LaGrande Architecture
SCMS - 18

David Grawrock
Security Architect
Intel

September 2003

Safer Computing Track – Fall IDF

Tuesday

- LT Overview
  SCMS-16
- TCG & TPM v1.2
  SCMS-17
- LT Architecture
  SCMS-18
- Tech Showcase
  Every Day
  Birds of a Feather Lunches
  Tuesday & Wednesday

Wednesday

- Privacy Method for Assuring Trust
  SCMS-19
- Opt-In Strategy
  SCMS-156
- Trusted Mobile KB Controller
  SCMS-24
- Software for LT
  SCMS-20
- Fundamentals for NGSCB
  SCMS-21
- Migrating Apps to NGSCB
  SCMS-22

Thursday

- TPM Recovery
  SCMS-25
- TCG Credentials
  SCMS-157
- TPM Mfg & Testing
  SCMS-180

= Overview
= Medium Technical
= Highly Technical

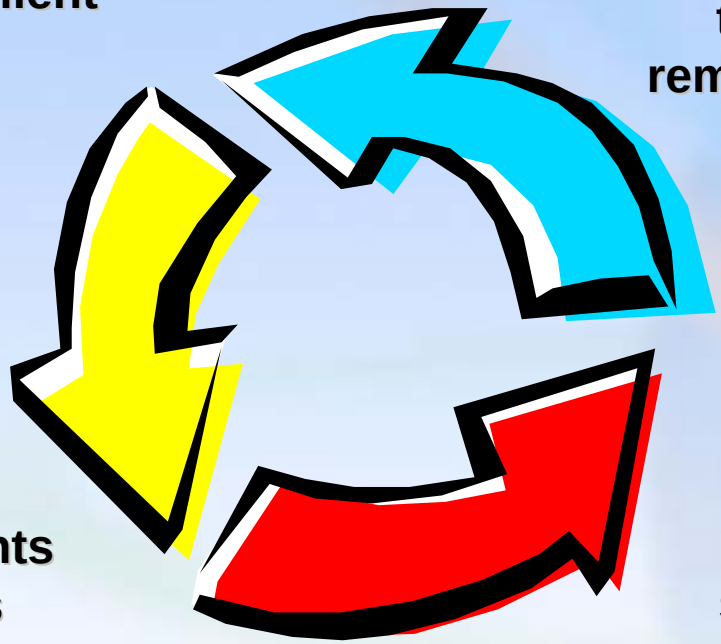
Agenda

- Security Opportunity
- Security Components
- Threat Definitions
- Hardware Solutions
- Break
- LT Architecture
- LT Features
- Threat Mitigations
- Secrets and Trust

Security Opportunity

High value of data created & stored on client

Financial incentive & technical means to remotely attack client PC



Ubiquitous connectivity (wired, wireless, remote)

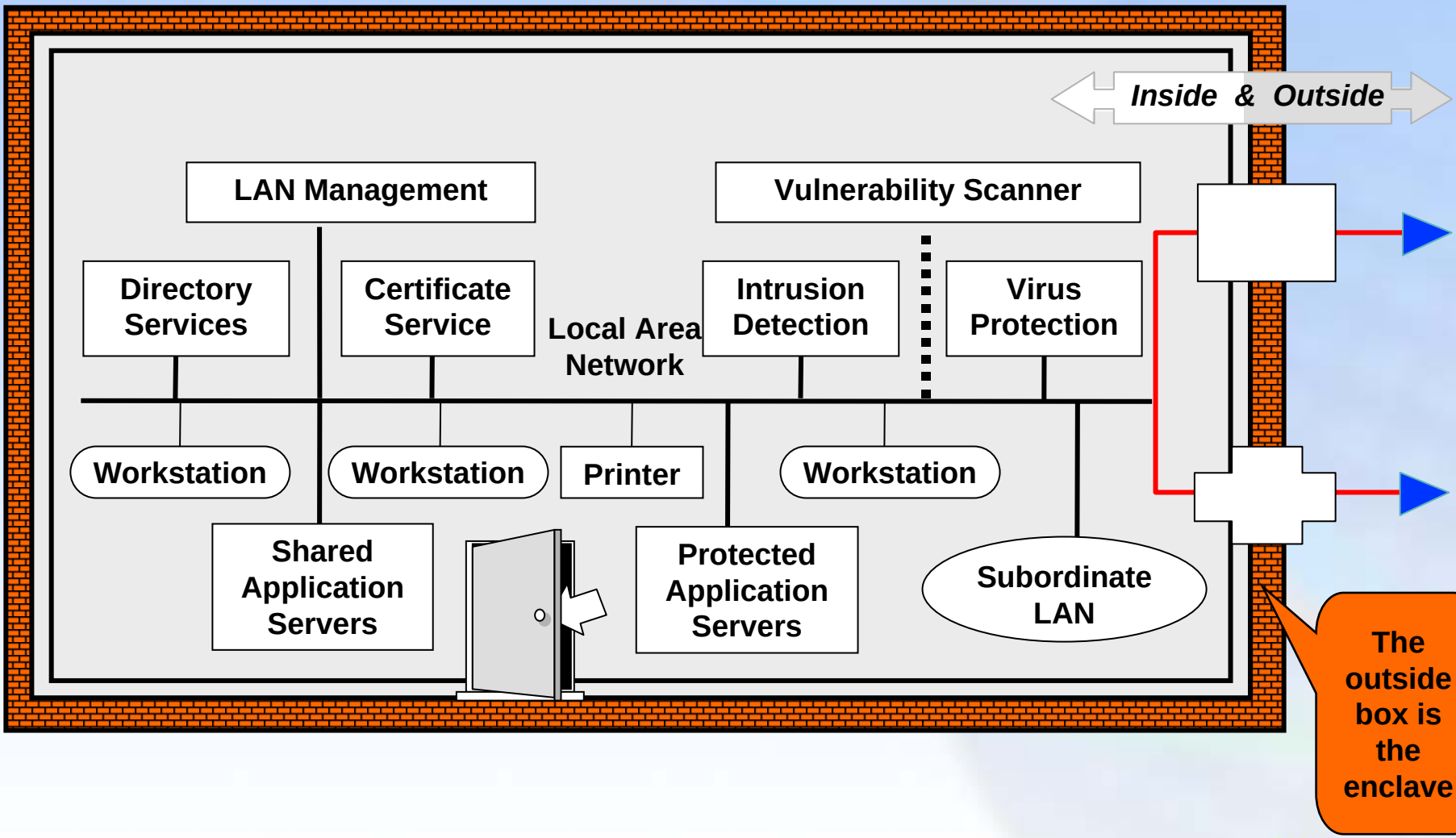
Increasing number of potential attackers

Potential attack points outpacing today's protection models

Increasing sophistication of readily available attack tools

A hardened client architecture can decrease the risk of serious financial loss due to compromised data

Local Computing Environment



Source: IATF Release 3.0

LaGrande Technology Objectives

Protect …	• Confidential data and communications • E-commerce transactions
From …	• Attack software on the system • Attack software on the network • Inadvertent exposure due to compromised software
Without compromising …	• Ease of Use • Manageability • Privacy • Performance • Versatility • Backwards compatibility

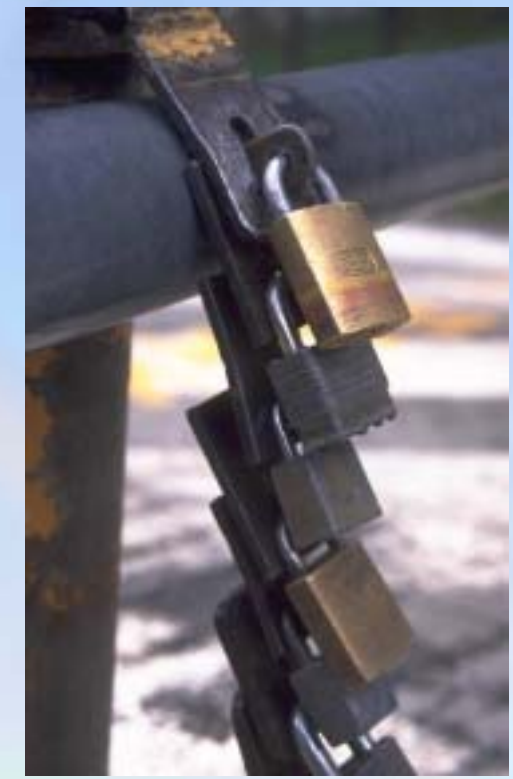
Terminology

- Trust
  - An entity can be trusted if it always behaves in the expected manner for the intended purpose

- Client
  - Any computing platform that users have access to and can initiate communication with other platforms

Security As A Component

- We are not providing standalone security
- We are providing key Intel Architecture building blocks



"There is more to life than increasing its speed."

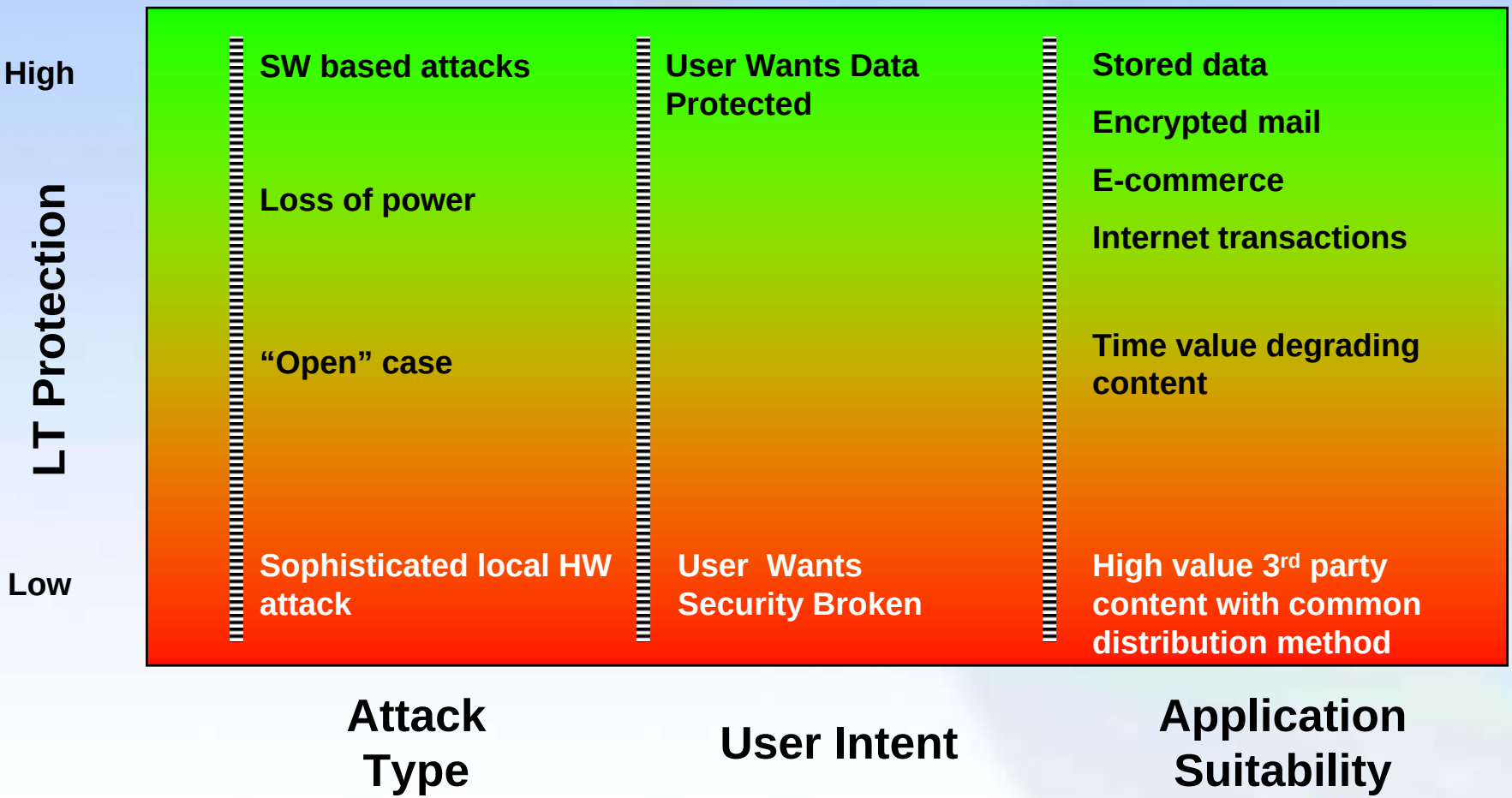
M. K. Gandhi

Greater data protection with the flexibility of PC computing

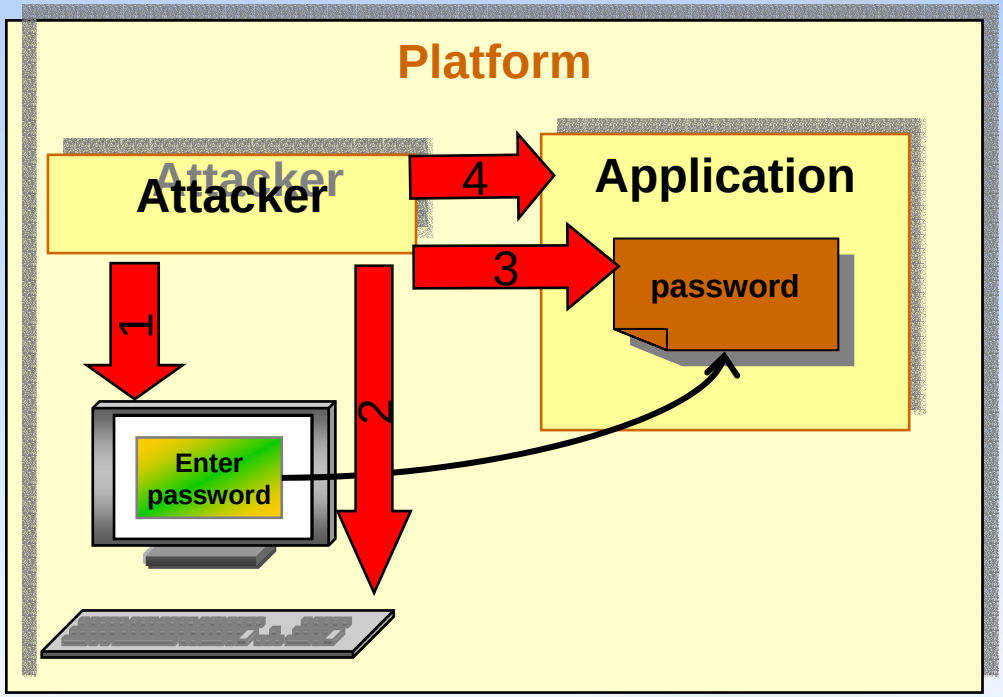
Agenda

- Security Opportunity
- Security Components
- Threat Definitions
- Hardware Solutions
- Break
- LT Architecture
- LT Features
- Threat Mitigations
- Secrets and Trust

Protection And Attack Matrix



Attacks on Login



Login process

- Display login screen
- User enters password
- Password validated
- Access granted

Attacks on application

1 Fake login screen

2 Sniff password from keyboard

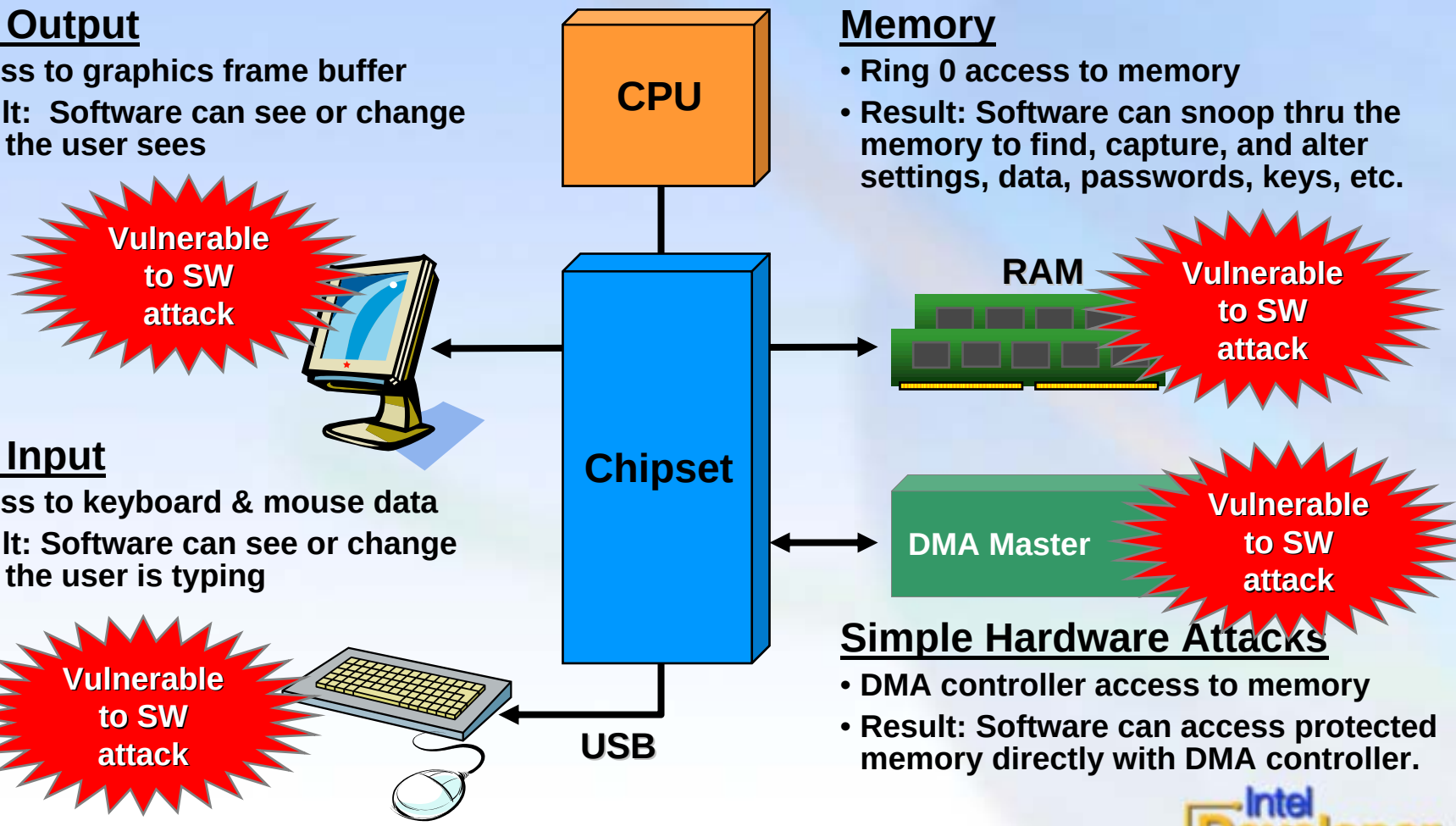
3 Read password in memory

4 Change application to ignore password entry

Vulnerabilities of the PC Today
Sample of Common Vulnerabilities

User Output
- Access to graphics frame buffer
- Result: Software can see or change what the user sees

Memory
- Ring 0 access to memory
- Result: Software can snoop thru the memory to find, capture, and alter settings, data, passwords, keys, etc.



User Input
- Access to keyboard & mouse data
- Result: Software can see or change what the user is typing

Simple Hardware Attacks
- DMA controller access to memory
- Result: Software can access protected memory directly with DMA controller.

Threat Model Recap

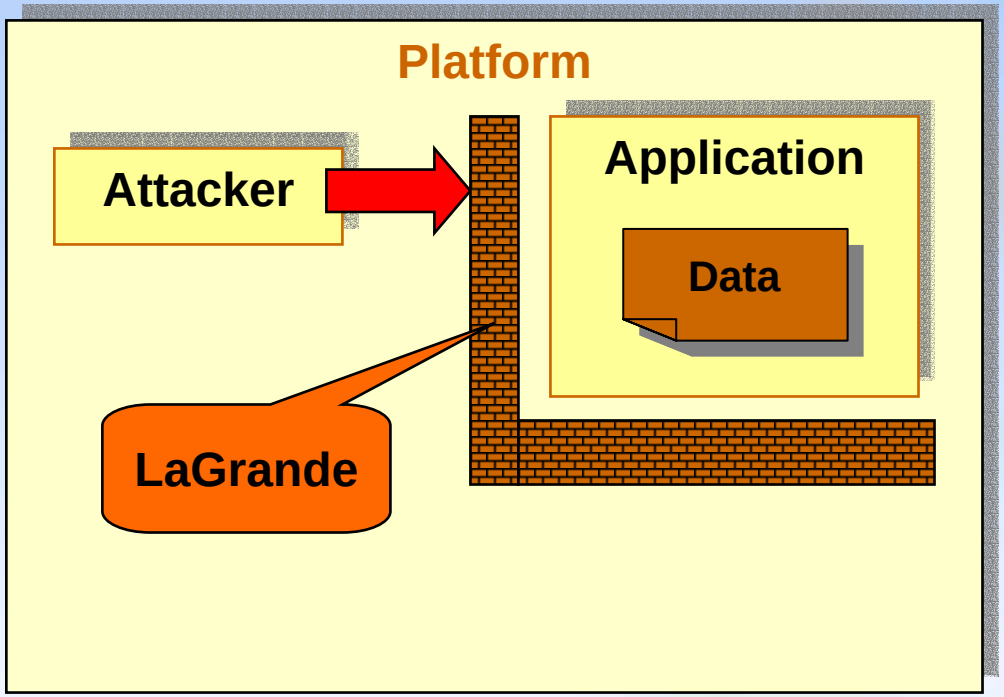
- Malicious Software can
  - Read memory
    - Expose secrets
  - Change memory
    - Change values of data or programs
  - Manipulate input and output
  - Change request for information

Threat Mitigation Has Four Principle Design Requirements

Hardware Solutions

- Protected Execution
- Attestation
- Sealed Storage
- Trusted Channels and Trusted Paths

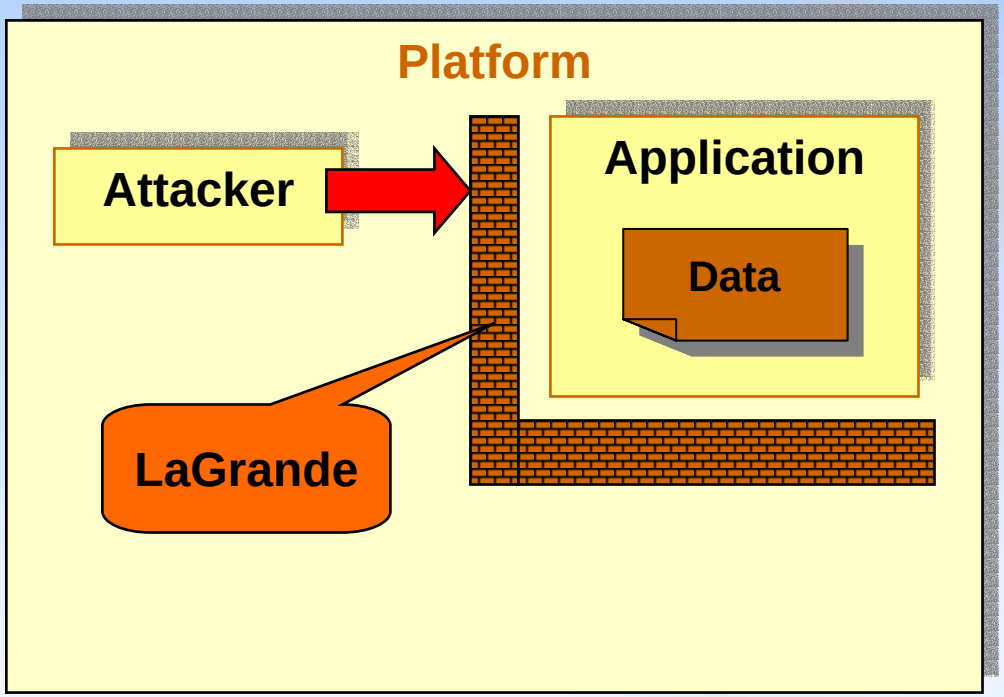
Protected Execution



- Many attacks simply read the memory of the application
- What is needed is some way to protect application from attacker
- Protected execution keeps resources of the application from the attacker

- Protected execution requires hardware support
- LaGrande Technology (LT) Protected Execution is an implementation of domain separation

What Is In The Brick Wall

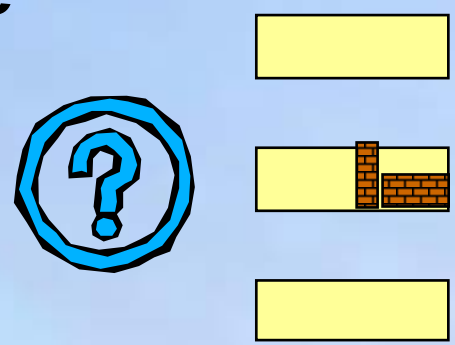


- Separation of
  - Execution processes
  - Memory pages
  - Devices
- Separation enforced by hardware
- Not new to computer science
  - Domain separation known since the 1970's

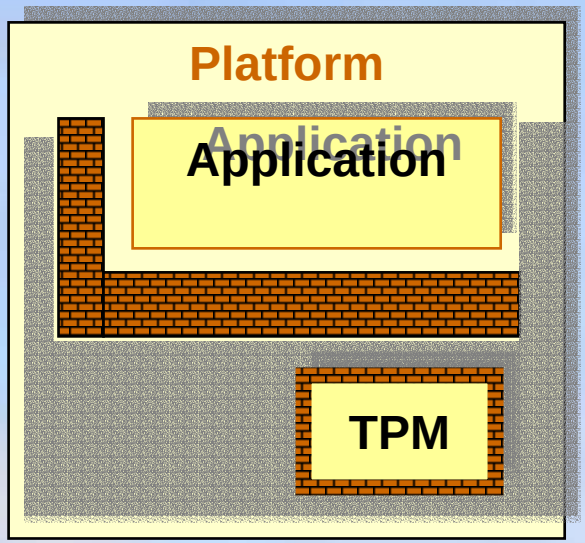
Intel Is Bringing Domain Separation to Mainstream PC Clients Via Protected Execution

Attributes In An Open Network

- Assume one wants to rely on the protected execution attributes
- Assume that the platform is in a group of platforms without protected execution
  - How does one differentiate between the platforms
- The information report needs to be reliable and verifiable
  - Most platforms have no mechanism to measure and report
- The need is to report the platform properties



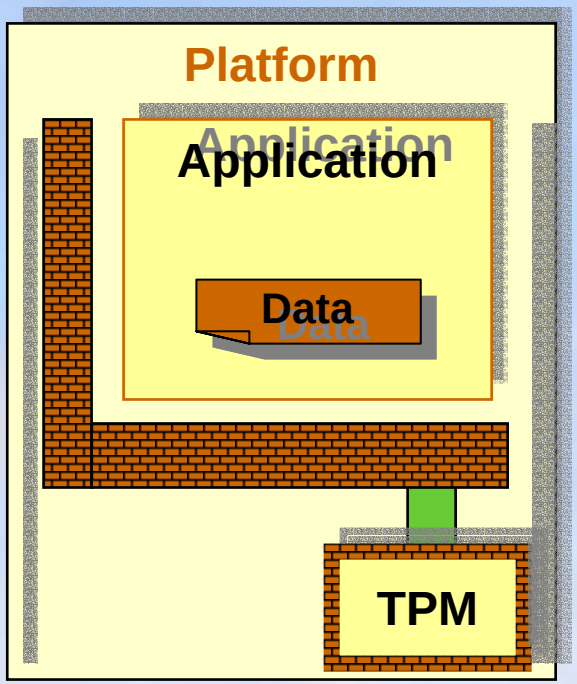
Attestation



- Prove platform properties
  - Hardware nature of platform
  - Current running configuration
    - How was the brick wall built
- Attestation requires
  - Accurate measurement
  - Storage of the measurement
  - Verifiable report of the measurement
  - A Trusted Platform Module (TPM) provides these capabilities
- Attestation device needs to provide the assurances that the storage and reporting mechanisms are properly protected
- Knowing what the brick wall is allows for the wall to report on applications protected by the wall

Trusted Platform Module (TPM)

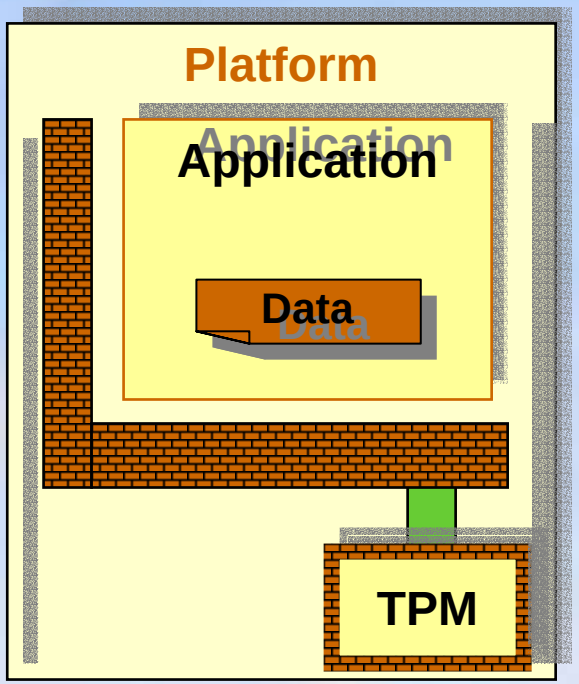
- Provides the attestation device for the platform
- Basic functionality
  - Random number generator
  - Repository for platform measurements
  - Uniqueness for reporting measurements
  - Non-volatile secure storage
  - Sealed storage
- LT requires TCG TPM version 1.2



- TPM designed to address users' privacy
- Measurement of brick wall creation stored in TPM

Sealed Storage

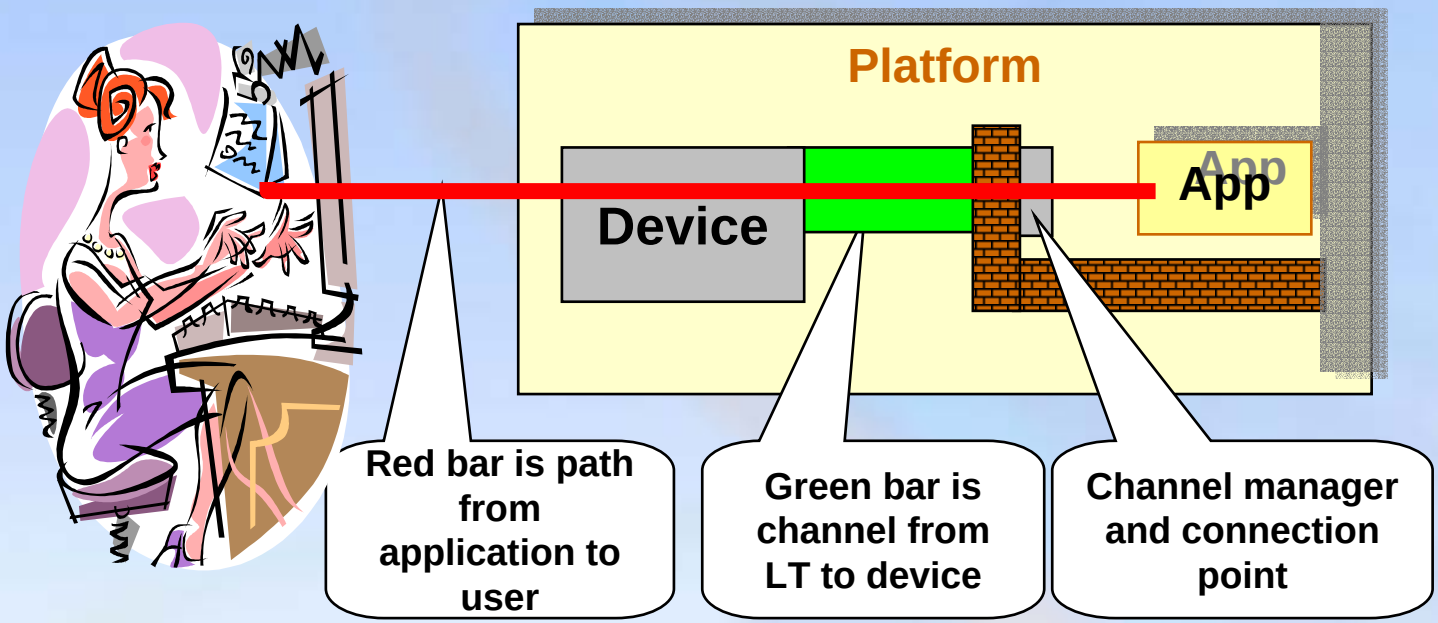
- Sealed storage is the combination of measurements and encryption
  - Seal some data such that the data is only available (unsealed) when the indicated measurement is present on the TPM
- Powerful technique to ensure that data is only available to a known environment
  - Sealing data to the brick wall ensures that the data is only available to the same brick wall
  - Changes in the wall change the measurement and make the data unavailable



TPM Provides Attestation and Sealed Storage

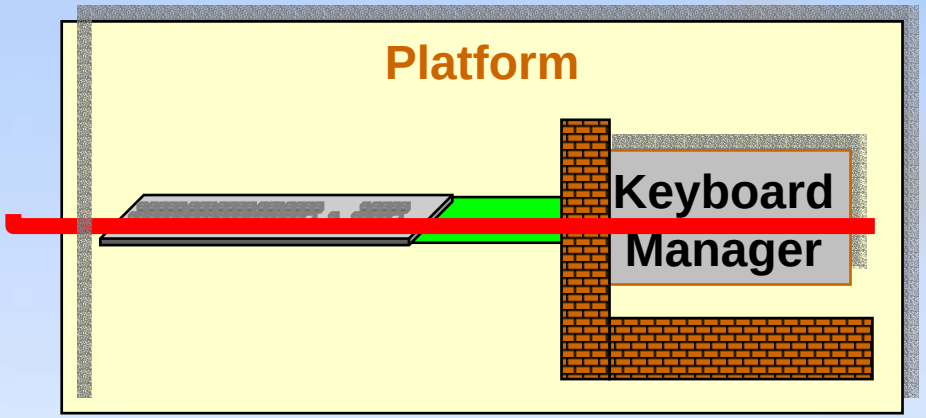
Trusted Channel

- Device typically input and output devices



- Secure channel between two computing entities
  - Brick wall and device
  - Brick wall provides connection point for protected applications
- Secure path between computing entity and human
  - Requires active participation of the human
  - Provides visual/aural indicator of the existence of a trusted channel

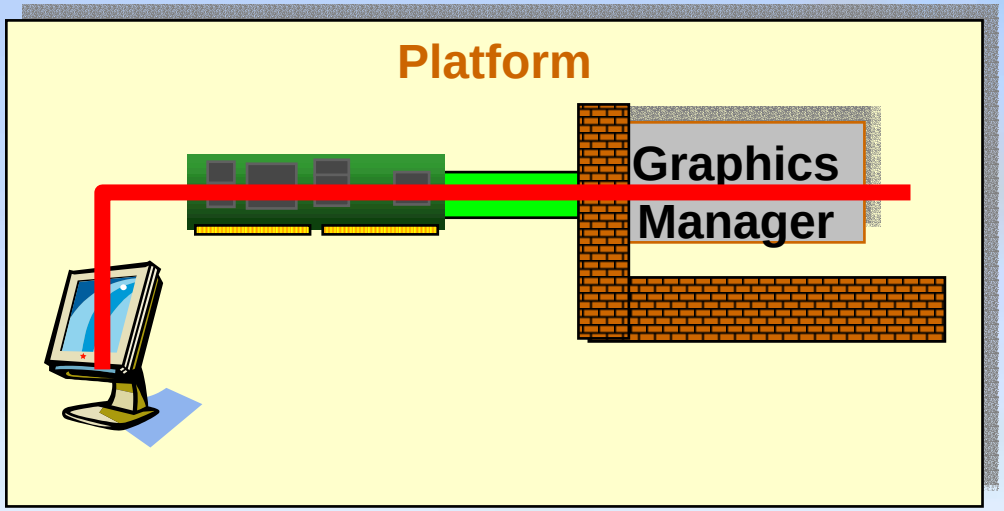
Protected Input



- Create trusted channel between keyboard and keyboard manager
- Mouse and mouse manager also need a trusted channel

- A LaGrande platform will provide the hardware hooks necessary to create the trusted channel
- OS needs to support use input manager in protected execution
- Need new input device that supports the creation of the trusted channel
- Many ways to solve the channel creation issues
- Application responsibility to create trusted path

Trusted Output Display Adapter

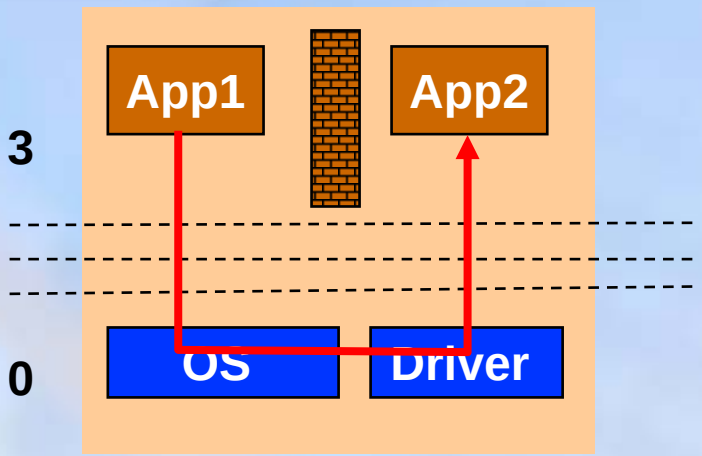


- Create trusted channel between graphics manager and display adapter
- Works for integrated and discrete graphic adapters

- OS needs to support having graphics manager in domain separated area
- Need display adapters that support the creation of the trusted channel
- Many ways to solve the channel creation issues
- Application responsibility to create trusted path

OS Requirements

- Domain separation design requires a small kernel with limited modifications
  - OS with device drivers in ring 0 breaks this requirement
  - Drivers allow app1 to gain access to app2 resources



- Easy to write brand new OS
  - Without backwards compatibility for applications and devices OS will not be successful in real world
- Building domain separation wall requires cooperation of the OS
  - Microsoft's* NGSCB* is one such type of OS that will use the properties of LT to properly create domain separation and maintain backwards compatibility

LaGrande Requires HW and SW changes

Entering The Password

1 Read password in memory

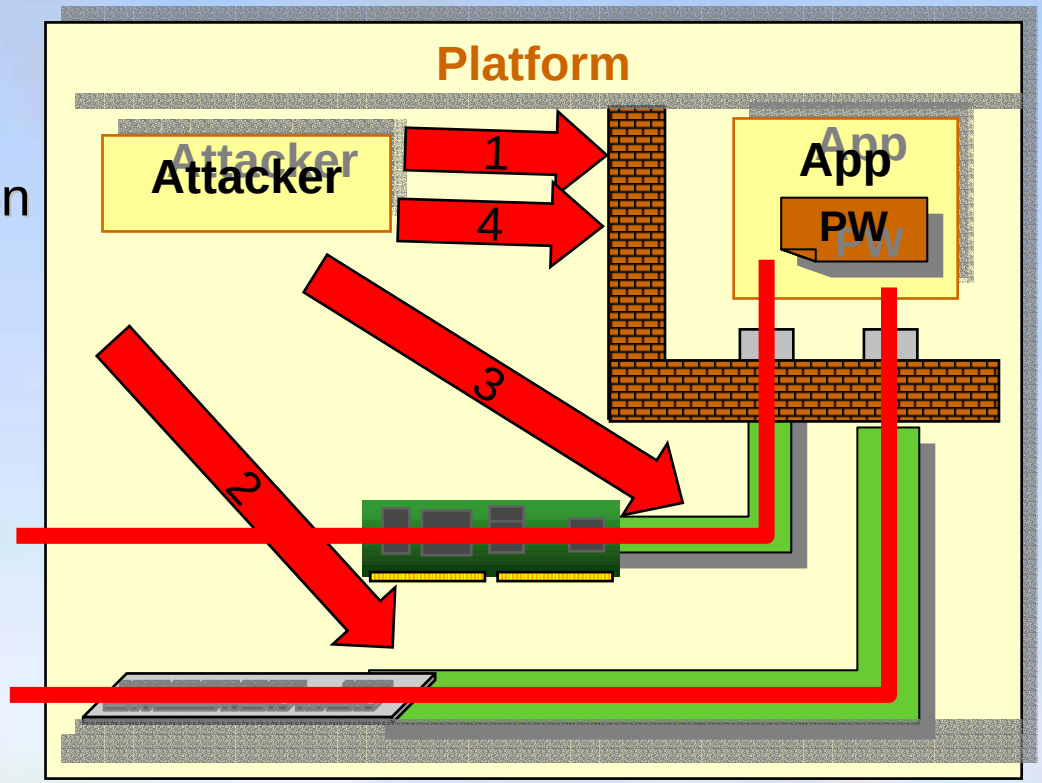
Defend using protected execution

2 Sniff password from keyboard

Defend using trusted input

3 Fake login screen

Defend using trusted output



4 Change application to ignore password entry

Defend using protected execution

Software Attacks Mitigated

Agenda

- Security Opportunity
- Security Components
- Threat Definitions
- Hardware Solutions
- Break
- LT Architecture
- LT Features
- Threat Mitigations
- Secrets and Trust

LT Security Features

Feature	Why is this Important?
Protected Execution **Platform subset where SW runs w/o interference or observation.**	• **Hardware implementation of domain separation**
Attestation **HW-based proof of current Protected Partition environment.**	• **Only install new secrets into an environment you believe will protect them.** • **Environment = HW and currently running SW.** • **Can't just ask SW – it can be spoofed**
Sealed Storage **Hold secrets for a specific Protected Execution environment.**	• **Ensure that the secret can only be "unwrapped" if the identical environment is re-launched.** • **Software-based encryption is insufficient – we must still protect the root key used by that software.**
Protected Input/Output **Ensure protected communication to and from the partition**	• **Provide infrastructure to create trusted channels to the input and output devices** • **Allow the protected partition the ability to create trusted paths**

LT Hardware Ingredients

LT = CPU + Chipset + TPM + Protected I/O

= LT specific enhancement

CPU Extensions

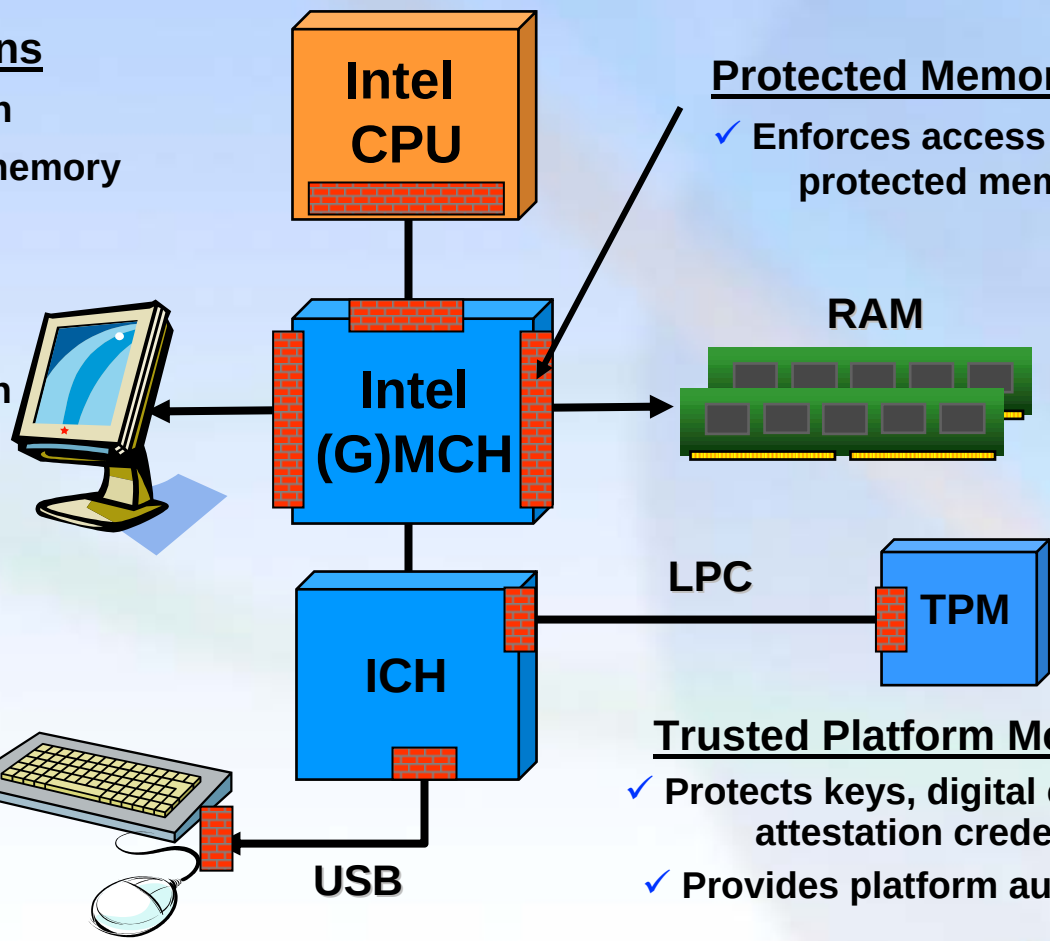
- ✓Enables domain separation
- ✓Sets policy for protected memory

Protected Graphics

- ✓Trusted channel between graphics & trusted SW (integrated or third party discrete graphics)

Protected Keyboard/Mouse

- ✓ Trusted channel between Keyboard/Mouse and trusted software



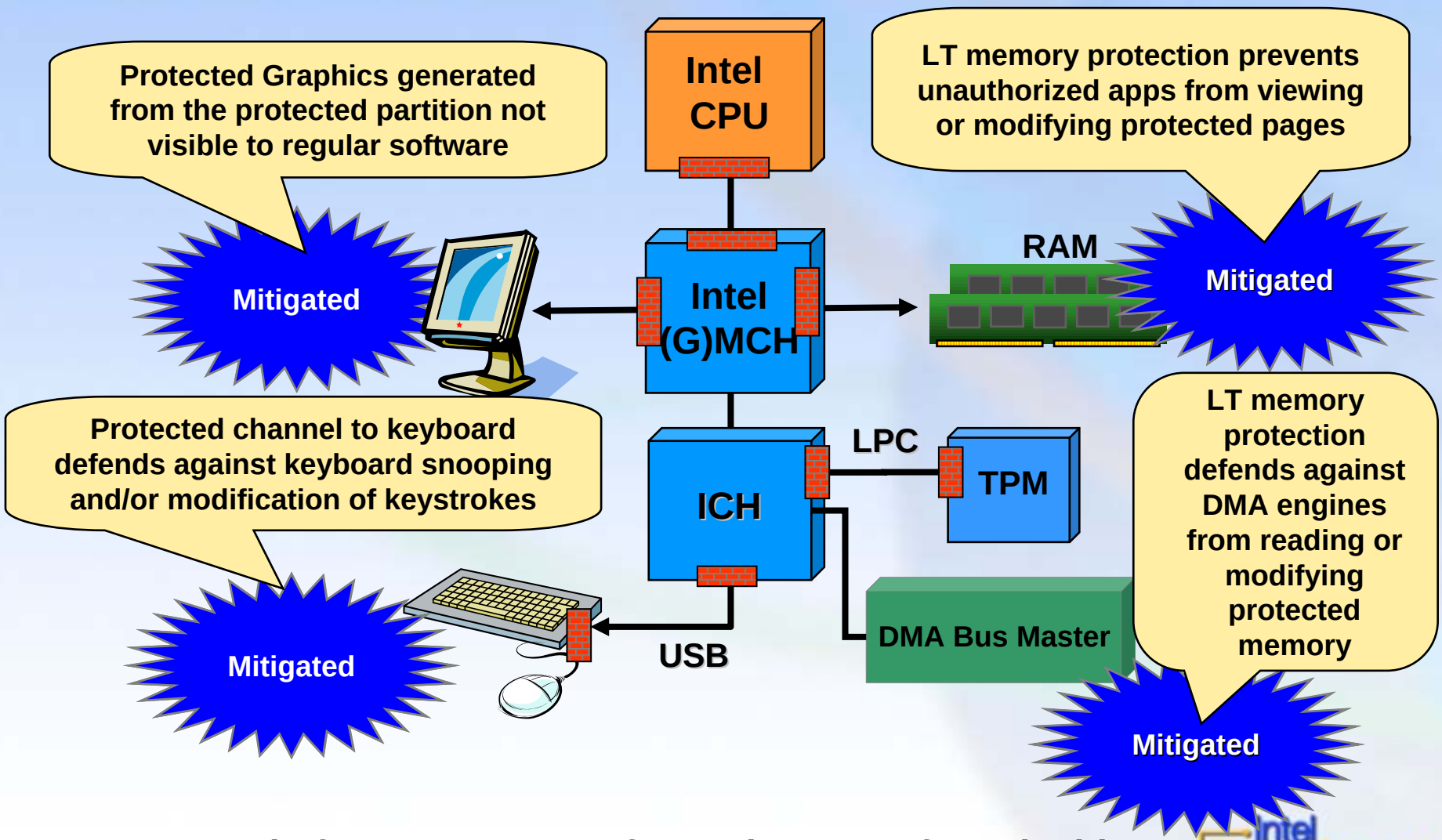
Protected Memory Mgmt

- ✓ Enforces access policy to protected memory

Trusted Platform Module v1.2

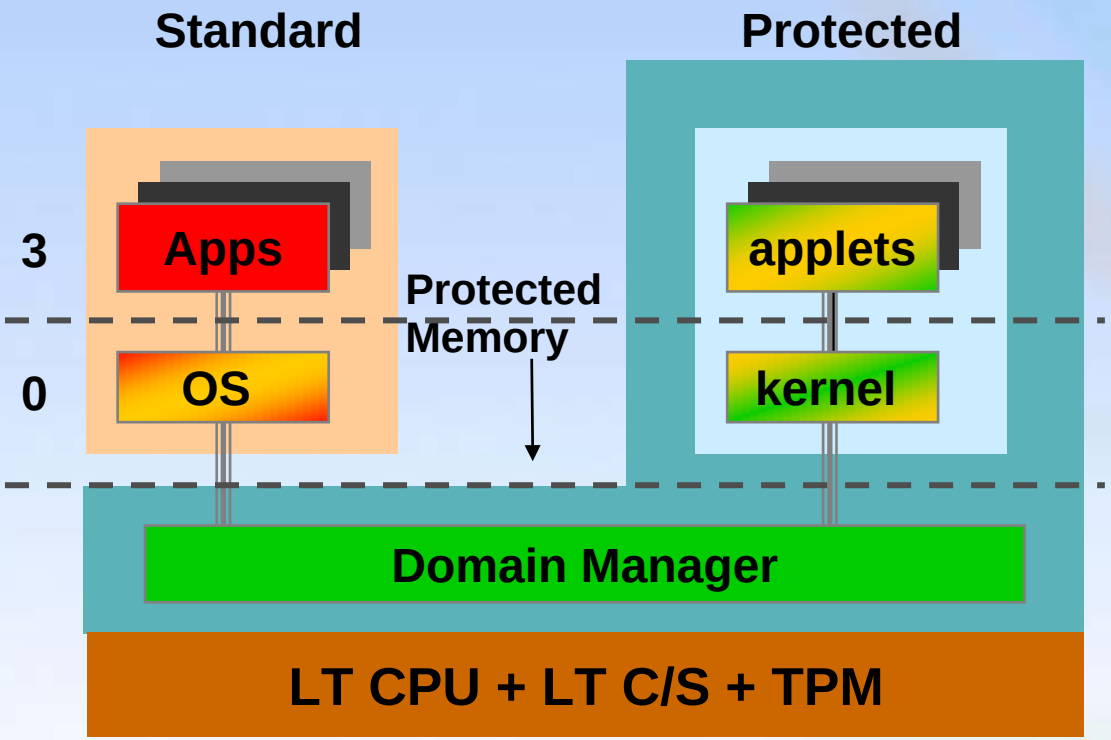
- ✓ Protects keys, digital certificates & attestation credentials
- ✓ Provides platform authentication

How LT Mitigates Vulnerabilities



LT platform creates a safer environment for valuable business data, transactions & processes

LT Protection Model

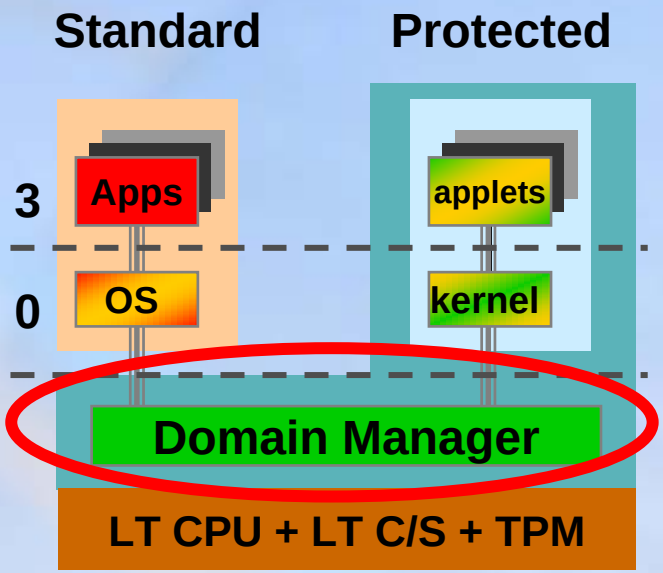


- Domain Manager provides domain separation
- LT can run any DM, OS and kernel
- Requires LT CPU LT Chipset (C/S) and TPM

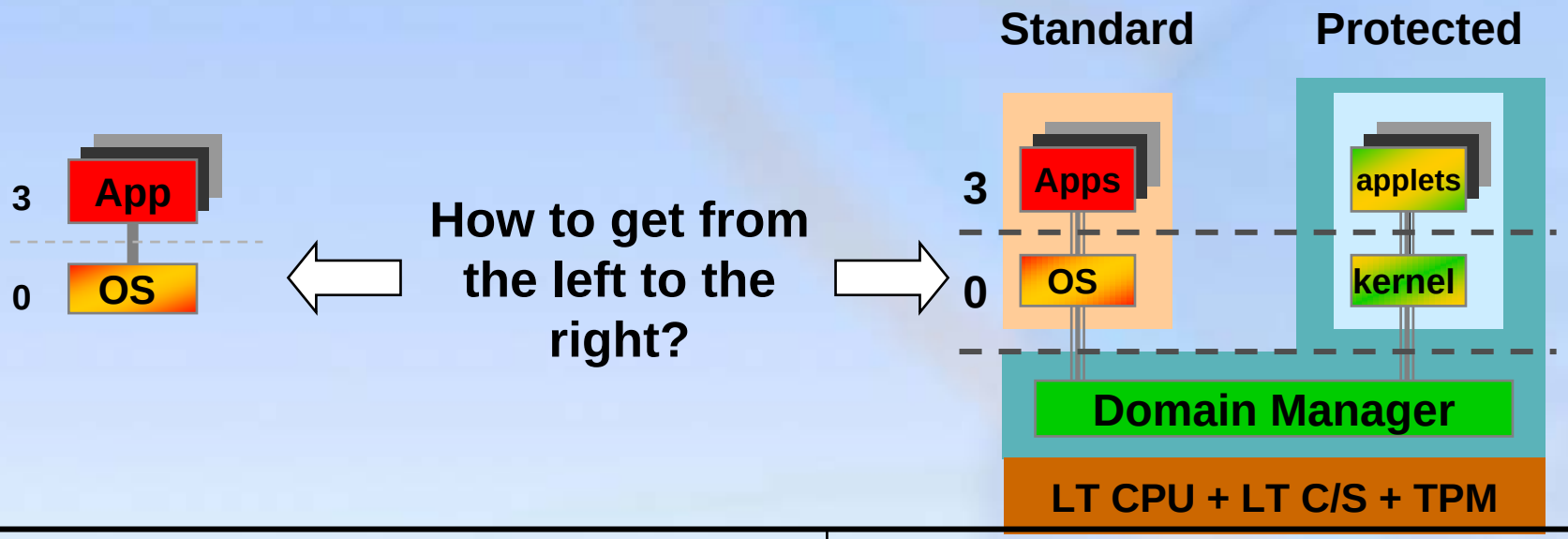
- Kernel provides protected partition services
  - may be designed to interact with a specific main OS

LT Security Foundation

- How do I determine if the Domain Manager (DM) protects my data?
  - Can I tell that a DM is loaded?
  - Can I tell if the "wrong" DM is loaded?
  - How do I prevent a "bad" DM from accessing secrets stored by my "good" DM?
  - DM protections are internal to the processor. How do we protect against non-processor accesses (e.g. DMA devices)?
- How can I prove to a remote server that I have an "acceptable" machine, so it will interact with me?
  - Both HW and SW capabilities

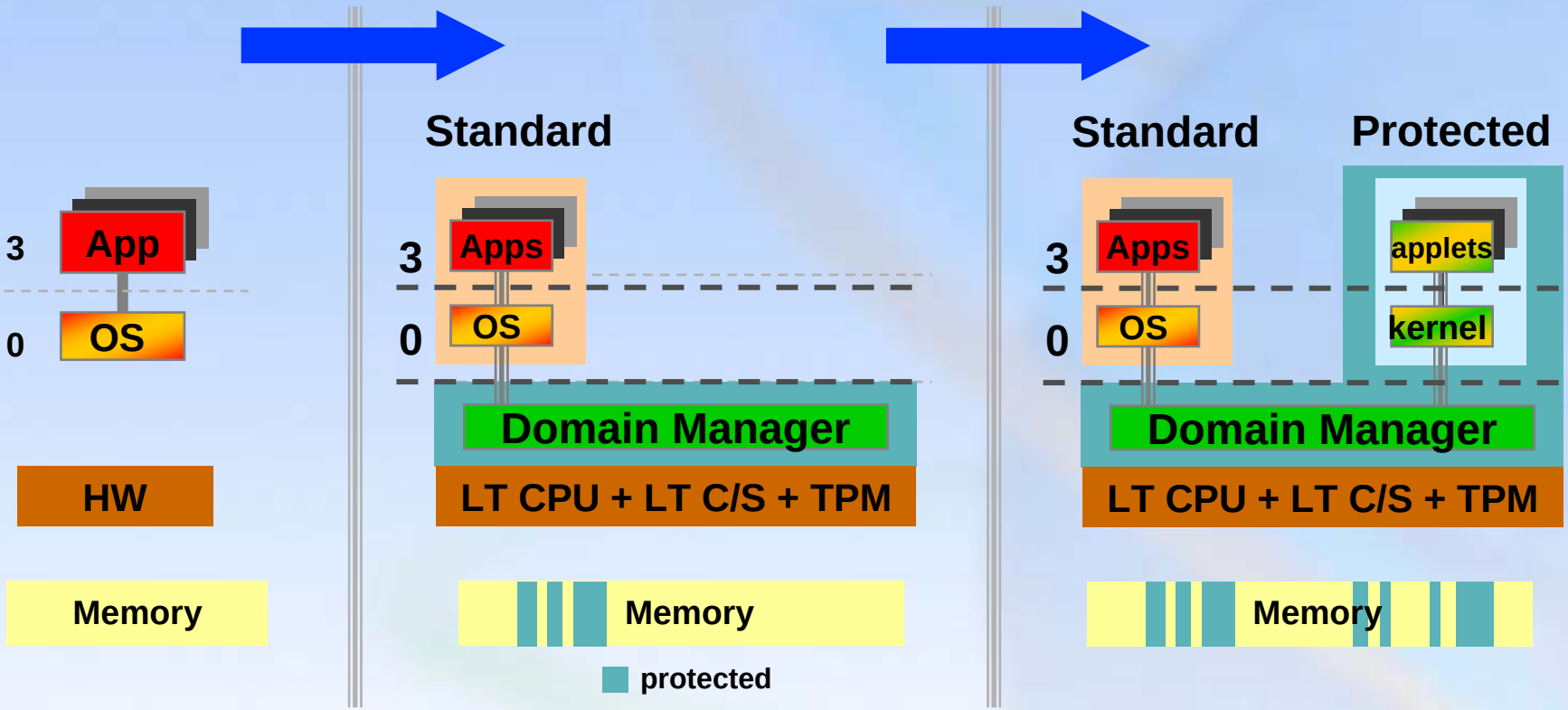


Launching Protected Domain



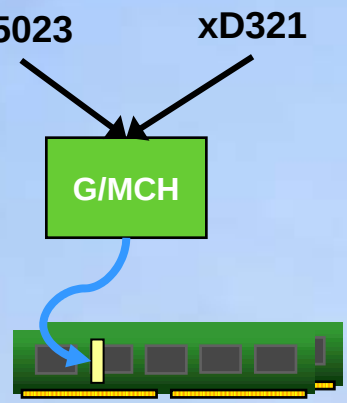
Problem	Solution
Launch LT without platform reboot	Late launch
Detect improperly configured hardware	ENTERACCS execution
Ensure no interference of launch	SENTER process
Detect tampering of Domain Manager	DM measurement by SENTER
Register DM	SENTER storing measurement in TPM

Launch Of Protected Domain



- Protected environment is only launched upon request
  - Possible to launch and take down LT multiple times without rebooting platform
- DM supplied in conjunction with launch request
- HW ensures proper launch of DM

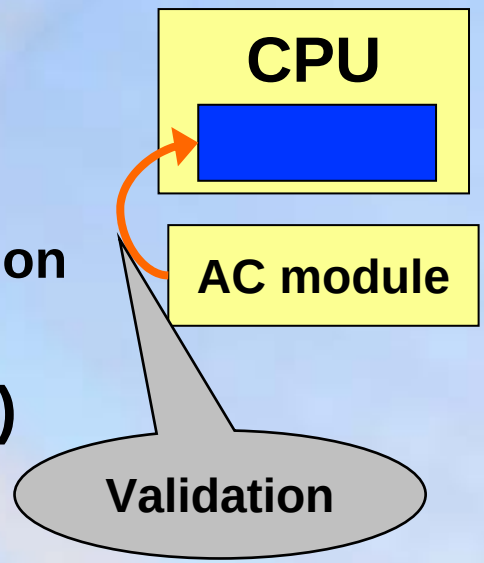
Memory Aliasing



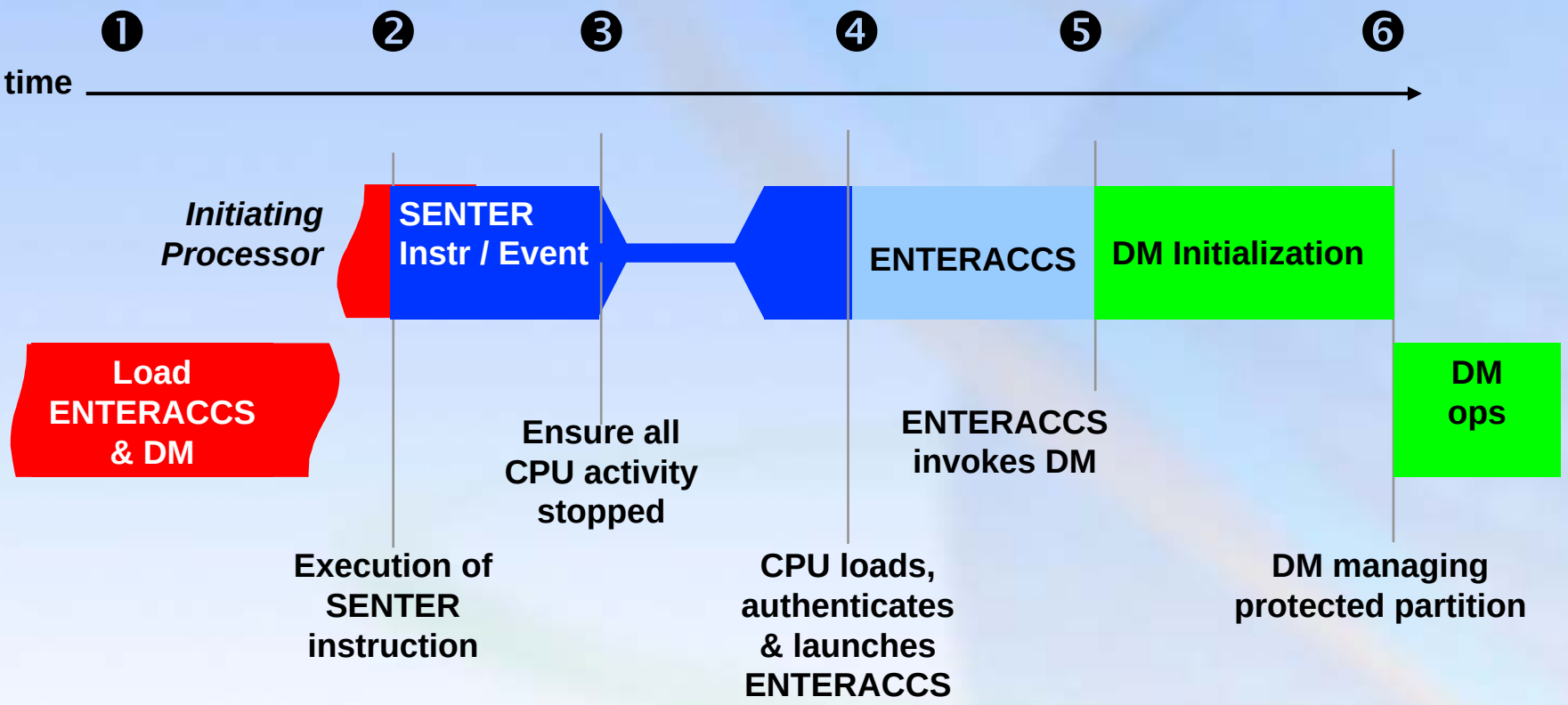
- **Problem**
  - **Memory controller requires flexibility to support wide range of memory types, sizes, speeds, configurations**
  - **BIOS may support multiple options for given DIMMs**
  - **Mismatch between DIMM characteristics & memory controller settings can allow address "folding"**
    - **Two FSB addresses mapped to same DIMM location**
  - **Testing for mismatch too complex for hardware**
- **Solution: DIMM Test authenticated code**
  - **Included in ENTERACCS module**
  - **Locks memory controller settings**
  - **Reads DIMM characteristics and memory controller settings**
  - **Tests for possible address "folding"**

Authenticated Code

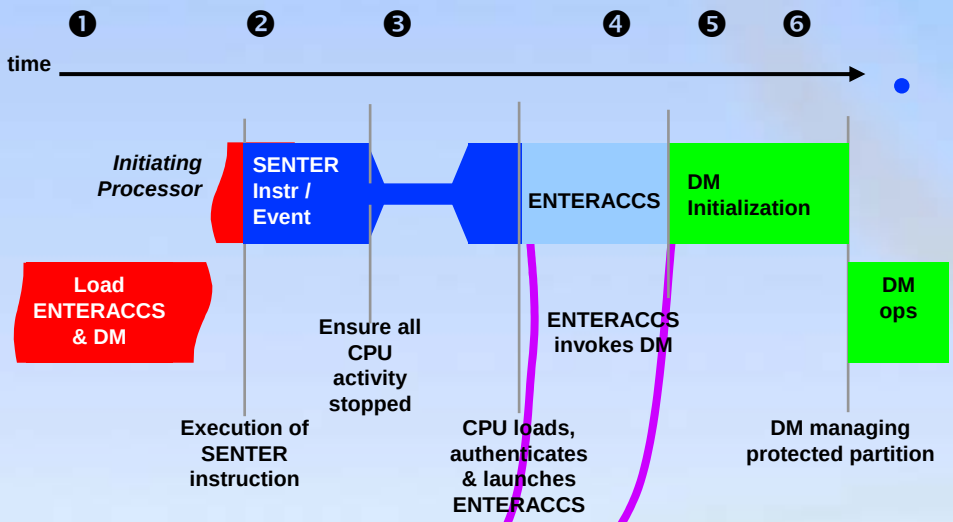
- How to detect improper hardware configurations
  - Configuration areas include chipset
  - RAM not trustable till chipset configuration validated
- Solution is Authenticated Code (AC)
  - Code is validated
    - Digital signature validation
    - Uses asymmetric keys
    - Chipset manufacturer signs code
  - Code runs in special hardware protected area
- AC code in use for many solutions
  - ENTERACCS, EXITAC etc.



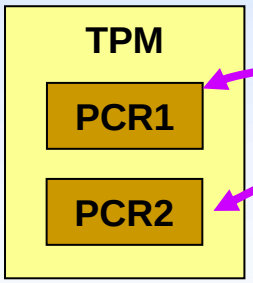
SENTER Process



DM and ENTERACCS Identity



- The SENTER measures and stores the identity (hash) of the ENTERACCS code in a TPM PCR
- The ENTERACCS code measures and stores the identity of the DM in a TPM PCR



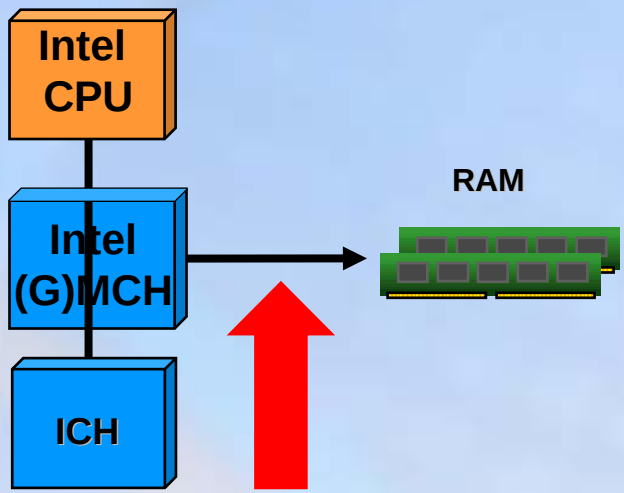
- ENTERACCS measurement assured by HW
  - Storage of measurement uses special TPM commands and bus cycles
- DM measurement assured by ENTERACCS
  - Storage of measurement uses normal TPM commands and special bus cycles

Agenda

- Security Opportunity
- Security Components
- Threat Definitions
- Hardware Solutions
- Break
- LT Architecture
- LT Features
- Threat Mitigations
- Secrets and Trust

RESET Protection

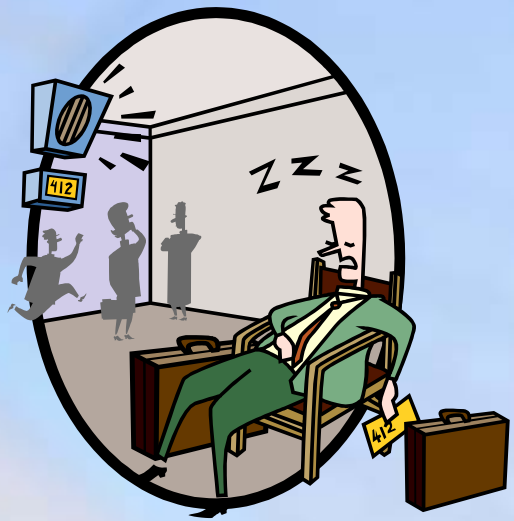
- **Problem: Unexpected system reset**
  - **CPU & chipset protections lost, but memory array may retain content**
    - **Secrets can be exposed**
  - **Can't re-establish protections**
    - **Page table may be corrupted**



- **Solution**
  - **Block access to memory on reset**
  - **If secrets were possibly in memory invoke EXITAC AC module**
    - **Need to handle power loss and other possible events**
    - **Only unblock after EXITAC execution completes**
    - **EXITAC writes set value to each byte of memory**

System Sleep Protection

- **Problem**
  - **Going to sleep may lose memory protections**
    - **Some states remove power to CPU but leave power to chipset**
  - **Power-up may be unable to reset the memory protections**
- **Recommendation: Encrypt before sleep, decrypt on resume**
  - **OS sleep routine requests DM takedown**
  - **DM encrypts all protected memory**
  - **Power removed**
  - **On power-up OS launches DM**
  - **DM decrypts memory pages and continues processing**



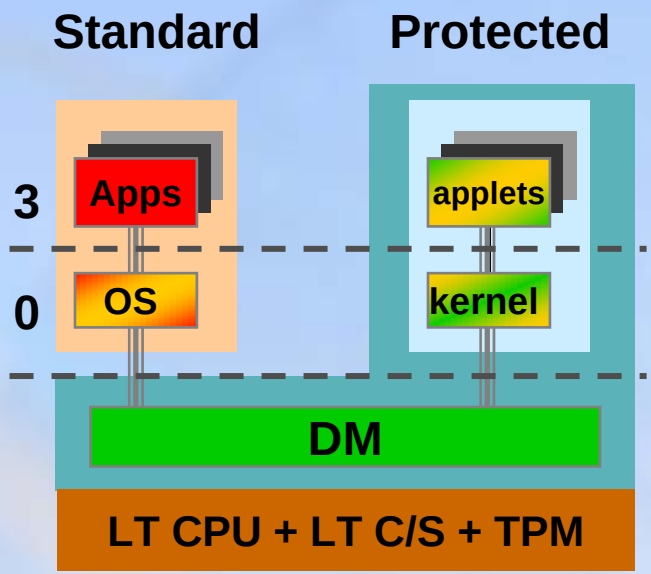
Agenda

- Security Opportunity
- Security Components
- Threat Definitions
- Hardware Solutions
- Break
- LT Architecture
- LT Features
- Threat Mitigations
- Secrets and Trust

Keeping Secrets

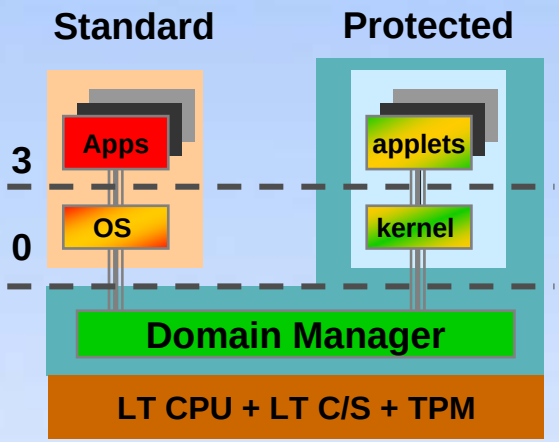
- Need to ensure that secrets held by one DM are not available to any other DM
  - Solution is to use sealed storage
  - This is reason that measurement of ENTERACCS and DM are so important
  - ANY change to ENTERACCS or DM results in new measurement value and secrets are not released
- LT assumes the existence of multiple domain managers on the platform
  - Can only execute one DM at a time
  - Sealed storage ensures that secrets held by one DM are not available to any other DM

Protecting Secrets



- **Problem**
  - Segregate data held by one DM from all other DM's on platform
- **Solution**
  - DM seals all data using the TPM and specifying the measured environment
    - The measured environment is the ENTERACCS and loaded DM on a specific platform
  - DM can reseal the data to a new environment when performing an update

Establishing Initial Trust

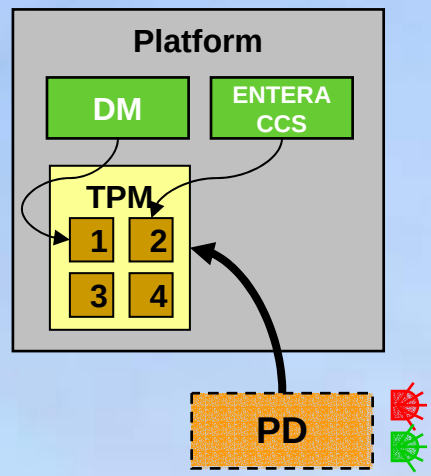


- **Assertion**
  - Once secrets are sealed to a specific environment, they will be accessible only to that environment
    - From the use of the sealing capabilities

- **Problem**
  - How do you decide that the initial environment is trustable?
  - How do you decide a currently running environment is trustable,
  - IF you don't yet have access to sealed secrets?
- **E.g. should I enter my pass phrase to the requesting environment the first time I use the machine?**

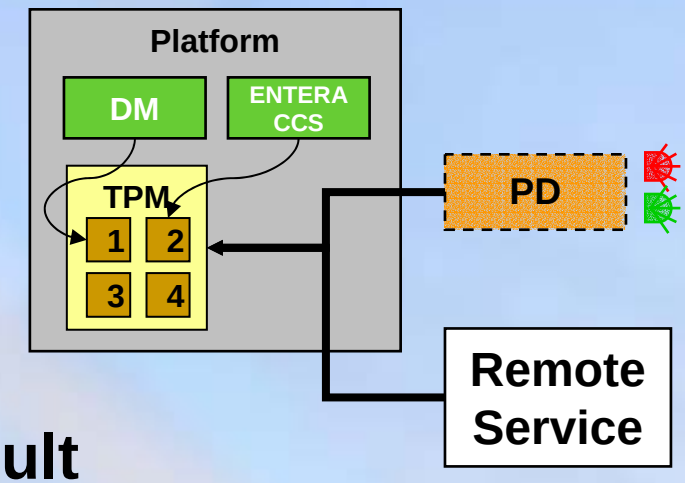
Initial Trust Mechanisms

- Method 1
  - User trusts system as delivered
- Method 2
  - User controlled portable device (PD)
    - Smartcard, USB, Cell phone …
    - Any connection mechanism (USB etc.)
    - Has output mechanism (screen, green light etc.)
  - PD loaded with measurements of valid configurations
  - PD performs attestation of system and reports pass or fail



Initial Trust – Remote Provider

- User connects to remote service
  - Web site …
- Service performs attestation and displays result
- Lots of ways to display the result
  - Simply display on web site
    - Could use another computer to review results
  - Send result by post card
  - Have user call 800 number
  - Many more
- Response comes “out of band” from the request
  - This allows the user to have confidence that attacker could not attack both sides of the platform validation



Summary

- LT is a versatile set of hardware enhancements to Intel processors, chipsets, and platforms
- LT creates a hardware foundation that helps protect data from software-based attacks
- LT uses well known security properties

Next Steps

Hardware Developers & OEMs	• Factor LT into Business platform planning • OEMs contact component vendors for specific product plans • Contact your Intel representatives for available specifications & schedules
Software Developers	• Factor LT into Business software planning • Contact your Intel representatives for information on Intel Early Access Program
IT	• Factor LT into enterprise security strategy and infrastructure planning • Communicate needs for LT to your PC vendors

Thank you for attending.

Please fill out the Session Evaluation Form.