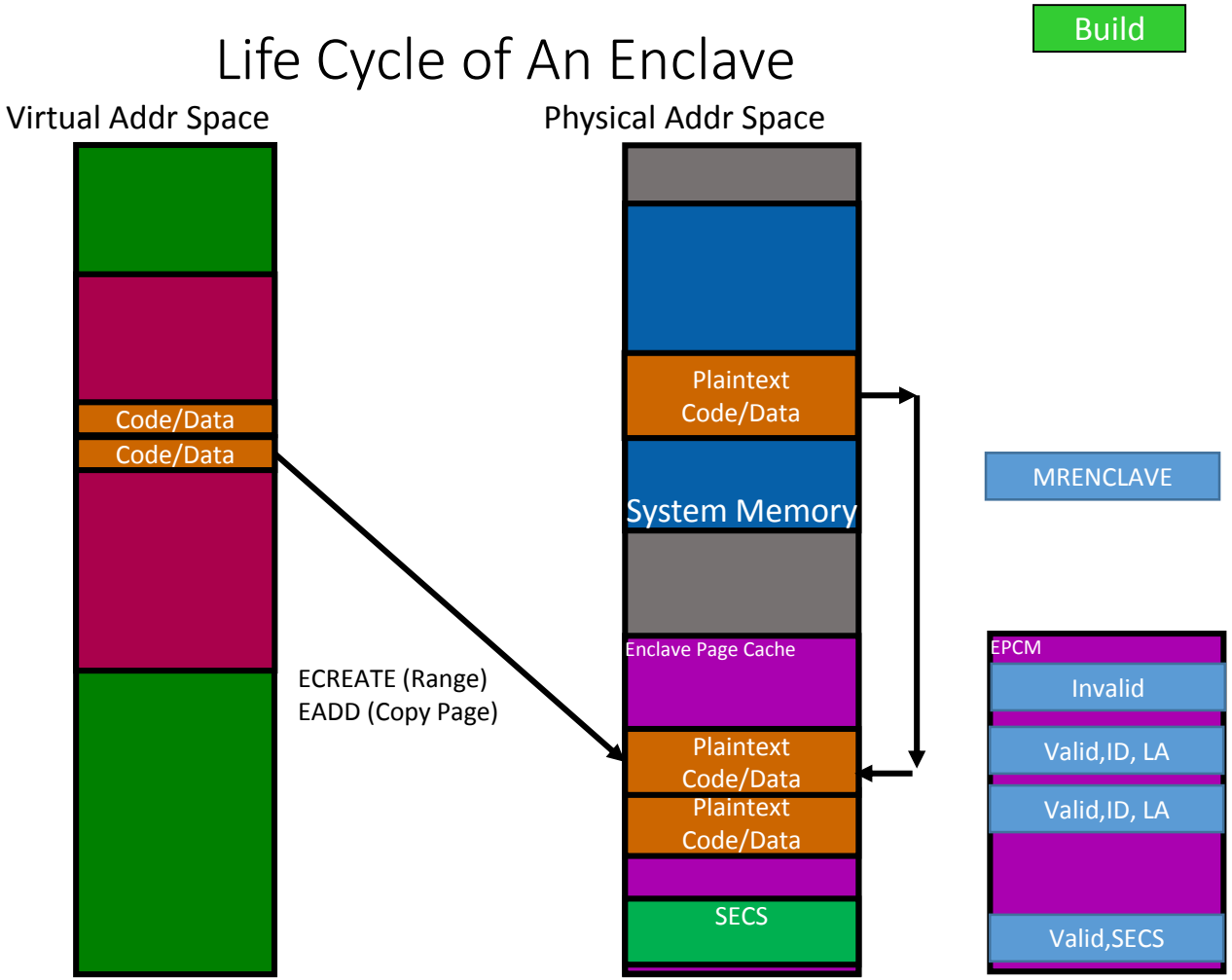
Life Cycle of An Enclave



Life Cycle of An Enclave

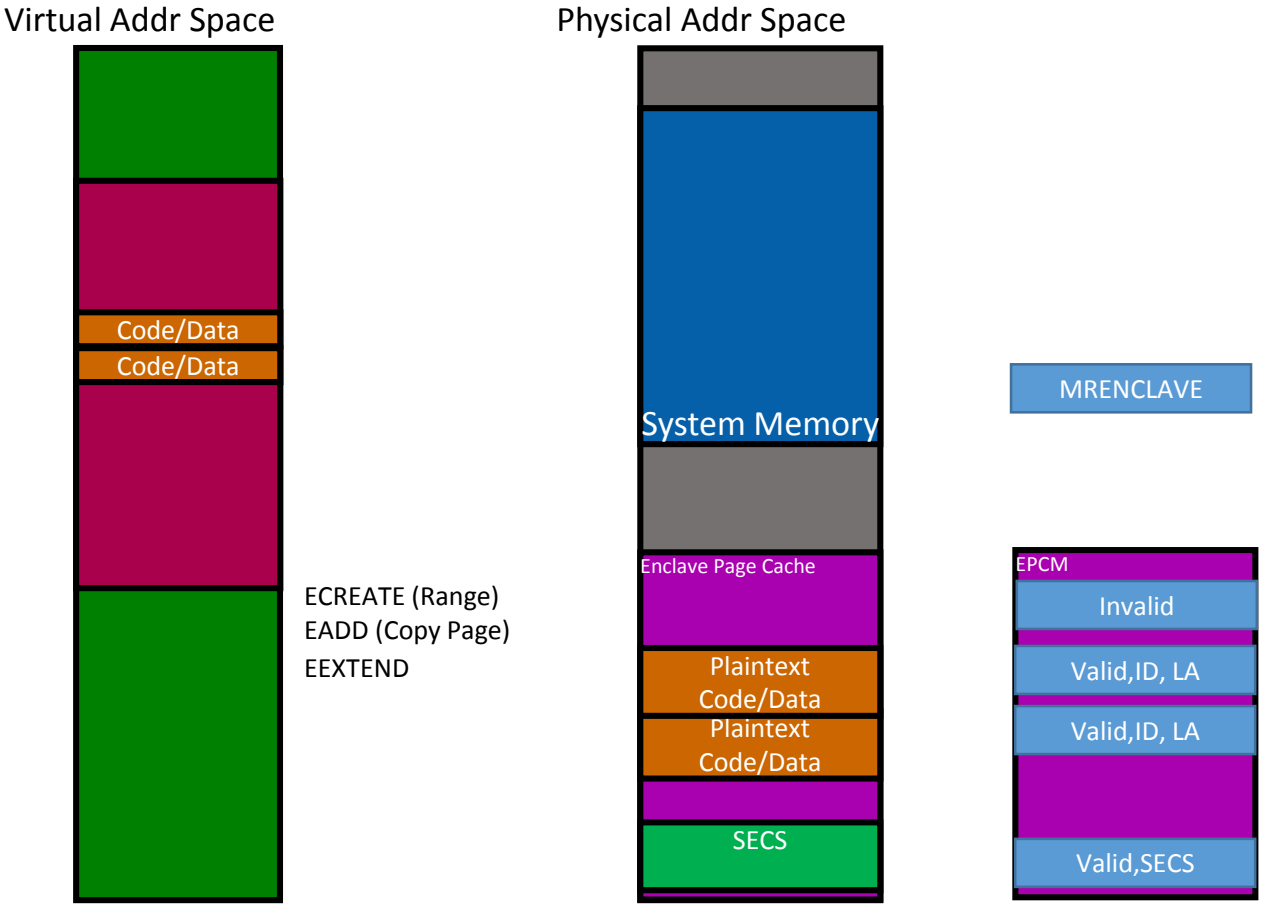


Life Cycle of An Enclave



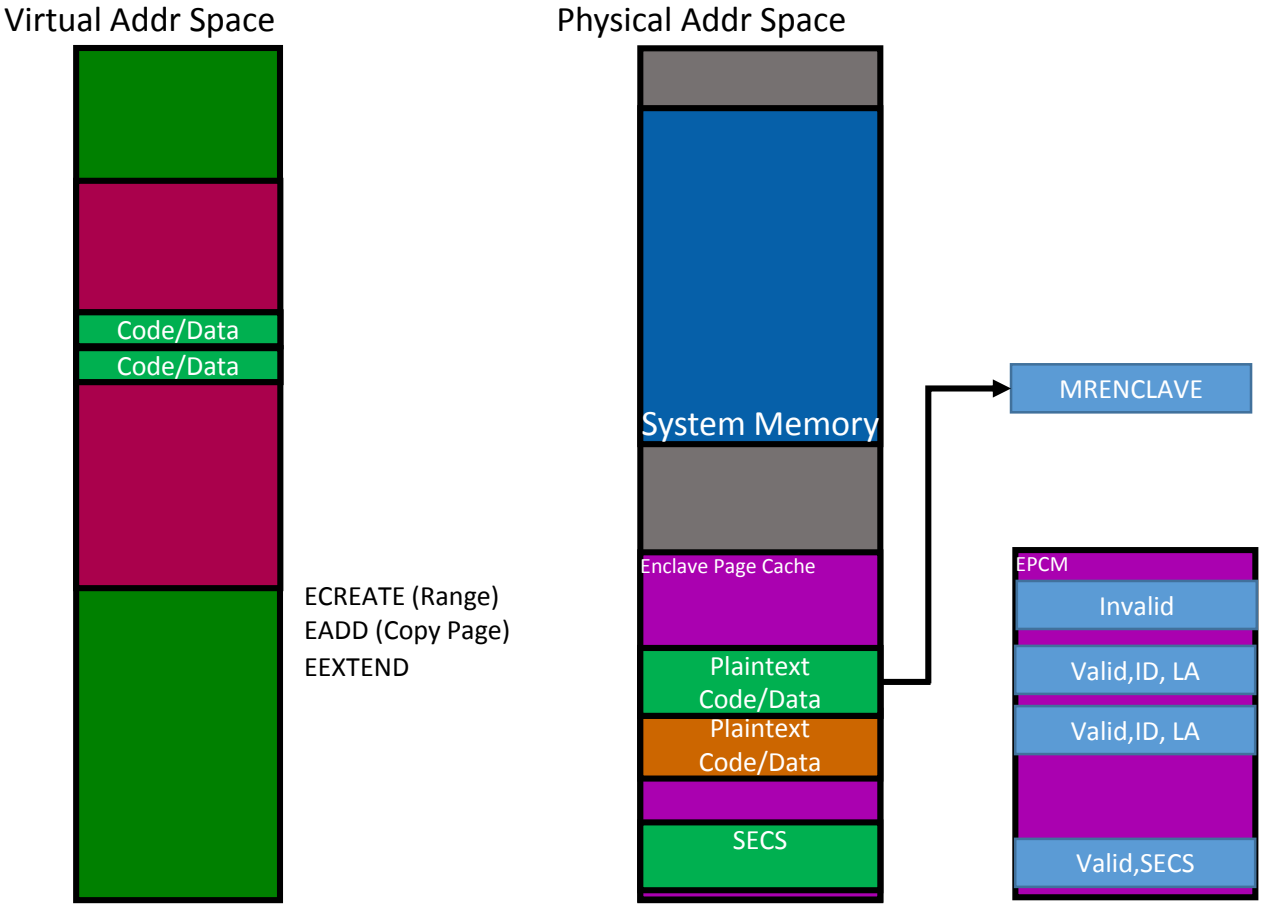
Life Cycle of An Enclave

Build

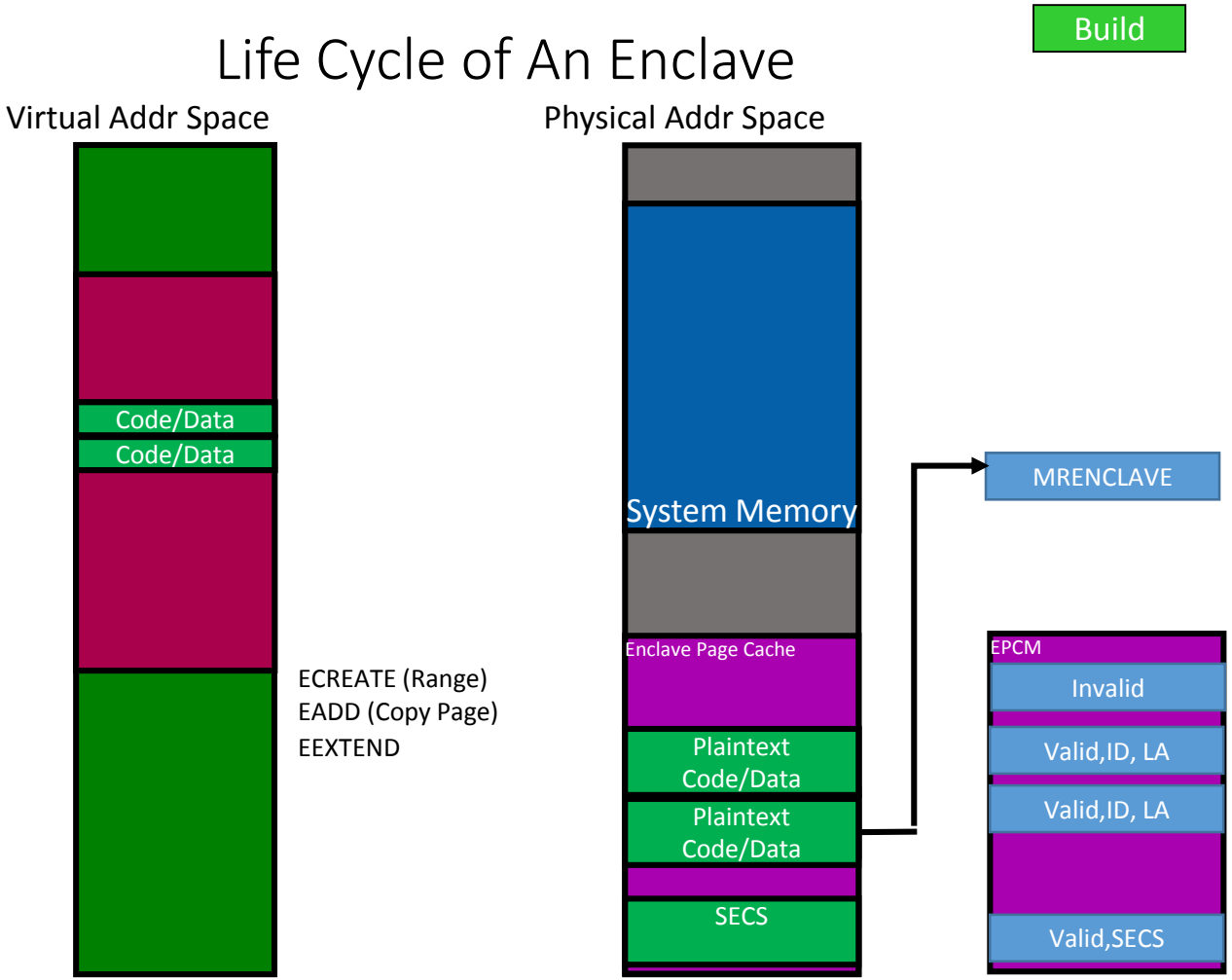


Life Cycle of An Enclave

Build

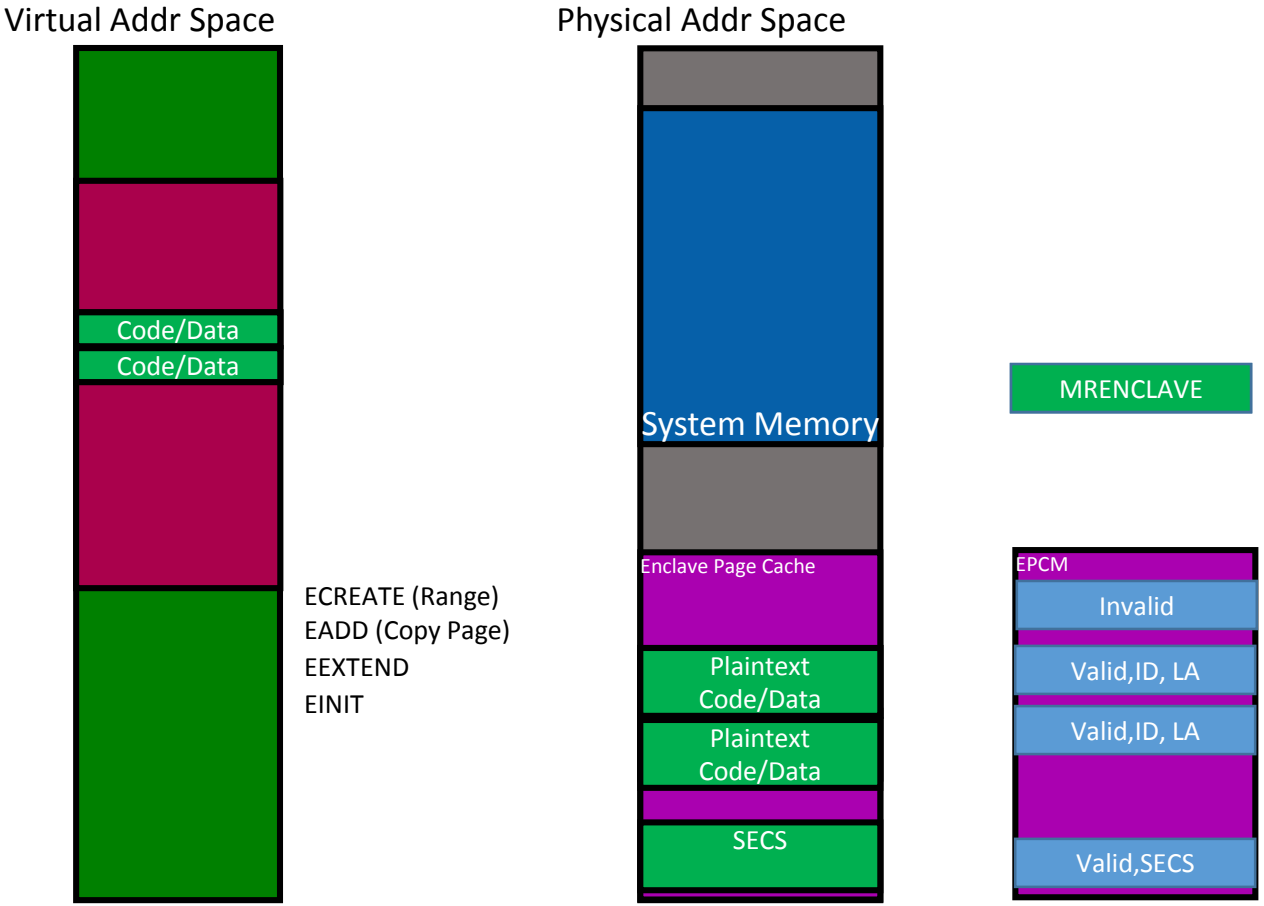


Life Cycle of An Enclave



Life Cycle of An Enclave

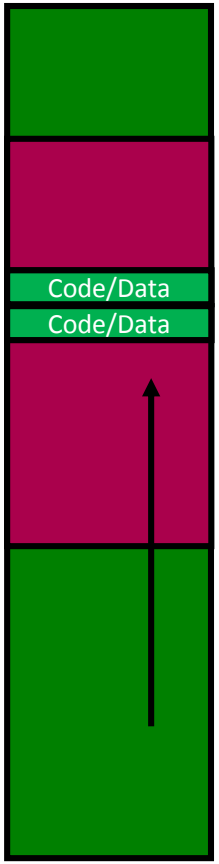
Build



Life Cycle of An Enclave

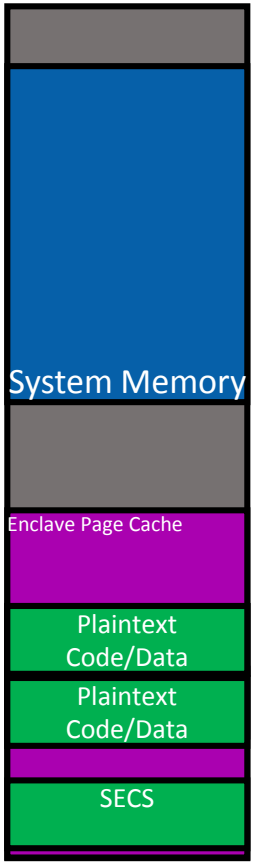
Build

Virtual Addr Space

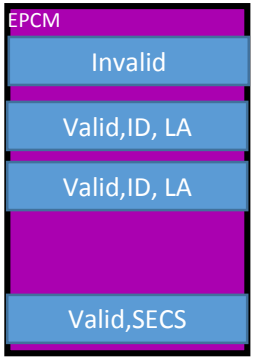


ECREATE (Range)
EADD (Copy Page)
EEXTEND
EINIT
EENTER

Physical Addr Space



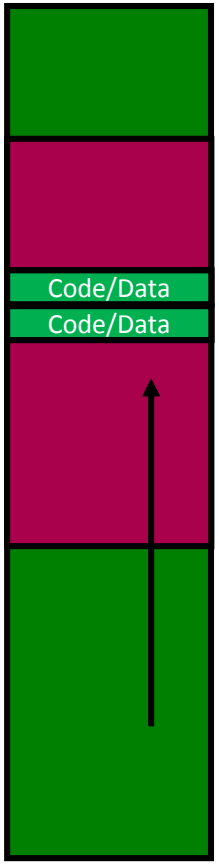
MRENCLAVE



Life Cycle of An Enclave

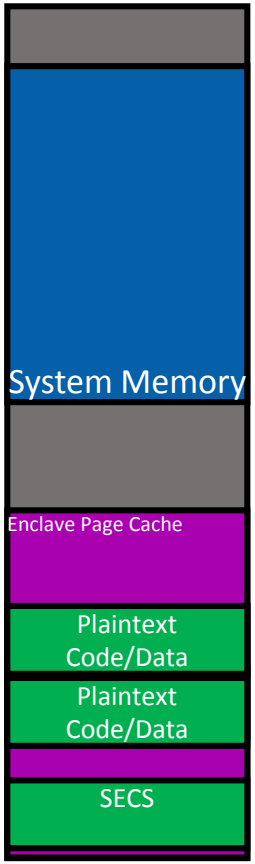
Build

Virtual Addr Space

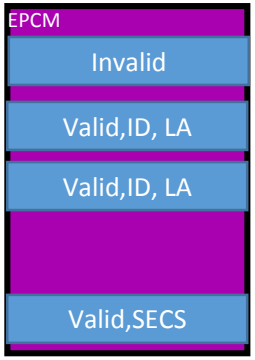


ECREATE (Range)
EADD (Copy Page)
EEXTEND
EINIT
EENTER

Physical Addr Space



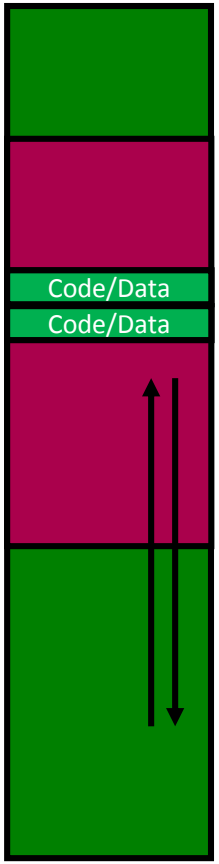
MRENCLAVE



Life Cycle of An Enclave

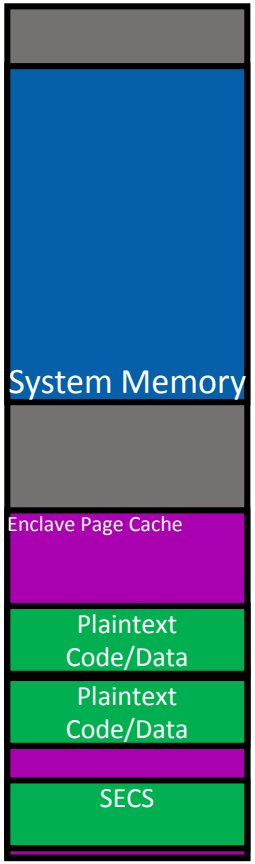
Build

Virtual Addr Space

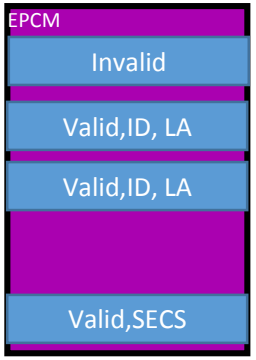


ECREATE (Range)
EADD (Copy Page)
EEXTEND
EINIT
EENTER
EEXIT

Physical Addr Space



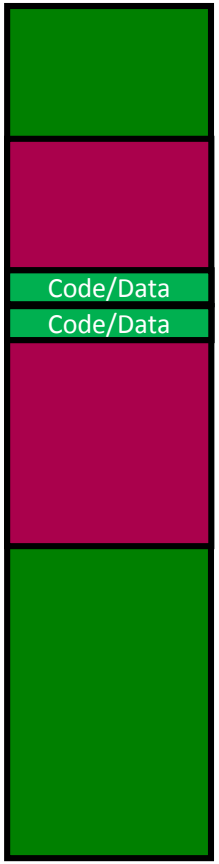
MRENCLAVE



Life Cycle of An Enclave

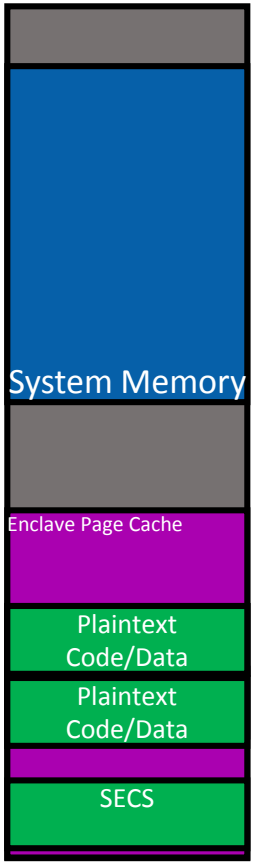
Build

Virtual Addr Space

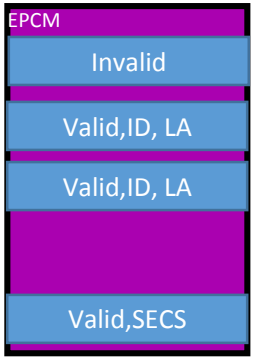


ECREATE (Range)
EADD (Copy Page)
EEXTEND
EINIT
EENTER
EEXIT

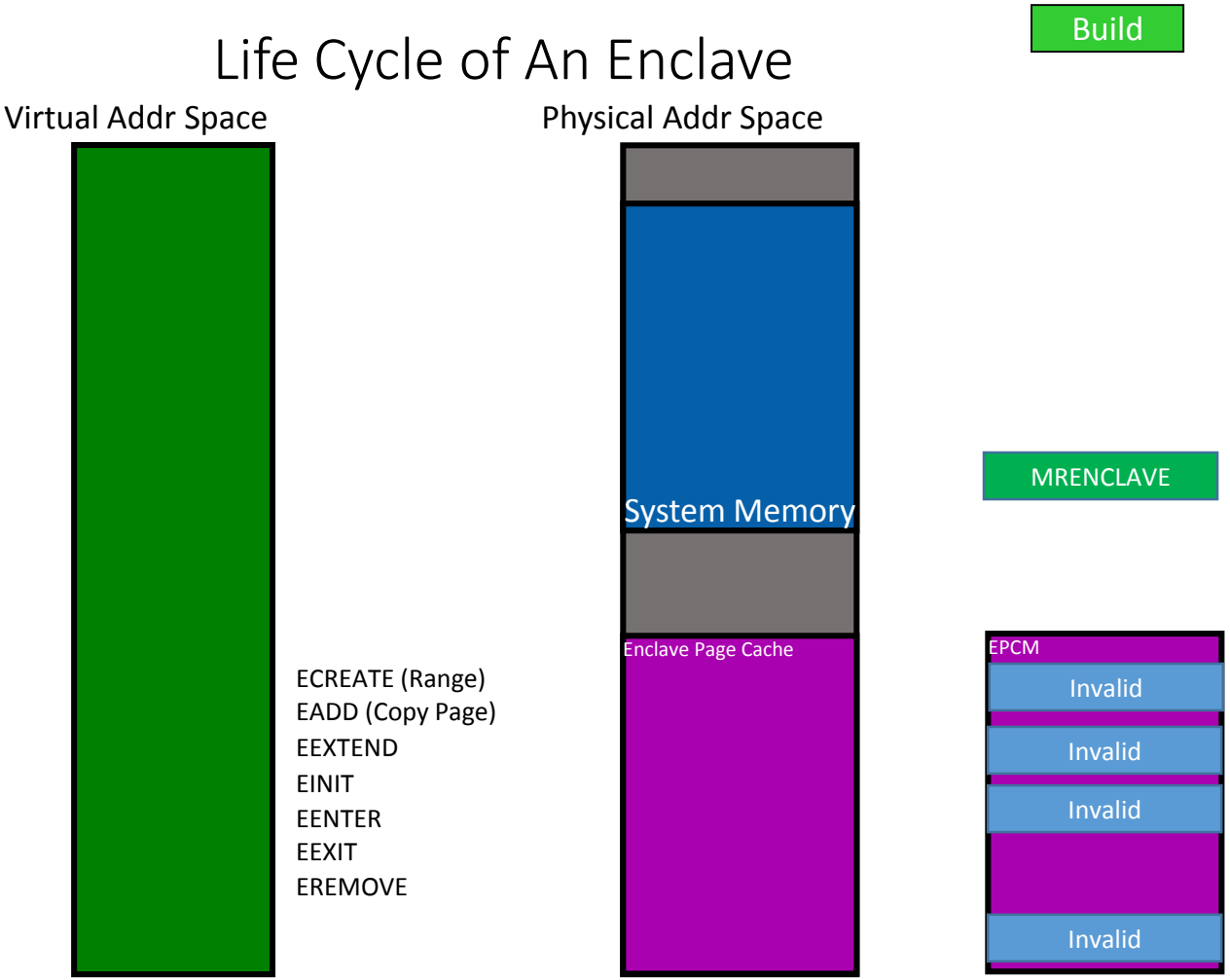
Physical Addr Space



MRENCLAVE



Life Cycle of An Enclave



SGX Paging Introduction

Requirement:

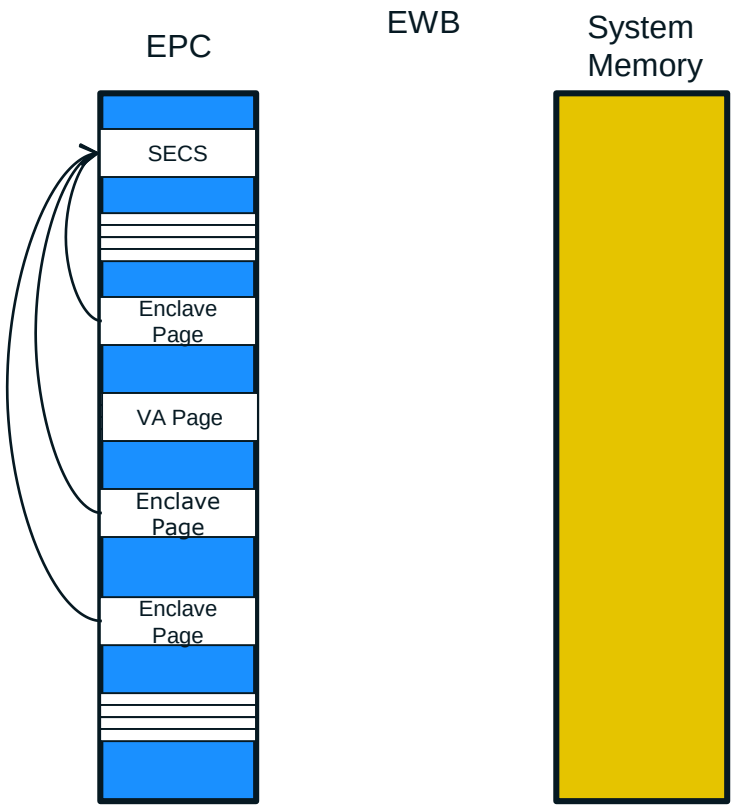
- Remove an EPC page and place into unprotected memory. Later restore it.
- Page must maintain same security properties (confidentiality, anti-replay, and integrity) when restored

New Instructions:

- EWB: Evict EPC page to main memory with cryptographic protections
- ELDB/ELDU: Load page from main memory to EPC with cryptographic protections
- EPA: Allocate an EPC page for holding versions
- EBLOCK: Declare an EPC page ready for eviction
- ETRACK: Ensure address translations have been cleared

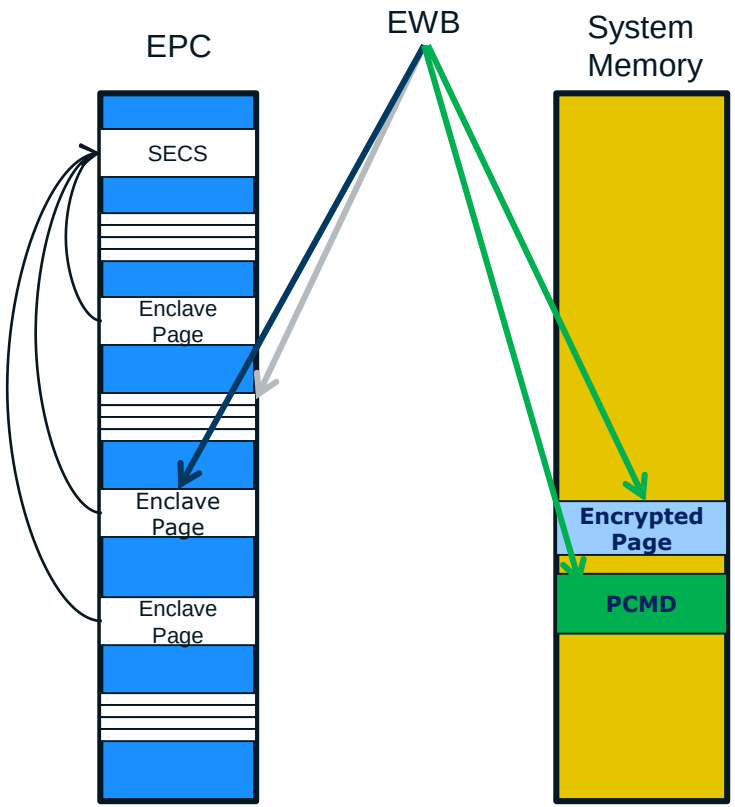
Page-out Example

BUILD



Page-out Example

BUILD

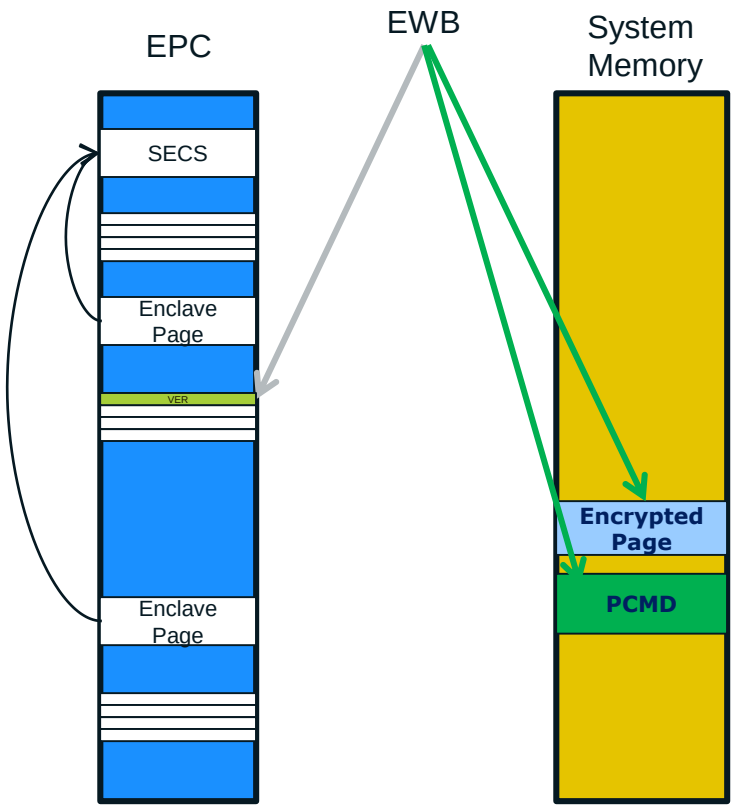


EWB Parameters:

- Pointer to EPC page that needs to be paged out
- Pointer to empty version slot
- Pointers outside EPC location

Page-out Example

BUILD



EWB Parameters:

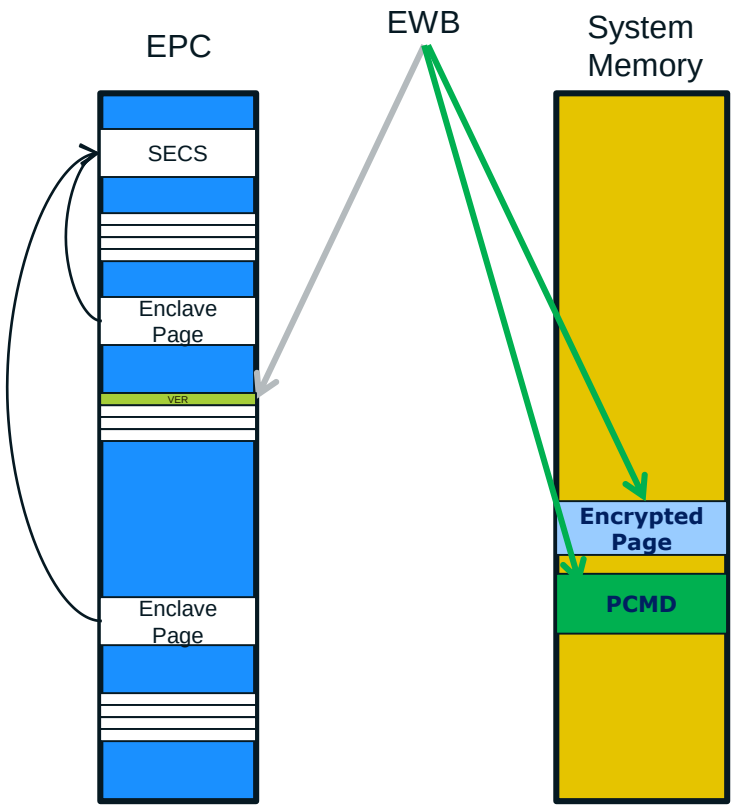
- Pointer to EPC page that needs to be paged out
- Pointer to empty version slot
- Pointers outside EPC location

EWB Operation

- Remove page from the EPC
- Populate version slot
- Write encrypted version to outside
- Write meta-data, PCMD

Page-out Example

BUILD



EWB Parameters:

- Pointer to EPC page that needs to be paged out
- Pointer to empty version slot
- Pointers outside EPC location

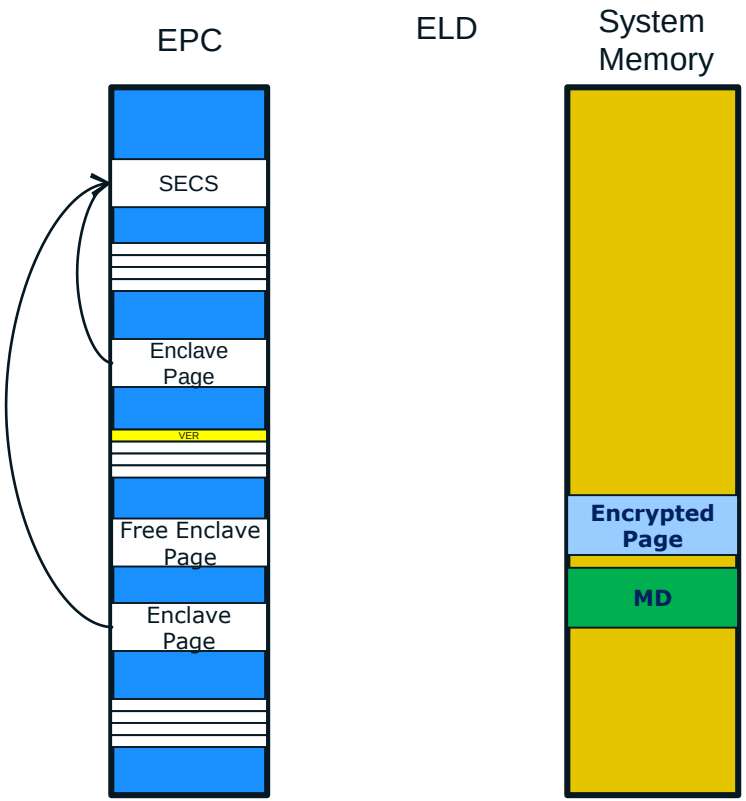
EWB Operation

- Remove page from the EPC
- Populate version slot
- Write encrypted version to outside
- Write meta-data, PCMD

All pages, including SECS and Version Array can be paged out

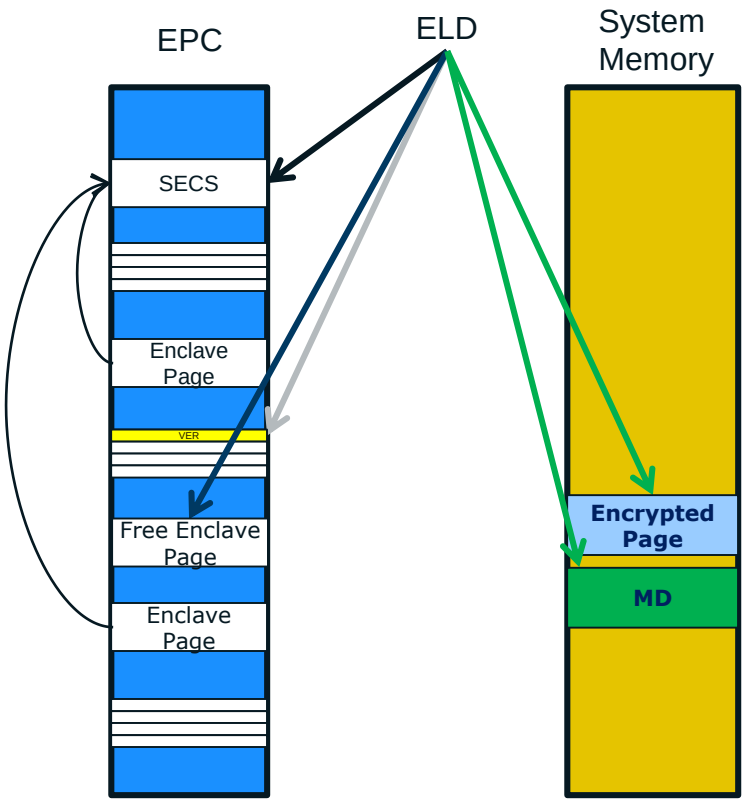
Page-in Example

BUILD



Page-in Example

BUILD

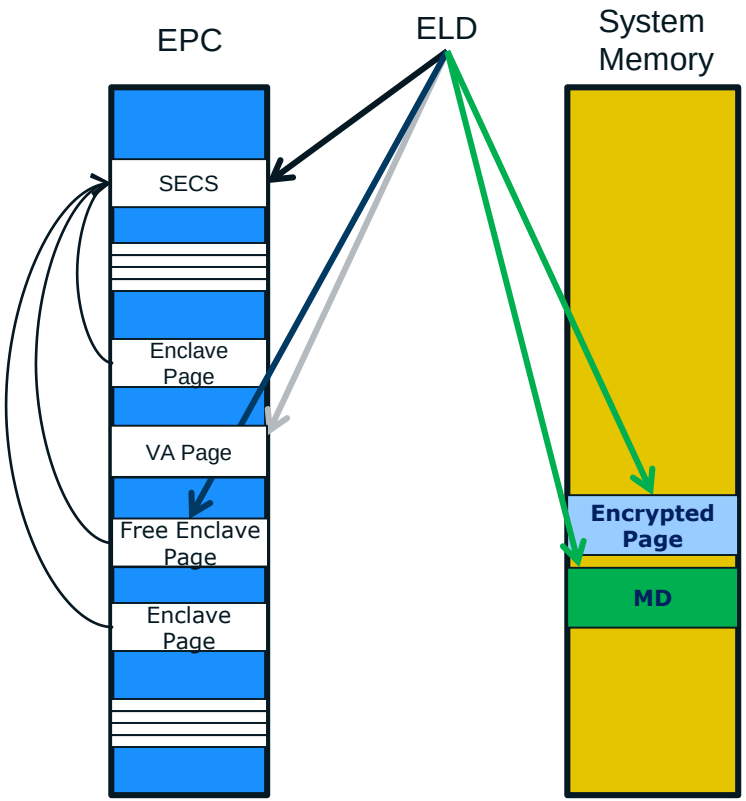


ELD Parameters:

- Encrypted page
- Free EPC page
- SECS (for an enclave page)
- Populated version slot

Page-in Example

BUILD



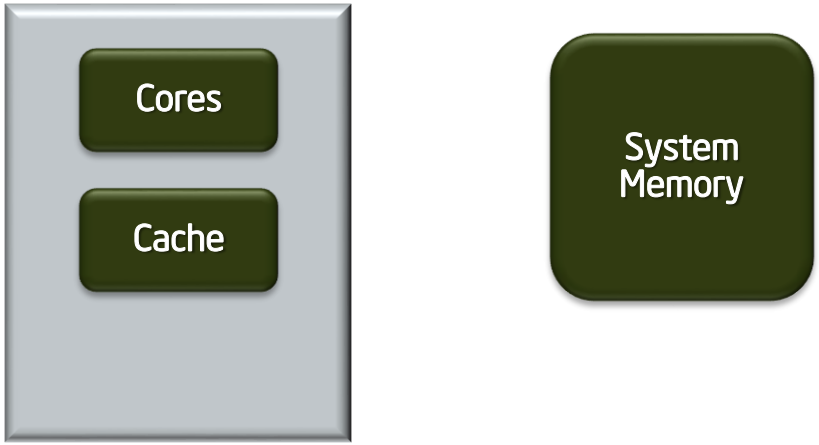
ELD Parameters:

- Encrypted page
- Free EPC page
- SECS (for an enclave page)
- Populated version slot

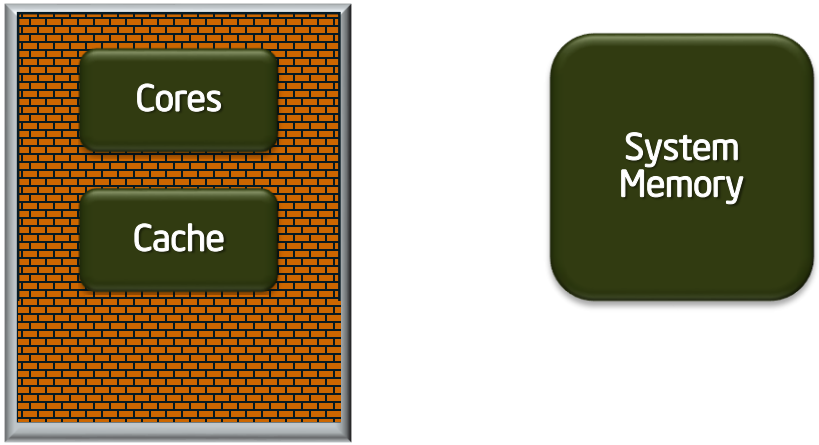
ELD Operation

- Verify and decrypt the page using version
- Populate the EPC slot
- Make back-pointer connection (if applicable)
- Free-up version slot

Protection vs. Memory Snooping Attacks

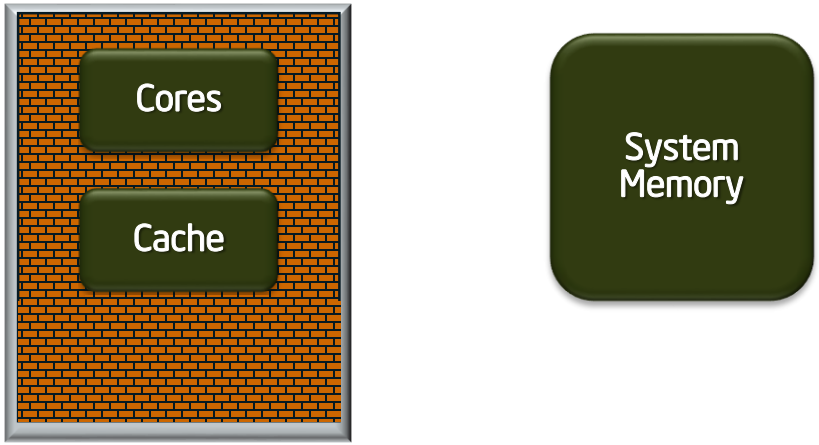


Protection vs. Memory Snooping Attacks



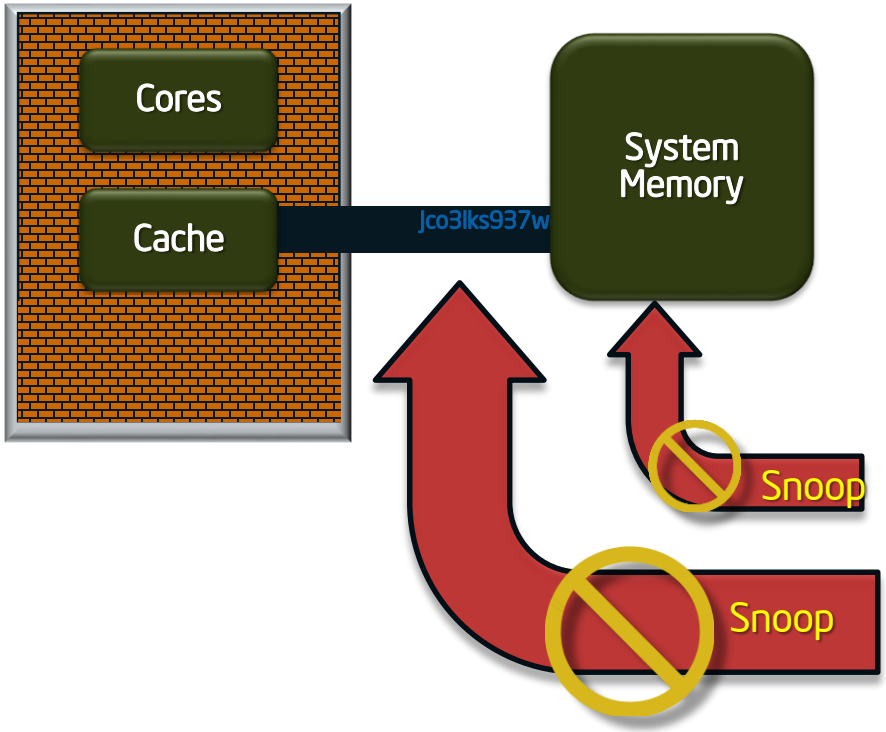
1. Security perimeter is the CPU package boundary

Protection vs. Memory Snooping Attacks



1. Security perimeter is the CPU package boundary
2. Data and code unencrypted inside CPU package

Protection vs. Memory Snooping Attacks



1. Security perimeter is the CPU package boundary
2. Data and code unencrypted inside CPU package
3. Data and code outside CPU package is encrypted and/or integrity checked
4. External memory reads and bus snoops see only encrypted data

SGX Technical Summary

- Provides any application the ability to keep a secret
  - Provide capability using new processor instructions
  - Application can support multiple enclaves
- Provides integrity and confidentiality
  - Resists hardware attacks
  - Prevent software access, including privileged software and SMM
- Applications run within OS environment
  - Low learning curve for application developers
  - Open to all developers
- Resources managed by SW
  - HW components are supported in a driver or OS

Thank You

intel®