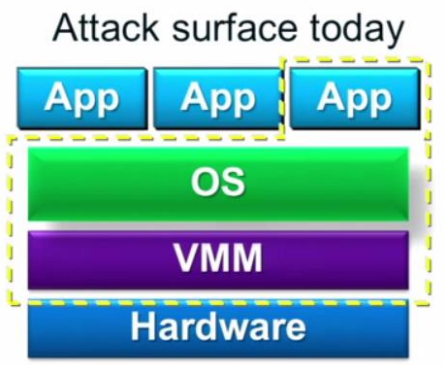
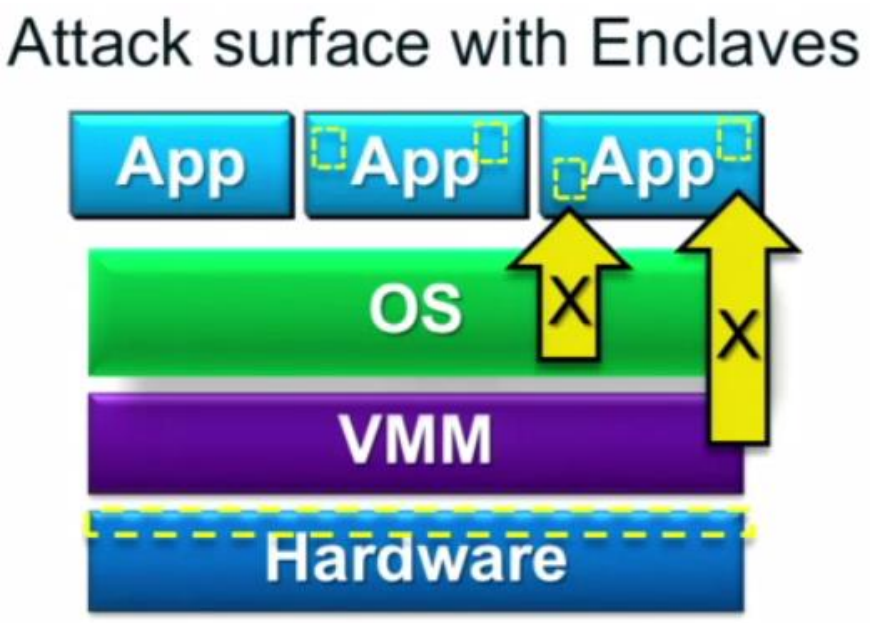
Why SGX?

SGX introduces the ENCALVE:

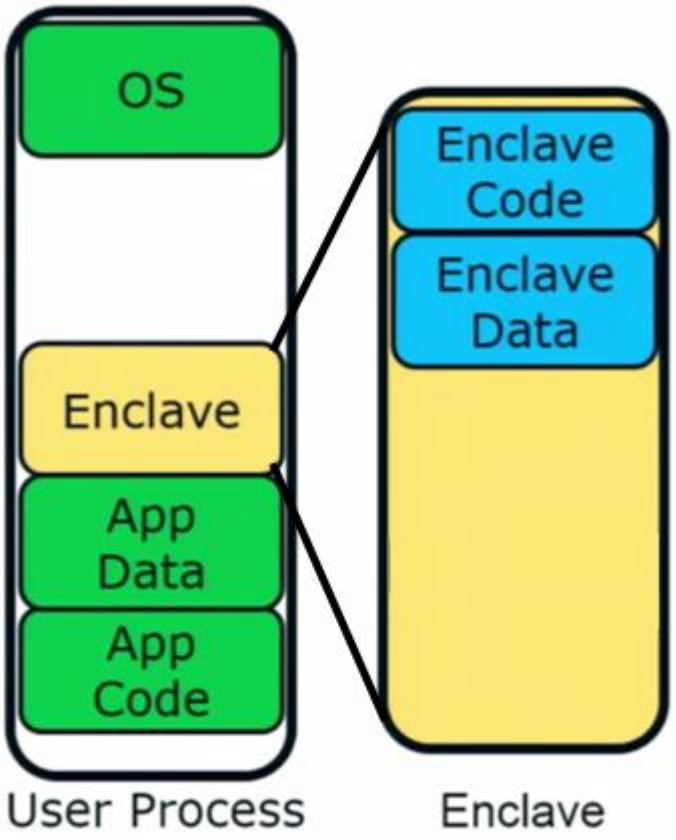
- A protected TEE (Trusted Execution Environment) container.
- Shrinks TCB (Trusted Computing Base) to HW and sensitive app logic.
- Extends HW TCB on to enclaves in ring-3.
- Runs TEE under untrusted OS or VMM.
- Supports continuous run of:
  - Trusted and untrusted apps.
  - Multiple enclaves.
  - Multi-threaded enclave.





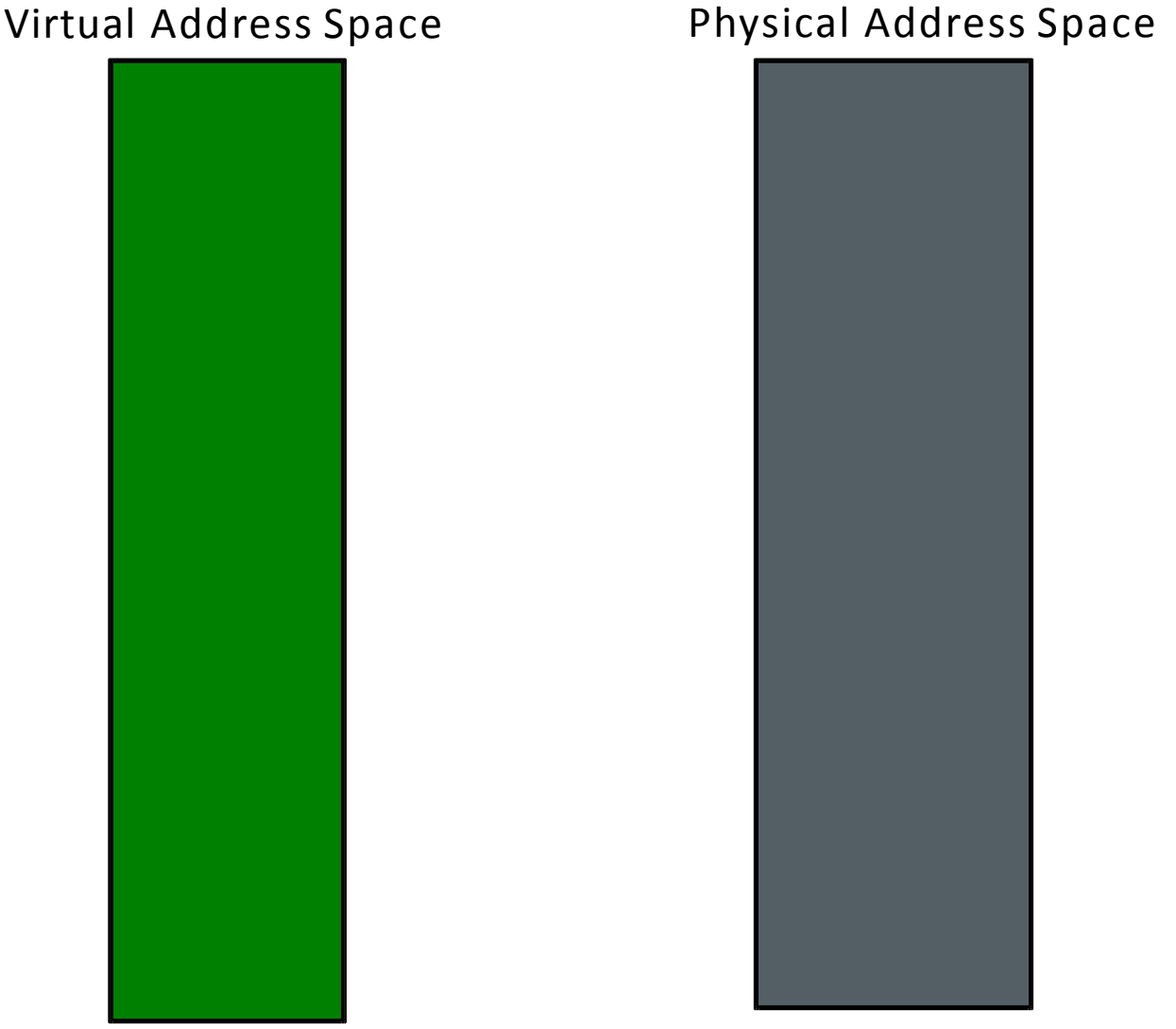
While maintaining application Integrity and confidentiality.

The Enclave – A trusted execution environment embedded in a process

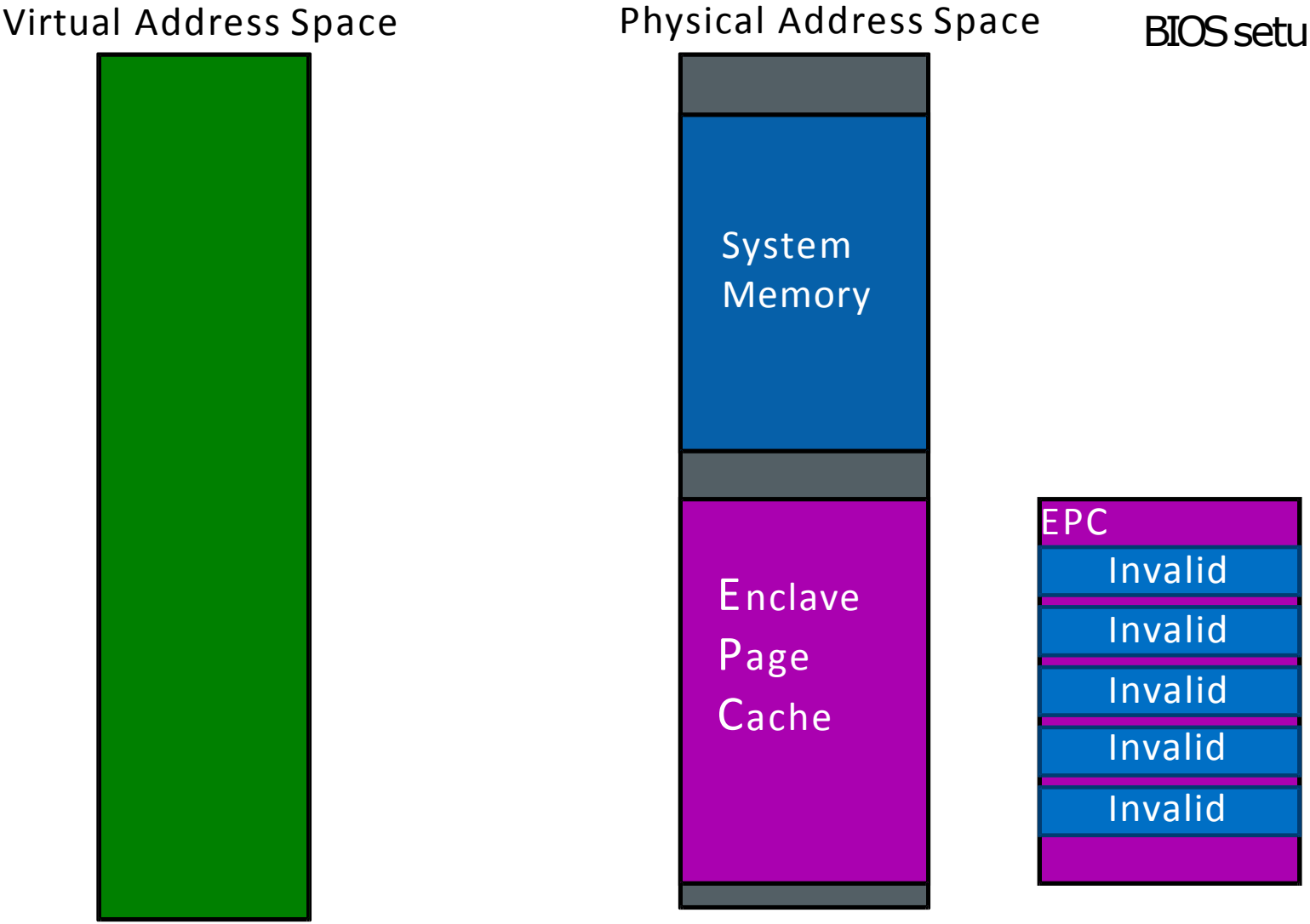


- Is part of the application & has full access to its memory.
- Measured and verified by HW on load.
- Operation visible only within CPU borders.
- Manageable by the OS, e.g:
  - CPU time slots
  - Paging and memory policy
- Secrets are provisioned or acquired only after enclave initialization.

Enclave Life Cycle

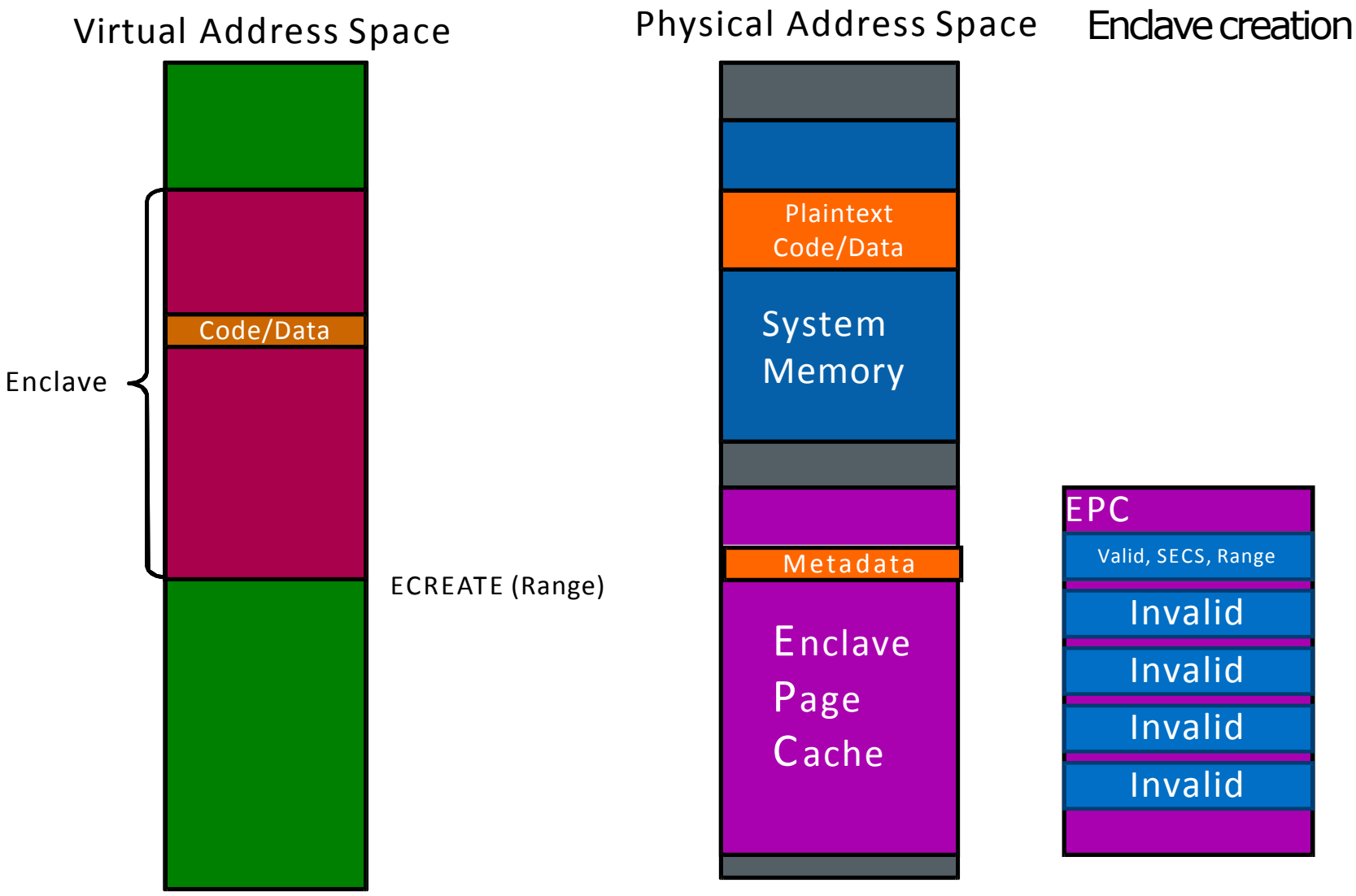


Enclave Life Cycle



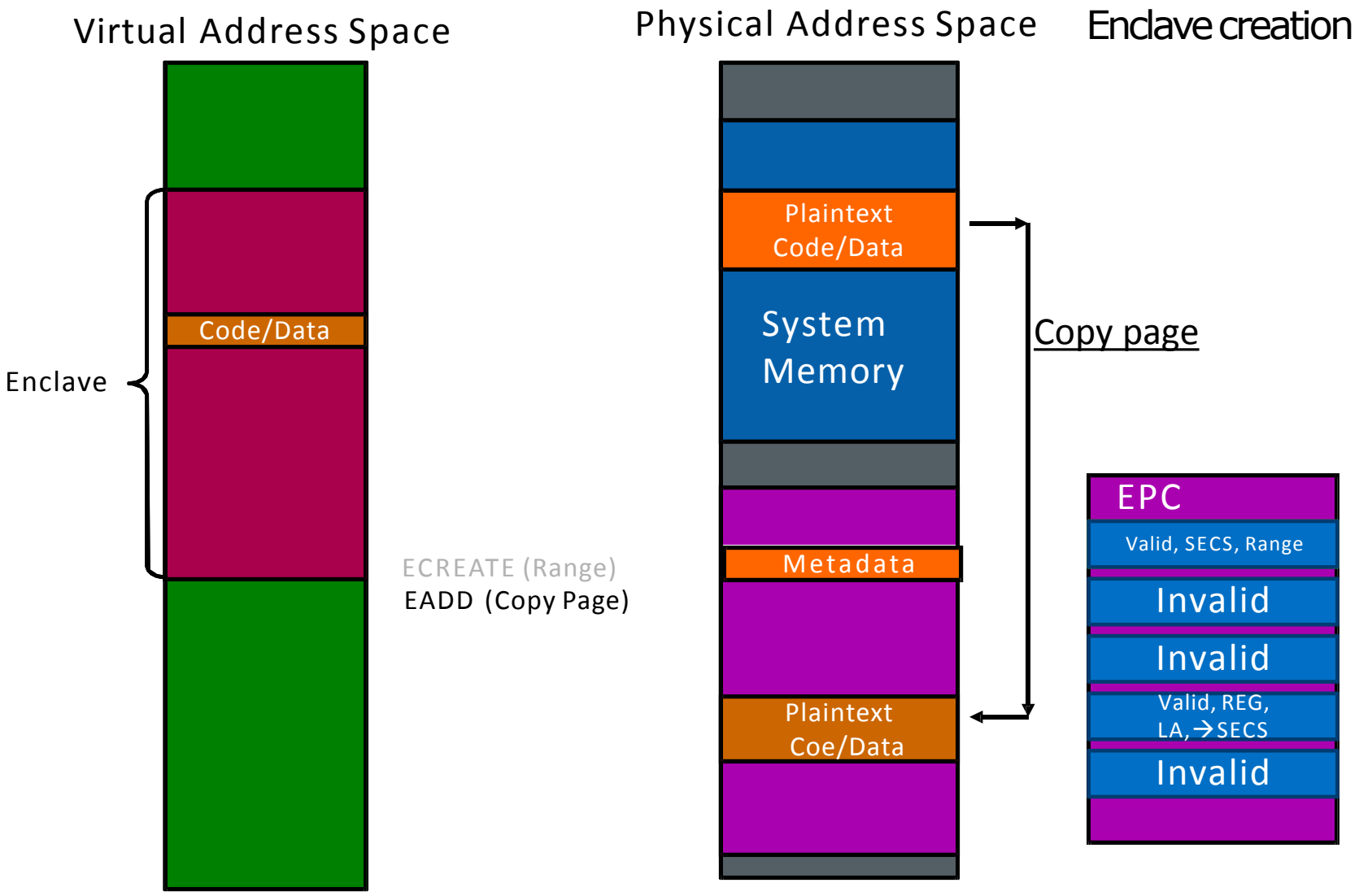
BIOS reserves memory address range for SGX use.

Enclave Life Cycle



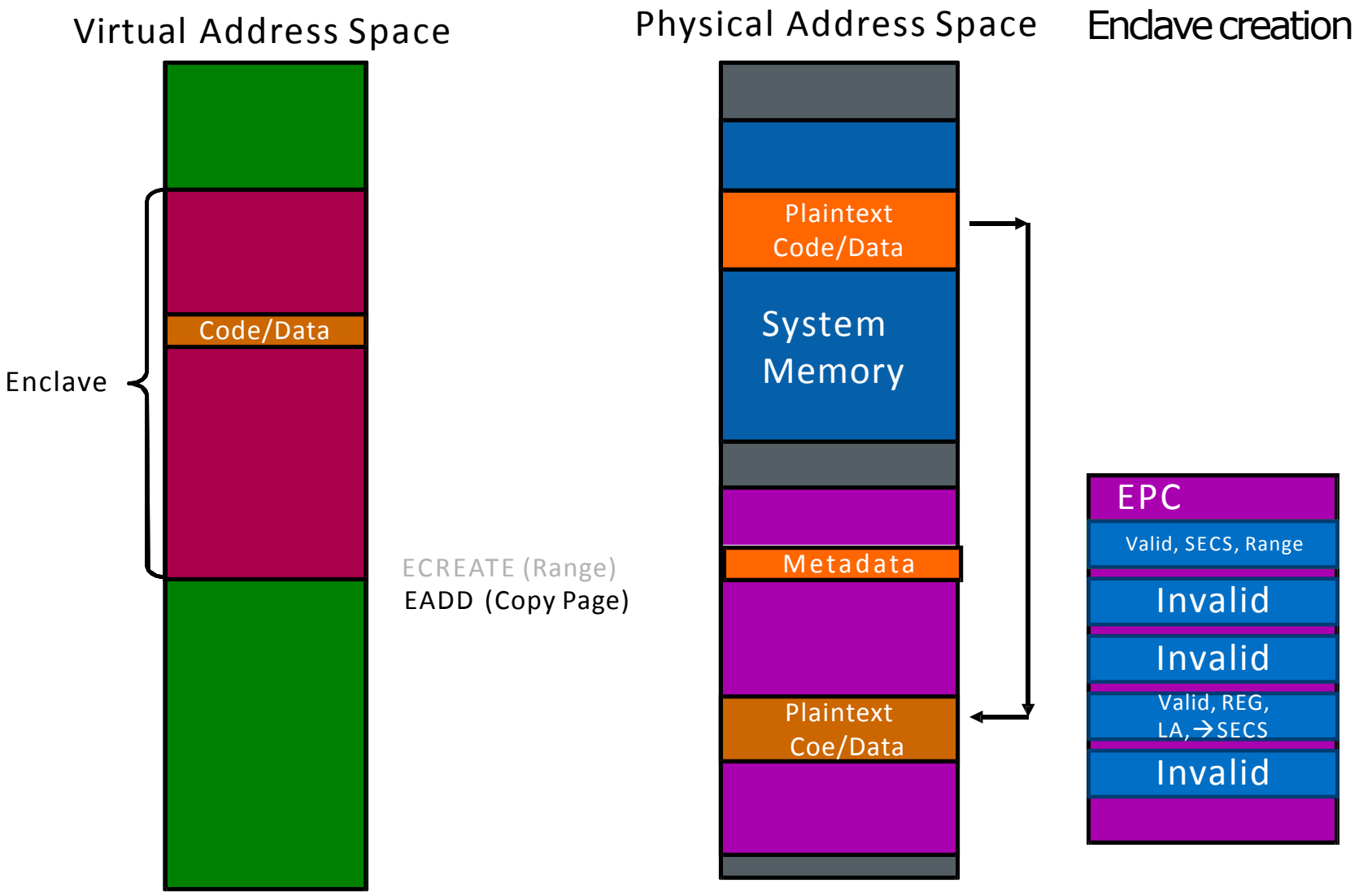
ECREATE - Stores enclave attributes (mode of operation, debug, etc.) in metadata.

Enclave Life Cycle



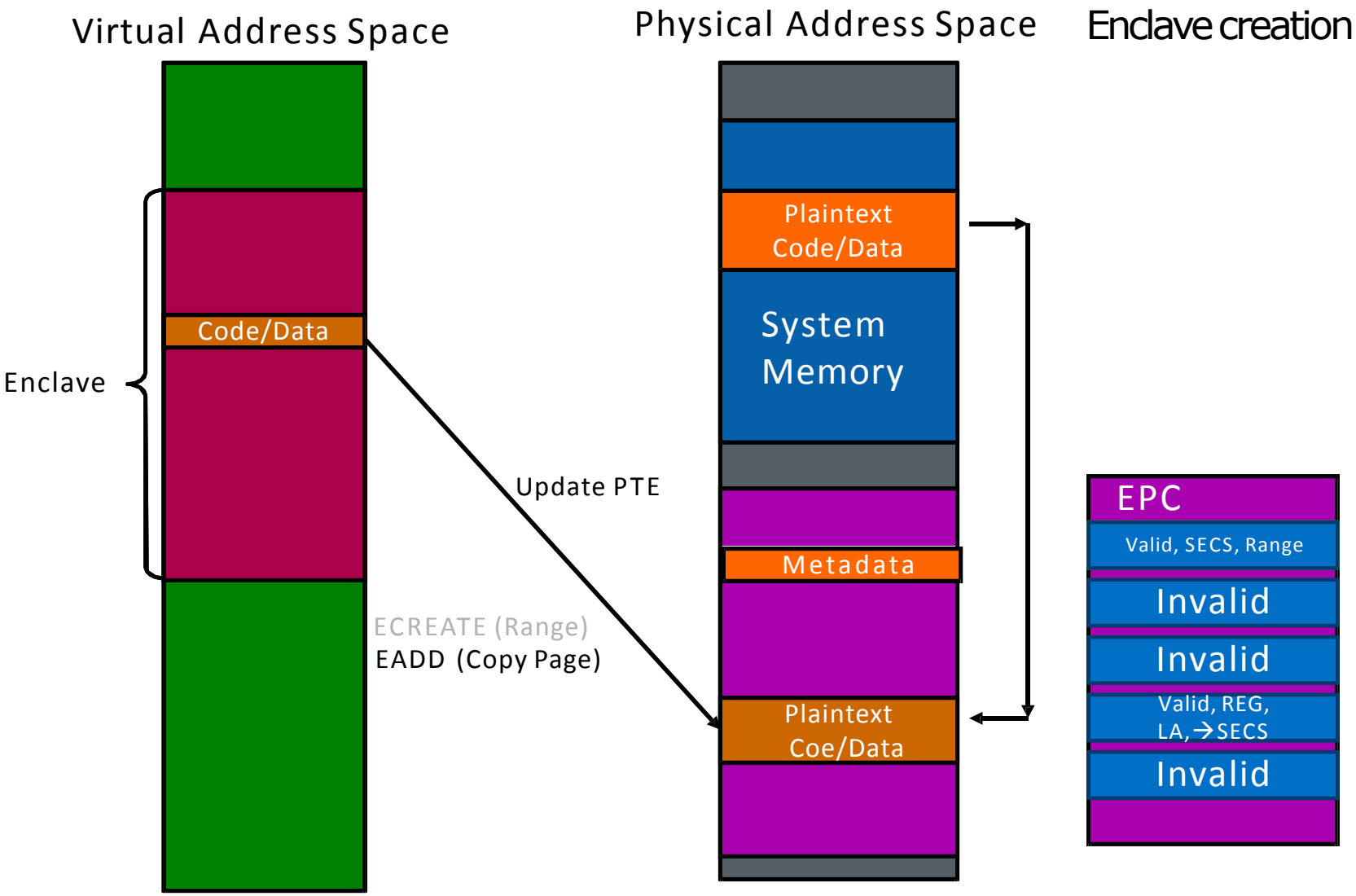
EADD – Commits new pages to enclave & updates security metadata.

Enclave Life Cycle

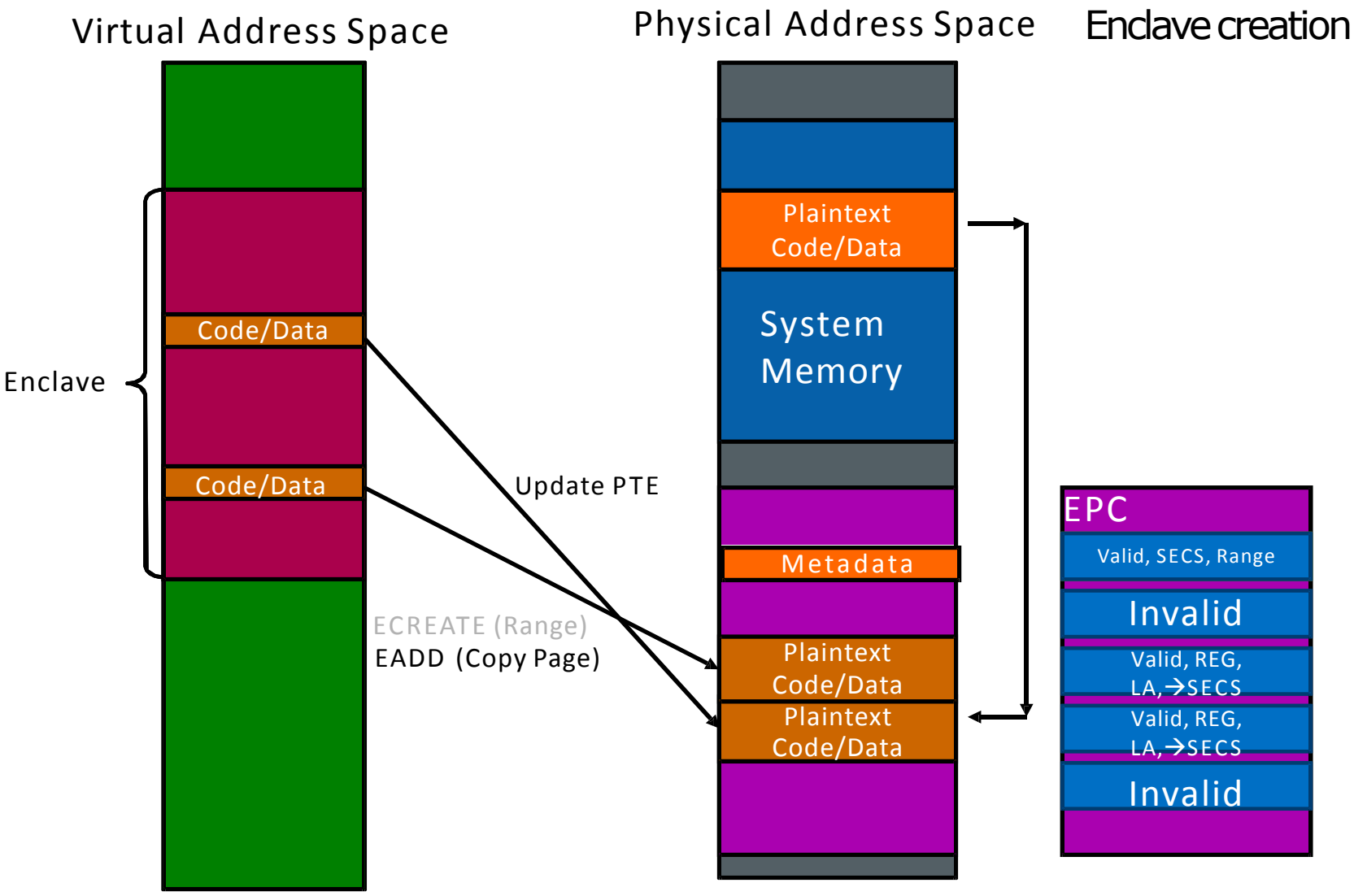


EADD – Commits code, data or SGX control structure page types.

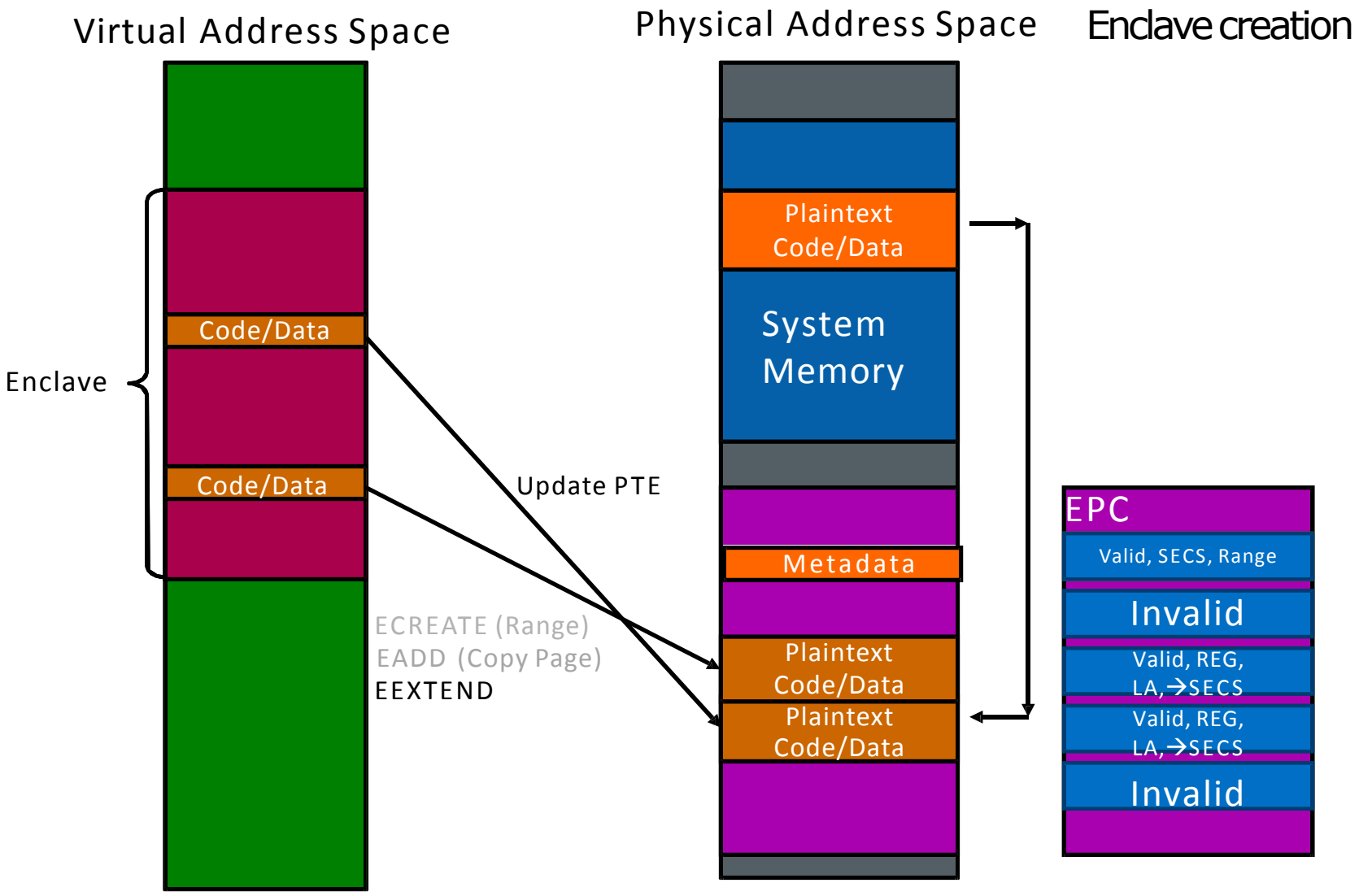
Enclave Life Cycle



Enclave Life Cycle

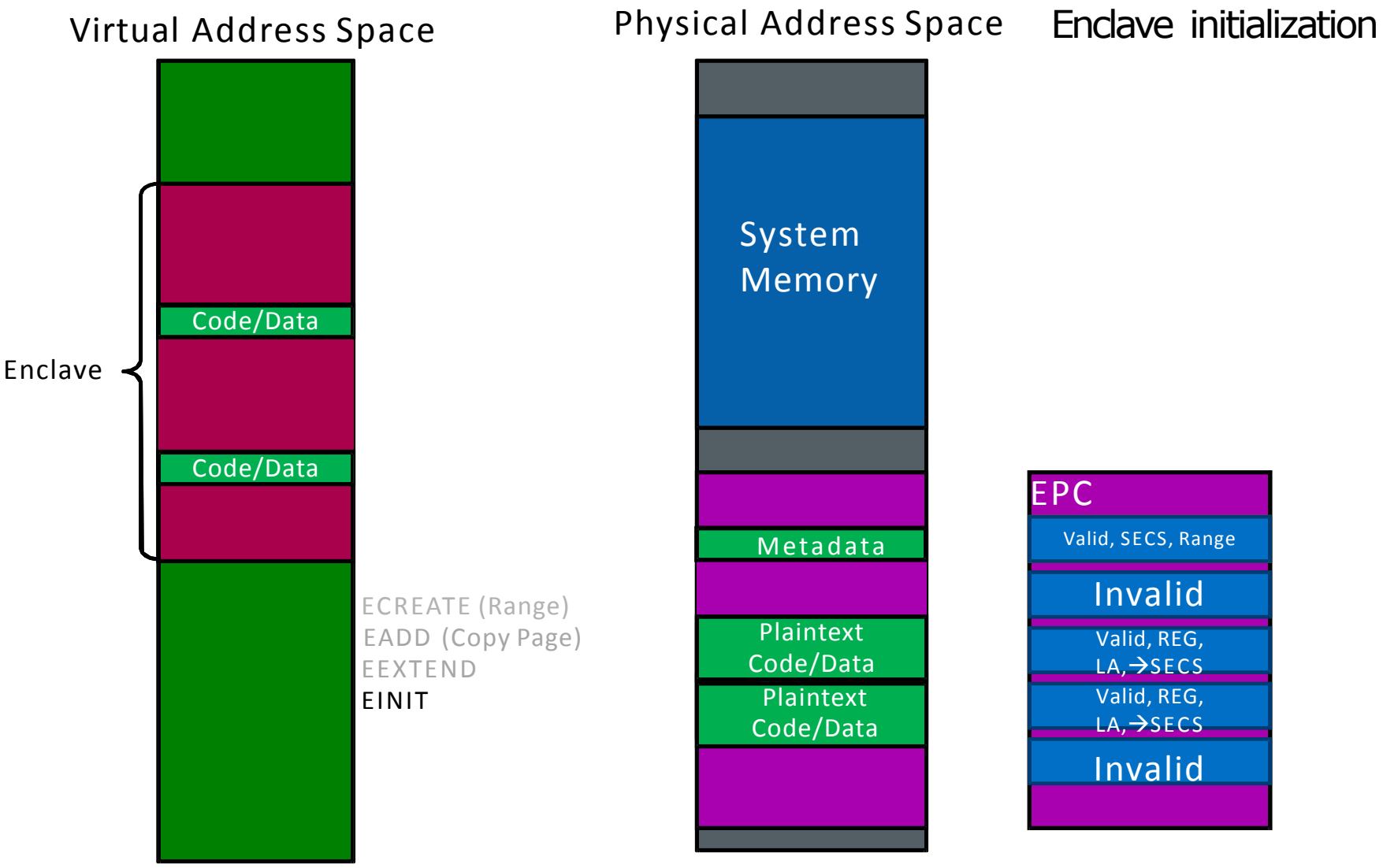


Enclave Life Cycle



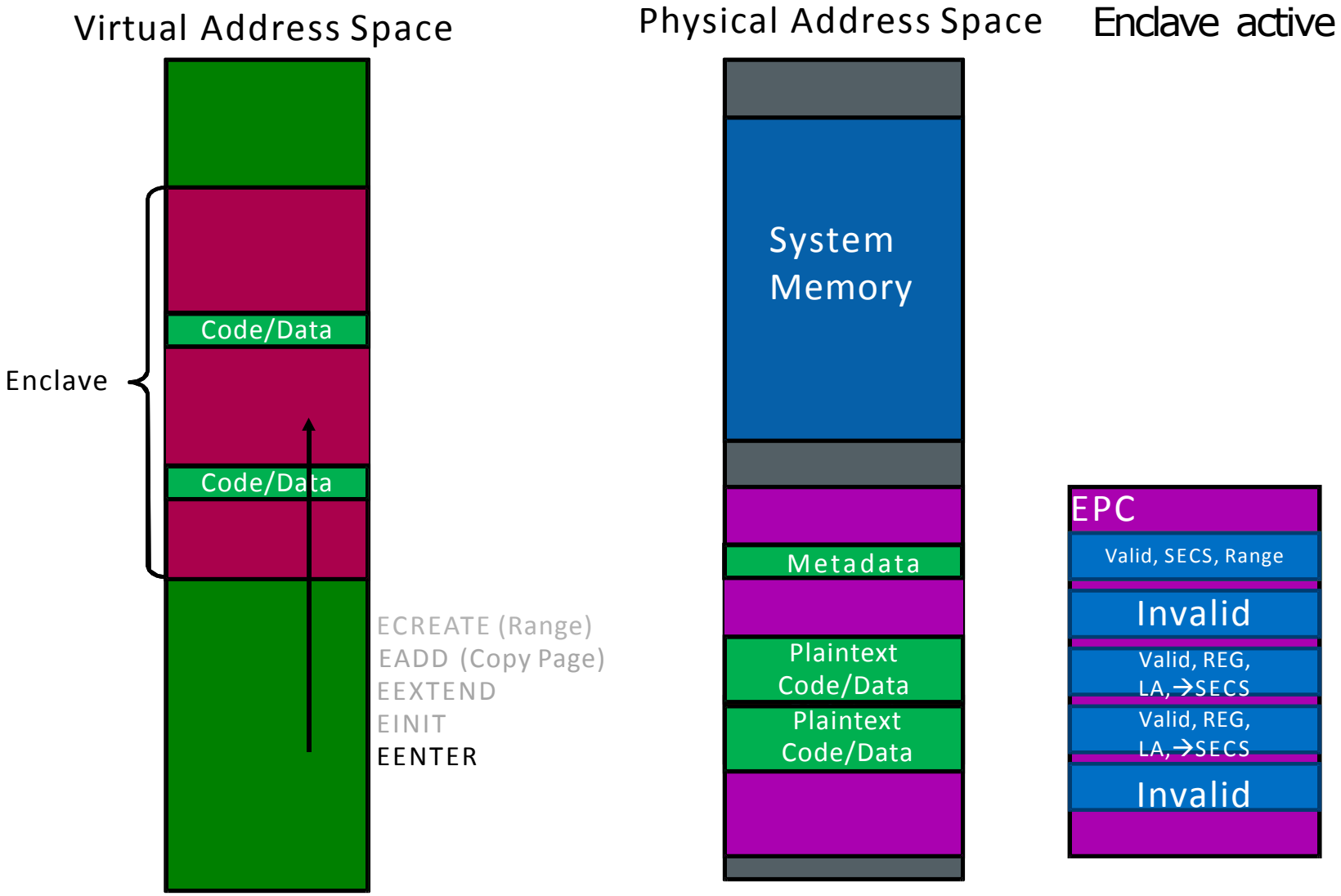
EEXTEND - Measures the enclave with SHA256. (detailed shortly)

Enclave Life Cycle



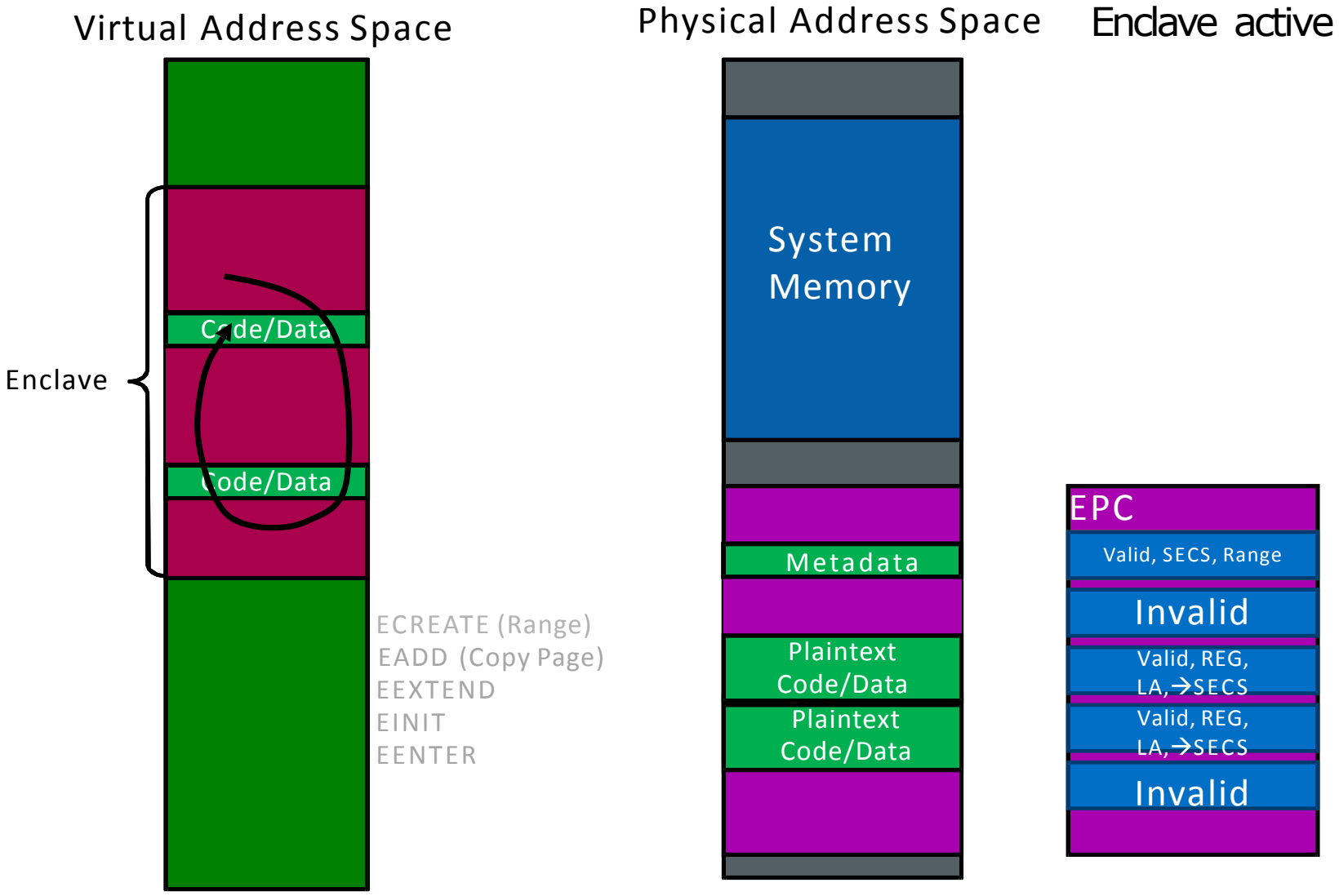
EINIT - Finalizes measurements, validates them & enables enclave's entry use.

Enclave Life Cycle



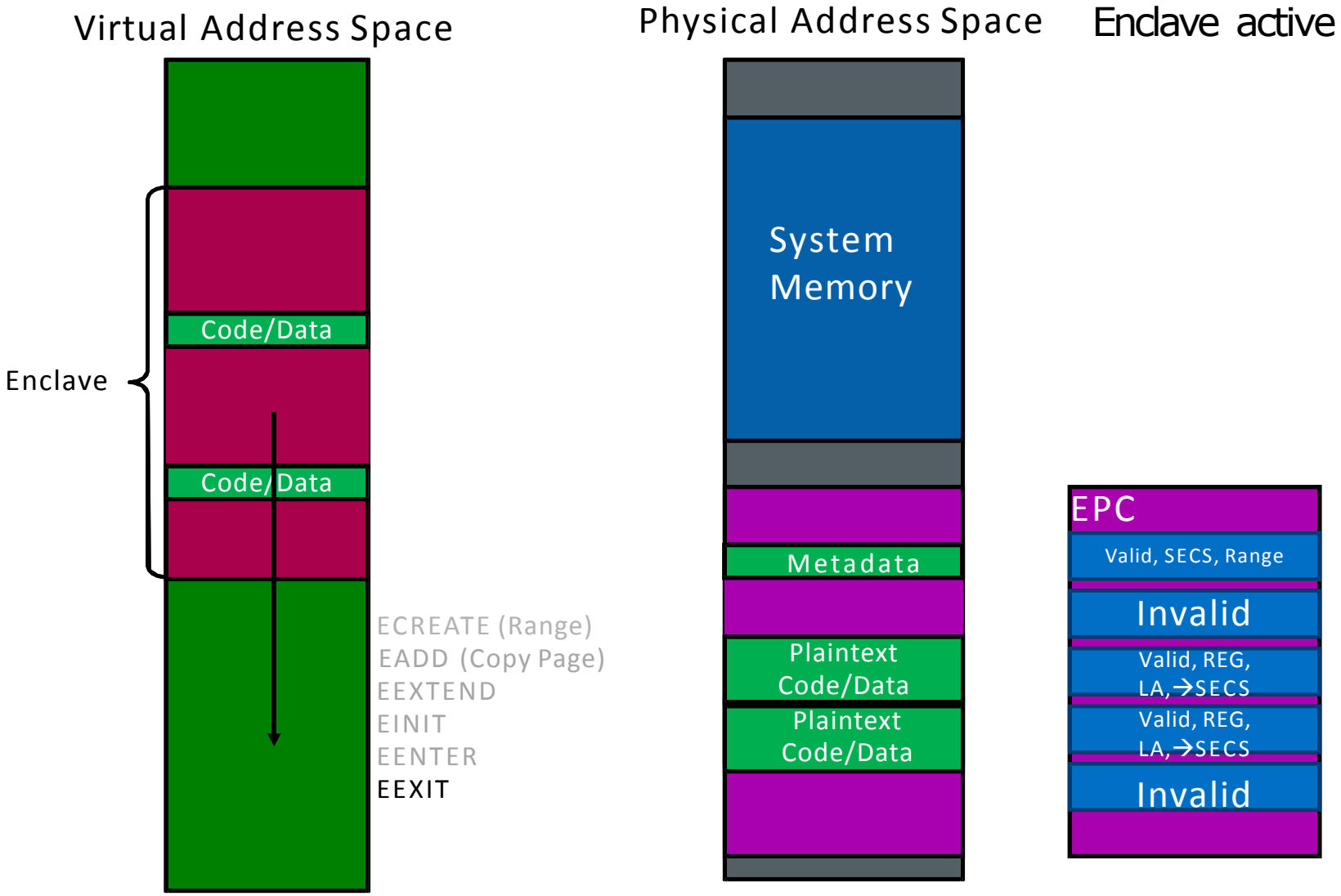
EENTER - Verifies enclave entry & sets CPU operation mode to “enclave mode”.

Enclave Life Cycle



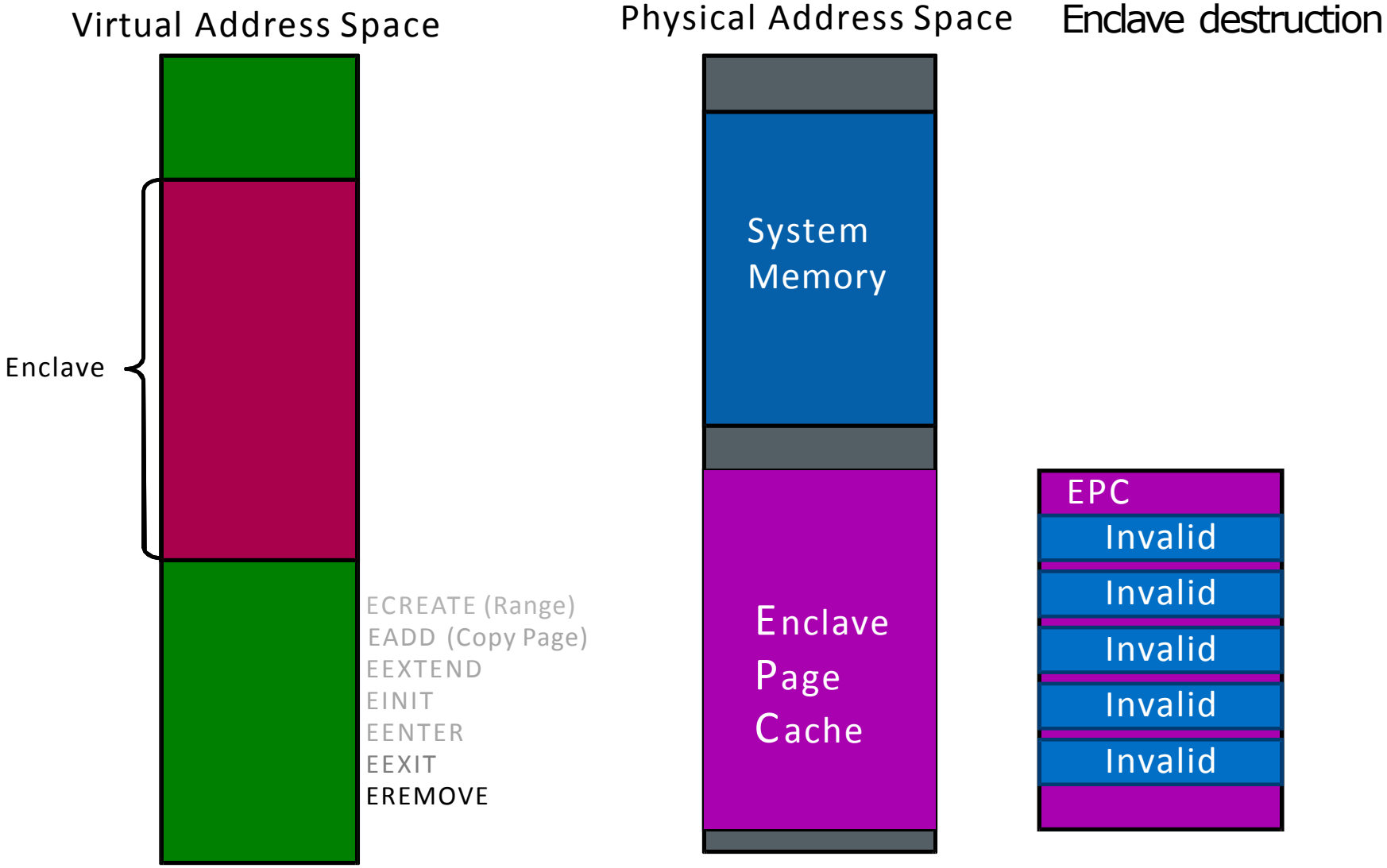
Enclave flow is executed using the secured address range, obscured from **all** other SW.

Enclave Life Cycle



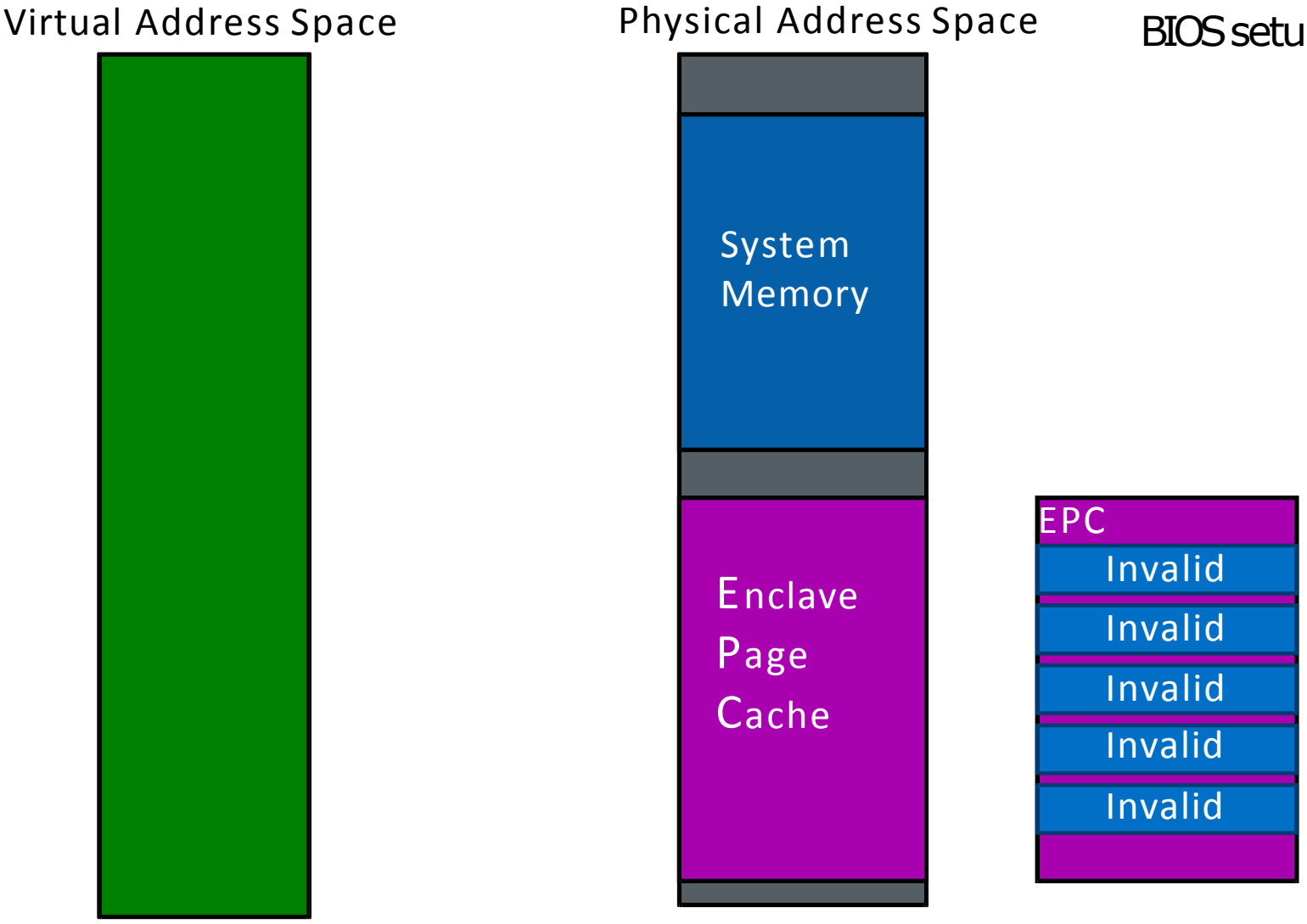
EEXIT – Clears CPU cache & Jumps out of enclave back to OS instruction address.

Enclave Life Cycle



EREMOVE – Clears enclave's trusted virtual address range reserved by ECREATE.

Enclave Life Cycle



Making use of SGX Enclave

Independent software vendor (ISV) enclave development process.

1. Identify sensitive parts of application.

2. Move sensitive logic to enclave project and compile.
    1. Multi-threaded enclave is supported by creating multiple TCS EPC entries.

3. Create accompanying enclave certificate (SIGSTRUCT)
    1. Measure enclave content using SHA-256
    2. Specify attributes
    3. Set ISV information
        1. Product ID
        2. Security version number – SVN
    4. Sign with ISV private key.

4. An enclave is in the clear before instantiation
    1. Sections of code and data could be encrypted, but their decryption key can't be pre-installed.
    2. Secrets come from outside the enclave after it satisfies 3rd part's trust.
    3. Subsequent runs use provisioned secrets again.

Enclave creation process

Loads enclave binary into the EPC and establishes identity.

1. ECREATE
    1. Finds a free EPC page and makes it the Enclaves SECS
    2. Stores enclave initial attributes (mode of operation, debug, etc.)
2. EADD
    1. Commits information (REG) or TCS as a new enclave EPC entry pages (4KB at a time)
    2. Creates and updates associated EPCM entry.
3. EEXTEND
    1. Measures 256Byte of info. into CryptoLog (thus runs 16 times to measure one page).
4. EINIT - ensures only measured code has enclave access!
    1. Gets as input: SIGSTRUCT, EINITTOKEN
        1. EINITTOKEN is generated using this CPU's Launch key, and contains: MRENCLAVE, MRSIGNER & ATTRIBUTES.

        (These values must match the corresponding values in the SECS)
    2. Verifies and creates enclave identities:
        1. Finalizes measurements for MRENCLAVE.
        2. Validates SIGSTRUCT with enclosed public key
        3. Uses ISV's signed SIGSTRUCT to validate MESIGNER and MRENCALVE.
        4. Checks that no Intel-only bits are set in SIGSTRUCT.ATTRIBUTES unless SIGSTRUCT was signed by Intel.
        5. Stores these and some other attributes in enclave's SECS.
    3. Enables enclave entry (mark as ready to be used)

Enclave Entry and Exit execution

CPU flow control under enclave mode

1. EENTER
    1. Gets enclave TCS address as parameter
    2. Verifies validity of enclave entry point
    3. Check that TCS is not busy and marked it as “BUSY”.
    4. Clears any translation remainder in cache (TLS)
    5. Sets Asynchronous Exit Pointer (AEP) parameter
    6. Change CPU mode of operation to “enclave mode”.
    7. If not debug, set HW so the enclave appears as a single instruction.
2. ERESUME
    1. AEP will mostly hold the ERESUME inst. address.
3. EEXIT
    1. Clear any translation remainder in cache (TLB)
    2. TCS is marked as “FREE”
    3. Jumps out of enclave flow back to OS instruction address (stored in RBX)
    4. Enclave developer is responsible for clearing registers state.