



Introduction to Intel® Software Guard Extensions(Intel® SGX)

Cyberday 2015, Technion, Israel

Ittai.anati@intel.com

July 2015

Legal Disclaimers

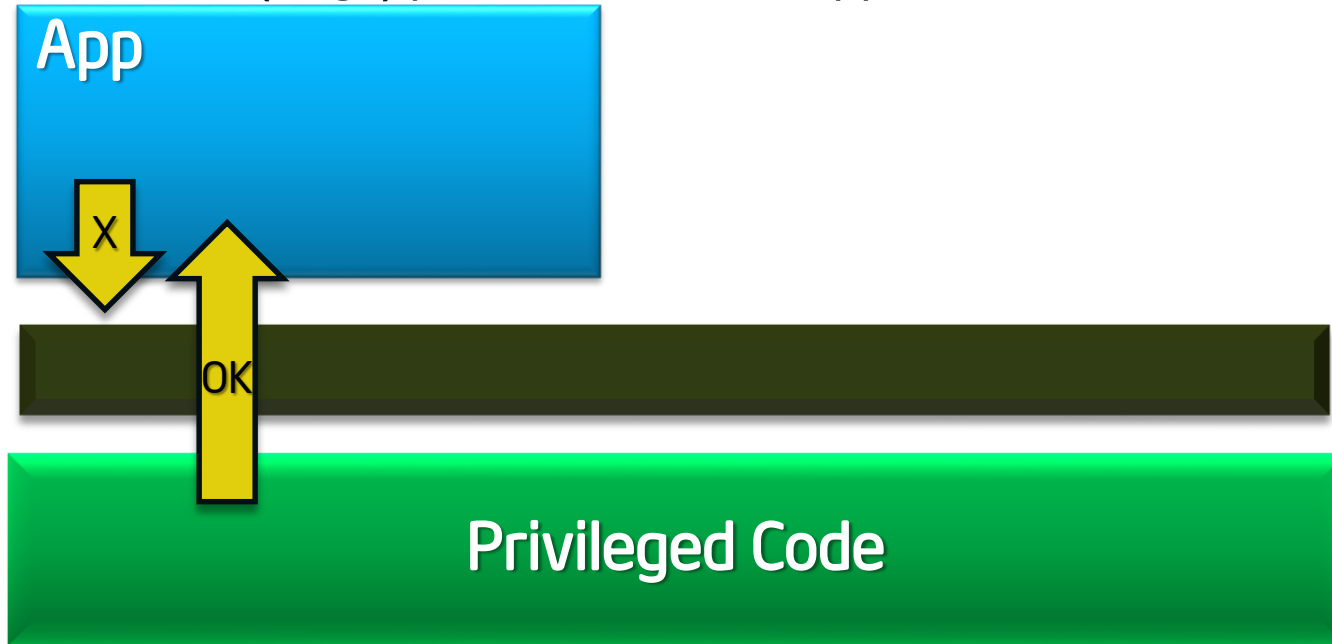
- The comments and statements are the presenters and not necessarily Intel's
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [intel.com](https://www.intel.com), or from the OEM or retailer.
- No computer system can be absolutely secure.

Slides are taken from the SGX tutorial slide deck from ISCA 2015, published at:

<https://software.intel.com/en-us/isa-extensions/intel-sgx>

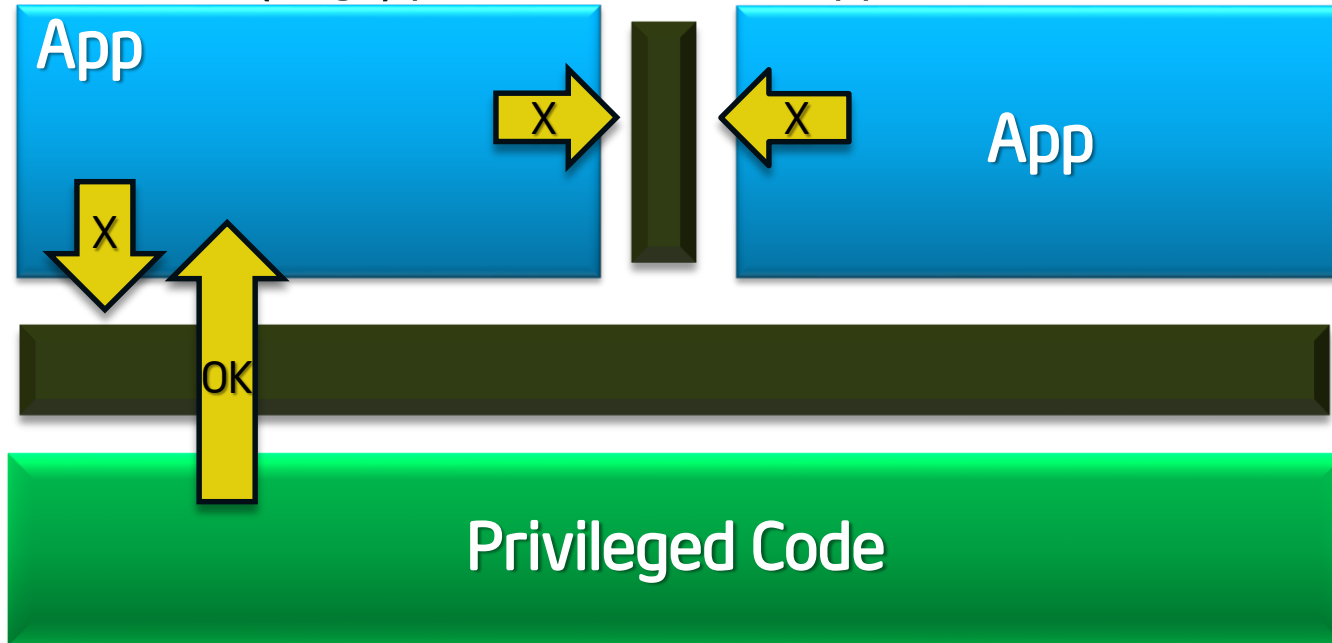
The Basic Issue: Why Aren't Compute Devices Trustworthy?

Protected Mode (rings) protects OS from apps ...



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

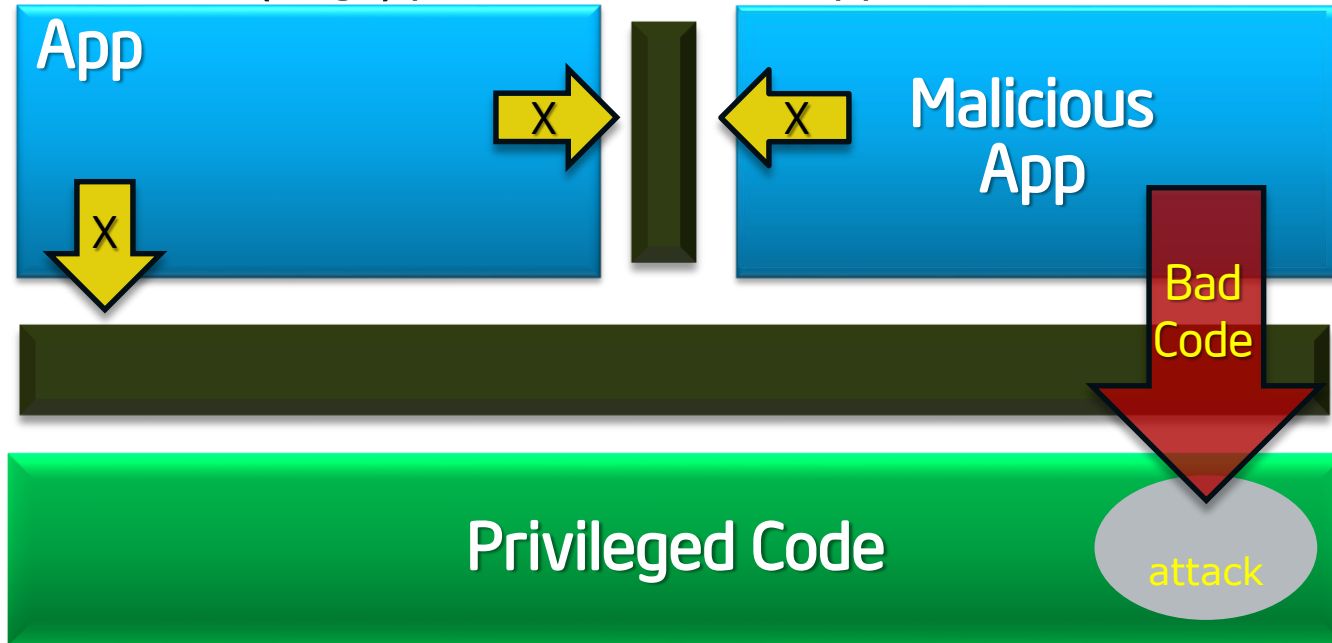
Protected Mode (rings) protects OS from apps ...



... and apps from each other ...

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

Protected Mode (rings) protects OS from apps ...



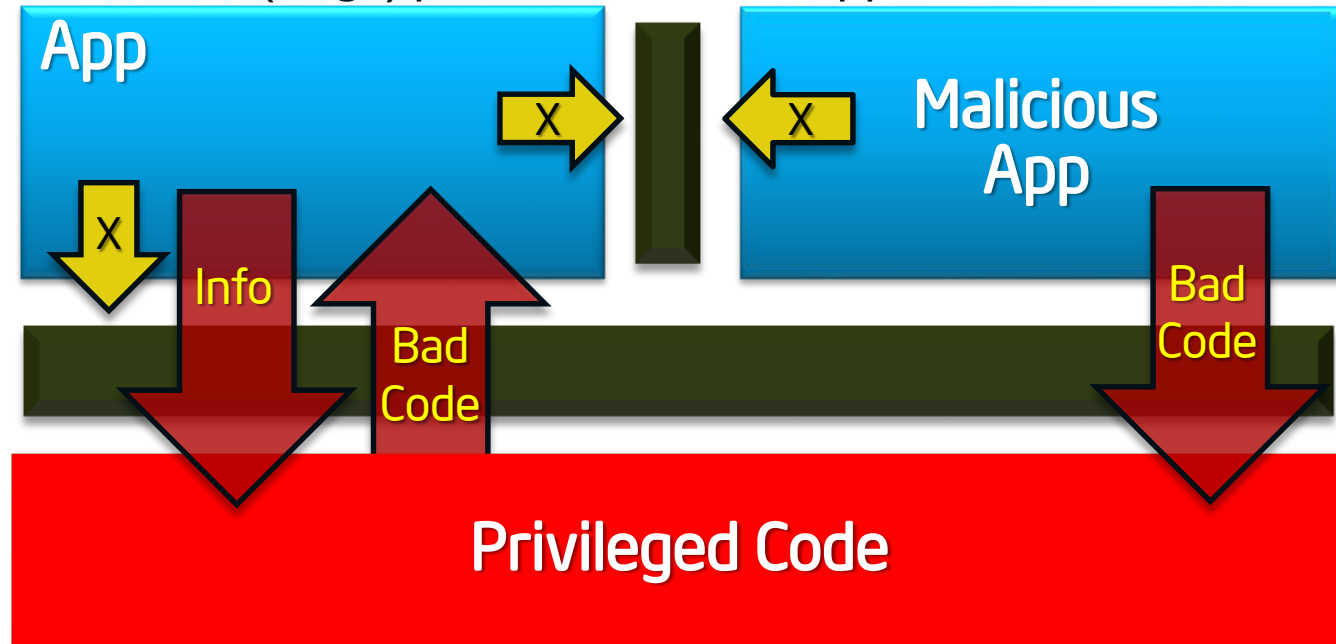
... and apps from each other ...

... 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

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

Protected Mode (rings) protects OS from apps ...



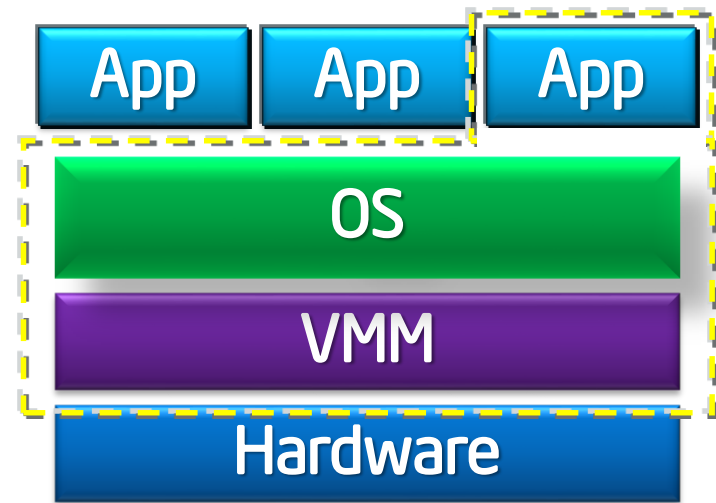
... and apps from each other ...


... 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 SGX

Attack surface today



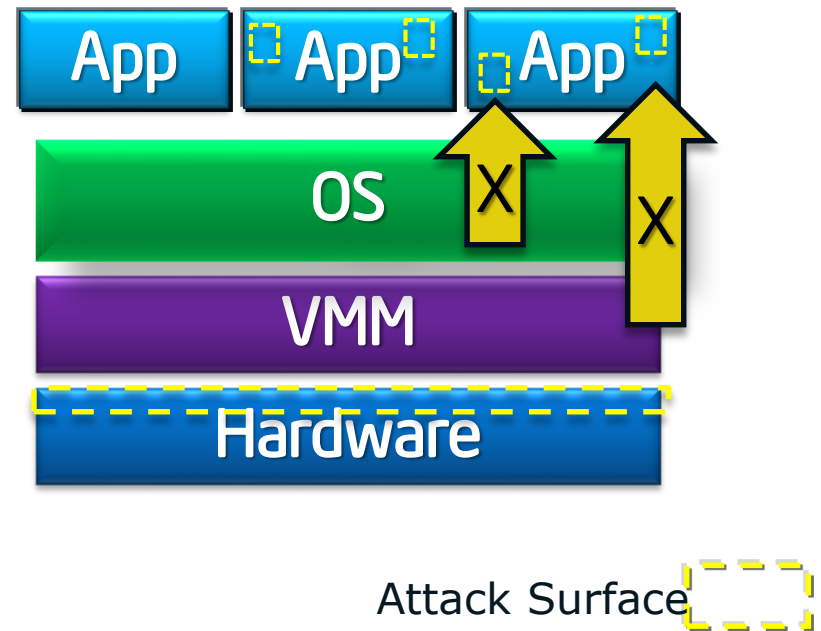
Attack Surface 

Reduced attack surface with SGX

Application gains ability to defend its own secrets

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

Attack surface with Enclaves



Reduced attack surface with SGX

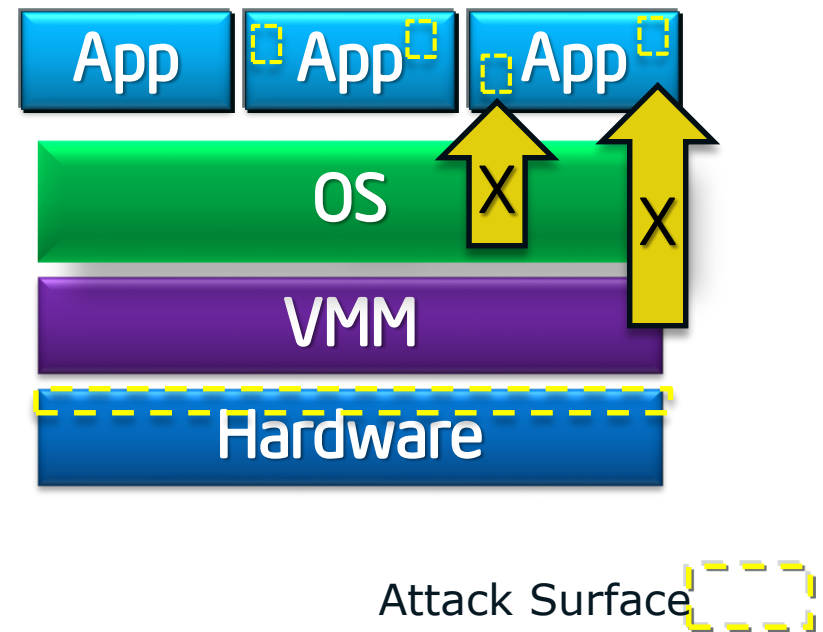
Application gains ability to defend its own secrets

- Small 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 Enclaves



Reduced attack surface with SGX

Application gains ability to defend its own secrets

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

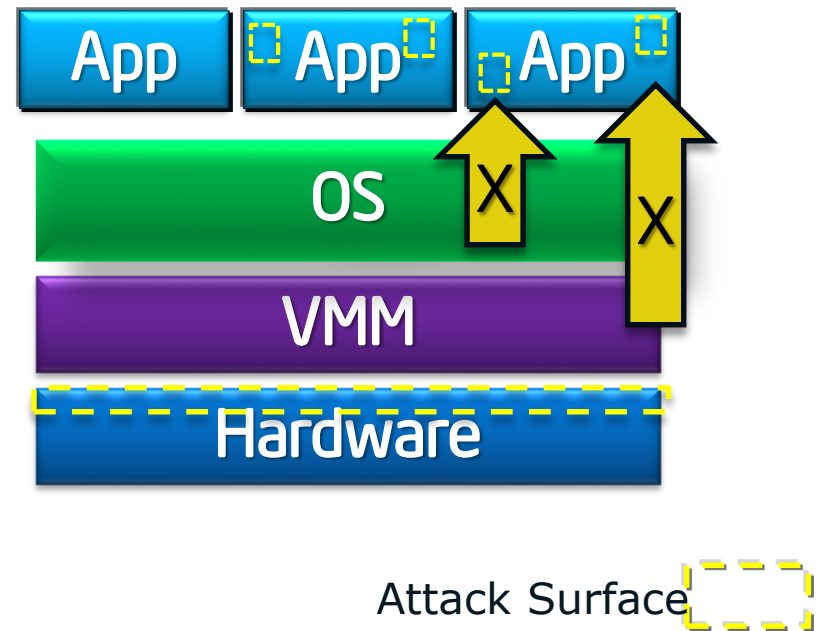
Familiar development/debug

- Single application environment
- Build on existing ecosystem expertise

Familiar deployment model

- Platform integration not a bottleneck to deployment of trusted apps

Attack surface with Enclaves



Scalable security within mainstream environment

How SE Works: Protection vs. Software Attack

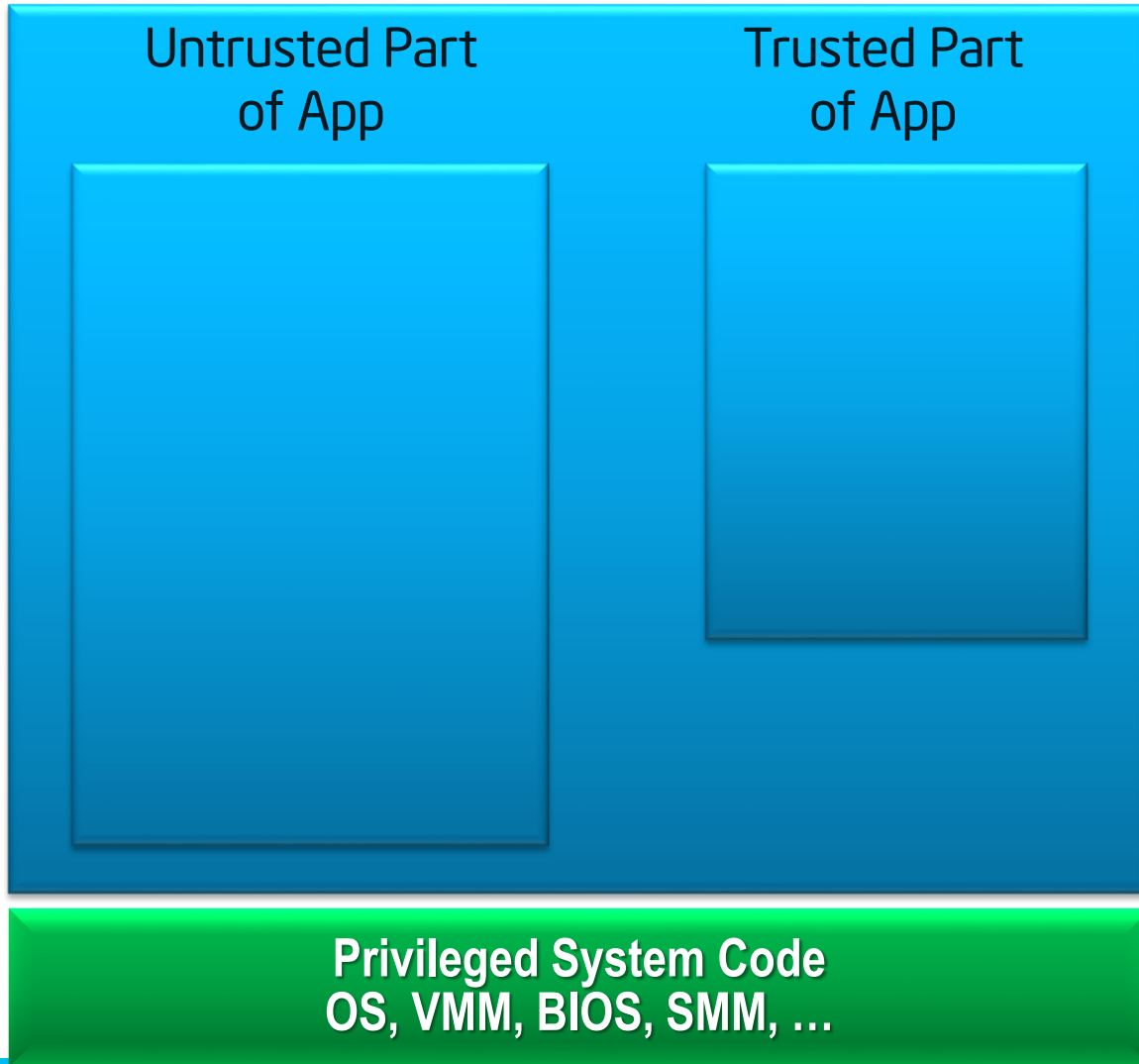
Application



Privileged System Code
OS, VMM, BIOS, SMM, ...

How SE Works: Protection vs. Software Attack

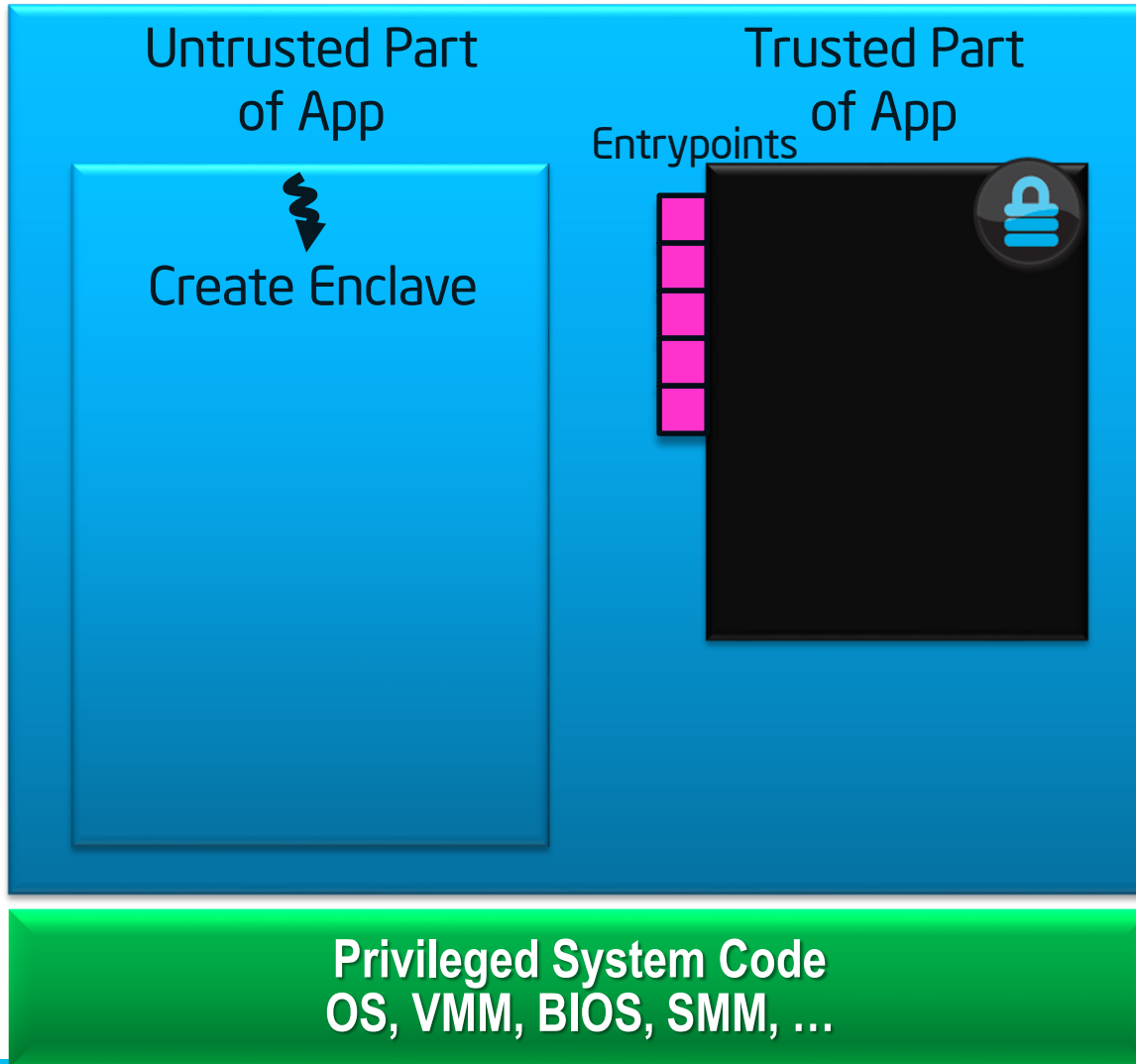
Application



1. App is built with trusted and untrusted parts

How SE Works: Protection vs. Software Attack

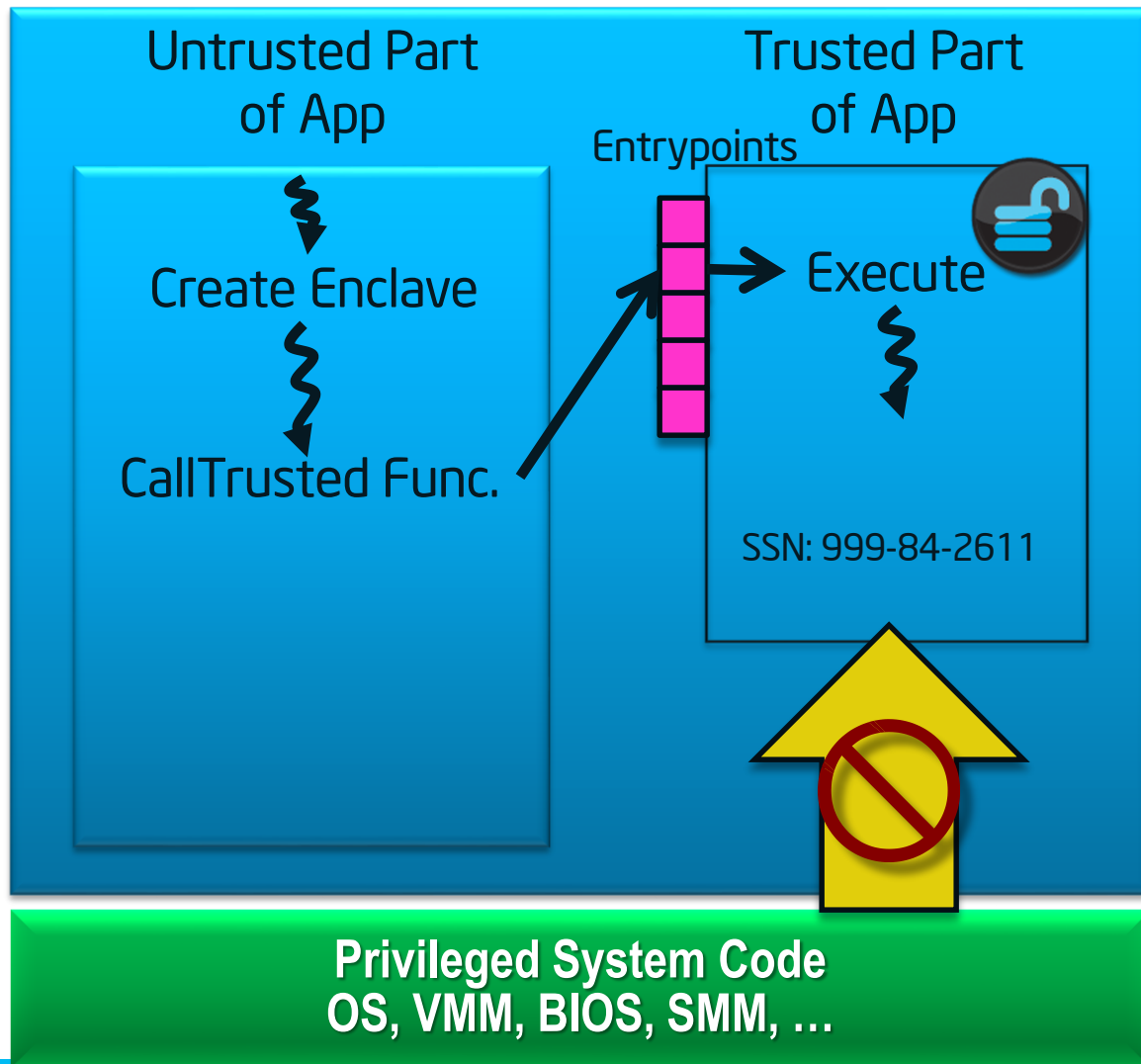
Application



1. App is built with trusted and untrusted parts
2. App runs & creates enclave which is placed in trusted memory

How SE Works: Protection vs. Software Attack

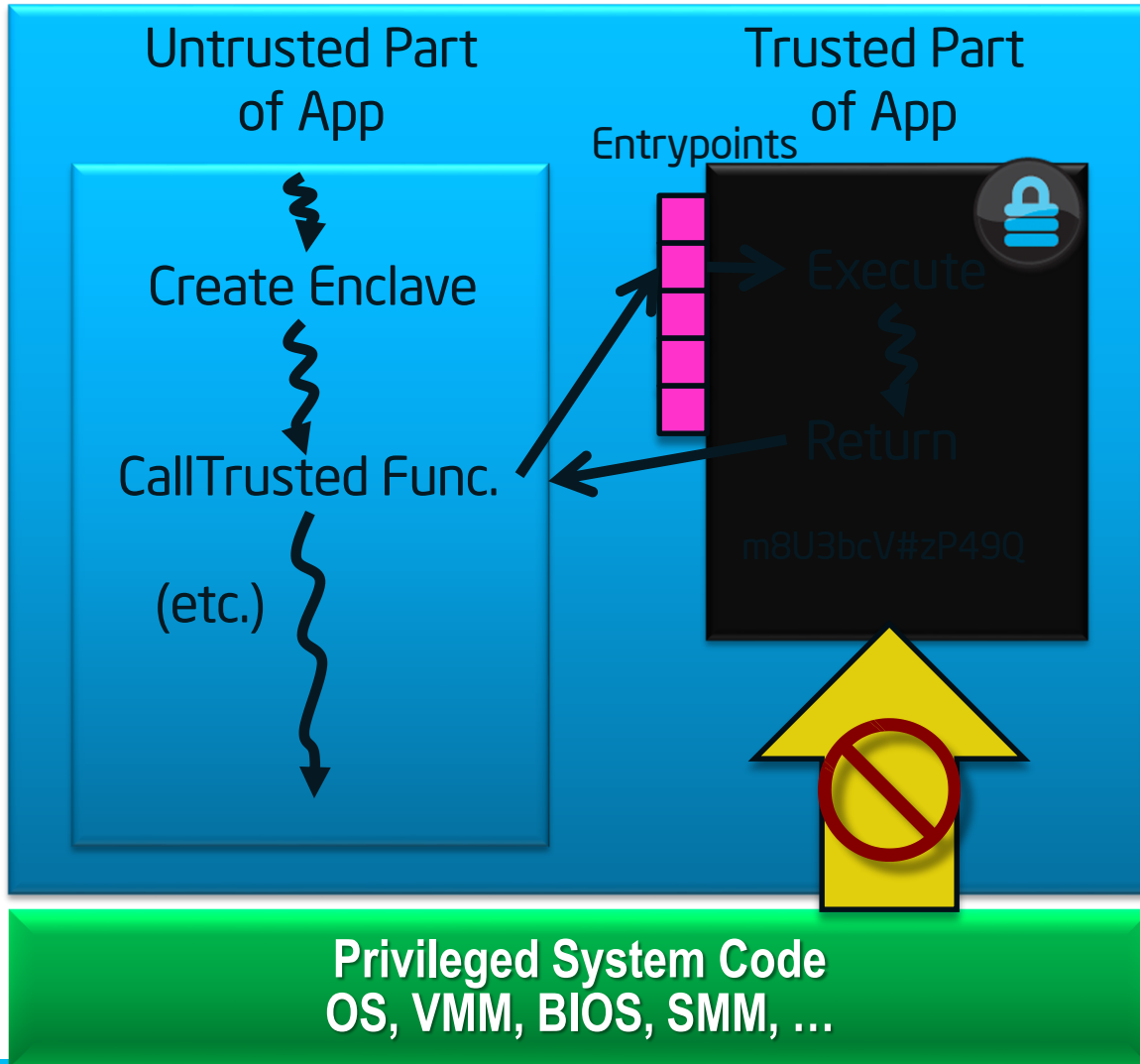
Application



1. App is built with trusted and untrusted parts
2. App runs & creates enclave which is placed in trusted memory
3. Trusted function is called; code running inside enclave sees data in clear; external access to data is denied

How SE Works: Protection vs. Software Attack

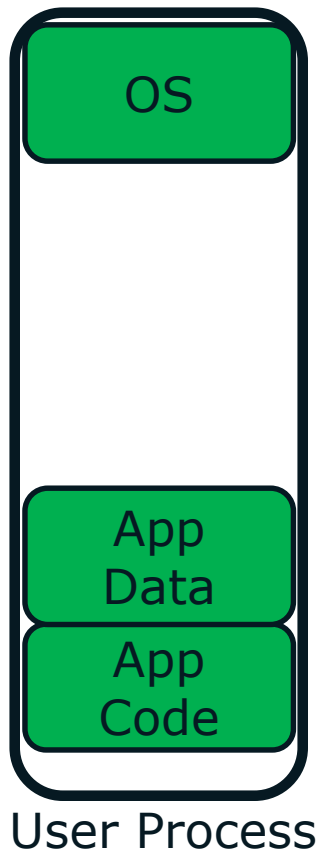
Application



1. App is built with trusted and untrusted parts
2. App runs & creates enclave which is placed in trusted memory
3. Trusted function is called; code running inside enclave sees data in clear; external access to data is denied
4. Function returns; enclave data remains in trusted memory

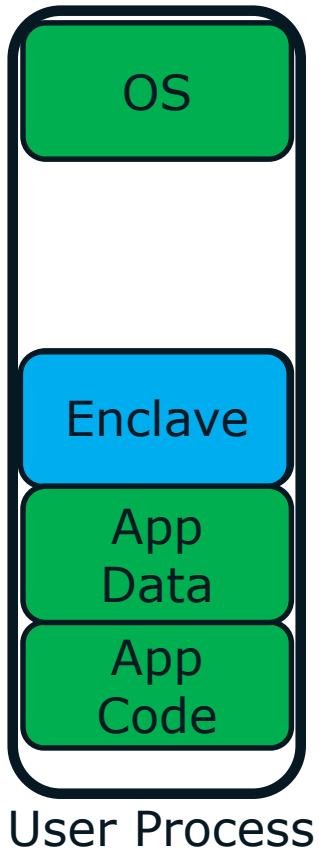
SGX Programming Environment

Trusted execution environment embedded in a process



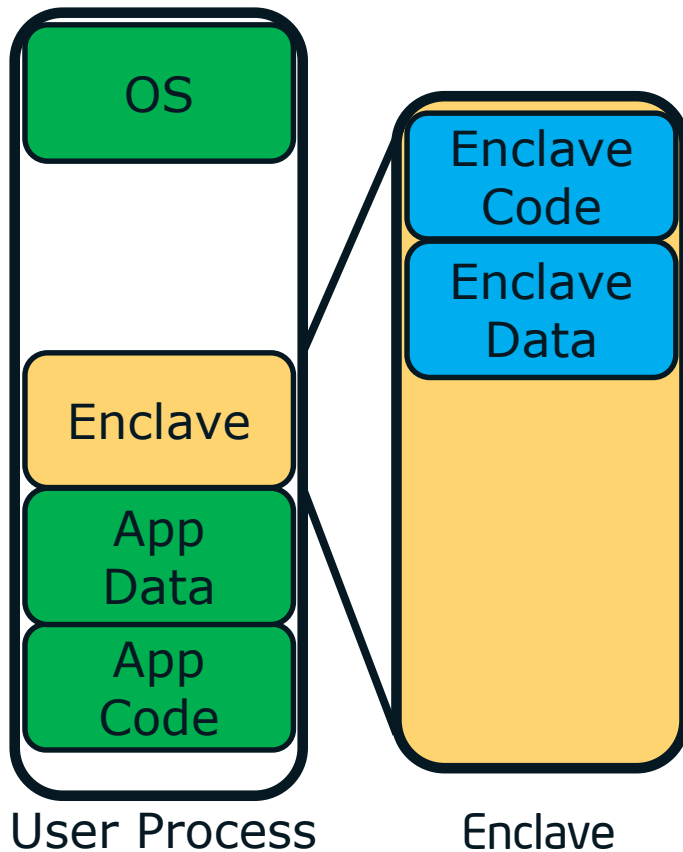
SGX Programming Environment

Trusted execution environment embedded in a process



SGX Programming Environment

Trusted execution environment embedded in a process



With its own code and data

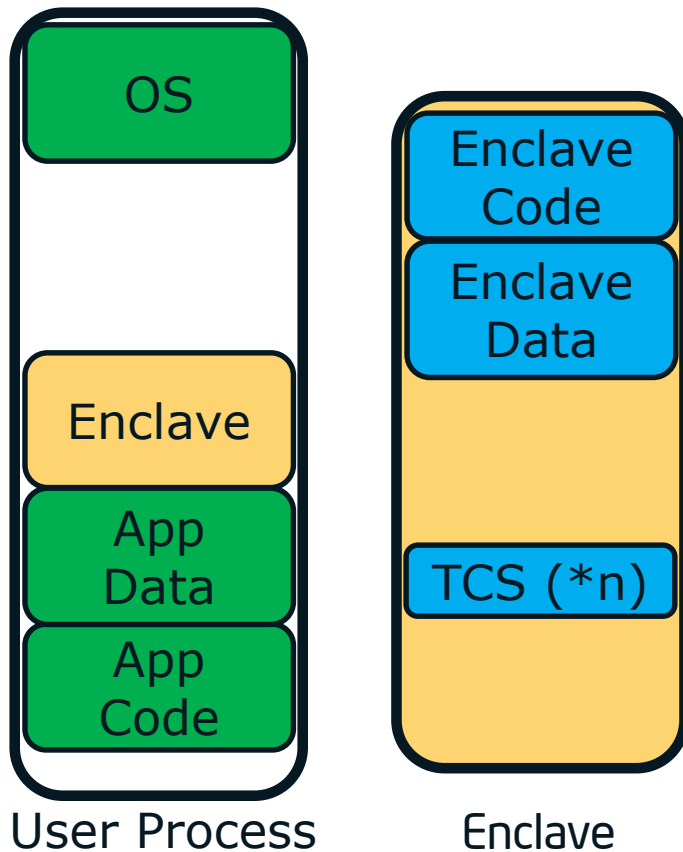
Provide Confidentiality

Provide integrity

With controlled entry points

SGX Programming Environment

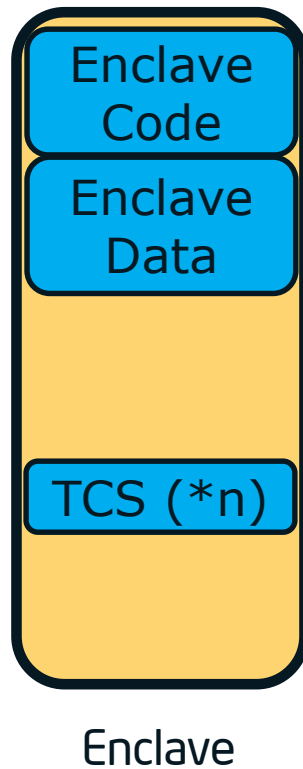
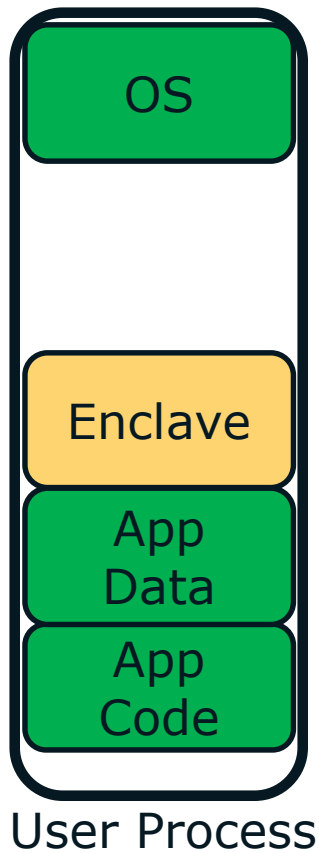
Trusted execution environment embedded in a process



With its own code and data
Provide Confidentiality
Provide integrity
With controlled entry points
Supporting multiple threads

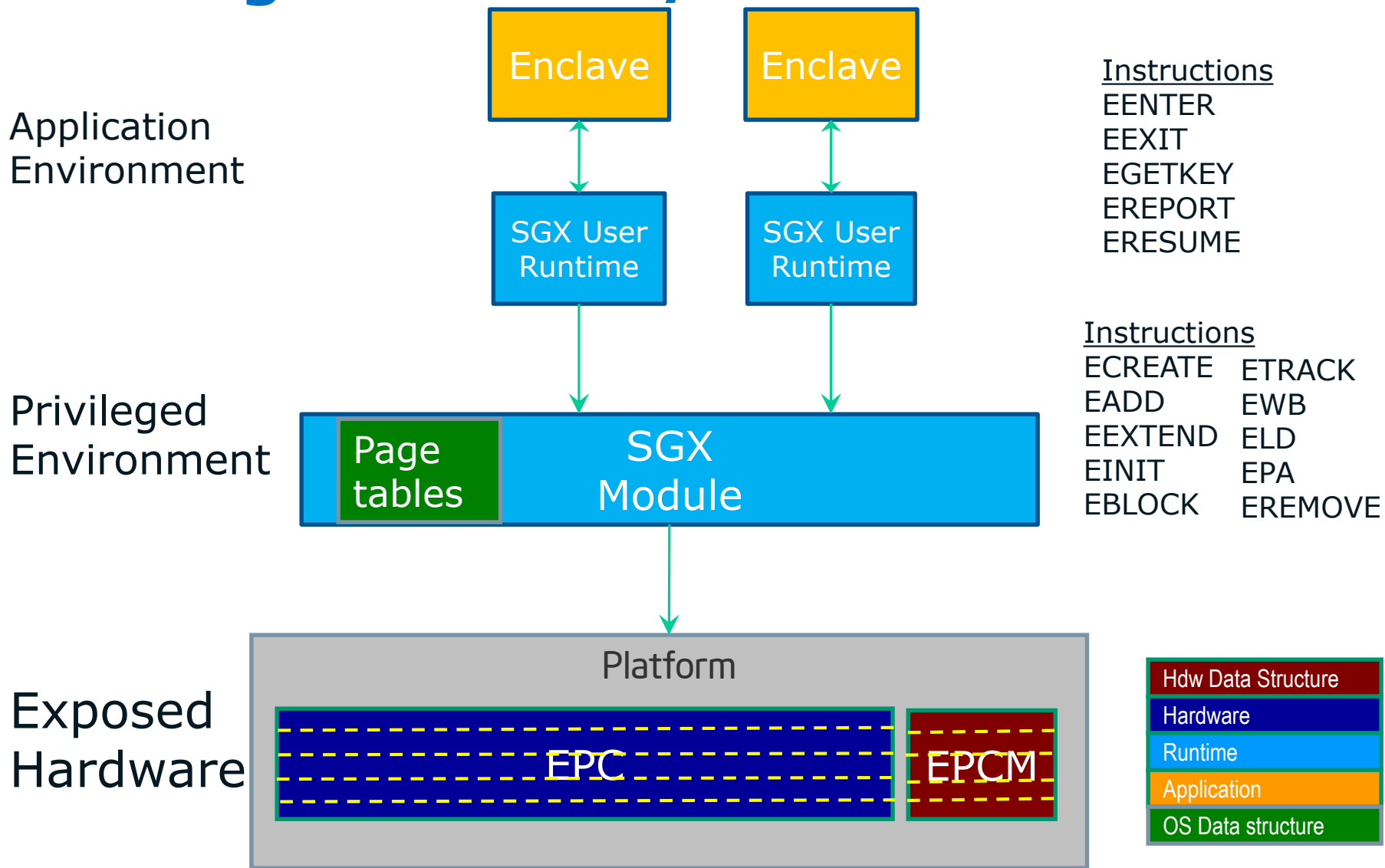
SGX Programming Environment

Trusted execution environment embedded in a process

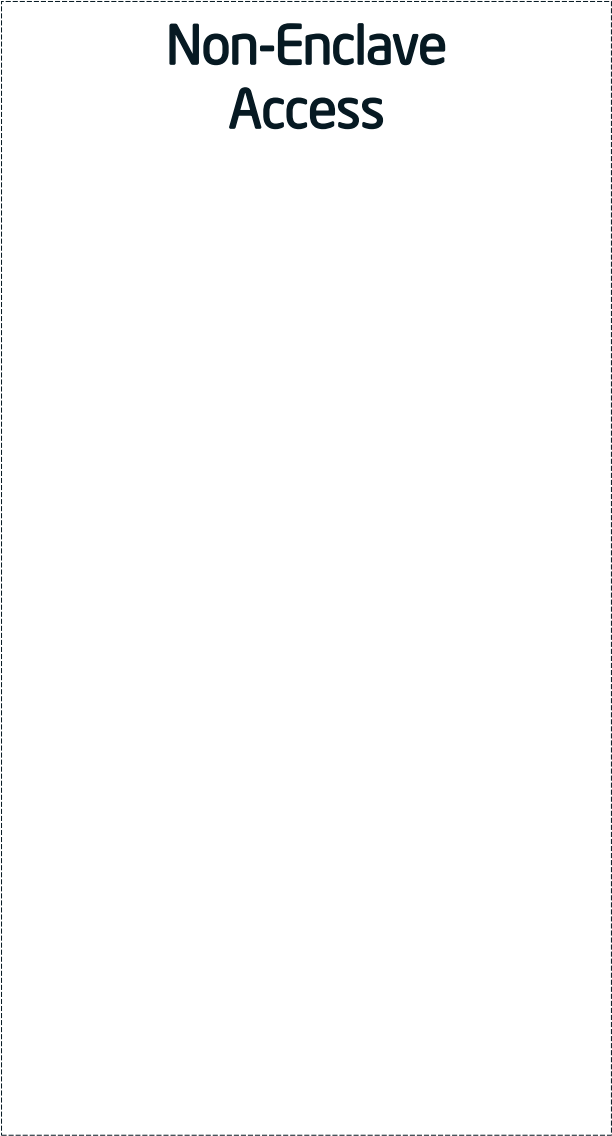
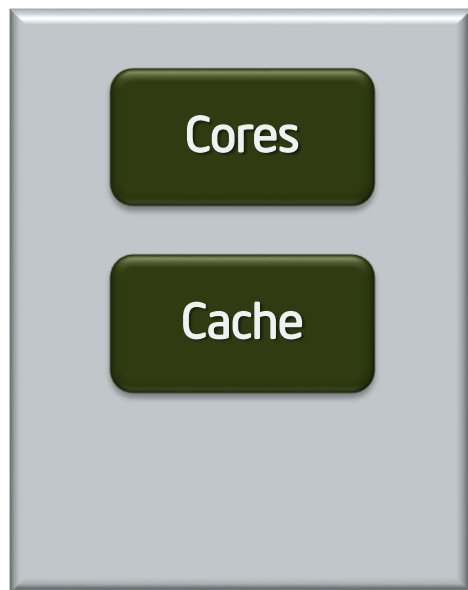


With its own code and data
Provide Confidentiality
Provide integrity
With controlled entry points
Supporting multiple threads
With full access to app memory

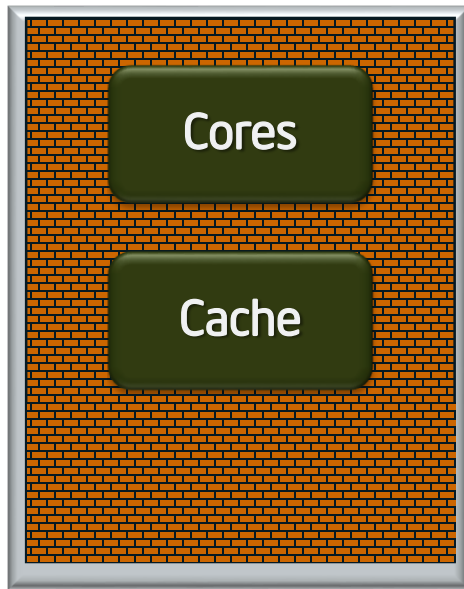
SGX High-level HW/SW Picture



Protection vs. Memory Snooping Attacks



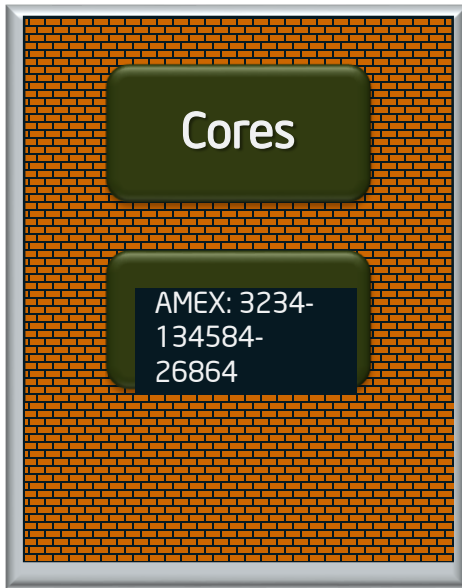
Protection vs. Memory Snooping Attacks



Non-Enclave Access

- Security perimeter is the CPU package boundary

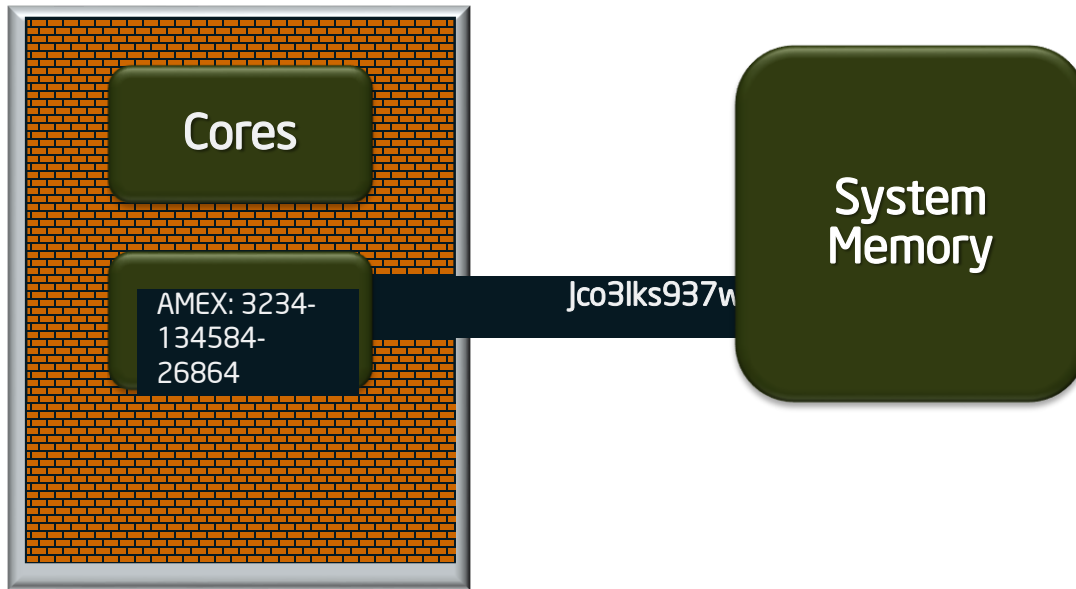
Protection vs. Memory Snooping Attacks



Non-Enclave Access

- Security perimeter is the CPU package boundary
- Data and code unencrypted inside CPU package

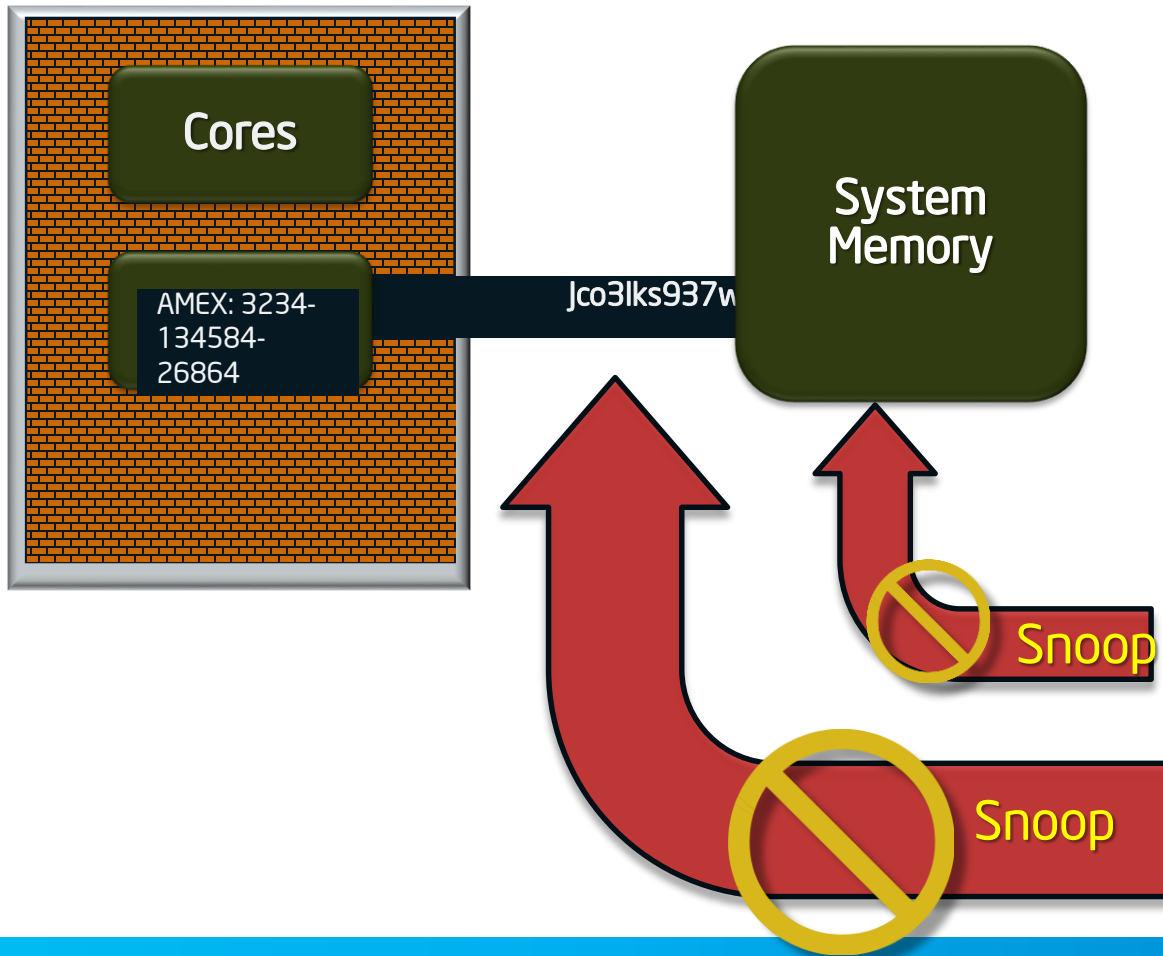
Protection vs. Memory Snooping Attacks



Non-Enclave Access

- Security perimeter is the CPU package boundary
- Data and code unencrypted inside CPU package
- Data and code outside CPU package is encrypted and integrity checked

Protection vs. Memory Snooping Attacks



Non-Enclave Access

- Security perimeter is the CPU package boundary
- Data and code unencrypted inside CPU package
- Data and code outside CPU package is encrypted and integrity checked
- External memory reads and bus snoops see only encrypted data

Critical Feature: Attestation and Sealing

Client Application

Remote Platform



Critical Feature: Attestation and Sealing



Remote Platform



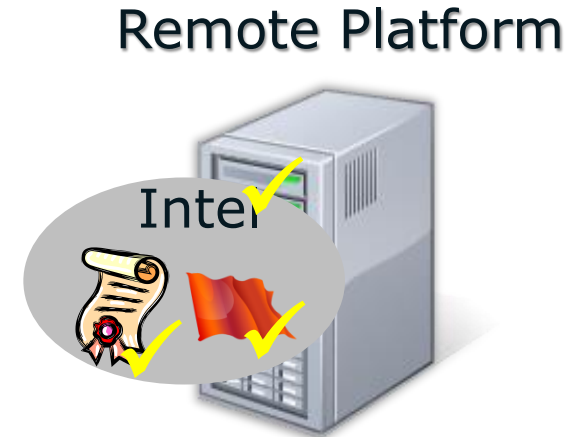
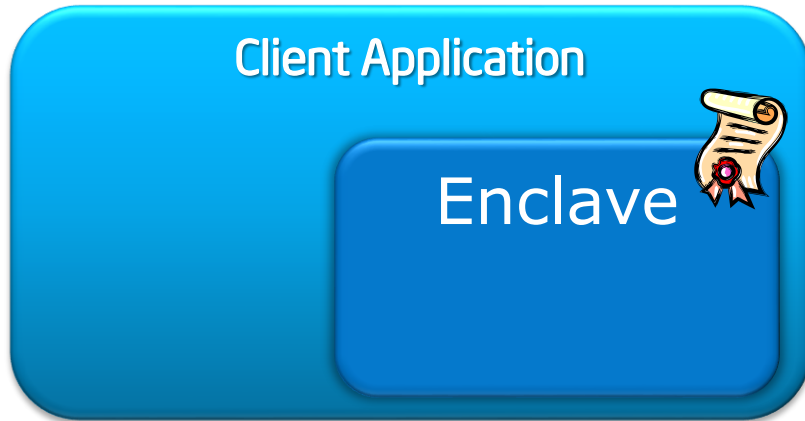
1. Enclave built & measured

Critical Feature: Attestation and Sealing



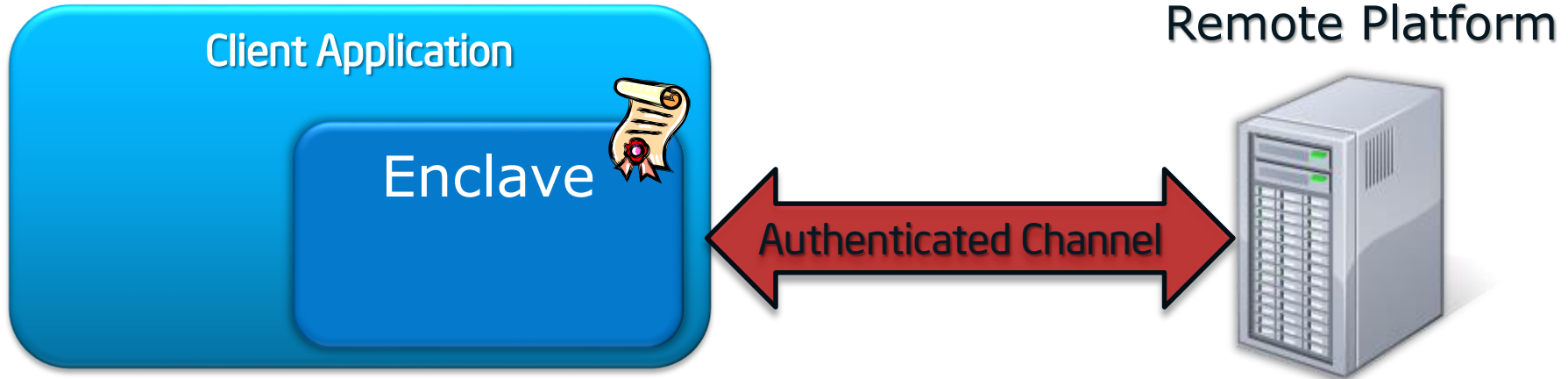
1. Enclave built & measured
2. Enclave requests REPORT (HW-signed blob that includes enclave identity information)

Critical Feature: Attestation and Sealing



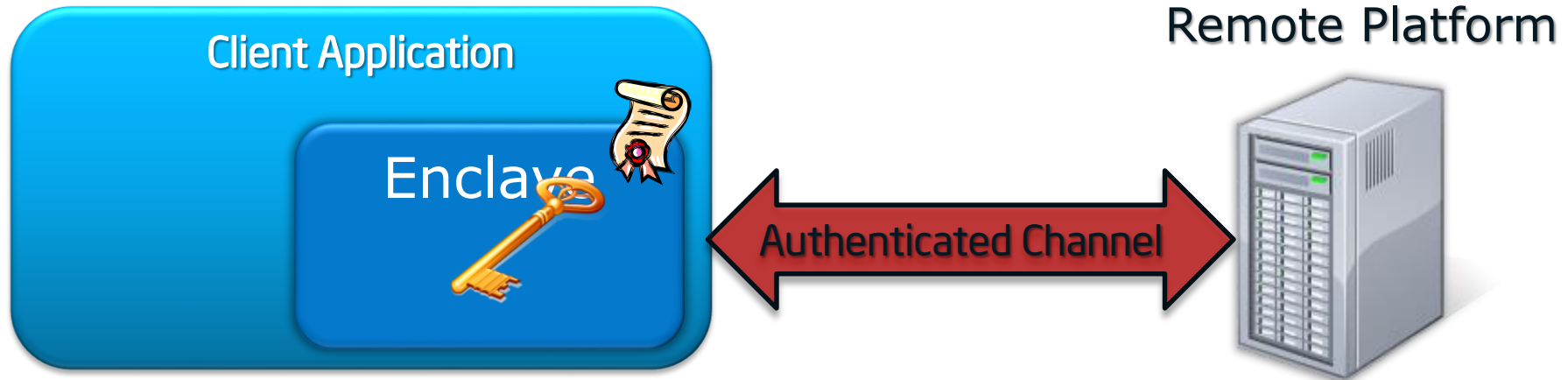
1. Enclave built & measured
2. Enclave requests REPORT (HW-signed blob that includes enclave identity information)
3. REPORT sent to server & verified

Critical Feature: Attestation and Sealing



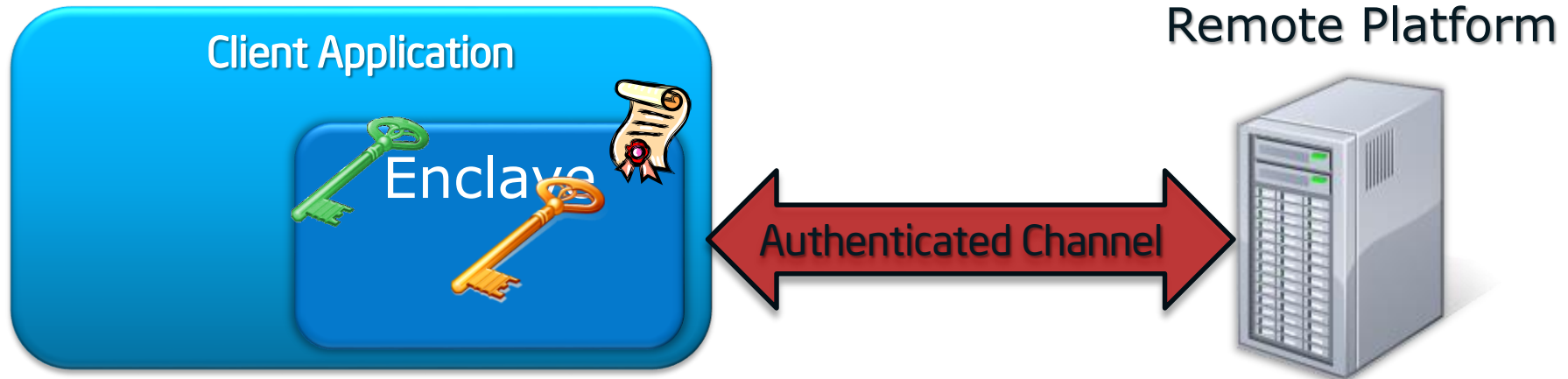
1. Enclave built & measured
2. Enclave requests REPORT (HW-signed blob that includes enclave identity information)
3. REPORT sent to server & verified

Critical Feature: Attestation and Sealing



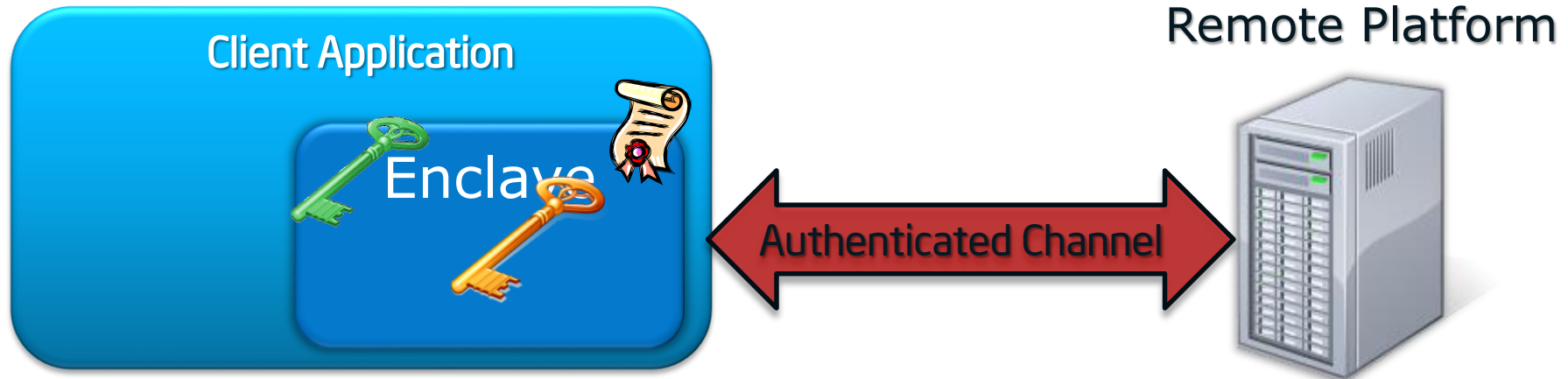
1. Enclave built & measured
2. Enclave requests REPORT (HW-signed blob that includes enclave identity information)
3. REPORT sent to server & verified
4. Application Key sent to enclave, first secret provisioned

Critical Feature: Attestation and Sealing

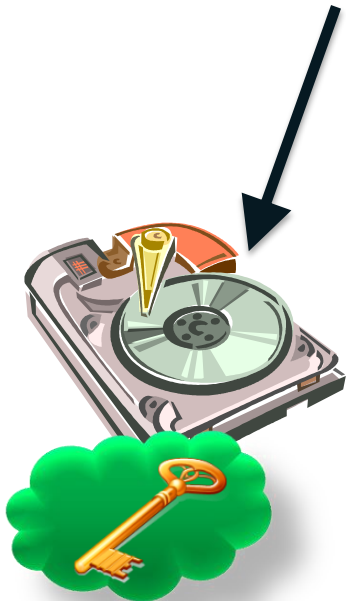


1. Enclave built & measured
2. Enclave requests REPORT (HW-signed blob that includes enclave identity information)
3. REPORT sent to server & verified
4. Application Key sent to enclave, first secret provisioned
5. Enclave-platform-specific Sealing Key generated (EGETKEY)

Critical Feature: Attestation and Sealing



1. Enclave built & measured
2. Enclave requests REPORT (HW-signed blob that includes enclave identity information)
3. REPORT sent to server & verified
4. Application Key sent to enclave, first secret provisioned
5. Enclave-platform-specific Sealing Key generated (EGETKEY)
6. Application Key encrypted via Sealing Key & stored for later (offline) use



Creating an Enclave - ISV

- Developer writes and compiles the enclave
 - Trusted functions at the enclave and rest outside
 - SGX1.0 will need to allocate all the memory upfront, SGX2.0 can dynamically allocate memory.
- Developer installs the enclave as DEBUG
 - Any enclave can be run as debug
 - Debug OPTIN is controlled per thread, and can be set via EDBGWR.

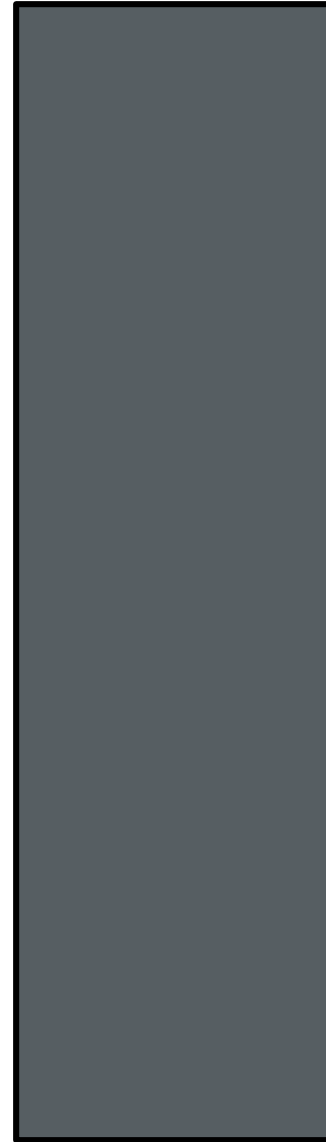
Creating SIGSTRUCT - ISV

- SIGSTRUCT – Enclave Signature Structure
 - Used to measure the enclave and attributes.
- Measuring the enclave content using SHA-256
- Specifying the attributes
- Setting the ISV information
 - Product ID
 - Security version number - ISVSVN
- Signing the App's SIGSTRUCT using the ISV private key with RSA-3072.

Life Cycle of An Enclave

Physical Address Space

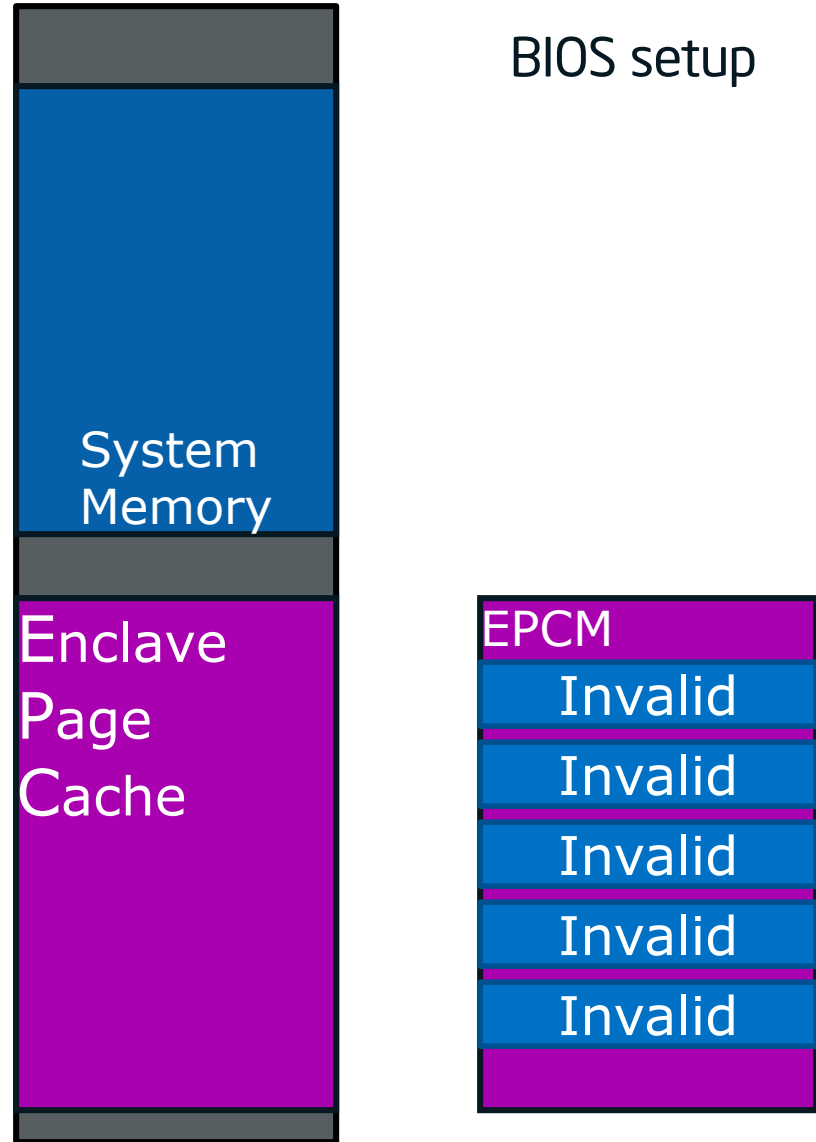
1/15



Life Cycle of An Enclave

Physical Address Space

2/15

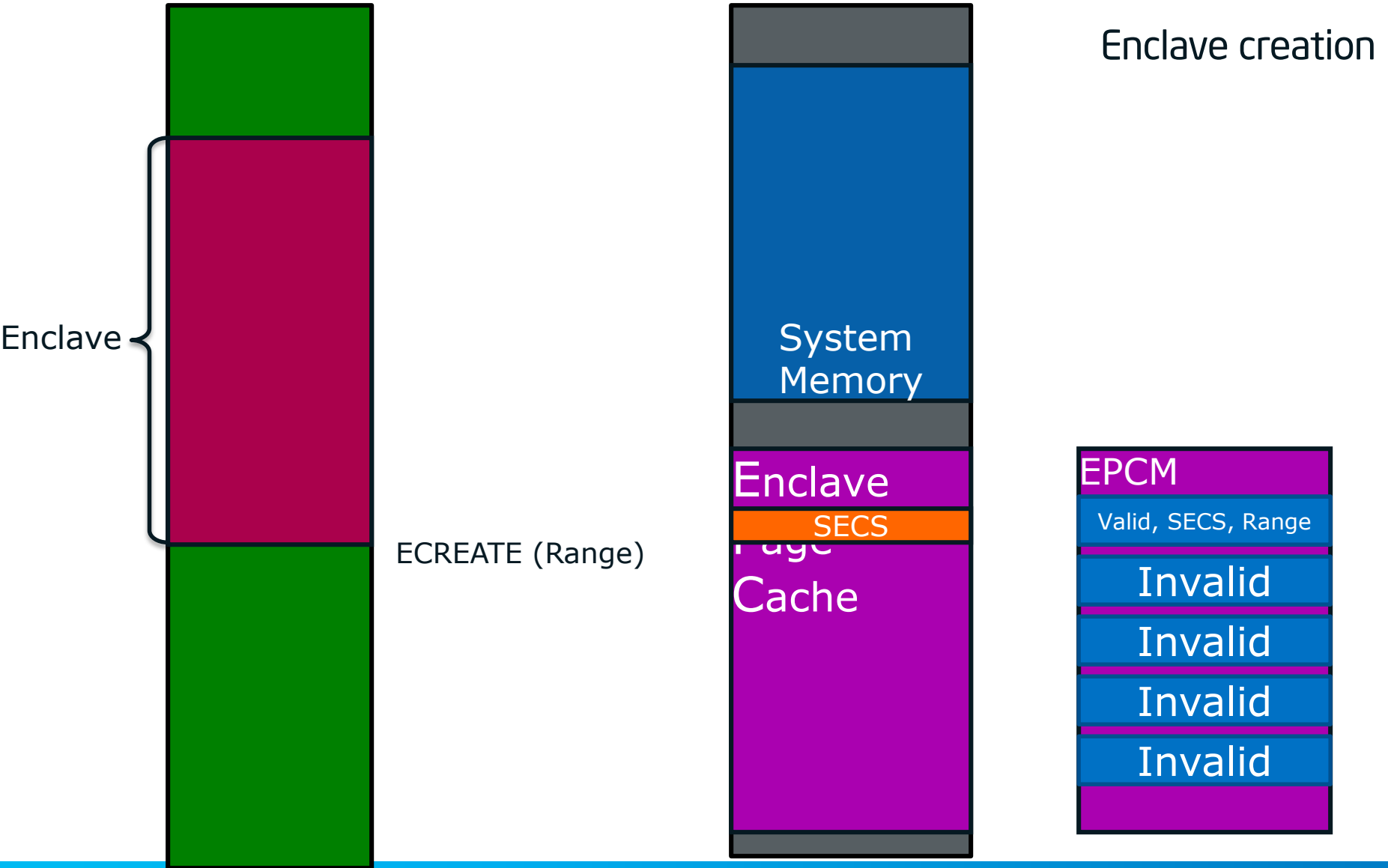


Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

3/15



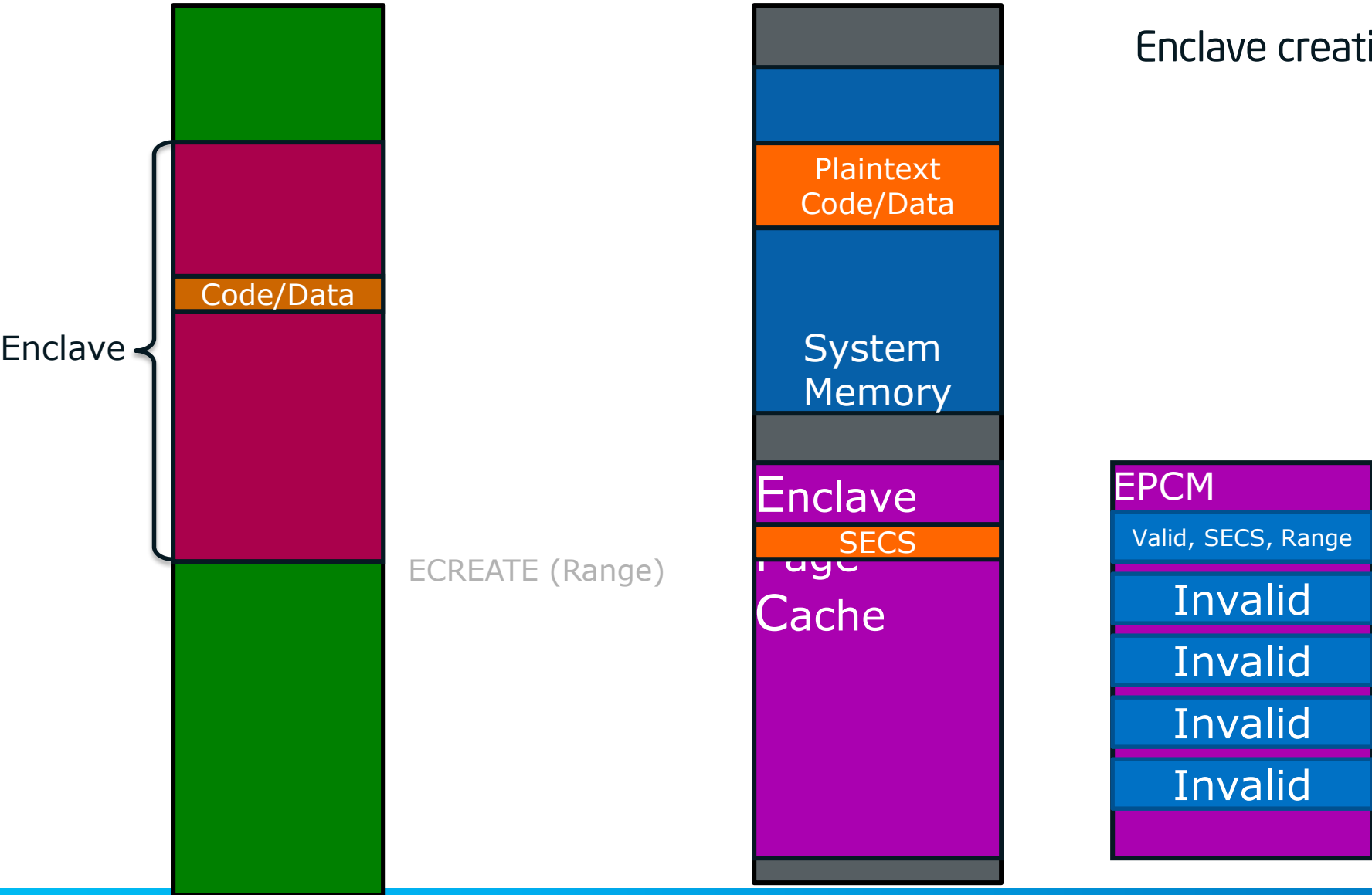
Life Cycle of An Enclave

4/15

Virtual Address Space

Physical Address Space

Enclave creation



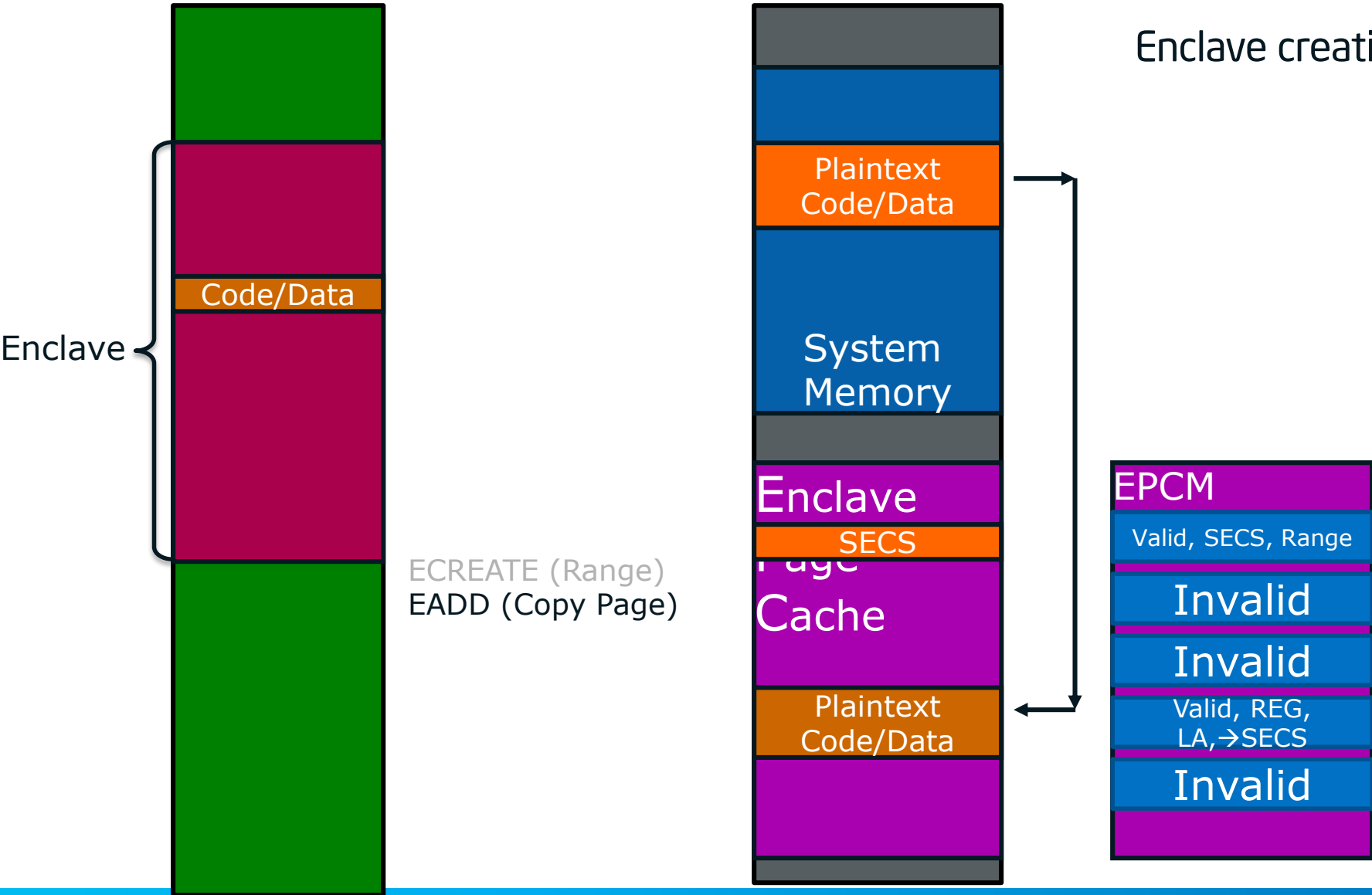
Life Cycle of An Enclave

5/15

Virtual Address Space

Physical Address Space

Enclave creation



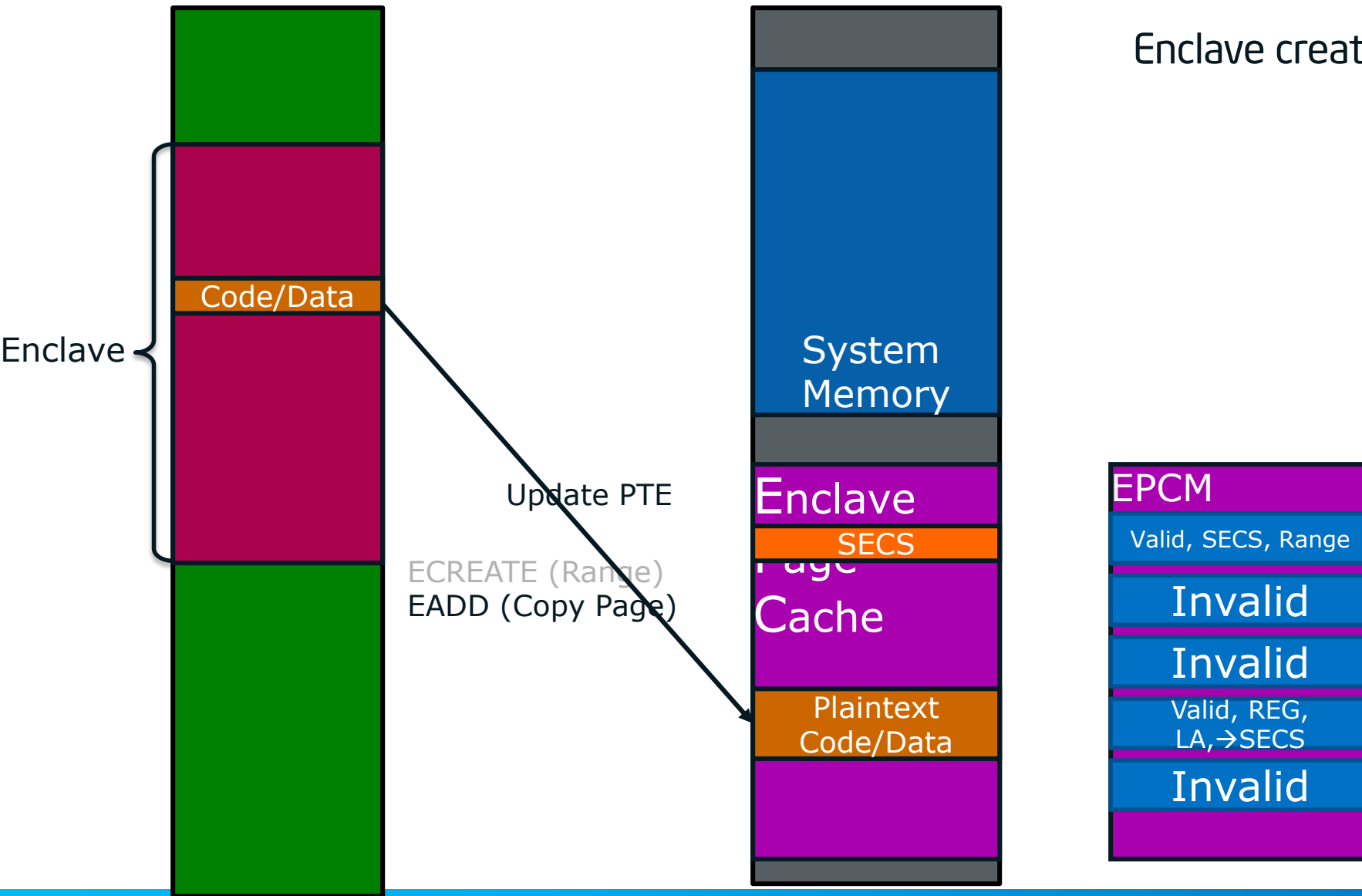
Life Cycle of An Enclave

6/15

Virtual Address Space

Physical Address Space

Enclave creation



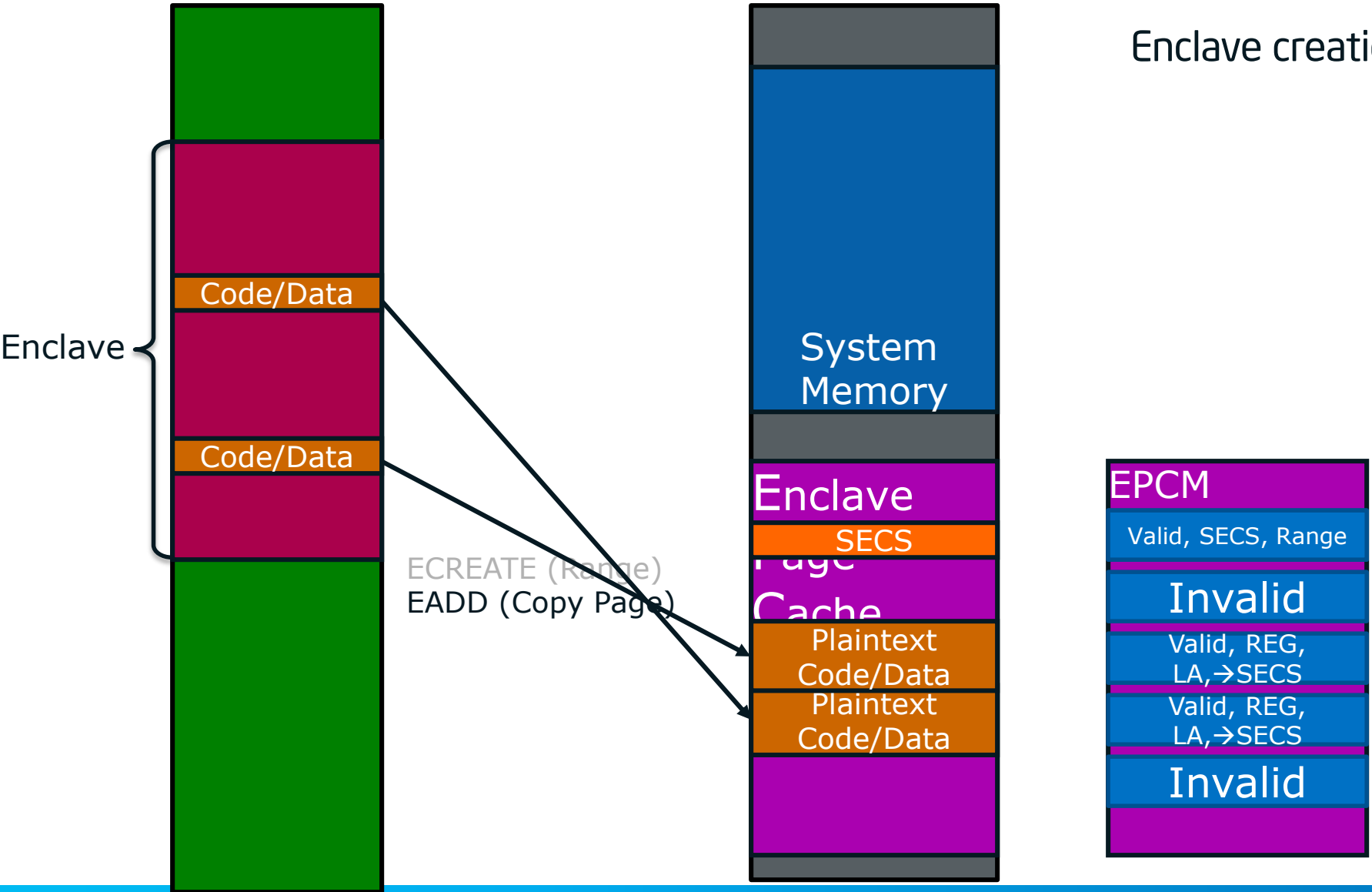
Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

7/15

Enclave creation

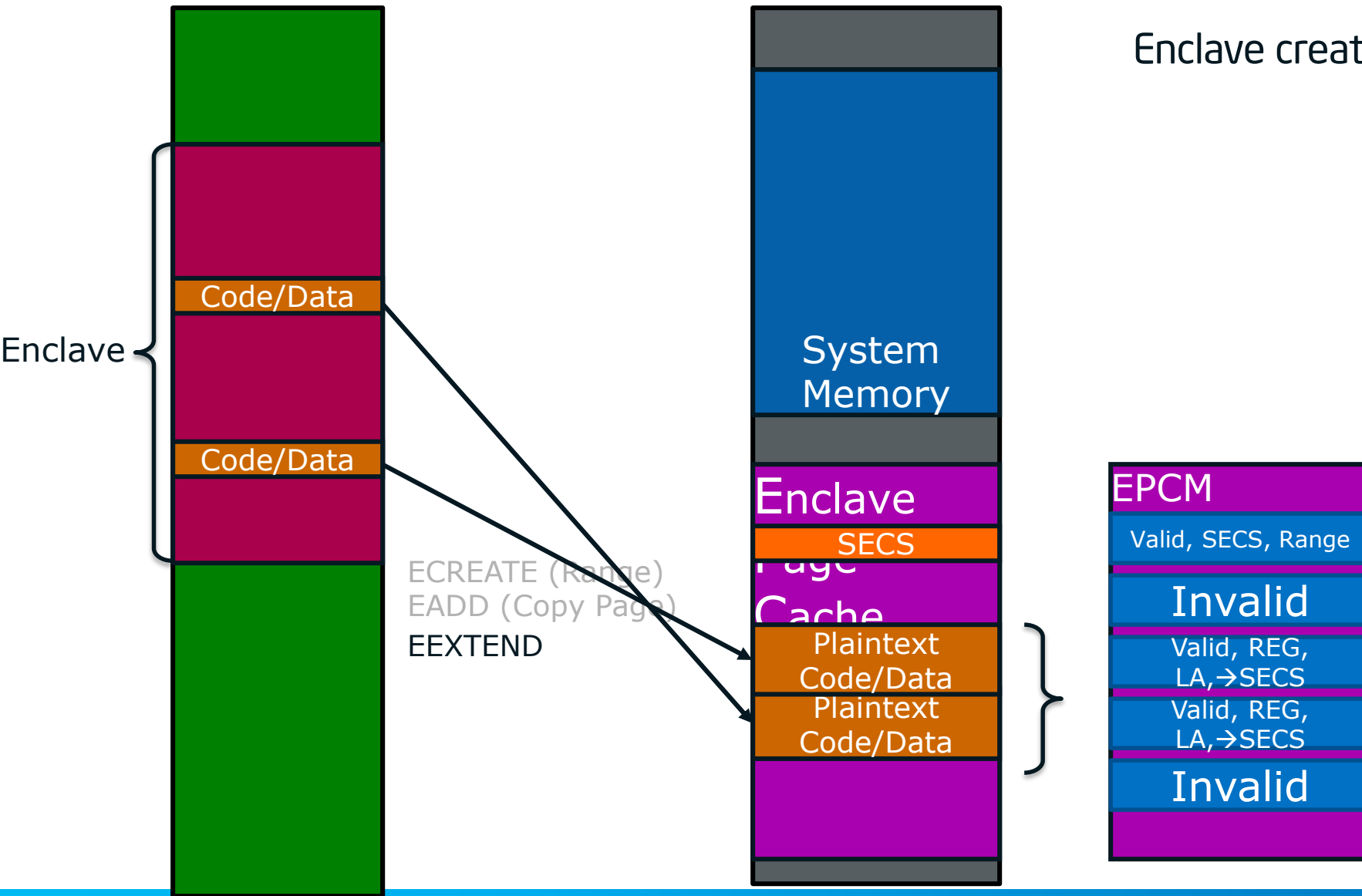


Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

Enclave creation

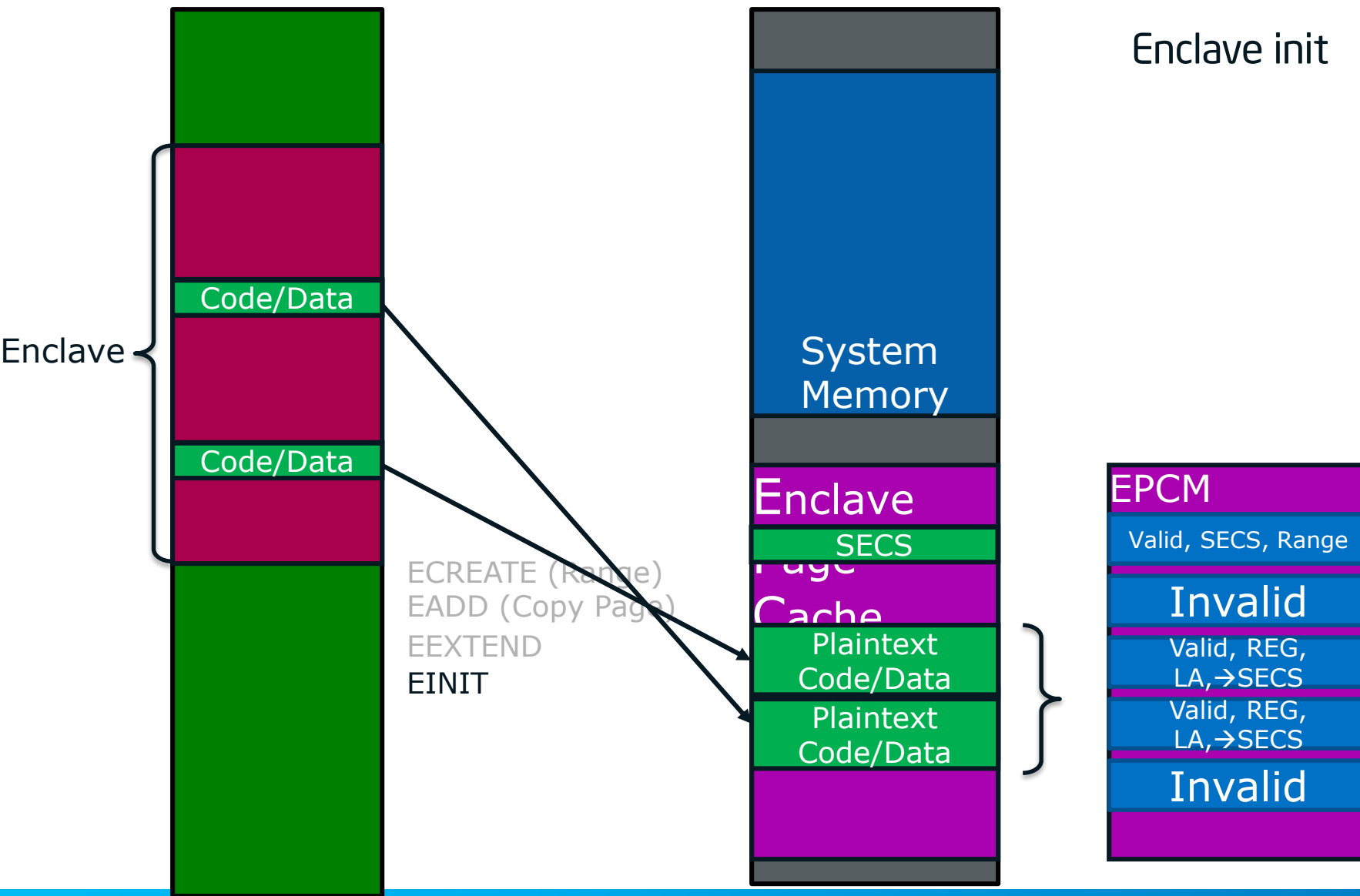


Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

9/15



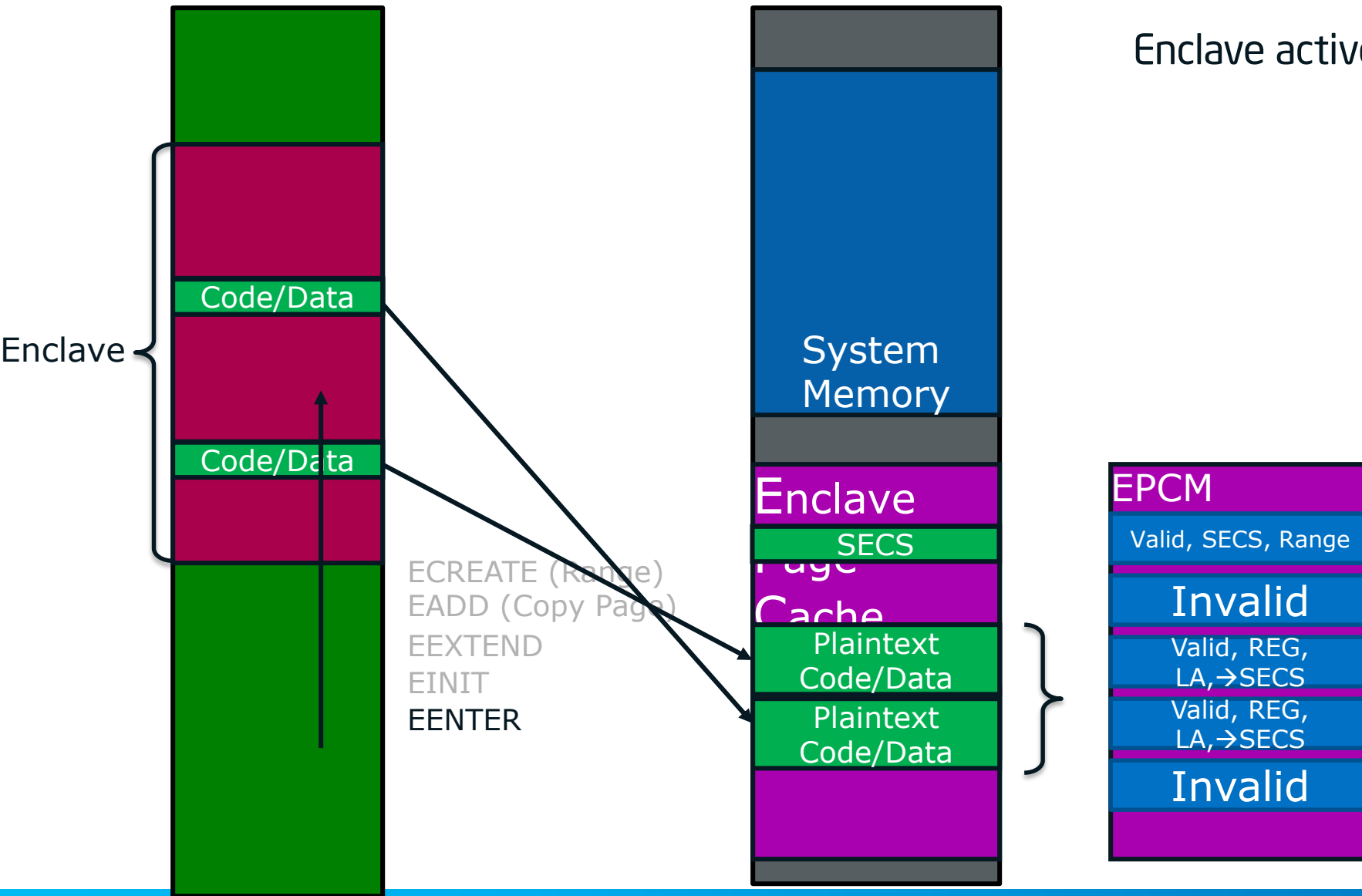
Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

10 / 15

Enclave active



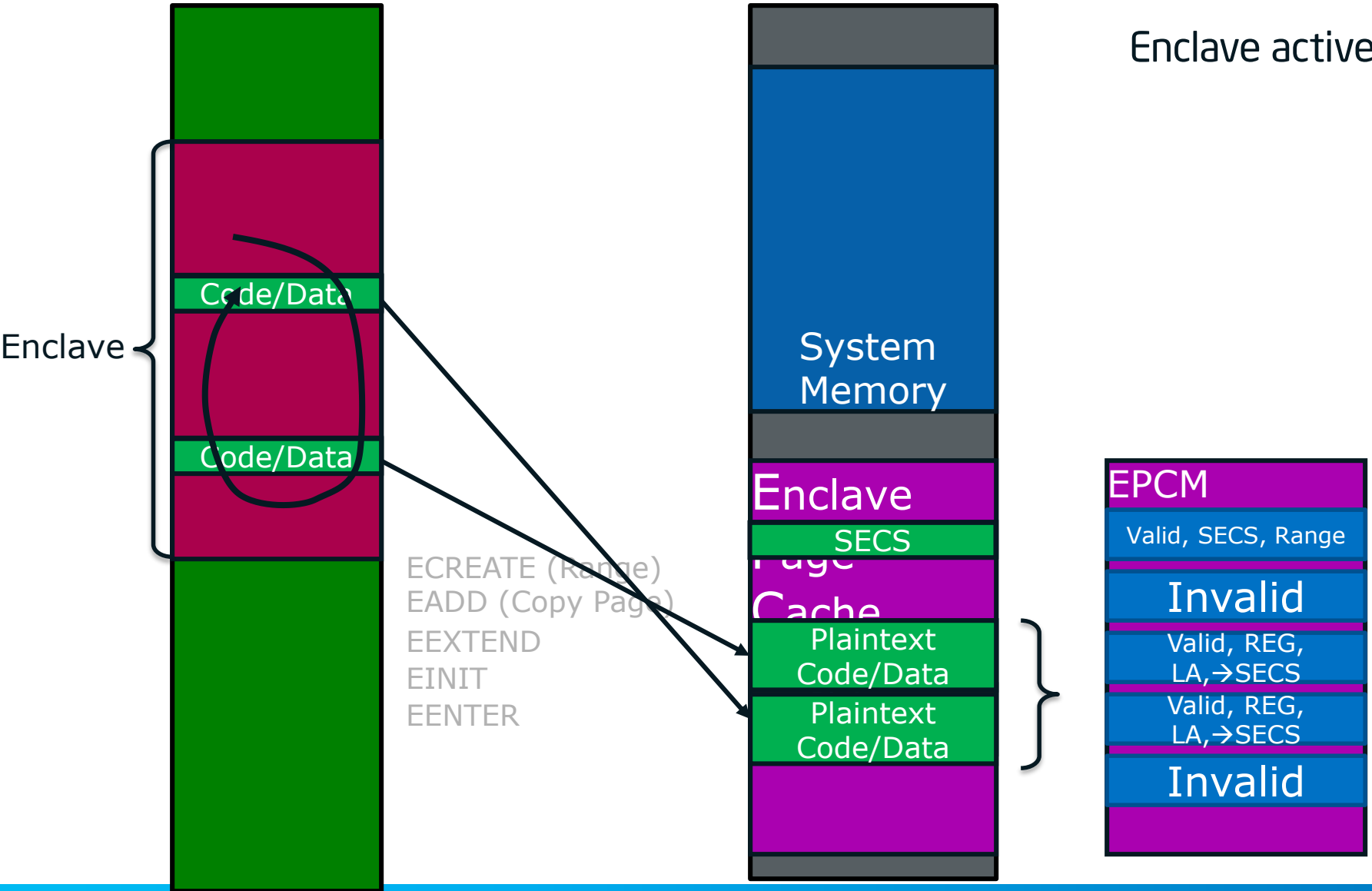
Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

11 / 15

Enclave active

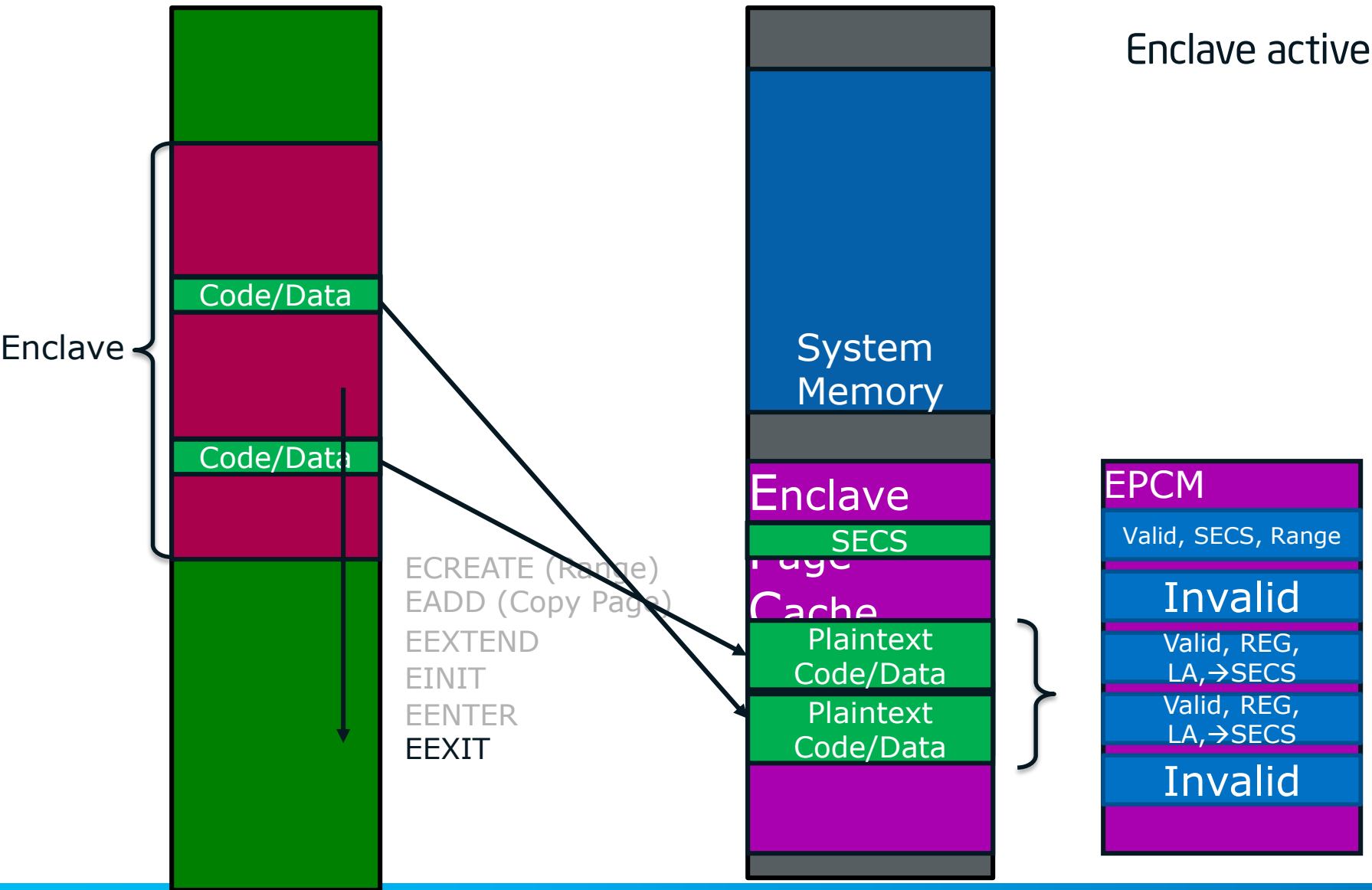


Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

12/15



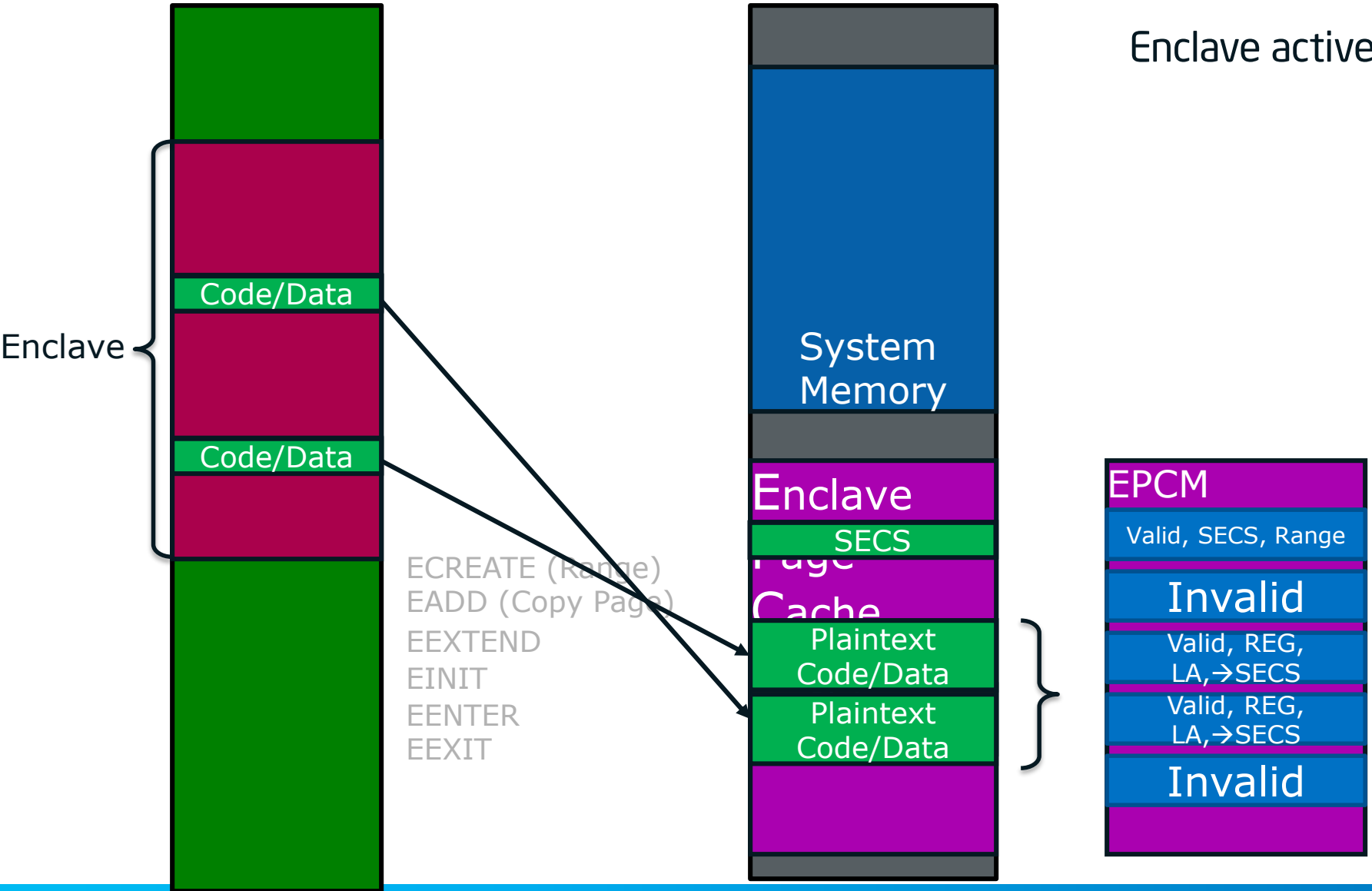
Life Cycle of An Enclave

Virtual Address Space

Physical Address Space

13 / 15

Enclave active



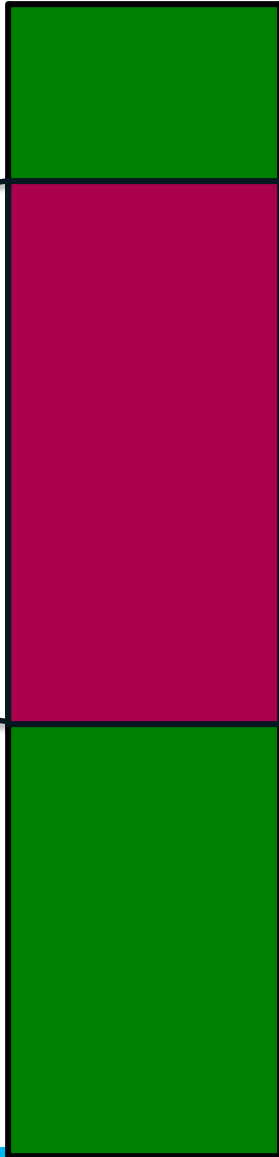
Life Cycle of An Enclave

Virtual Address Space

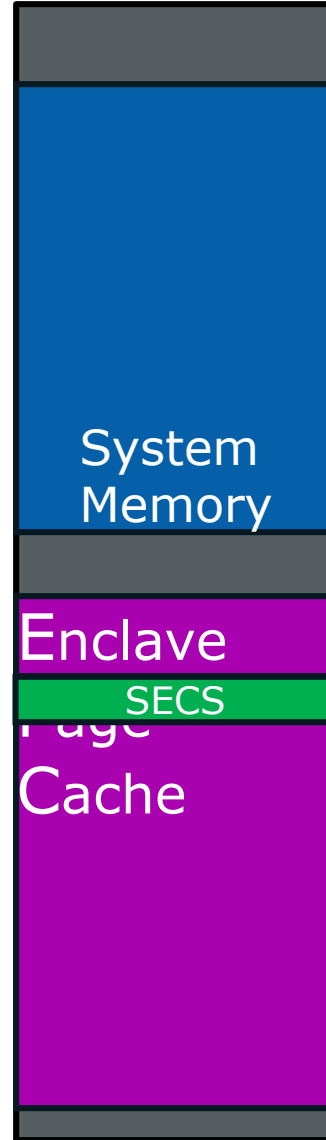
Physical Address Space

14 / 15

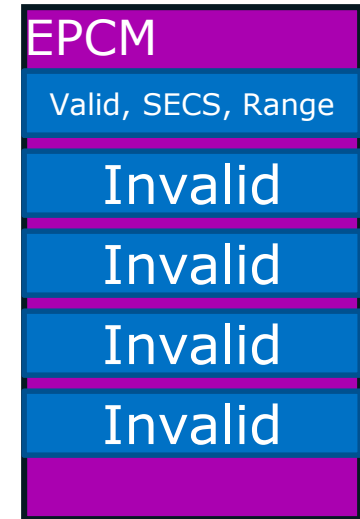
Enclave



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



Enclave destruction

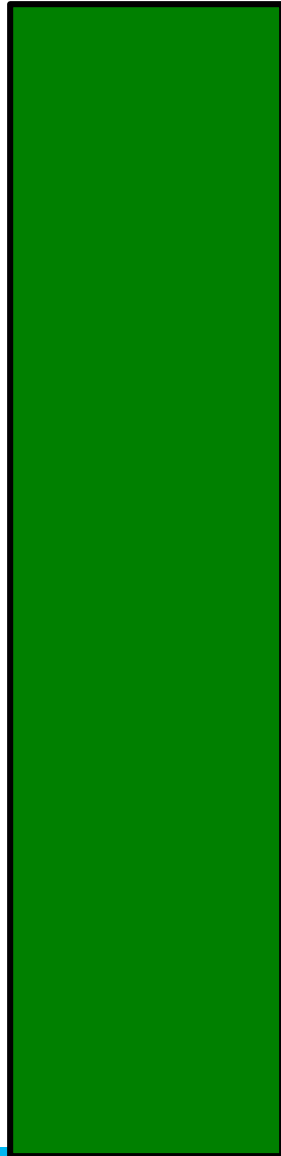


Life Cycle of An Enclave

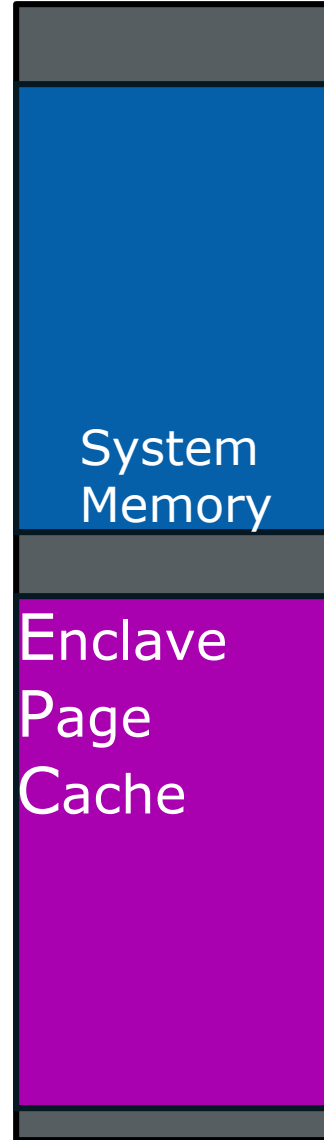
Virtual Address Space

Physical Address Space

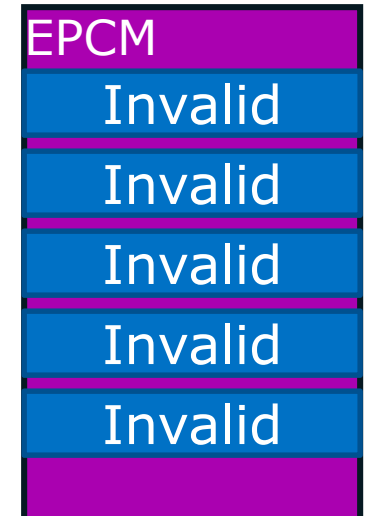
15 / 15



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



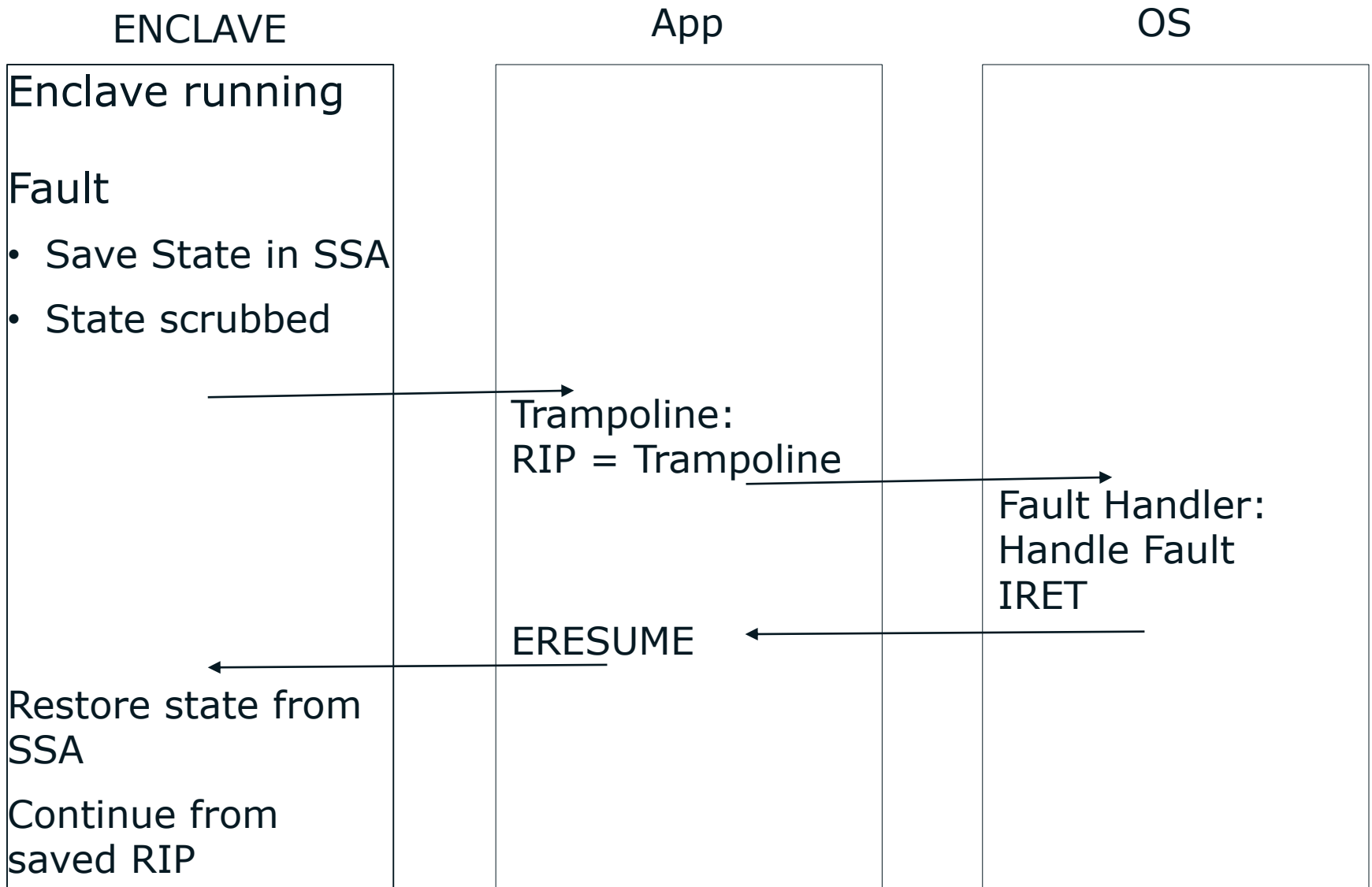
Enclave destruction



Handling Exceptions

- Asynchronous Exit (AEX)
 - Faults, exceptions and interrupts initiate the Asynchronous Exit flow.
 - During AEX, enclave register state is stored in the enclave's active SSA frame and initialized to a known value prior to leaving the enclave
 - The RIP is initialized to an area referred to as the trampoline code
- SSA
 - Each enclave thread has a dedicated State Save Area frame entry that is pre-defined by the ISV for that thread

Handling Exceptions



Attestation

SGX provides LOCAL and REMOTE attestation capabilities

Local attestation allows one enclave to attest its TCB to another enclave on the same platform

Remote attestation allows one enclave to attest its TCB to another entity outside of the platform

Sealing

“Sealing”: Cryptographically protecting data when it is stored outside enclave.

Enclaves use EGETKEY to retrieve a persistent key that is enclave & platform specific

EGETKEY uses a combination of enclave attributes and platform unique key to generate keys

- Enclave Identity
- Enclave Sealing Authority & Product Identity

Enclave is responsible for performing the encryption with an algorithm of its choice.

Summary

- Intel® SGX provides outstanding data protection and a simple programming model
 - An enclave limits the size of the TCB
 - Enclaves are protected in face of a compromised OS/VMM.
- Developers may focus on securing the smaller TCB
- Enclaves run within the application process
 - be built and debugged with familiar tools.
- The Intel SGX SW stack and tools should simplify development even more.

Links

Joint research poster session: <http://sigops.org/sosp/sosp13/>

Public Cloud Paper using SGX2:

https://www.usenix.org/sites/default/files/osdi14_full_proceedings.pdf

Programming Reference for SGX1 & SGX2:

<http://www.intel.com/software/isa>

HASP Workshop:

<https://sites.google.com/site/haspworkshop2013/workshop-program>

ISCA 2015 Tutorial Link:

<http://sgxisca.weebly.com/>



Thank You