Power Transitions in Intel® Software Guard Extensions (Intel® SGX) Applications for Windows

Scope

This article provides guidelines on handling power transitions for Intel® Software Guard Extension (Intel® SGX) enabled applications running on Microsoft* Windows*. General information on Intel SGX can be found on the Intel SGX portal at https://software.intel.com/en-us/sgx.

Exception Handling and Power Handling

In today's mobile computer systems, maximizing battery life is a top priority. Minimizing device power consumption in idle scenarios and eliminating it entirely in standby scenarios are fundamental objectives toward realizing this goal of longer battery life.

To realize these system-power objectives, the Microsoft* Windows* OS supports power states such as S0 to S5 and S0ix (connected standby) states. When the system resumes from a lower power state to the working power state (S0), an application should resume where it left off. To achieve this objective, the application must first store its context information before going to a lower power state or sleep state (S1 through S4). In a similar way, the context information for the application must be restored when returning to a working power state (S0).

Modern operating systems provide mechanisms to enable applications to be notified of major power events on the platform. When the computer enters a lower power state, the OS suspends to RAM or saves to disk context information for future restoration.

For Intel SGX, power transitions from an S0/S1 state to an S2-S5 state cause the protected memory encryption key for an enclave to be destroyed. This makes the enclave effectively unreadable; therefore, it must be recreated on a system resume. Enclaves that need to preserve secrets across S2-S5 power states must save their state information to a disk.

However, applications need to work around two situations to accomplish this goal:

- The Intel SGX architecture does not provide a means of directly messaging power-transition events into enclaves. So applications register callback functions for such events. When a callback function is invoked, the application can call the enclave specifically to save the secret state to disk.

- The OS does not guarantee that the enclave will be given enough time to seal its entire internal state. So enclaves that need to preserve state across power transitions must periodically seal enclave-state data outside the enclave (that is, to a disk or cloud) in anticipation of a future power transition.

Upon re-instantiation of the application, enclaves are subsequently rebuilt from scratch. Applications must retrieve their protected states from the disk or cloud. To minimize the overhead caused by constantly sealing secrets and storing the encrypted data to a disk or cloud, the enclave writer should design their application enclave to keep as little state information as possible inside the enclave so that the application can effectively manage a power-transition event. The less state information stored inside the enclave, the quicker the enclave will be able to backup this information outside the enclave and to recover from a power transition.

Methodology (Power Transition)

Windows provides APIs for applications to receive and handle power-event notifications from the OS. Windows applications can handle the following the power events:

- When a power source changes, such as transitioning between alternating-current (AC) and direct-current (DC) power
- When a battery's remaining charge level reaches its minimum threshold
- When the OS resumes from sleep mode
- When the OS requests low-power mode

The OS notifies applications of these power events by using the `WindowProc` callback function. However, the application running in Intel SGX enclaves adopts a different approach, one based on a specific error code returned.

In this approach, during an enclave call (ECALL) to process the secret, if a power transition occurs that caused the enclave to be lost, the ECALL to the enclave returns the error code: SGX_ERROR_ENCLAVE_LOST. The application running within the Intel SGX enclave identifies, via the error code, that a power transition has already happened. This means that to continue to process the secret, the enclave must be rebuilt. Because the enclave has already incrementally sealed its data before the power cycle occurred, the newly built enclave can retrieve the sealed data.

Power-Transition Handling with Intel SGX

Power transitions can happen while processing two types of ECALLs in applications running in Intel SGX enclaves. These are:

- An initialization ECALL after enclave creation
- A normal ECALL to manipulate secrets within the enclave

Initialization ECALL after Enclave Creation

Figure 1 shows the flow for handling of a power transition during initialization ECALL after enclave creation. Figure 2 shows how these calls are handled in both the untrusted and trusted code parts of applications.

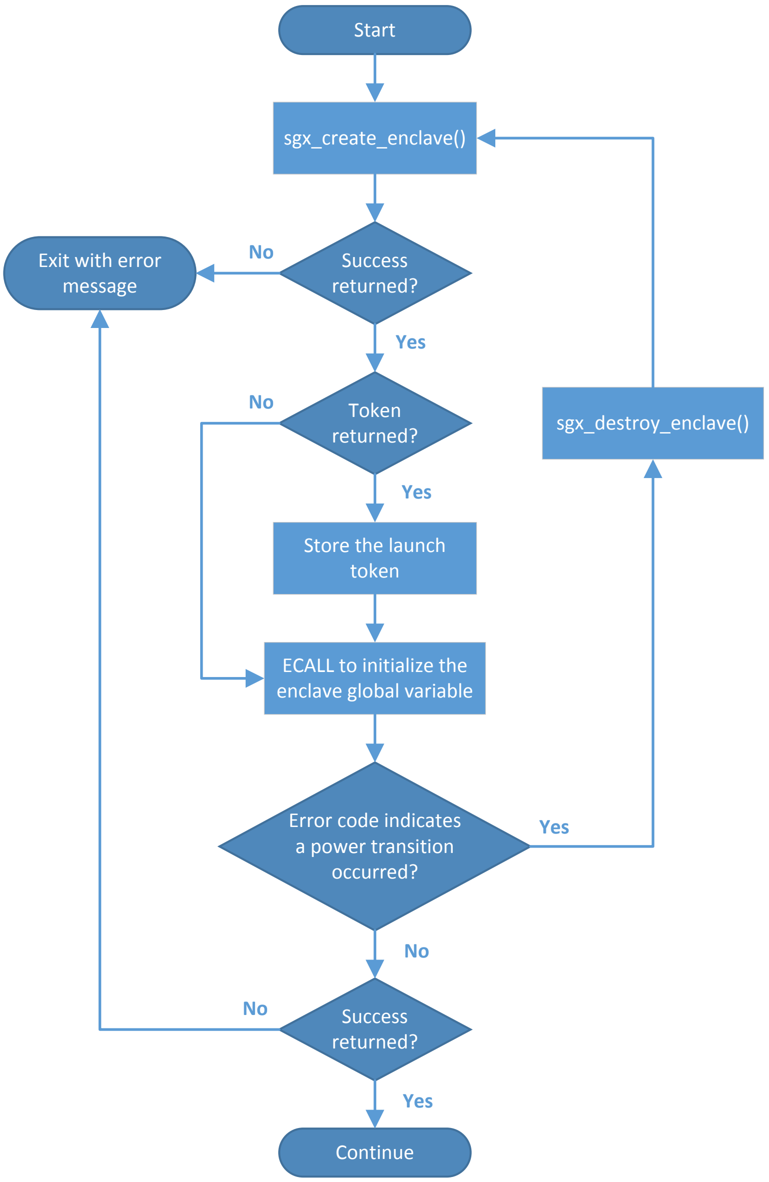
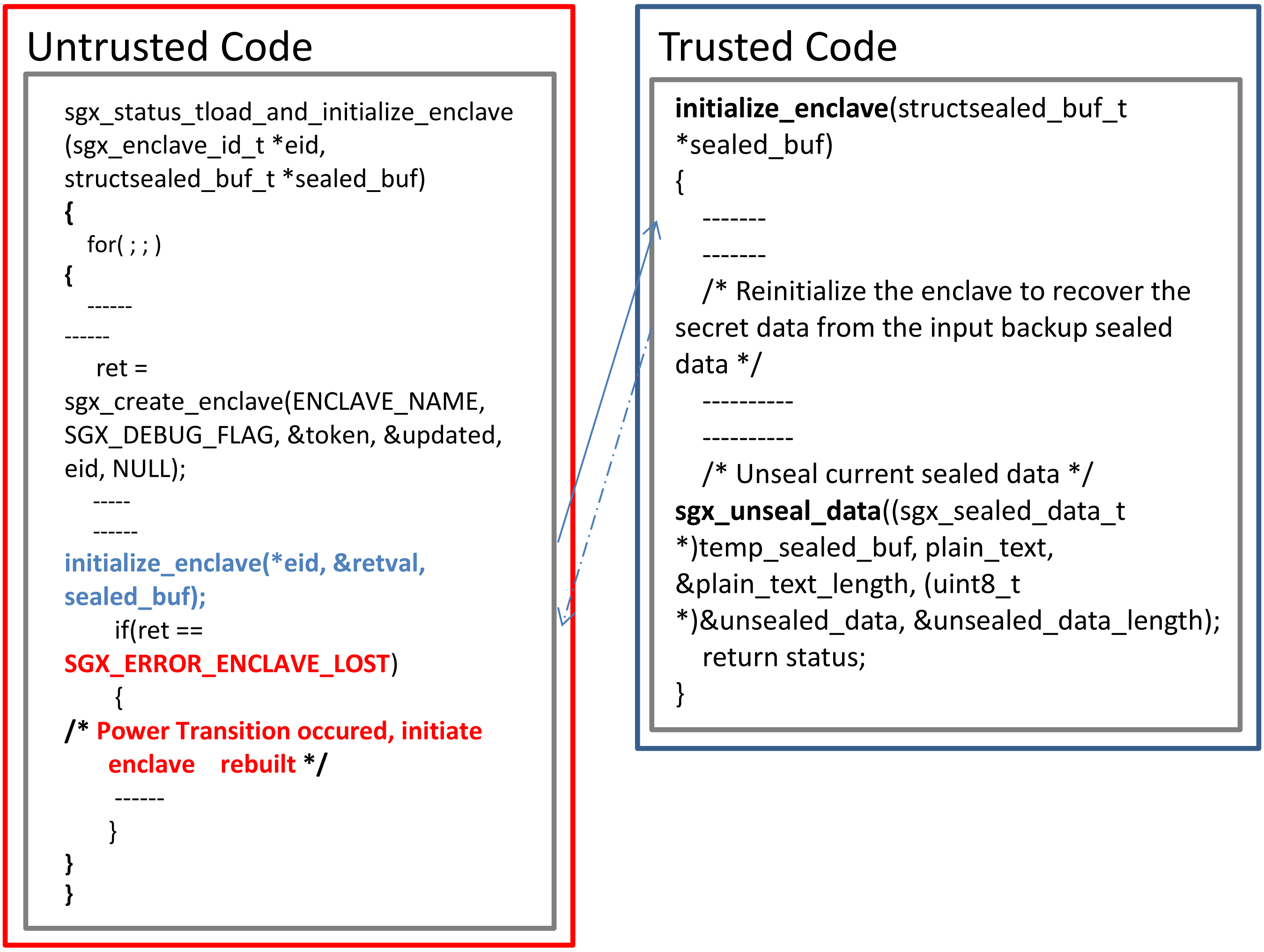


Figure 1. Operations that take place in the course of a power transition during enclave initialization



Untrusted Code

```
sgx_status_tload_and_initialize_enclave
(sgx_enclave_id_t *eid,
structsealed_buf_t *sealed_buf)
{
   for( ; ; )
{
   ------
------
    ret =
sgx_create_enclave(ENCLAVE_NAME,
SGX_DEBUG_FLAG, &token, &updated,
eid, NULL);
    -----
    ------
initialize_enclave(*eid, &retval,
sealed_buf);
      if(ret ==
SGX_ERROR_ENCLAVE_LOST)
      {
/* Power Transition occured, initiate
     enclave    rebuilt */
      ------
     }
}
}
```

Trusted Code

```
initialize_enclave(structsealed_buf_t
*sealed_buf)
{
   -------
   -------
   /* Reinitialize the enclave to recover the
secret data from the input backup sealed
data */
   ----------
   ----------
   /* Unseal current sealed data */
sgx_unseal_data((sgx_sealed_data_t
*)temp_sealed_buf, plain_text,
&plain_text_length, (uint8_t
*)&unsealed_data, &unsealed_data_length);
   return status;
}
```

Figure 2. Code called in the course of a power transition during enclave initialization

There is no need to handle power transitions in the `sgx_create_enclave` function in Intel SGX application code. This is because power-transition handling is already implemented in the uRTS (untrusted Run-time System) library. When error code SGX_ERROR_ENCLAVE_LOST is returned during enclave initialization, the Intel SGX application identifies that the power transition has happened. The Intel SGX application subsequently destroys the existing enclave and rebuilds the enclave.

Normal ECALL to Process Secrets within the Enclave

Figures 3 shows the flow for handling a power transition for the most common ECALL type into an enclave. Figure 4 shows how this call is handled in both the untrusted and trusted code parts of applications.

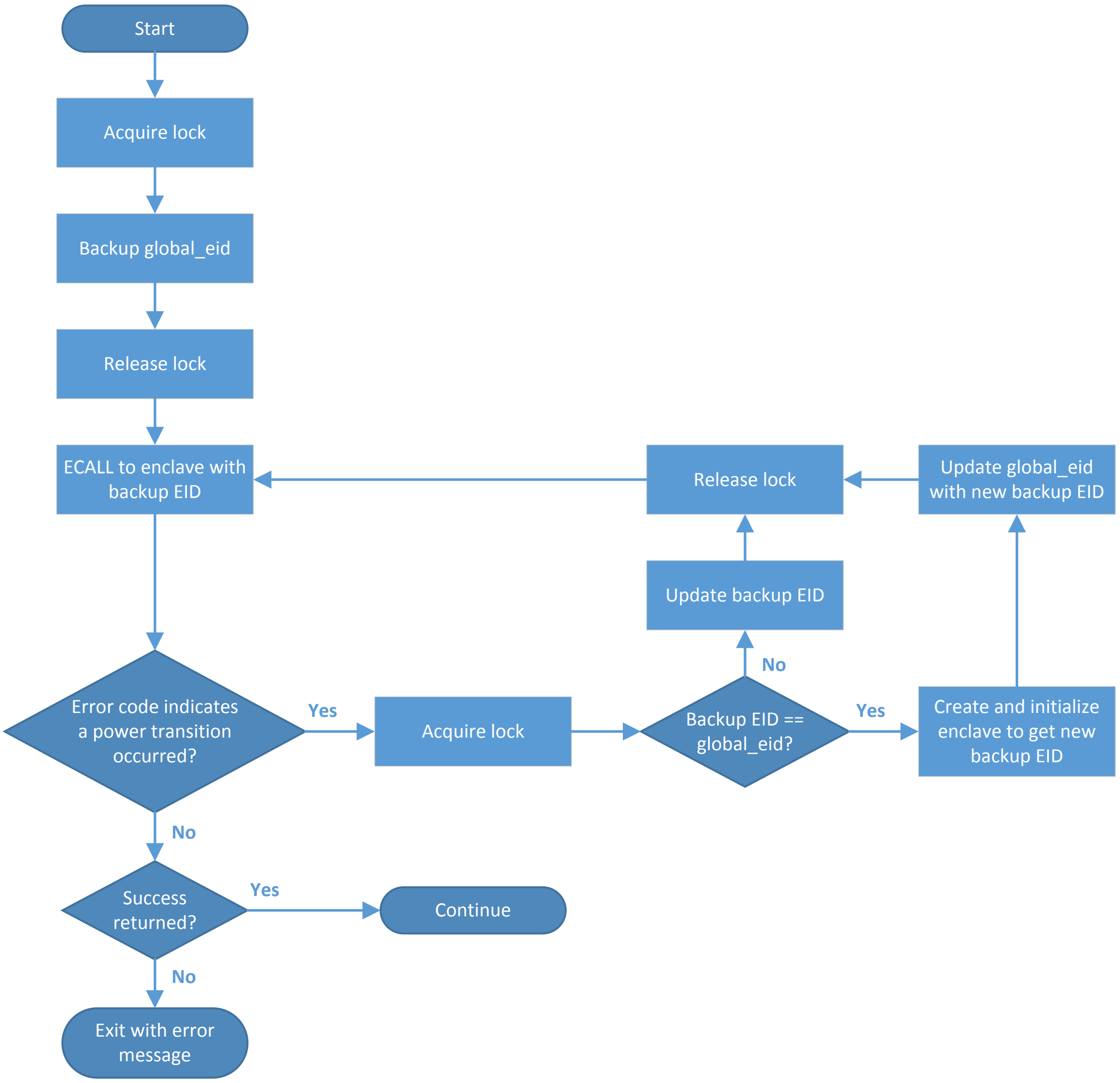
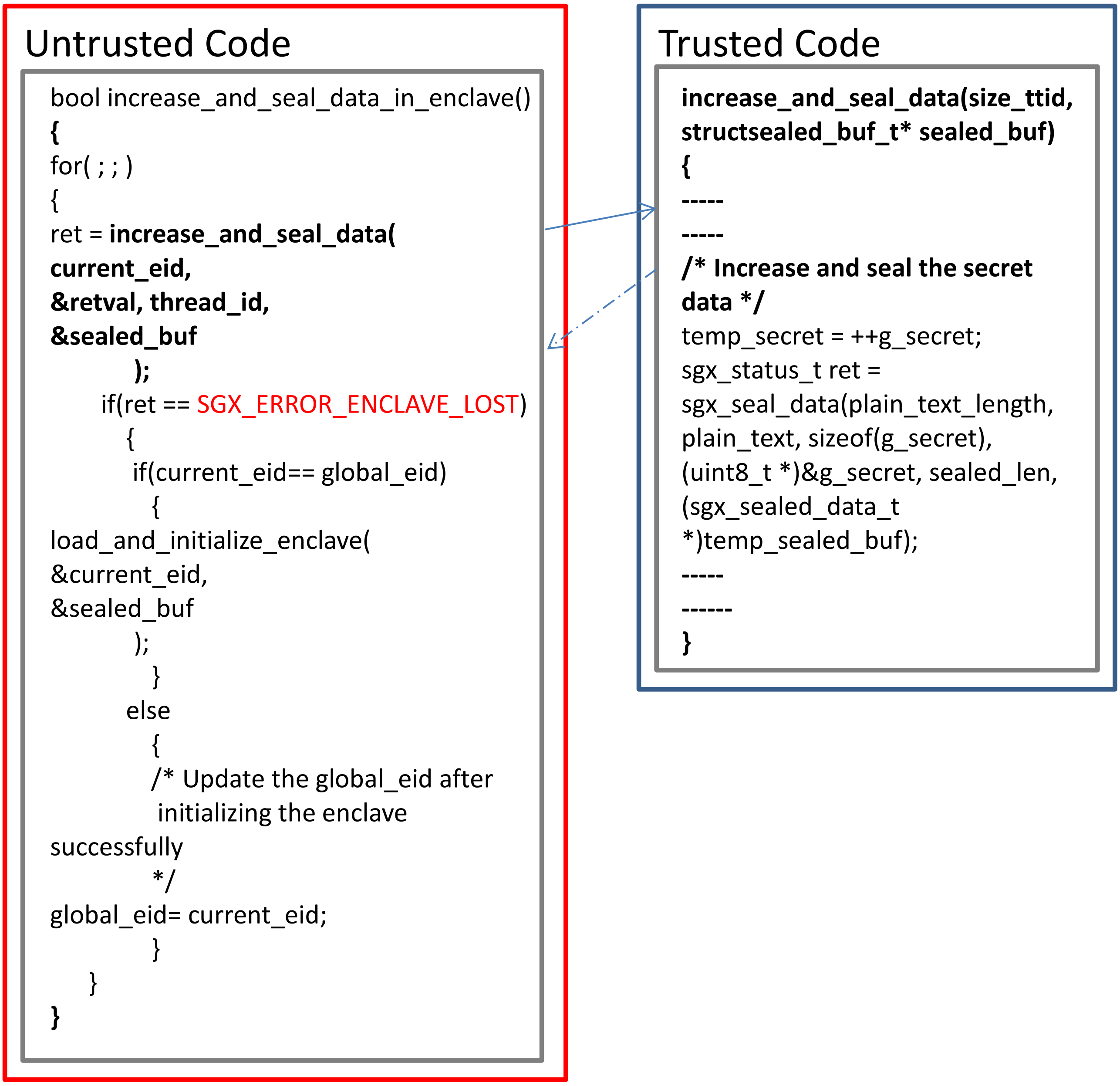

Start
Acquire lock
Backup global_eid
Release lock
ECALL to enclave with backup EID
Release lock
Update global_eid with new backup EID
Update backup EID
No
Error code indicates a power transition occurred?
Yes
Acquire lock
Backup EID == global_eid?
Yes
Create and initialize enclave to get new backup EID
No
Success returned?
Yes
Continue
No
Exit with error message

Figure 3. Operations that take place in the course of a power transition during ECALL to process secrets



```
bool increase_and_seal_data_in_enclave()
{
for( ; ; )
{
ret = increase_and_seal_data(
current_eid,
&retval, thread_id,
&sealed_buf
          );
      if(ret == SGX_ERROR_ENCLAVE_LOST)
         {
          if(current_eid== global_eid)
            {
load_and_initialize_enclave(
&current_eid,
&sealed_buf
          );
            }
         else
            {
            /* Update the global_eid after
             initializing the enclave
successfully
            */
global_eid= current_eid;
            }
     }
}
```

```
increase_and_seal_data(size_ttid,
structsealed_buf_t* sealed_buf)
{
-----
-----
/* Increase and seal the secret
data */
temp_secret = ++g_secret;
sgx_status_t ret =
sgx_seal_data(plain_text_length,
plain_text, sizeof(g_secret),
(uint8_t *)&g_secret, sealed_len,
(sgx_sealed_data_t
*)temp_sealed_buf);
-----
------
}
```

Figure 4. Code called in the course of a power transition during ECALL to process secrets

In the code example, the Intel SGX application has one main thread and two or more child threads. The main thread creates and initializes the enclave the first time. It handles the power transition that occurs during loading and initialization. Each child thread is then responsible for issuing normal ECALLs to process the secret. If, during an ECALL, the application identifies a power transition and the ECALL returns SGX_ERROR_ENCLAVE_LOST, the application creates and initializes a new enclave and retrieves the sealed data.

Once the new enclave is created, the Enclave ID (EID) global variable is updated. The EID is used by any thread that needs to issue an ECALL for processing the secret. This global EID can be updated by any one of the running child threads when the new enclave is created.

When the SGX_ERROR_ENCLAVE_LOST error code is returned, the application knows that a power transition has occurred. If the current EID and the stored global EID are equal, then the current thread rebuilds the enclave, updates the global EID with the newly obtained EID, and unseals the stored data. Otherwise, if there is a mismatch between the global EID and current EID, then the application knows that:

- Another thread encountered a power transition
- The other thread already rebuilt the enclave
- There is no need to rebuild the enclave
- The current EID update (with the updated global EID) is correct

Summary

Intel SGX applications cannot depend on Windows power-transition event notification for sealing secret data, because the OS cannot guarantee that it can give enough time for a given enclave to seal its secret data to disk. And the `WindowProc` callback function cannot be used for notification. So Intel SGX applications uses a specific methodology for dealing with power-transition events, to achieve minimal data loss.

- Intel SGX applications identify the occurrence of power-transition events based on the error code SGX_ERROR_ENCLAVE_LOST. Because the error code returns only when a power-transition event has already occurred and the OS has resumed, an Intel SGX application's secret data must be sealed periodically by the enclave when the OS is running. This allows stored secret data to be retrieved from disk (or the cloud) after the enclave is rebuilt.
- Also, to minimize the overhead of regularly saving secret data to a disk or cloud, the secret data that an enclave stores should be kept minimal.

References

1. "Intel SGX SDK Users Guide for Windows OS" – 2016 Intel Corporation.
2. Intel. "Application Power Management for Mobility." – March 2002 Intel Corporation. https://software.intel.com/sites/default/files/m/4/2/6/apmm_wp.pdf.