Volume 12 Issue 04 Published December 23, 2008 ISSN 1535-864X DOI 10.1535/itj.1204.01

Intel Technology Journal

Features Intel's recent research and development

Articles

Preface

Richard Bowles, Publisher
David King, Managing Editor

The wide variety of uses of Intel® vPro™ technology is the scope of this Intel Technology Journal (Vol. 12, Issue 4). Content architect for this issue is John Vicente and he has assembled an excellent sampling of the ways that Intel vPro technology can be deployed to maintain system and data security, to improve manageability, and to support emerging compute models.

Security

The Journal begins with a thorough explanation of the security mechanisms built into the firmware, memory, and chipsets that are components of Intel vPro technology.

This cluster of technologies ensures that Intel's manageability framework is robust to attacks and that only authorized users can access the functions through local or remote management features.

A second article looks at the shifting requirements for the protection of data. In particular, the author explains and evaluates different ways to protect data at rest. Encryption provides the foundation, but should encryption be undertaken by software running on a client machine, by the chipset surrounding the processor, or by technology embedded in the storage device?

Manageability

"Innovating Above and Beyond Standards" is a rich article that puts the evolution of improved system management into an historical context. The authors point out that there is a natural tension between innovation and standardization and, at the same time, the two can and should coexist. The article provides a historical roadmap of related manageability standards and explains their association to current components of Intel® Active Management Technology (Intel® AMT).

While Intel vPro technology is built in, "Configuring Intel Active Management Technology" enumerates the steps in the installation process and shows the ways in which functions and features can be setup and customized to specific user and IT organization needs.

Three additional articles put Intel vPro technology to work. One explores remote repair capabilities, another looks at managing mobile power consumption, and a third examines the use of Intel vPro technology for reducing power consumption.

Emerging compute models and innovative uses

"Building Robust, Dynamic Virtual Clients" provides an introduction to client virtualization. The authors provide an overview of emerging virtual computing usages, associated challenges, and a synopsis of key capabilities to support Dynamic Virtual Client (DVC) computing. They go on to describe the next steps in the maturing of DVC technologies.

The streaming client compute model is the topic of the next article. In this model, operating systems, data storage, and application execution occur on a server and the results are streamed to client systems. The objective is to gain the manageability and business continuity advantages while maintaining the flexibility and mobility of personal computers.

Technology developed for one class of purposes can often be put to other uses. The final article in this Journal demonstrates how innovative software can extend the functionality of Intel vPro technology in a variety of directions. Creating virtual serial ports and virtual storage drives are two examples.

We hope you enjoy this issue of the Intel Technology Journal.

Foreword

Gregory Bryant, Vice President, Business Client Group; General Manager, Digital Office Platform Division, Intel

The introduction and integration of the PC into the business environment in the early 1980s resulted in the de-centralization of computing. A whole new industry sprang up focused on providing information technology (IT) organizations with software tools to manage their growing inventory of desktops and notebooks. The advent of the internet and wireless has freed workers from their offices and cubicles. IT shops around the globe are beginning to feel the strain as modern day software solutions are stretched to their limits.

Enter Intel® vPro™ technology. Created with IT in mind, critical functions that have challenged traditional software solutions are now moving into silicon. Solutions that rely on these functions can be assured that the capabilities they depend on to successfully control fleets of PCs, even in the most remote locations, are present even if the agent has been disabled, the operating system is no longer functioning or the system has been compromised. Intel is now shipping its third generation of Intel® Core™2 processor platforms enabled with Intel vPro technology. Every major OEM and more than 80+ solution providers are shipping platforms and products enabled to take advantage of the platform's capabilities. Businesses across the world are deploying and utilizing Intel vPro technology to save energy, reduce CO_2 emissions, return their workers to full productivity faster, and more.

I am very proud to present an inside view of Intel vPro technology in this special issue of ITJ. I think you will find the breadth of topics and the details provided fascinating, useful and inspiring. There are exciting new compute models emerging and many new innovations on the horizon. Intel is committed to delivering the capabilities that IT needs the most, ensuring that PCs with Intel vPro technology will continue to be the best business PCs for years to come.

Enjoy!

Technical Reviewers for Q4 2008 ITJ

Tim Abels, Business Client Group
Olga Adamovsky, Mobile Wireless Group
Gal Alkon, Business Client Group
Gareth J. Bevan, Financial Enterprise Services Group
Herman D'Hooge, Corporate Technology Group
Dori Eldar, Business Client Group
Sarah Flanagan, Corporate G&A Group
Lukas M. Grabiec, Corporate G&A Group
Steve Grobman, Business Client Group
Clyde Hedrick, Business Client Group
Ajith K. Illendula, SSG Enabling Group
Erik J. Johnson, SSG Support Group
Mukesh Kataria, Chipsets Products Group
Divya Kolar Sunder, Corporate Technology Group
Arvind Kumar, Digital Enterprise Group
Omer Levy, Business Client Group
Ajay Mungara, SSG Enabling Group
Mike Murphy, SSG Enabling Group
Michael Navon, Chipsets Products Group
Lisa Neal-Graves, Corporate G&A Group
Stefanie Neuhierl, Corporate G&A Group
Randy Nystrom, Financial Enterprise Services Group
Craig T. Owen, Business Client Group
Tom Quillin, Business Client Group
Yasser Rasheed, Business Client Group
Christie Rice, Financial Enterprise Services Group
Aharon Robbins, Chipsets Products Group
Tal Roth, Chipsets Products Group
Adi Shaliv, Business Client Group
Randal F Templeton, SSG Enabling Group
Jeff M. Tripp, Business Client Group
Moshe Valenci, Chipsets Products Group
John Vicente, Business Client Group
Vincent Von Bokern, Chipsets Products Group
Yael Yanai, Chipsets Products Group

Advanced Security Features of Intel® vPro™ Technology

Omer Levy, Business Client Group, Intel Corporation
Arvind Kumar, Business Client Group, Intel Corporation
Purushottam Goel, Business Client Group, Intel Corporation

Keywords: Intel® vPro™ technology, security, authentication, audit log, random numbers, authorization, TLS, blob service, measurement, monotonic

Abstract

Intel® vPro™ technology creates a powerful platform in which security and manageability go hand in hand. Since manageability is a crucial aspect of an enterprise's network, it is extremely important to secure the manageability infrastructure against attacks from outside and inside the network. In this article, we delve into some of the most interesting security features that make Intel vPro technology a far more secure (and thereby differentiated) offering than any other security technology on today's market.

First we start with an overview of some of the basic security features of Intel vPro technology. Then, for most of the remainder of this article, we delve into some of the advanced and more complex security aspects of Intel vPro technology that truly put this technology above its competitors when it comes to providing security. We start with a detailed discussion of a few of the foundational security aspects of Intel vPro technology, provided by hardware mechanisms, such as true random numbers and monotonic counters. We then discuss the details of a secure storage service, which is an immensely useful firmware mechanism that allows storage of secrets on the nonvolatile flash memory, such that they cannot be read or tampered with, even if the flash part is physically attacked. Next we discuss the mechanism of firmware measurement. In this mechanism, the firmware provides a measurement of the code running on the internal processor. Finally, we discuss audit logging. This mechanism helps to mitigate the "rogue insider" problem. We show how the audit log, enabled with Intel vPro technology, is designed to prevent such an insider from abusing the power of this technology, and then covering his or her tracks.

Introduction

Any new technology, such as Intel® Active Management Technology (Intel® AMT), which is an integral part of platforms, with Intel vPro technology, is under constant threat of attack by adversaries, for fun, fame, and/or profit. Attackers could operate remotely and communicate with Intel AMT over the wired or wireless network interfaces, or they could be present physically at the keyboard of the computer. Attackers could place some malicious program in the computer's operating system (OS) that works on their behalf. In this article, we explain some of the newer and more advanced details of the security protections designed into Intel AMT. These protections ensure that Intel AMT is well guarded from attacks by malicious entities (people or programs) operating remotely or locally. However, providing robust security in any system often comes at a cost. This cost usually is felt in terms of reduced convenience and less ease of use. We also describe some of the tradeoffs between security and ease of use of the Intel AMT computer.

Attack Surfaces

As with any security analysis, we start with analyzing the attack surfaces. Intel AMT provides very unique and powerful computer-manageability features that provide ample benefits to the IT administrator of the enterprise. Just like most other powerful capabilities in any system, the adversaries can and will attempt to misuse Intel AMT to attack the computer. The extent of the damage caused depends on the nature of the attack. For example, it could be something relatively innocuous that just creates a nuisance for the end user of the computer, or it could be something serious, such as disabling some of the security protections offered by Intel AMT.

The first attack surface is the Intel AMT internal processor itself. If the attacker is able to execute arbitrary code on the Intel AMT processor, he or she can access the secrets stored on the platform and also bypass several of the protection mechanisms. Such an attack is of course difficult to mount, but very rewarding for the attacker. Therefore, the Intel AMT execution environment is an obvious attack surface.

Next is the fact that Intel AMT is a network entity, and it is therefore important to make sure that any network communication between the Intel AMT platform and the remote management console is secured, such that no secrets are revealed to a network eavesdropper. Anyone who accesses Intel AMT through the network interface should not be able to reveal any secrets without authentication.

There are also local attackers. Should one of the enterprise's platforms, enabled with Intel vPro technology, be stolen, the thief might attempt to discover the enterprise's secrets, by using any of the Intel AMT local management interfaces or by physically accessing the nonvolatile flash memory storage.

Finally, an insider who has legitimate authentication credentials to access Intel AMT might abuse his or her position and cause damage to the system, or he or she might gain access to the computer's secrets through a backdoor into the computer (more on this later).

In the following sections we describe the specific protections designed into Intel AMT to prevent these attacks.

Security Overview

This section describes the protections available in Intel AMT from its very first generation. These constitute protections such as isolated execution, code integrity, storage protection, network security, authentication, and access control.

Isolated Execution Environment

Intel AMT runs on an internal processor integrated into the platform chipset. The firmware code for Intel AMT is stored on internal Read Only Memory (ROM) in the chipset, and on a portion of the platform's nonvolatile flash memory device (simply referred to as flash). The flash also contains the nonvolatile configuration data for the Intel AMT firmware.

The Intel AMT processor uses some Random Access Memory (RAM), internal to the chipset, for runtime storage; however, most of the runtime storage comes from an area in the platform Dynamic RAM (DRAM), called the Unified Memory Architecture (UMA). The integrated DRAM controller, once it is configured properly by the system Basic Input/Output System (BIOS), ensures that the host does not have read and write access to the UMA region of system memory. (The host in this case refers to the main platform processor and OS software that are visible to the end user, for example, the Intel® Core™2 Duo processor, and Windows Vista*, respectively.)

All of the above ensure that the Intel AMT processor has an isolated execution environment: when using platform resources, the processor maintains a closed system that does not interfere with the operation of the main platform, and cannot be intruded upon by the OS and software applications running on the main processor.

Firmware Signing

The Intel AMT processor begins its execution from the internal ROM. The content of the ROM is determined before the Intel® chipset is manufactured and is created as part of the chipset. Since the ROM code cannot be modified after the chipset is manufactured, the ROM code makes an appropriate root of trust for the Intel AMT subsystem.

The bigger portion of the Intel AMT code is located on the platform flash. The ROM code is responsible for loading the code from the flash, only after verifying its authenticity. This process ensures that only authentic code produced by Intel is loaded onto the Intel AMT processor.

The signing method for the flash code is based on public/private key cryptography. The private part is kept safely in Intel's data centers. When Intel produces a firmware version for Intel AMT, a digital signature for the code image is produced in the signing facility (within one of Intel's data centers) by using the private key. This digital signature is then stored on the production platform flash, along with the firmware code. The corresponding public key is embedded in the ROM code. When the ROM loads, it uses the public key to verify that the signature on the flash matches the code it is about to load. Only if the signatures match, will the ROM load the code onto the flash.

A similar process occurs during the firmware update process (a process to update the firmware from an older version to a newer version). Each updated

firmware image, which may be publicly available and also placed on the website of the original equipment manufacturer (OEM), is also signed by using the Intel private firmware signing key, as explained earlier. The Intel AMT processor will update the code on flash, only if the image is properly signed.

Flash Security

The platform flash part is shared among various platform consumers. It contains the Intel AMT code and data (including secret/sensitive data such as keys and access control lists), and also the BIOS code and other platform configuration parameters. It is important to make sure that the Intel AMT portion (code and data) is accessible only to the Intel AMT processor.

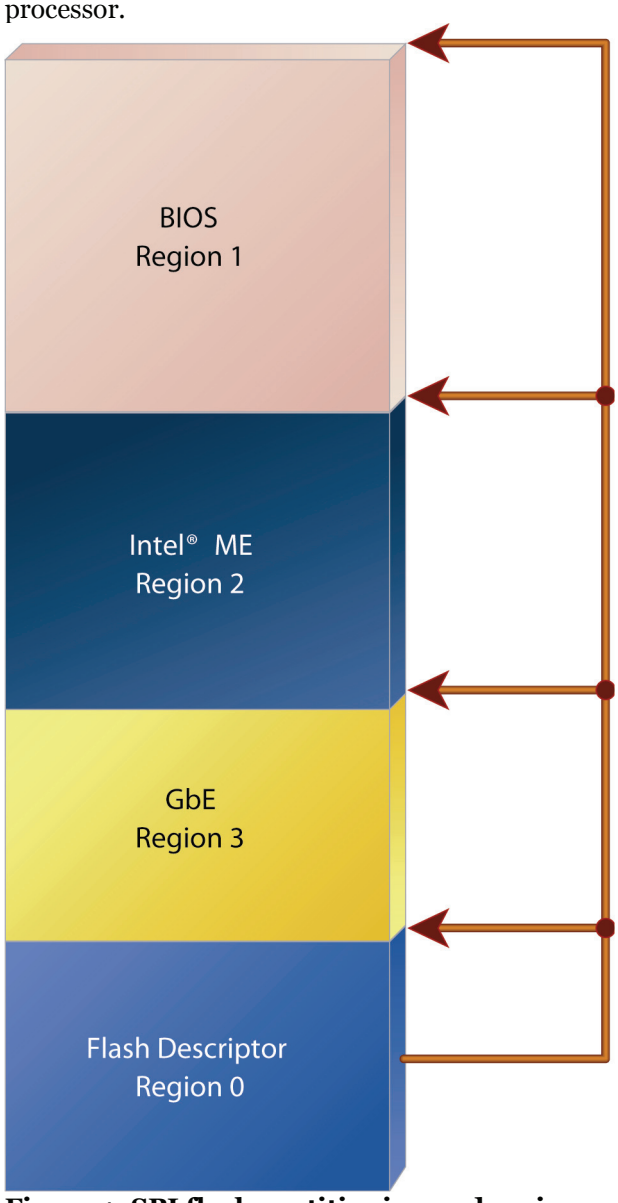

Figure 1: SPI flash partitioning and region owner. Source: Intel Corporation, 2008

At the beginning of the flash (that is, at address 0) there is an area called the flash descriptor. This area describes the partitioning of the flash into regions, and defines the owner of that region (see Figure 1). The flash controller is responsible for enforcing the partitioning of the flash according to the contents of the flash descriptor. This ensures that the Intel AMT region of the flash is accessible by the Intel AMT processor only.

The flash descriptor also contains the access capabilities to the flash descriptor itself. If the flash descriptor remains open to write operations, the protection on the other regions is of course useless. Therefore, at the end of the manufacturing process, the OEM must lock the flash descriptor region. From that point on, it cannot be rewritten. This precludes any possibility of access to the flash from regular host-based software.

The flash descriptor also contains the access capabilities to the flash descriptor itself. If the flash descriptor remains open to write operations, the protection on the other regions is of course useless. Therefore, at the end of the manufacturing process, the OEM must lock the flash descriptor region. From that point on, it cannot be rewritten. This precludes any possibility of access to the flash from regular host-based software.

Network Security

The Intel AMT firmware implements standards-based network security. This approach increases the ability of the enterprise IT administrator to use Intel AMT in a secure manner by using standards-based software, thereby avoiding any compatibility issues.

Access to Intel AMT through the network is based on the Hypertext Transfer Protocol (HTTP). The HTTP enables the HTTP server (the Intel AMT subsystem in this case) to require the client (the management console) to identify itself in some way, before providing the service. Intel AMT supports two forms of identification. The first is called “HTTP digest”: it requires the HTTP console to supply a username and a password as forms of identification [1]. As the name implies, the password is passed in a “digested” form, that is, hashed, such that it is not visible to a network eavesdropper [2]. The second form of identification is called “HTTP Negotiate” [3]. HTTP Negotiate can be based on various authentication protocols, such as NTLM, and Kerberos. Intel AMT supports Kerberos-based HTTP Negotiate. This form of identification is specifically suited to (but not limited to) Microsoft Active Directory* domains. HTTP Negotiate requires the management console to retrieve a Kerberos ticket from the Active Directory

server; the ticket encapsulates the details of the administrator (his or her username and the Active Directory groups he or she belongs to), and it signs them with a key known only to the HTTP server.

The Intel AMT firmware holds an Access Control List (ACL). Each entry in the ACL contains a HTTP Digest username/password pair, or a HTTP Negotiate Active Directory user. The ACL entry also contains the list of resources and features in the Intel AMT subsystem that this specific user has access to. In both forms, the Intel AMT platform first checks that the user is authenticated; it then checks that the user is authorized to access the specific resource/feature; and only then does it act on the specific user request.

The Intel AMT subsystem also allows the use of secure HTTP (HTTPS) to ensure the confidentiality of the contents of the message. In this mechanism, HTTP is implemented on top of the Transport Layer Security (TLS) protocol, rather than on top of the plain Transmission Control Protocol (TCP) [4]. TLS allows encryption and integrity-protection of all messages from the client to the server and back, PKI-certificate-based authentication of the server to the client, and optional certificate-based authentication of the client to the server.

In addition, Intel AMT supports some advanced methods of network security, such as link-layer authentication, by using 802.1x (especially useful in WiFi* networks), and client compliance, by using Network Access Control (NAC) mechanisms.

All of these methods are optional; it is up to the IT administrators to determine the required network security level for their manageability/security requirements. Many of the options mentioned require some backend infrastructure and maintenance mechanisms to be supported by the IT: for example, a periodic replacement of all secrets provisioned onto the Intel AMT platform.

It should also be noted that the initial configuration of the secrets provisioned onto the platform with Intel AMT is itself performed in a secure manner. A detailed explanation of the provisioning process can be found in "Intel AMT Configuration" [5], which also appears in this issue of the *Intel Technology Journal*.

Advanced Security Features of Intel® AMT

We now cover some of the more advanced hardware and firmware security features, built into Intel AMT, that were not in its first version: these features were added to the later version to make Intel AMT more secure and resilient to attacks. The new hardware includes new features, such as a true random number generator (RNG), secure storage of sensitive data, a measured launch of Intel AMT firmware, and secure audit logging.

True Random Number Generator

Many cryptographic algorithms and mechanisms make use of random numbers, including several of the Intel AMT mechanisms described previously. The important feature of RNG is its entropy, that is, the measurement of the inability of an external viewer to predict the next number that will be generated by the RNG, even if the viewer knows all the previously-generated random numbers by that generator. Many implementations use a pseudo-RNG (PRNG), a deterministic algorithm that produces the next random number based on the current generator's state. These algorithms maintain a high level of entropy, as long as the initial state (also called "the seed state") of the PRNG is not known [6]. For example, some PRNG implementations seed themselves according to the value of one of the platform clocks. This value is considered to be fairly unpredictable (due to the high resolution of the clock), and therefore makes a good seed for the PRNG. However, given that a large number of platforms power up at the same time, a time that may be known to within a few minutes or seconds, this could help a potential attacker to narrow down the possibilities and therefore crack the PRNG seed state, thereby predicting the next numbers generated by the PRNG. Conversely, an attacker could learn from the numbers generated by one hacked platform to break other platforms in the enterprise (known as a BORE attack: "Break Once, Run Everywhere").

Intel vPro technology contains a true random number generator (TRNG) hardware device (see Figure 2). The TRNG is based on two resistors that produce a thermal noise. The noise is amplified and provided as input to a frequency-modulated, low-frequency oscillator. Combined with a high-frequency oscillator, a nearly-random bit stream is produced. A voltage regulator controls the above hardware components to avoid any bias based on voltage. In addition, a logic block attempts to correct the bit stream of any bias that may have been inserted (for example, due to the non-perfect duty cycle of the oscillator), by using a standard anti-bias correction algorithm.

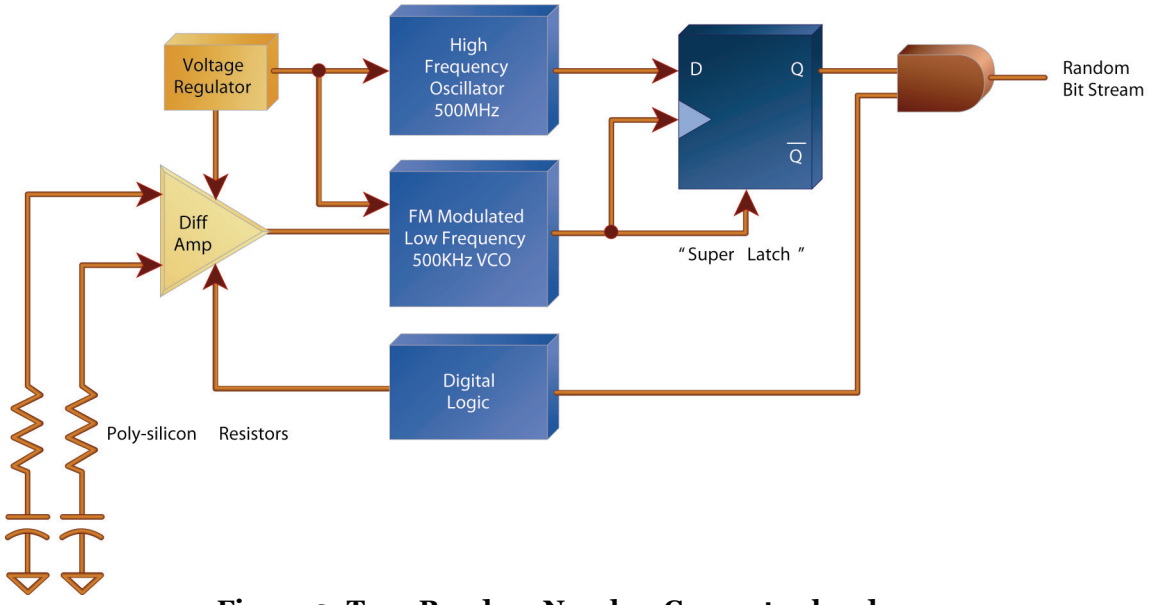

Figure 2: True Random Number Generator hardware
Source: Intel Corporation, 2008

One reason not to use a TRNG for Intel AMT usages (such as in TLS) is that a TRNG takes relatively longer than a PRNG to generate random bits. In reality, Intel AMT uses a PRNG whose state is occasionally reset to a state generated by the TRNG. This creates a powerful high-quality RNG that is able to keep up with the high usage of random numbers in the Intel AMT subsystem.

Secure Storage of Sensitive Data–Blob Service

As mentioned earlier, the flash part contains the configuration data for Intel AMT, which stores some of the Intel AMT secrets. The flash controller prevents software applications and drivers running on the host processor from accessing the flash part. However, an attacker may be able to steal a platform, pull out the flash, and read its contents by using a flash reader. In this way, an attacker could secure a backdoor to the enterprise network by reading the secrets, or even modifying them before returning the flash to the compromised system.

The Blob Service is a firmware mechanism that allows for protection of sensitive data on the flash part. The pieces of information protected by the Blob Service are called *blobs*. In this context, protection of a blob may take the following forms:

- Encryption to prevent the attacker from reading the content of the blob.
- Integrity to prevent the attacker from modifying the content of the blob.
- Anti-replay to prevent the attacker from reading an encrypted/integrity-protected blob (with a value known to the attacker), and later reusing it as-is while overriding a value unknown to the attacker.

Chipset Key

Encryption of secrets is achieved by using standard encryption techniques, but the interesting feature is the key that is used for the encryption. The encryption key needs to be stored in some non-volatile form, but the flash itself is obviously not a good place to store it (otherwise the attacker would first read this key from the flash and then use it to decrypt the rest of the protected data on the flash). Rather, the Intel AMT hardware contains a key that is unique to each system, and it is known to the Intel AMT firmware only. This key is called the *chipset key*.

The chipset key is actually a set of 128 fuses. Each fuse can be blown or un-blown, corresponding to a 0 or 1 value. The 128-fuse-set thus creates a 128-bit encryption key.

The status of each of the fuses (0 or 1) is determined at manufacturing. A random subset of the fuses is blown on the chipset manufacturing line, while the rest remain un-blown. Thus, a random unique value is created for each chipset.

A similar technique is used to generate the integrity key for the integrity part of the Blob Service.

Monotonic Counters

The Intel AMT hardware contains a few registers that implement simple counters. The counters are incremented by the firmware. Those registers are unique in the sense that they are powered by the platform coin battery (also known as the RTC battery, as it powers the platform Real Time Clock). Therefore, the counters retain their value as long as the battery is functional, which typically is for a few years.

To implement anti-replay, the value of the counter is incremented, then appended to the blob before applying the integrity algorithm. When the blob is read by the firmware, the value of the counter in the blob is compared to the value in the register. If they match, only then is the value considered valid. As long as the counter register is not reset (either by wraparound of the counter or by the replacement of the coin battery), the value of the counter is unique and therefore the anti-replay is achieved.

The algorithm described here requires a separate hardware register for each blob that needs to be anti-replay protected. In fact we can take this method one step further. Only one blob (let us call it the *counter blob*) in the system will be protected by one hardware counter only. But the counter blob can contain counter values for other blobs. Whenever an anti-replay protected blob is modified, its private counter is incremented; this means that the counter blob is modified, which requires incrementing the hardware counter. Therefore, the counter blob helps us reduce several counters to a single counter, maintained by the hardware and protected by the coin battery.

When the battery is replaced, the anti-replay protected blobs are invalidated. In some cases, this will require the user to reinsert some of the secrets protected by the anti-replay blob service.

Note that we assume that every anti-replay protected blob is also integrity-protected. This assumption makes perfect sense: an anti-replay blob contains a unique value that prevents it from being replayed. If the blob is not integrity-protected, the unique value can be modified, and therefore the anti-replay quality is also lost.

A sample list of data blobs protected by Intel AMT, by using the Blob Service, is given in Table 1.

Table 1: Sample list of data blobs

Intel® AMT Data Structure	Encrypted	Integrity Protected	Anti-Replay Protected
Usernames and hashed passwords	No	Yes	No
Permissions and Access Control Lists	No	Yes	No
Certificates	No	Yes	No
Kerberos keys and attributes	Yes	Yes	No
Private portions of Asymmetric Key Pairs	Yes	Yes	No
Integrated TPM secrets	Yes	Yes	Yes

Measured Firmware

Someone not familiar with the concept of measured launch may ask what it means, and why it is necessary. We begin by trying to answer these questions. We would love to be able to say that the Intel AMT firmware is free of design and implementation bugs. The reality is that there is no such thing as bug-free code. Even some of the most critical systems, such as airplane cockpit software or air-traffic control software, have been known to have flaws. Therefore, the entire computer industry depends on a perpetual cycle of bug-fixing and software updates (also known as patches) to fix the known flaws. By doing this, we eliminate the known flaws at least. However, finding new vulnerabilities is not an easy task. Attackers (and researchers, who are good guys) spend a lot of time reverse engineering software to discover new flaws. The most common way of compromising software is to attack it by using malware: malware makes use of known vulnerabilities in a system, usually a system that has not been kept up to date with patches. It is, therefore, very important to keep any software updated with the latest updates, especially the security updates.

What's even worse is the inability to be able to know which version of the software is running. Bugs in the design or code make the software vulnerable to being exploited by malware. The first thing the smartest malware tries to do is to hide itself.

Software running on the host processor, such as the BIOS or virtual machine monitor (VMM) loader, needs to be able to measure any code running on the platform. This measurement can be used to find out if every piece of code running on the platform can be found on a "whitelist" (a list of accepted versions) of software that was previously verified by the platform owner [7].

The existing features in the chipset that come closest to providing this functionality are the firmware signing and flash protection mechanisms discussed earlier.

As stated, the Intel AMT processor verifies the firmware signature at initialization time, by using the verification logic in the ROM code, and the measurement done during the signature verification process is not recorded anywhere in the platform for subsequent evaluation. Therefore, the host processor has no role to play in this verification, thereby leaving the host-based VMM loader or the BIOS with no capability to assess the validity of the firmware at any later point in time, after the platform powers on. Consequently, the host software cannot enforce any policy that depends on the evaluation of the Intel AMT firmware measurement to enforce certain system behavior.

The other problem with signed firmware images is that all images (belonging to a particular chipset generation) are valid on that chipset, because they are signed by the Intel code-signing private key (the corresponding public key being embedded in the ROM). Therefore, the existing firmware signature verification mechanism makes no distinction between the various versions of the firmware images that may have been produced by Intel for that product generation/family.

The Intel AMT firmware measurement mechanism solves these issues by providing a direct mechanism for reading the firmware measurement, by using host-based software (such as BIOS, VMM Loader, OS, or OS agent), without imposing the burden of the knowledge of firmware address location or offsets in the flash, on the host software. The design of the architecture ensures that the measurement is always completed, recorded, and locked inside the Intel AMT processor, before Intel AMT firmware execution begins. This measurement is also available while the Intel ME is powered on. Therefore, it is not possible to overwrite this measurement value after Intel AMT firmware execution begins. This aspect of the firmware is guaranteed by the hardware of the Intel AMT processor. So, even if there is a vulnerability (known or unknown) that somehow exists in the Intel AMT firmware, it is impossible for malware to exploit this vulnerability and thereby modify the previously-recorded measurement value. By definition, the measurement will be different for each firmware version, thereby giving the host a mechanism to cryptographically assert the measurement of the firmware image. The Intel AMT firmware measurement value is readable by the host software (such as BIOS, or VMM Loader) via the PCI configuration space of the Host Embedded Controller Interface (HECI) device (the Intel AMT processor is visible to the host OS as a HECI device).

Audit Log

Intel vPro technology creates a powerful tool for the network administrator to control the network entities. However, being in possession of a powerful tool comes with risks: the risk of erroneous use of this tool, and more importantly, the risk of malicious use of this tool. Rogue insiders are becoming a real threat to worldwide governments and enterprises, as demonstrated by a recent San Francisco network lockout [8].

A legitimate insider such as a network administrator already has very powerful credentials to access sources of business-critical information in an enterprise. Unfortunately, if such administrators turn against the enterprise, they become rogue insiders, and prevention of malicious use of a privileged system is nearly impossible. However, the risk can still be mitigated, by using deterrence mechanisms, and this is where auditing capabilities comes into play. [9, 10]

The Intel AMT audit log (see Figure 3) is an internal log that captures the administrator's operations in the system, and also captures unauthorized accesses to the system. When a security breach is discovered, the audit log can assist in tracking down the administrator that may have caused the breach.

The auditing capability cannot prevent system administrators from misusing the system, but it will prevent them from covering their tracks. The Intel AMT auditing subsystem allows an enterprise, or the authorities, to follow the steps of the administrator responsible for misusing the system, in a manner that is provable and undeniable. Having such a mechanism in place can deter an administrator from misusing the system in the first place.

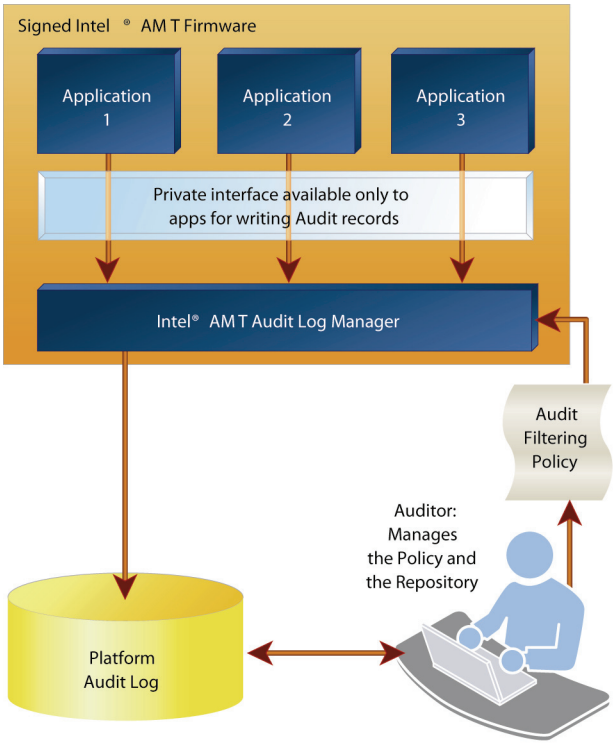

Figure 3: Intel® AMT audit log
Source: Intel Corporation, 2008

Separation of Duties

An audited system has both an administrator and an auditor role. The auditor controls the audit log policies and contents. In many cases, an enterprise will outsource their auditing services to an impartial third-party company that provides auditing services. A separation of duties is required to create a true audited system, one that cannot be tampered with by the internal enterprise administrators.

The separation of duties concept is adhered to in the credential mechanism embedded in the audit log subsystem. While the administrator might usually be omnipotent where the system is concerned, the audit log is outside of his or her purview. The administrator should not have the credentials to clear the log, modify the auditing policies, or modify the auditor's access credentials. The auditor, in turn, should only be given enough privileges to manage the audit log. Conversely, administrative operations in the system are typically outside of the auditor's purview. This is the concept of "two person controls." Thus, in order to compromise a network, and escape undetected, the administrator and auditor would have to collaborate.

Audit Log Records

A record in the audit log represents an administrative operation on the system. The record contains the following information:

- Identifier of the operation being logged.
- Access control credentials (username) that were used for the operation.
- IP address of the management console that initiated the operation.
- Timestamp of the operation.
- Additional information specific to the operation, if applicable.

Posting an Event to the Log

For an enterprise to claim it maintains a security log, the sequence of records in the audit log must match what transpired in the system. This means that if the administrator initiated an operation that should be logged in the audit log according to the policy, and the log entry could not be written because the log was full (an extremely rare scenario, which as we explain later, we try to prevent from occurring at all costs), then the operation fails. When the log needs to be retrieved, the auditor can be certain that no auditable operations occurred in the system other than those written in the log.

Auditing Policy

The auditing policy defines which administrative operations should be logged in the audit log. Operations can be defined in the policy as "critical" or "noncritical," and how operations are defined is crucial to a proper balance of security and usability within an enterprise. Critical events will always be logged; if a critical operation occurs and the log is full, the operation will fail. Noncritical operations will be logged only if the log is at least 20 percent empty. When the log is nearly full and a noncritical operation occurs, the entry will not be logged and the operation will not fail. Noncritical operations are logged as space permits. The last 20 percent of the log is reserved for critical operations only. A true, foolproof audit is therefore performed only on critical operations.

The Audit Trail

Due to the limited capacity of the Intel AMT flash, and the usability concerns described earlier, the auditor needs to clear the log periodically. Before clearing the log, the auditor requests an audit trail: this is the current contents of the log, signed by the firmware auditing service in a way that can later be verified by the auditor. The auditor can store the trail in long-term storage, such that if a breach occurs later, the old log can still be retrieved. The signature can attest to the fact that the logs have not been

tampered with while in long-term storage. Figure 4 illustrates the structure of the audit log trail.

Two potential problems may arise with this approach. The first is the ability of an administrator to delete an entire signed trail from long-term storage. This issue may be addressed by adding an incremental counter to the signed trails. This allows the auditor to make sure that all signed logs are in place. In addition, the enterprise administrator should not have access to long-term storage, but a discussion of this issue is beyond the scope of this article.

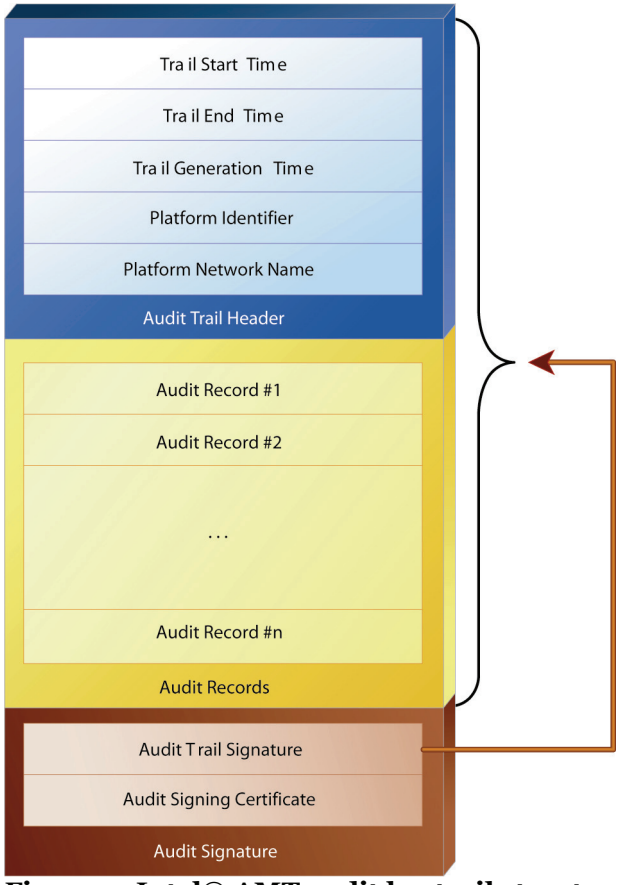

Figure 4: Intel® AMT audit log trail structure
Source: Intel Corporation, 2008

A second issue that may arise from this approach is the revocation of the keying material used to sign the audit trail. It is recommended that an additional signature be added to the auditing software by means of a temporal certificate being added to the trail before it is stored in long-term storage. When this certificate is replaced periodically, the logs in long-term storage should be re-signed. This adds another layer of authentication in case the keying material used to sign the audit trail is compromised or revoked.

This kind of an auditing system provides very robust protections against rogue-insider attacks.

Conclusion

Because security and manageability go hand in hand, the secure implementation of Intel AMT is critical to reducing the total cost of ownership of the platform. In this article, we just touched on a few of the security features implemented in the Intel AMT platform; however, we hope we helped illuminate some of the more interesting and powerful ones.

References

[1] "HTTP Digest RFC." At http://www.ietf.org/rfc/rfc2617

[2] "HTTP definition in RFC 2616." At http://www.w3.org/Protocols/rfc2616/rfc2616.html

[3] "HTTP Negotiate." At http://msdn.microsoft.com/en-us/library/ms995329.aspx

[4] "TLS RFC." At http://www.ietf.org/rfc/rfc4346.txt
802.1x: http://www.ieee802.org/1/pages/802.1x.html

[5] Dori Eldar et al. "Intel AMT Configuration." *Intel Technology Journal*, Volume 12, Issue 4, 2008.

[6] At http://en.wikipedia.org/wiki/Pseudorandom_number_generator

[7] "Intel Trusted Execution Technology." At http://www.intel.com/technology/security/

[8] At http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2008/07/14/BAOS11P1M5.DTL&tsp=1

[9] "Windows Event Logs." At http://technet.microsoft.com/en-us/library/bb726966.aspx

[10]"XDAS." At http://www.opengroup.org/security/das/xdas_int.htm

Authors' Biographies

Omer Levy joined Intel in 1999 during his studies in the Hebrew University of Jerusalem, Israel, where he received a B.Sc. degree in computer sciences and mathematics. He was part of the development of various software and firmware projects, including the first generations of Intel AMT firmware, before joining the Intel AMT architecture team as a firmware architect.

Arvind Kumar is Principal Engineer and Chief Manageability Architect at Intel. During his 14 years with the company, Arvind has worked on different manageability products, including server, blades, desktop, and mobile systems, and he is currently driving long-term common manageability architecture. Arvind represents Intel in DMTF and has been active in industry manageability initiatives such as Common Diagnostics Model (CDM), Systems Management Architecture for Server Hardware (SMASH), and Desktop and mobile Architecture for System Hardware (DASH).

Purushottam Goel graduated from the University of Roorkee, India (now called the Indian Institute of Technology) with a bachelor's degree in electrical engineering and went on to obtain a master's degree from the computer engineering department of BITS, Pilani, India. He worked at the Bangalore R&D Center of Novell, Inc. from 1996 to 2000 working on various projects, most notably on the security components of the NetWare* operating system. Subsequently he tried his luck in a couple of startups, before joining Intel in 2002. Purushottam is one of the key architects of Intel AMT. He is responsible for all security features and aspects of the product, including the design of the provisioning and setup mechanism of Intel AMT.

Storage Protection with Intel® Anti-Theft Technology - Data Protection (Intel® AT-d)

Ned Smith, Principal Engineer, Intel Corporation

Keywords: Data-at-rest protection (DAR), full disk encryption, key management, embedded security.

Abstract

Data-at-rest (DAR) encryption, embedded in peripheral controller hardware, combines the security, reliability, and performance benefits of storage device encryption, but it doesn't impact the enterprise services necessary for low-cost operation and worker productivity. This kind of encryption is highly adaptable to the needs of new-use models that cache data in high-speed flash memory or that stream data over network-attached storage, while utilizing the established enterprise management infrastructure.

Introduction

On October 8, 1871, a fire began in a small wood shed in Chicago, Illinois, and for a variety of reasons it spread to destroy more than 2000 acres of prime downtown real estate. This fire became known as the Great Chicago Fire, one of the most infamous fires of the 19th century. Following this disaster, the city's fire-prevention standards were reformed, and building codes were later put into place.

Preventing data theft and accidental disclosure of proprietary information is analogous to good fire-prevention measures. It requires a small upfront investment to ensure devastating losses won't occur later.

Data-at-rest (DAR) encryption technology prevents the unauthorized use of data stored on lost or stolen storage devices, thereby preventing these data from being spread on the Internet or other networks. DAR encryption is like placing your valuables in a fireproof safe; even if the surrounding building burns to the ground, the valuables inside are still safe. It is inevitable that stored data will get lost or stolen. The only reliable way of protecting data at rest from these threats is encryption.

In the United States, the Health Insurance Portability and Accountability Act [1] and the Sarbanes-Oxley Act [2], as well as numerous state regulations, constitute the "building codes" that mandate adoption of DAR encryption. More recently, the state of Nevada, for example, has strengthened regulations to require encryption unconditionally [3]. Many other state laws allow user notification as a substitute for encryption.

Moreover, similar regulations exist in other countries: the European Union's Data Protection Directive [4], Japan's Personal Information Privacy Act [5], and Canada's Personal Information Protection and Electronic Documents Act. All of these laws were put in place to help protect DAR.

Challenges to Protecting DAR

In response to these regulations, IT professionals face the challenging task of incorporating DAR encryption technology [6] into their systems to protect sensitive data. Corporate data are increasingly mobile, distributed, and prolific. Data are routinely taken out of physically secured facilities to accommodate workers who travel or have flexible working habits. Data are also distributed geographically as corporations' business interests take them into other cities, states, and countries. Data are accumulating at a high rate and are being stored on a widening variety of storage media. All of these forces drive the evolution of new storage media, higher bandwidth subsystems, and network-connected storage that blend data-in-flight (DIF) technology with DAR technology—a combination that did not exist in the past. As a result, IT professionals, seeking to apply comprehensive DAR protection to their systems, must make tradeoffs between worker productivity, effectiveness of the solutions, and cost to the corporation.

Privacy regulations hold company executives responsible for malfeasance, resulting in the need for companies to audit DAR system operations. Audit

trails offer evidence of compliance with regulations and therefore protect corporations from potentially expensive and damaging law suits.

Worker productivity is impacted if data are not readily available. Consequently, access to data storage resources must be made available, but also controlled, a process that often involves the integration of encryption technology into identity management systems. Moreover, encryption keys are integral to the use of encryption technology. The management of these keys requires that they be integrated into identity and resource management systems in order for DAR technology to be effective.

All of these services, audit, storage, and identity management, integrated into a traditional manageability infrastructure, contribute to the deployment and operational costs of support services for businesses and thus are a significant part of the Total Cost of Ownership (TCO) of DAR solutions.

IT managers also need to consolidate DAR services to benefit from economies of scale, through centralized user identity, authorization, and asset and key management services. Centralization offers greater assurance that information security policies are uniformly applied while also minimizing system maintenance costs.

Approaches to DAR on Client Platforms

DAR technology for client computers generally falls into four categories, identified by where encryption is applied: 1) software-only, 2) storage devices, 3) storage controller, and 4) remote storage.

Software-Only Encryption

Central processing unit (CPU) cycles are used in software-only encryption to perform encryption operations. The DAR module must be inserted into the data storage path: this is done either above the file system, by hooking file system read and write interfaces into the DAR module; or below the file system, by intercepting device reads and writes at the driver layer, such as at the Advanced Host Controller Interface (AHCI) driver (see Figure 1).

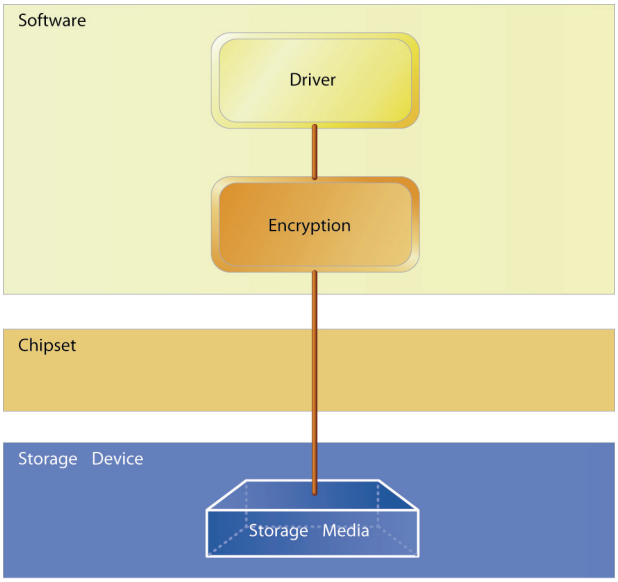

Figure 1: Software-only encryption model

Software-only encryption can be easily made to work with a variety of storage interfaces and media types, especially if configured by an operating system (OS) vendor. However, system and application errors can cause the encryption function to be bypassed or the audit trail to be omitted. Encryption overhead is borne solely by the CPU, which can affect performance for the end user.

Storage Device Encryption

Storage device encryption (see Figure 2) works by means of a hard-drive microcontroller, or by means of dedicated encryption hardware that is integrated into the drive controller. Device encryption also depends on external software to provide user authentication, key management, and all other support services. Storage device encryption is transparent to the OS and applications. In the case of storage device encryption, the overhead is not borne by the CPU, further minimizing the impact to the OS and applications. Encryption key management, audit, and access-control software is written in such a way that it assumes data are encrypted, but it cannot be certain that this is the case. Likewise, the storage device logic relies on external services and software to function properly, but it cannot be certain that this is the case.

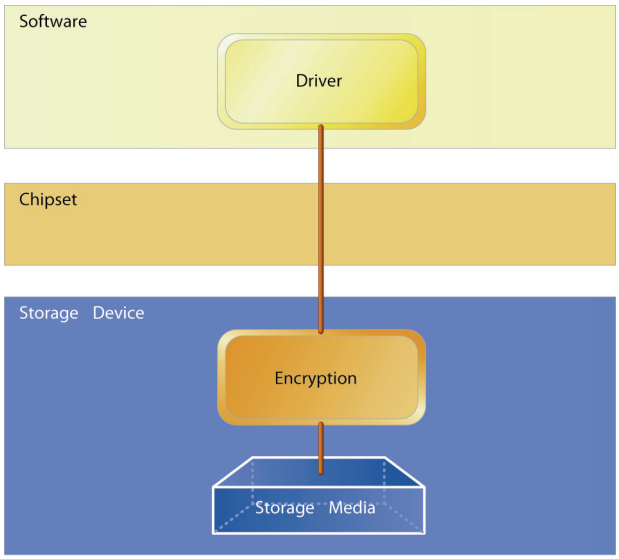

Figure 2: Storage device encryption model

Storage Controller Encryption

Storage controller encryption (see Figure 3) utilizes host controller hardware or dedicated encryption hardware to encrypt. The host controller decodes commands in the storage data stream to locate data packets that are then encrypted and repackaged before being sent to the storage device. Storage controller encryption depends on software or firmware for user authentication, encryption key management, and support services.

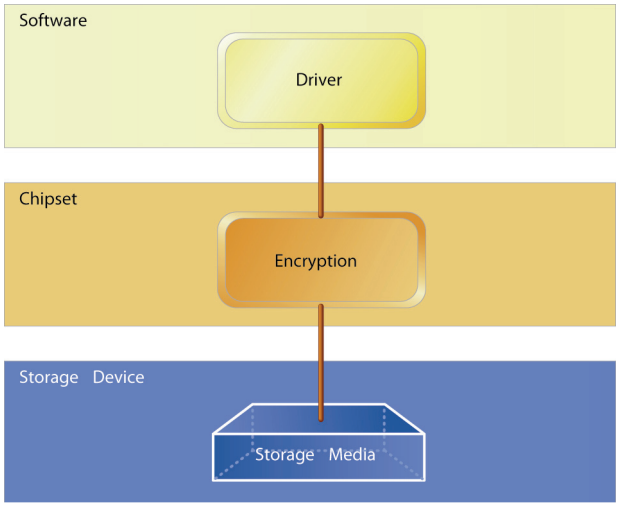

Figure 3: Storage controller encryption model

Remote Storage Encryption

Remote storage encryption (see Figure 4) relies on storage protocol redirection over a network interface, such as Intelligent Drive Electronics Redirection (IDE-R) or Internet Small Computer System Interface (iSCSI). Any of the previously-mentioned DAR encryption techniques can be applied prior to network redirection. Data are protected by using policies and authorizations pertaining to the client—as if locally stored. Local storage may be used to cache data that are later synchronized with a remote storage device. Network security may be applied in addition to local storage encryption to protect against certain types of man in the middle (MITM) and denial of service (DOS) attacks that are unique to networks.

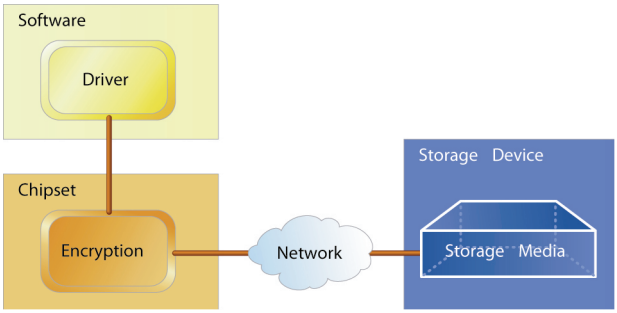

Figure 4: Remote storage encryption model

The Intel Data-at-Rest Solution

Intel® Anti-Theft Technology - Data Protection (Intel AT-d), employs a storage controller encryption technique to encrypt Serial Advanced Technology Attachment (SATA) data streams. Intel AT-d is part of a suite of technologies under the Intel® vPro™ brand that includes firmware for improved management, network connectivity, and system performance. Intel AT-d leverages the manageability features of the Intel Management Engine (Intel ME) to ameliorate many of the challenges inherent in the protection of DAR.

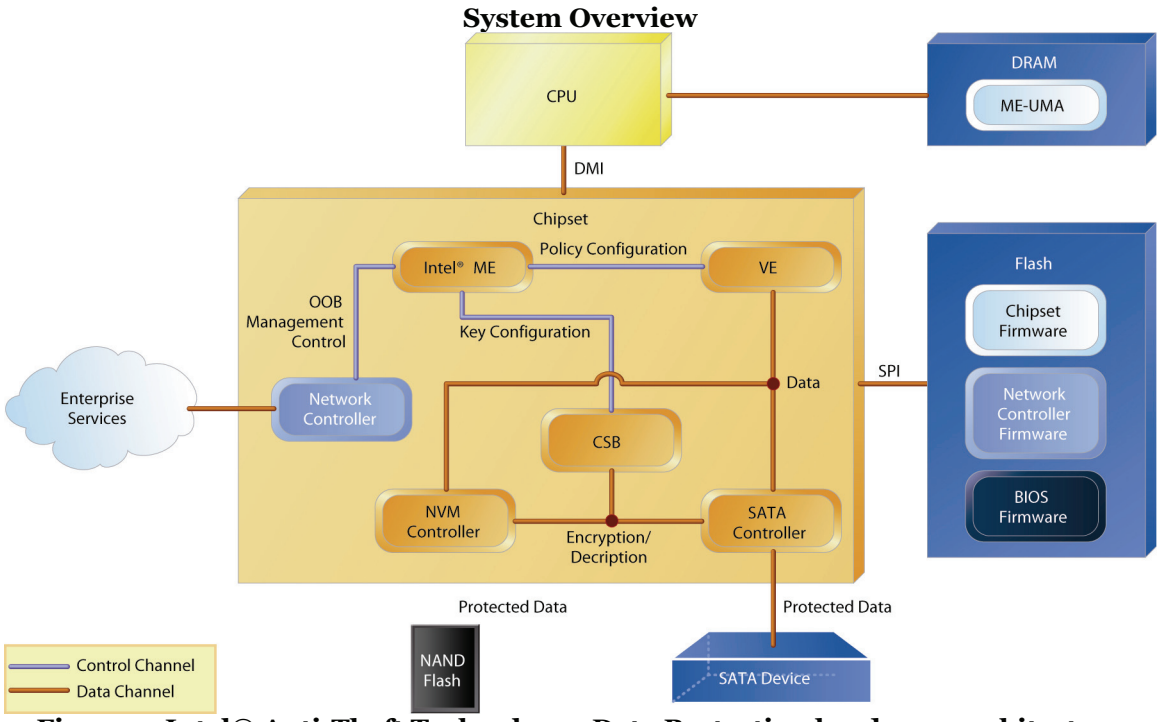

Figure 5: Intel® Anti-Theft Technology - Data Protection hardware architecture

Figure 5 shows the components of the chipset that implement DAR protection. The Crypto Services Block (CSB) is a hardware implementation of the Advanced Encryption Standard (AES) and it supports key sizes of 128 and 256 bits. The Virtualization Engine (VE) is comprised of a general-purpose controller that performs SATA command decoding and other accelerated operations by using dedicated silicon. The Intel ME controls the behavior of the VE and CSB by configuring policies and keys. The Intel ME also collects audit events, manages user authentication, and interfaces with enterprise services. The integrity of the firmware for both the Intel ME and VE is ensured by means of a digital signature before it is stored on the Serial Peripheral Interface (SPI) flash memory. User Access Control Lists (ACL) are encrypted and stored in a data region of SPI flash memory. A portion of the SATA device is reserved for DAR metadata that contains the Disk Encryption Key (DEK) that encrypts data on the disk. Data can be cached for performance improvements on platforms containing Nonvolatile Memory (NVM). Data stored in NVM persist after power is removed from the system; hence, they must be encrypted to be protected from theft. When the platform is fully powered, a portion of DRAM, known as ME-UMA, is available for use by Intel ME. The host OS is not able to access ME-UMA memory, in general, because of a memory isolation mechanism that is configured by the Basic Input Output System (BIOS) and then locked before the OS runs.

Encryption is applied to a data write operation as follows:

1. The AHCI driver issues a data write command to the VE.

2. The VE decodes the command and identifies the data portions to the SATA controller.

3. The SATA controller routes data on the fly through CSB, which encrypts the data by using the key previously supplied by the Intel ME.

4. The CSB returns the data to the SATA controller for transmission across the SATA interface to the storage device.

A read command is the reverse of a write command, but with decryption. Before any encryption or decryption can be performed, the Intel ME must insert the DEK into a memory register in the CSB. The CSB contains registers for up to six encrypted SATA devices. When storage devices are enumerated, the VE checks for the existence of Intel AT-d metadata on the disk. If found, the metadata are returned to the Intel ME to be decrypted. The

DEK is wrapped (further encrypted) by a key that is derived from two values dynamically obtained. The first value is a passphrase obtained from a user. The second value is a chipset key that was embedded in hardware at the time of manufacture. Each chipset key is unique to a specific platform, but the key cannot be read externally. The passphrase and chipset key are supplied to a Public-Key Cryptographic Standard (PKCS#5) key derivation function that outputs the DEK wrapping key. Because the wrapping key is derived each time the system is powered up, a thief must know the user passphrase and have access to the chipset key in order to obtain the DEK.

Storage devices protected with Intel AT-d are therefore bound to the platform that encrypted the data. This prevents an attacker from putting the drive into another platform on which an attack tool kit could perform a variety of cracking techniques.

Audits are triggered whenever a function is performed that could affect compliance with privacy regulations. Auditable events include enabling/disabling encryption, changing encryption keys, modifying key strengths, successful and failed user log-on attempts, key recovery, and remote unlock operations.

Intel AT-d is compatible with Intel® Matrix Storage Technology implementation of the Redundant Array of Inexpensive Drives (RAID). The RAID abstraction is applied after the Intel ME unlocks the drives participating in the RAID array.

All data on an Intel AT-d protected drive are encrypted, except for the Intel ME metadata and pre-boot authentication (PBA) metadata areas (see Figure 15), which remain unencrypted. This includes Master Boot Record (MBR), RAID metadata, and OS and user data. Fully encrypting the drive protects sensitive data included in paging and configuration files, and it prevents offline attacker manipulation of system files by a toolkit.

Fully encrypting the drive also presents challenges to IT organizations. An encrypted MBR prevents recovery operating systems from loading until the disk is unlocked. Encrypted RAID metadata can prevent initialization of the RAID array as well. User authentication, therefore, must occur before any pre-boot service that requires drive access.

User authentication is facilitated by host software in the pre-boot environment. A pre-boot authentication module interfaces with users to obtain passphrase or other credentials that are given to the Intel ME for validation, evidenced by successful unwrapping of the DEK.

The Extensible Firmware Interface (EFI)[7], the next generation of system firmware, supports driver dependency declarations, useful for staging system bring-up actions to account for DAR encryption constraints. The EFI framework supports many other features that make it easier for vendors of multifactor authentication components to work together to provide a rich and robust pre-boot authentication solution. This allows vendors to specialize by using best-of-breed capabilities in their products.

Crypto Services Block

The crypto services block (CSB) implements in silicon the AES algorithm using the LRW mode (see Figure 6 and [8] for more on this mode). There are three cryptographic engines that can be multiplexed across a variable number of SATA ports.

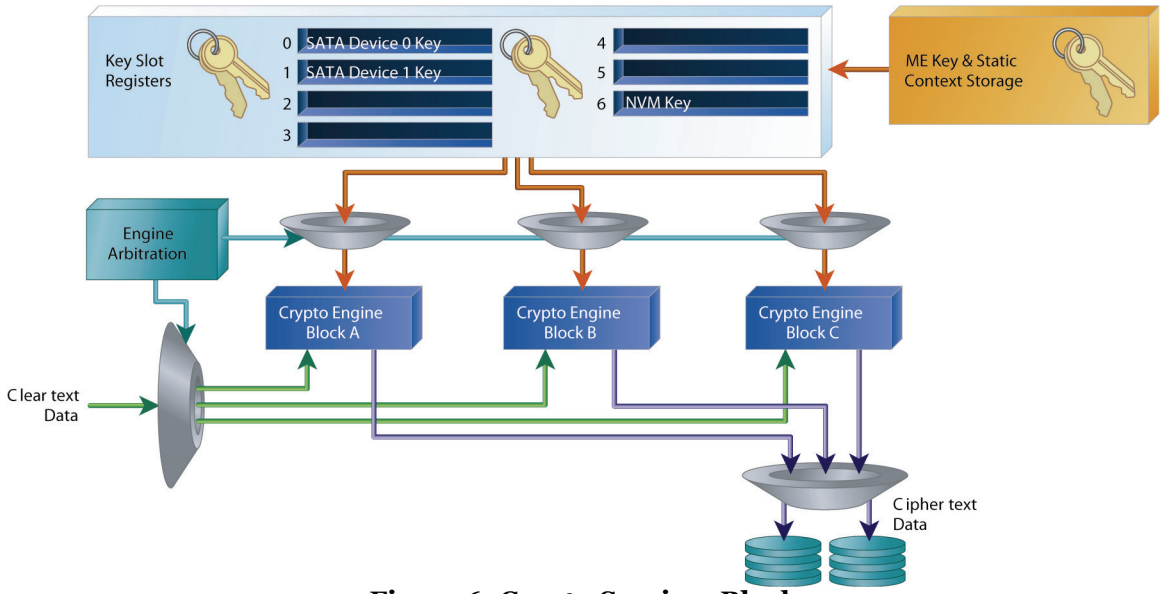

Figure 6: Crypto Services Block

There are six key slot registers that store disk encryption keys for fast access from any of the three crypto engines. Key slots are populated by the Intel ME at platform initialization and when new drives are detected. An Engine Arbitration module ensures the data input stream, encryption key, encryption engine, and the data output device are scheduled before processing write operations.

Each crypto engine block operates at or near line rates to ensure that it doesn't introduce any latency. This eliminates the need to multiplex a single stream over multiple crypto engines.

Virtualization Engine

Data enters the VE over the Desktop Management Interface (DME) and a virtual host controller such as the AHCI, which is used for SATA devices. The VE can support multiple virtual host controllers, but typically only a single virtual AHCI is usually needed.

The VE itself is an ARC* International 32-bit microcontroller that runs a ThreadX* real-time OS by using Express Logic* that runs firmware developed by Intel. The VE uses on-chip SRAM and can access system DRAM that is isolated from host CPU cores. SATA command-decode operations and other functions can be performed by dedicated silicon designed to accelerate command processing.

The CSB is accessed by the SATA Controller as data packets are available for encryption or decryption. The encrypted packets (assuming write operations) are then routed to the SATA interface by the physical host controller. The SATA storage device is unaware that data are encrypted; therefore, in theory, any SATA-compliant storage device could support Intel AT-d encryption (see Figure 7).

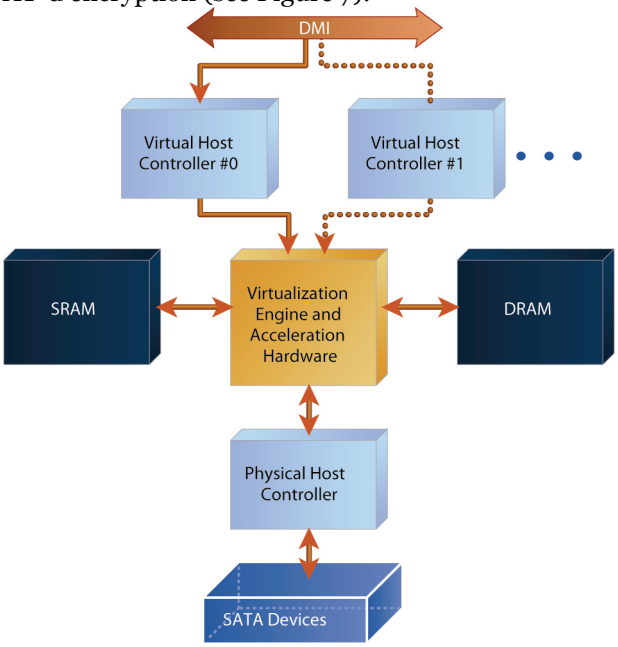

Figure 7: Virtualization Engine

Intel® Management Engine

Much like the VE, the Intel ME (see Figure 8) is an ARC International, 32-bit microcontroller that runs the ThreadX, real-time OS. Firmware developed by Intel implements key management, access control, and other support. The Intel ME can use both on-chip SRAM and DRAM that is isolated from the host CPU. Persistent data are stored in flash memory accessible by the SPI bus. All Intel AT-d metadata stored in SPI flash and on the drive are encrypted by using a platform container key (PCK) that uses

counter mode AES (AES-CTR) encryption (see [9] for more information on CTR).

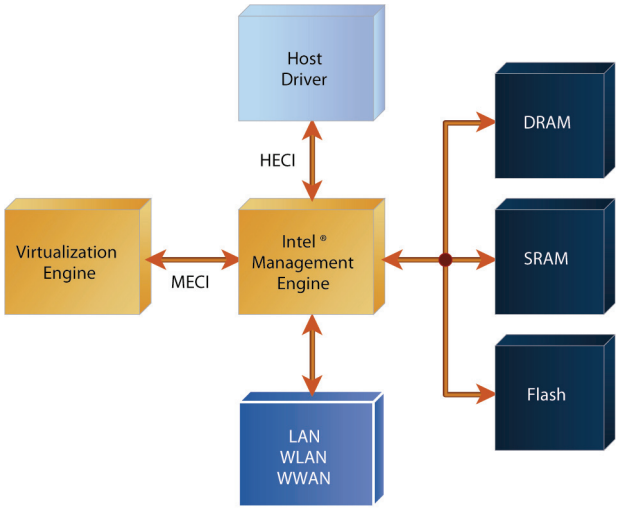

Figure 8: Intel® Management Engine

The Intel ME can control aspects of VE operation directly over the Intel ME Command Interface (MECI), which is an internal bus. The Intel ME may also access on-board networking interface devices during low-power states, such as Sleep mode, in addition to normal operation. The Intel ME shares network resources with the host OS, but the host is unaware of this unless special monitoring tools are used. The HECI is used by host drivers to communicate directly with the Intel ME.

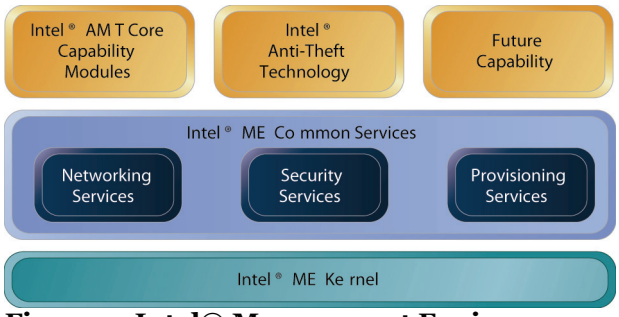

Figure 9: Intel® Management Engine common services

The Intel ME firmware is modular, that is, core capabilities function independently of others, and new capabilities may be easily added.

Functionality that is commonly used by multiple capabilities is called Intel ME Common Services (CS) and consists of three major parts: Networking Services, Security Services, and Provisioning Services (see Figure 9).

Networking services comprise a Transmission Transport Protocol/Internet Protocol (TCP/IP) stack, Transport Layer Security (TLS), Hypertext Transport Protocol (HTTP), Simple Object Access Protocol (SOAP), Web Services for Management (WS-MAN), and a host-based TLS interface called the Intel Local Manageability Service (LMS).

The TCP/IP stack supports either IPv4 or IPv6, depending on which technology generation is running. For IPv4, the host OS will share the same network address with the Intel ME. For IPv6, the Intel ME has its own IP address that is not shared with the host. We explore in more detail the services needed to support DAR encryption in the "Intel® Anti-Theft Technology - Data Protection Support Services" section of this article.

Security services provided by the Intel ME CS include the following:

- User authentication consisting of both HTTP Digest and Kerberos [10]
- Domain authorization using Microsoft Active Directory*
- Secure time
- Auditing

Provisioning services support two deployment modes: zero touch and one touch. With zero touch, deployment certificate anchor keys are embedded in the firmware, allowing well-known certificate authority keys to be used to validate IT credentials that can then be used to take ownership of the platform. One touch mode configures organizational certificates, symmetric keys, and trusted hosts that may be used to complete setup and deployment tasks remotely.

Intel At-d ME firmware implements the Storage Encryption Service (SES) that includes management of key storage hierarchy, drive migration, setup, configuration, drive conversion, user authentication, and access to devices.

Key Hierarchy and Management

Intel AT-d key hierarchy is a three-tiered hierarchy that uses either derived keys or externally escrowed keys to wrap a device group key (DGK) that in turn wraps one or more device encryption keys (DEK), one per encrypted device.

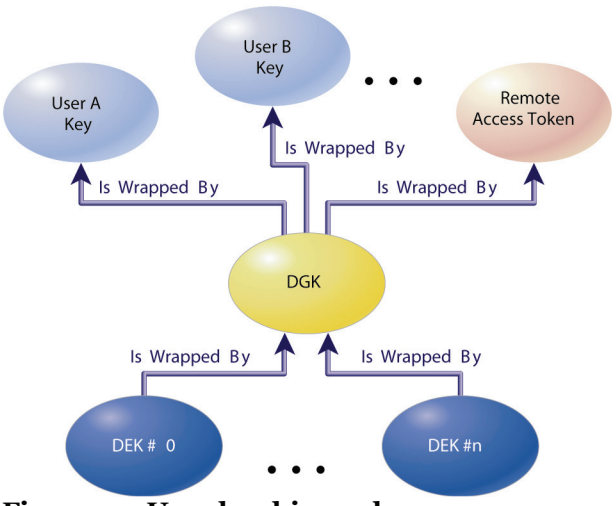

Figure 10: User key hierarchy

The user key hierarchy is shown in Figure 10. Both the DGK and DEK are generated on the platform by using both a pseudo-random number generator (PRNG) and a true random number generator (TRNG) to seed the PRNG. The key hierarchy is stored as metadata in both SPI flash memory and the storage device.

Table 1: Key generation formulas

Key	Generation Formula
DEK	PRNG(TRNG() = seed)
DGK	PRNG(TRNG() = seed)
Fuse Key	Manufacturer blown fuses
PCK	$HMAC_{SHA256}$(FuseKey, Intel® Management Engine fuses, key-string)
User Keys	$HMAC_{SHA256}$(PCK, rand, PKCS#5(name, pin, string))
Security Questions (SQ) Keys	$HMAC_{SHA256}$(PCK, rand, PKCS#5(name, SHA256(answer1, answer2, answer3)))
RCK	PKCS#5(recovery token)

User keys are derived dynamically from user-supplied credentials. Up to six local accounts are supported. The PKCS#5 key derivation function combines user name, password, and a salt-string that is hashed, by using SHA-256 with the platform's unique fuse key to produce the wrapping key. The fuse key has 256-bits equivalent entropy to counter brute force cryptographic attacks.

A variant form of the user-derived key is based on the familiar authentication method of combining the answers to three security questions by using SHA-256 to create a value that is substituted for the password value described earlier.

For scenarios in which an administrative console is used to remotely unlock the drives, a random number is generated and combined with the fuse key value to produce a remote access token that wraps the DGK.

Each principal wraps a copy of the DGK so as to avoid requiring a combination of principals to be present before access is granted.

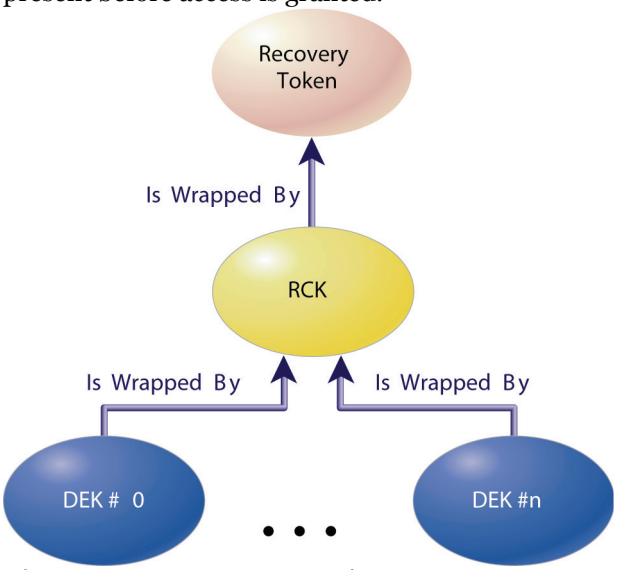

Figure 11: Recovery key hierarchy

A recovery token may be derived via PKCS#5 of a migration passphrase or it may be a random number generated by the recovery service. The migration token is used to wrap a Recovery Key (RCK) that in turn wraps the DEKs for each device (see Figure 11). The RCK can be random number generated by the recovery service or derived from the recovery token.

The generation of the RCK does not incorporate the fuse key value in order to facilitate cross-platform migrations. The migration token is stored safely in a key recovery server or other secure storage. Should drives require migration to a different platform, the RCK can be used to obtain copies of DEKs that are not bound to the old platform.

Access Control

Access to disks is controlled at the platform level, meaning all AT-d protected disks are made accessible when a user successfully authenticates. Finer-grain access control is possible, but it requires managing additional DGK keys. As the name implies, each DGK key is a member of a different group. By dividing drives into groups and wrapping their DEKs with the DGK keys, respectively, it would be possible to restrict access based on the user's group affiliation.

Access to disks should be local, remote, and unattended. In local access, Advanced Technology Attachment (ATA) commands are blocked by the VE before an authorized user has logged on. When a successful logon occurs, the Intel ME notifies the VE of the status change; then the VE permits access (see Figure 12).

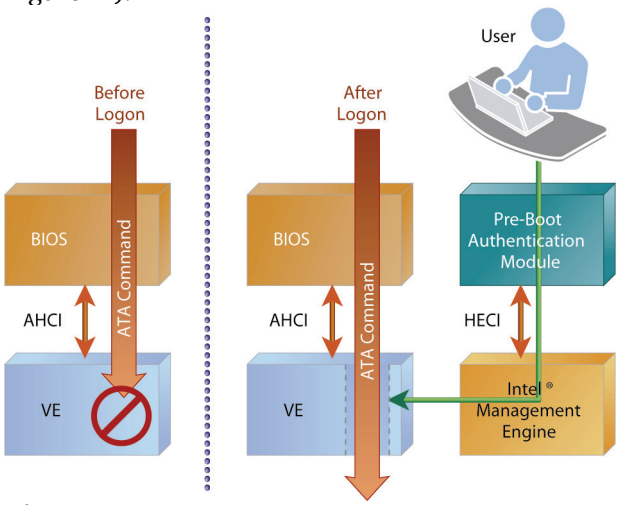

Figure 12: Local access control enforcement

Remote access control is similar to local control except the remote user must be authenticated by the Intel ME by using administrative credentials. Credentials could be in the form of a Kerberos ticket or an X.509 certificate. Authorization data in the credentials may also constrain the actions of the remote user, further preventing access. The remote user must also present a remote access token (see Figure 10) to the Intel ME so that the Intel ME can unwrap keys.

Unattended access is sometimes desired if the system is set to automatically reboot after a power failure or other event. The BIOS and Trusted Platform Module (TPM)[11], or other similar secure storage device, are required to support this scenario. The TPM is used to protect an unattended access PIN that is stored locally (see Figure 13). The BIOS will calculate system state and extend TPM configuration registers during reboot. If the system state matches the system state expected by the TPM, then the PIN is released. BIOS can supply the password to the Intel ME as if in the local access scenario (see Figure 10).

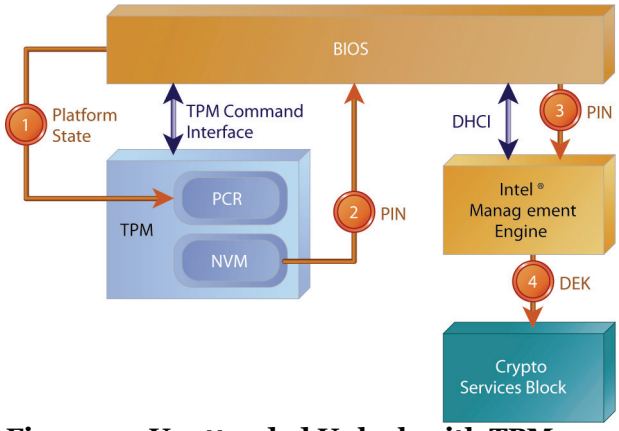

Figure 13: Unattended Unlock with TPM

If unattended access is required and an OS is stolen, it may be possible for encrypted drive contents to be read. Therefore, careful consideration should be given to this risk, before a system is configured for unattended access.

Many hard drives support drive locking by using ATA security commands. Locking the drive does not protect data from determined thieves; however, it does prevent casual data snooping. A locked drive also prevents the VE from accessing the device as the VE cannot completely initialize the device when it is locked. The VE does not have enough information to expose the device to the BIOS, without knowing the password. However, if the device remains hidden, the BIOS isn't programmed to issue the ATA security command to unlock the device (see Figure 14).

If user authentication occurs very early in pre-boot, even before drive identification, then user supplied Hard Disk Drive (HDD) passwords can be used to unlock drives transparently.

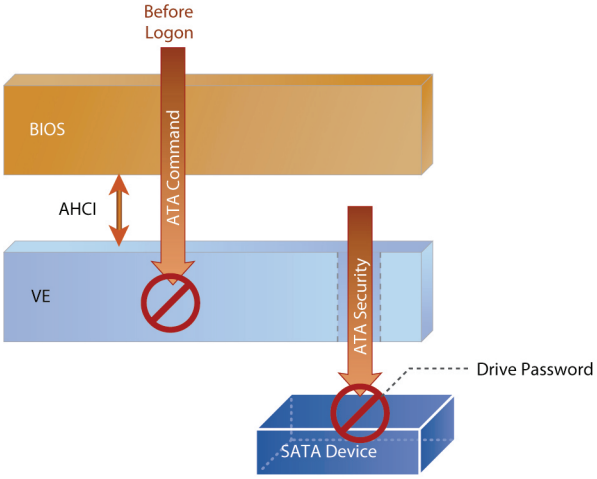

Figure 14: ATA Security Drive Unlock with Virtualization Engine

Drive Geometry

The VE virtualizes storage devices, so that multiple virtual drive partitions can be recognized. The vast majority of Intel AT-d platforms use a single virtual HDD partition (see Figure 15).

Contained within the first virtual HDD are all the traditional drive geometry elements. Beginning at Linear Block Address (LBA) zero, we have the Master Boot Record (MBR). This is followed by the drive data, such as the OS files and user files. Some systems have hidden partitions that may be used by BIOS or other system utilities. The Host Protected Area (HPA) can be used to store an emergency recovery OS (ROS), a multimedia utility, diagnostics utilities, or other programs. Systems with Intel Matrix Storage Technology subsystems that implement RAID place RAID metadata at the end of the virtual drive. By doing this, the RAID optional ROM can easily locate metadata at system initialization.

There is a single DEK for the drive that spans each virtual HDD, resulting in all virtual HDDs being encrypted with the same key.

The Virtual Drive Definition (VDD) data is placed at the end of the physical drive, LBA-n. VDD data contain drive geometry, marking the beginning and end of each virtual HDD. The VDD also identifies the start and end locations of the Intel ME metadata area. The VDD and Intel ME metadata are not encrypted by Intel AT-d. However, the contents of these areas are protected by the VE and Intel ME.

The Intel ME metadata consists of an AHCI file system block, Intel AT-d metadata, PBA code, and PBA metadata. The AHCI file system is used by an Intel ME firmware storage driver. Intel AT-d metadata contains the wrapped DEK, device configuration data, drive conversion status information, and the drive migration package. The migration package also contains a copy of the DEK wrapped with the RCK. Finally, Intel ME metadata contains PBA executables and a storage area.

Access to the PBA area is permitted via the VE by using the VE Command Interface (VECI), or via the Intel ME by using the Intel AT-d Host Command Interface (DHCI); which uses HECI. The VE can ensure that access requests outside the PBA ranges are prevented given that PBA code executes on the host processor.

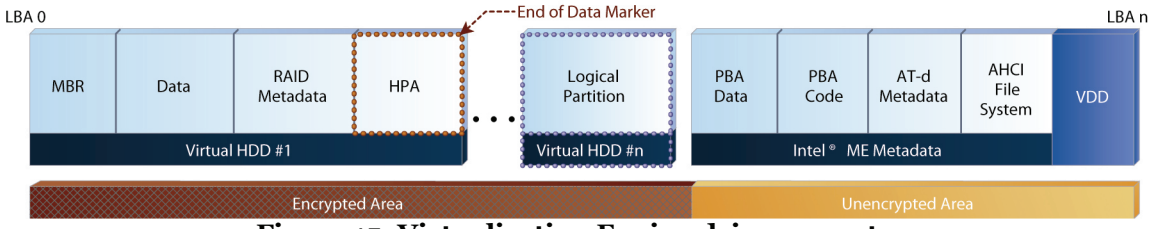

Figure 15: Virtualization Engine drive geometry

A challenging situation may arise when the HPA is used to store a backup copy of the BIOS, or when it is used to store BIOS extensions—and the HPA is encrypted. During pre-boot, before the disk has been unlocked, such features are unavailable.

The VE can expose the PBA data area to the BIOS for storage of a backup BIOS image or, in the case of Extensible Firmware Interface (EFI) BIOS, it can store EFI drivers that do not fit into the system flash memory. To accomplish this, the BIOS boot block or other BIOS drivers must recognize when the VE is accessing PBA data, versus accessing a typical hard drive. Two options exist for this scenario: either the BIOS will maintain two interfaces, one for each storage area, or the VE switches context underneath the AHCI interface when the drives are unlocked. BIOS may yet require a separate storage interface for PBA data, after drive unlock, if the BIOS need to access both areas of the drive simultaneously.

Figure 16 shows a backup EFI partition table located at the end of the drive, in the Intel ME metadata, that can be used to locate an unencrypted EFI partition (n) and an encrypted partition (x). Partition (n) is used before drives are unlocked for BIOS recovery, critical pre-boot authentication drivers, and any other BIOS code that must execute prior to the drive unlock event. Subsequent to drive unlock, partition (x) is available for EFI use.

Since EFI Partition (n) and the backup EFI partition table are located within the Intel ME metadata region, the BIOS must use a block I/O driver that is aware of drive virtualization, and software must be aware of the content stored in EFI Partition (n).

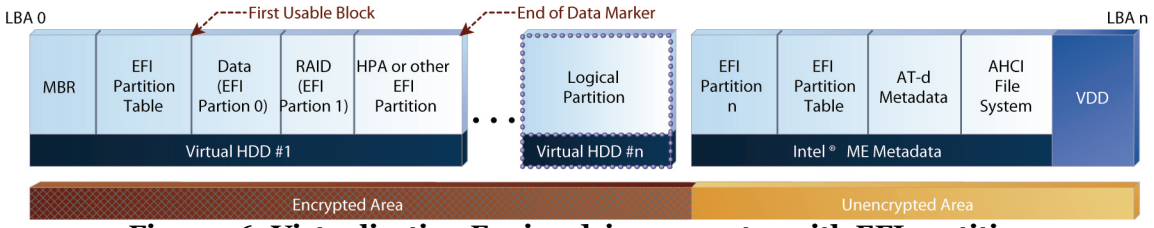

Figure 16: Virtualization Engine drive geometry with EFI partition

Drive Migration

Each encrypted drive contains a copy of a migration package that is used to move the drive to another platform. Recall that user ACLs wrap DEK by using keys that are bound to a particular platform's fuse key. The migration package, however, is not wrapped with a fuse key derivative. An escrowed key, also called the Migration Token (MT), is used instead. The MT can be stored by a key management service or simply in a personal storage device.

The Intel ME performs drive migration by doing the following operations:

- Generates RCK by using an MT for the previous platform.
- Decrypts the migration package.
- Reads the device configuration profile and DEK.
- Creates a migration blob for the new platform by using a new RCK.

Drives can be migrated between Intel AT-d platforms easily. Migration to other platforms requires an extra step that copies data to an unencrypted drive or partition.

Setup and Configuration

Before Intel AT-d can be used, Intel AT-d metadata must be instantiated and populated on the storage device and Intel AT-d platform flash. Device metadata partitioning is best created during manufacturing so that RAID metadata or user data is in the right place and does not have to be relocated. If the device metadata aren't partitioned during manufacturing, these data might be allocated to the end of the drive where Intel ME metadata need to be located. Platform manufacturers can also configure Intel ME CS for zero-touch provisioning and remote enrollment of local users and administrators.

Intel AT-d can be enabled by an administrator through the Intel ME BIOS Extension (Intel MEBX) utility in which user accounts and devices are configured for use. The first user created is a local administrator. The local administrator account is authorized to configure security policies, device recovery and migration, and user management. User accounts are also created with default credentials that a user can change upon first use.

If the drive had been previously in use and contains cleartext data, these data are converted (encrypted) in the background. The data conversion process is driven by the Intel ME. A systematic walk through each storage block on the device is read, encrypted, and rewritten over the cleartext. Flash memory storage devices require an additional step that clears blocks relocated by memory wear, leveling algorithms. Data conversion occurs in the background so users can continue working. The conversion algorithm is fault tolerant, so it will survive power failures.

The Intel ME performs data conversion operations. It does not have access to partition tables or file allocation tables (FAT) that may be helpful in distinguishing between data blocks and unallocated blocks; therefore, much of the conversion time may actually be spent encrypting empty blocks. Computers that don't have encryption enabled may have unallocated blocks containing deleted files that could be viewed by using data recovery tools. Therefore, it may not be safe to use drives in

environments where there is a risk of theft or loss until data conversion has completed.

Pre-boot Authentication

User authentication is a prerequisite for Intel AT-d drive unlock. Initialization of services, such as RAID, that can contain user data in metadata areas, must be encrypted. Since much of RAID initialization occurs in pre-boot, user authentication must also occur in pre-boot. In Figure 17, an abbreviated sequence of BIOS initialization steps is shown. Following PCI device enumeration and before RAID option ROM initialization, Intel AT-d with PBA option ROM runs and the user is prompted for log-on information.

The popularity of multifactor authentication is increasing because of its improved security properties and improved usability. Usually, the integration of multifactor authentication into pre-boot authentication must interface with several different authentication devices that are connected over a variety of I/O buses, therefore creating a need for a rich PBA environment. Version 2.2 of the EFI environment defines a rich PBA environment (see Figure 18). Each authentication device is abstracted through a credential-provider interface; other modules that abstract user identity and user profiles also exist. These, combined with drivers for accessing Intel AT-d services contained in Intel ME, offer tremendous flexibility in enabling multiple vendors, specializing in different aspects of pre-boot authentication, to create a compelling full-featured PBA application on top of an EFI framework.

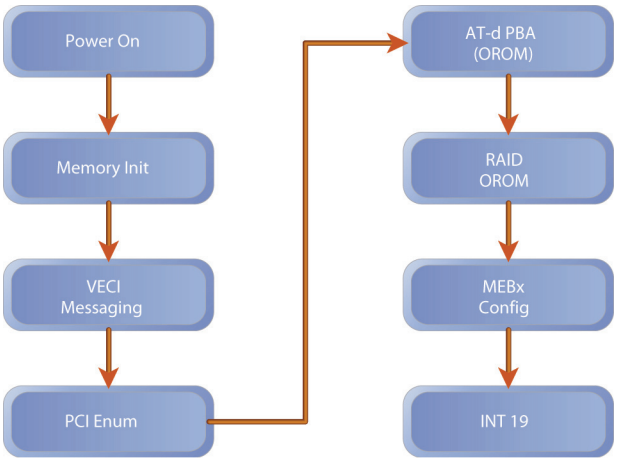

Figure 17: Legacy BIOS pre-boot authentication architecture

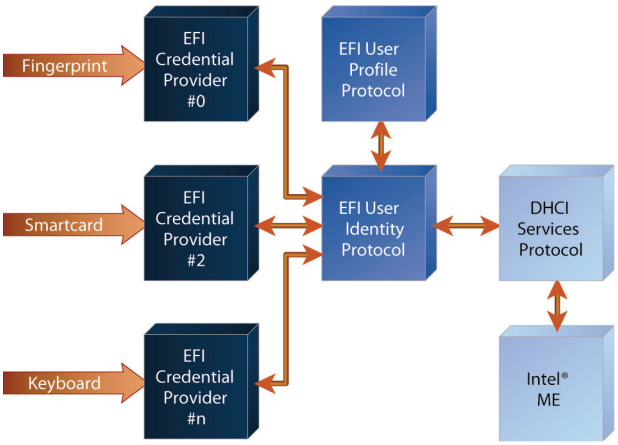

Figure 18: EFI pre-boot authentication architecture

Enterprise users may value an extensible pre-boot authentication environment because of the prospect of integrating domain authentication and single-sign-on. Many enterprises deploy Microsoft Active Directory for centralized user management. Linking user identity management for multifactor authentication with Kerberos domain authentication has desirable security properties, especially if deployment costs can be minimized.

Intel AT-d drive unlock that uses enterprise domain authentication has an added security advantage: existing IT procedures for managing user accounts also extend to local drive access, independent of OS version or type, something that challenges IT departments trying to deploy data-protection policies uniformly.

Performance Considerations

The benefit of improved DAR management through Intel ME and VE has a small but measurable impact on data throughput performance. Although data encryption and SATA packet manipulation can be performed nearly at line rates, they are not instantaneous. Short latencies are introduced to roundtrip I/O operations, where roundtrip refers to the time the I/O operation enters the host driver queue until the completed operation returns back to the host driver. For sequential block operations, VE latency is added to HDD latency (see Figure 19).

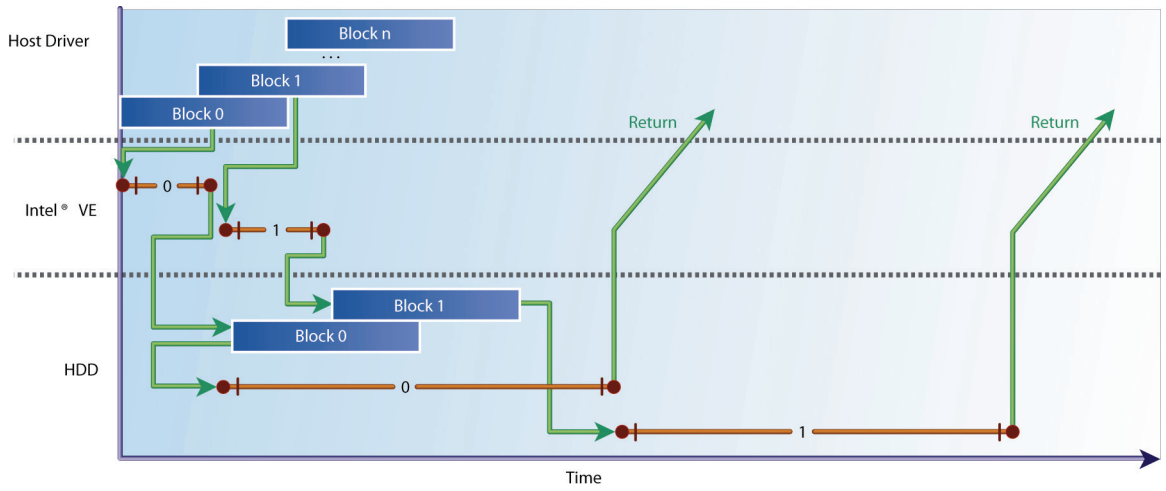

Figure 19: Single device I/O latency with Virtualization Engine

Multi-threaded I/O operations can be performed by using a RAID configuration. Block (n) processing can begin before Block (0) is completed by the HDD, which improves overall I/O latency significantly. However, the greatest opportunity for improving I/O performance rests with improving storage devices. The use of Solid State Devices (SSD) can dramatically improve performance, especially for small transfer sizes. The use of SSDs in a RAID array dramatically improves transfer rates for both large and small transfer sizes.

Data conversion performance is constrained by two factors: Intel ME horsepower and background host I/O activity. The ARC microcontroller operates at much slower speeds than typical host CPUs. Data conversion on the Intel ME is single-threaded, and it may be interrupted by higher-priority Intel ME tasks. Host I/O activity takes priority over data conversion; therefore, data conversion rates may vary.

Intel® AT-d Security Properties

Intel AT-d provides several security enhancements to traditional DAR solutions, because it is integrated into platform hardware. Intel AT-d uses a hardware RNG seed; the seed is supplied to a key generation algorithm implemented in Intel ME firmware that is compliant with the Federal Information Processing Standard (FIPS). Derived keys rely on entropy and uniqueness properties contained in a chipset key that is created at chipset manufacturing time by blowing fuses. It cannot be modified later.

Generated keys are protected in chipset memory when in use, making them immune to infamous Cold Boot attacks [12] where DRAM can be read even after the system is powered off.

Chipset firmware that implements critical key management, audit, and authentication operations is protected during execution in a hardware-defined isolated environment. Firmware integrity is verified both at the time it is provisioned to the platform and each time it is loaded. If firmware and metadata stored in platform flash memory are tampered with, such tampering is detected and the system may not execute.

Encryption and portions of SATA command decoding are implemented in silicon that has no external dependencies; hence, proper operation is ensured.

Integration of DAR support services in platform hardware ensures data protection policies are consistently applied regardless of OS, application, or storage-device choices. This lowers many operational costs, including IT security, audit, and risk assessment.

Intel® AT-d Support Services

Well-maintained systems are subject to periodic updating and reconfiguration by manageability personnel. The Intel ME supports remote administration of Intel AT-d by using an embedded network stack that includes TLS and the Kerberos authentication protocol, originally developed by the Massachusetts Institute of Technology (MIT).

Among the enterprise services needed to support DAR protection are these:

- Audit and compliance management.
- Key recovery management.
- User identity management.
- Platform configuration management.

The audit and compliance management service responds to audit log threshold events generated by the platform. The service archives client audit log files and resets the high water mark so that auditing con continue.

Before Intel AT-d encryption can begin, a copy of the DEK is stored in a key recovery service. Should the DEK on the drive become corrupted or lost, the key recovery service can restore it. Another copy of the DEK can be made by using a portable USB storage device. It allows data to be recovered when a key recovery service is not available.

There are many enterprise-class, user-identity management frameworks in use today. Common frameworks include Microsoft Active Directory, MIT-Kerberos, Public Key Infrastructure (PKI), Novell* Directory Services (NDS), and a variety of Web-based solutions. Intel AT-d maintains user account information for one local administrator and up to five users. The user accounts can be integrated with virtually any identity-management framework that supports the Intel AT-d programming interface. The WS-Man protocol is used to transport DHCI over a network to manageability consoles or gateway servers.

Manageability frameworks are used to perform a variety of management and administration duties remotely. Intel AT-d encryption introduces a dependency on management consoles that requires disks be unlocked before actions that involve access to storage media can be performed. Remote disk unlock is achieved by obtaining an unlock token from the key management service or other administrative service. The unlock token is used by the Intel ME to unwrap the DEK keys used to decrypt each drive. Following a drive unlock operation, the remote manageability processes can function normally.

Remote access poses a challenge for computers in satellite offices or in remote locations outside a corporate firewall. The Intel ME can traverse a corporate firewall with remote presence server technology that establishes a TLS Virtual Private Network (VPN) between the Intel ME and corporate Intranets (see Figure 20).

The computer, enabled with Intel vPro™ technology, contacts the Management Presence Server (MPS) by using pre-configured network domain information. TLS-VPN credentials, embedded in the client, support mutual authentication. Client-manageability traffic is forwarded to corporate Intranet servers over TCP/IP. Enterprises that support Kerberos Key Distribution Center (KDC) services can negotiate server tickets for the Intel ME, thereby allowing the management console to interact with computers containing Intel vPro technology, by using IT-managed privileges.

The MPS can proxy client credentials so that "Kerberized" services in the corporate network can be accessed with privileges appropriate for computers operating outside the corporate firewall.

The use of Kerberos tickets for service access is important, because authorization information can pass through the MPS. The auditing service can present domain credentials to the Intel ME, authorizing the administration of Intel AT-d audit logs, with the knowledge that other servers would be denied access by the Intel ME.

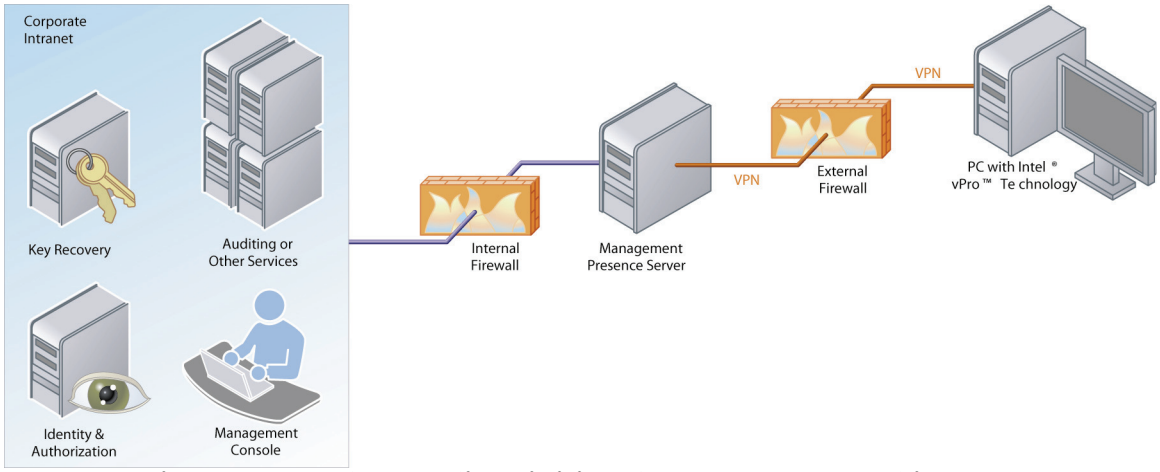

Figure 20: Intel® ME client-initiated remote access architecture

Occasional connection to DAR support services is necessary for enterprise-class operation of Intel AT-d. When not connected, however, operation continues normally by relying on the cached state maintained in the Intel ME data area of SPI flash memory. Cached values include the following:

- Audit logs
- User ACLs
- Remote access credentials
- Disk unlock token (optional)

As network connectivity options continue to improve, there are scenarios where connectivity cannot be achieved. In these scenarios, platform original equipment manufacturers (OEMs) can trade connectivity for cache size.

Consolidation of DAR services can reduce TCO by eliminating standalone or incompatible vender proprietary services (see Figure 21). Further cost reductions are achieved by eliminating overlapping functionality. For example, multiple user identities for the same person can be replaced by a single integrated identity management system.

How Intel® AT-d Can Improve Storage Device Encryption

For a given generation of Intel AT-d, a limited number of storage interfaces and devices may be supported. Information security objectives seek comprehensive DAR protection regardless of interface or media type. Subsequently, it may be desirable to augment Intel AT-d storage controller protections with storage device encryption. However, to maximize IT investment in DAR infrastructure, encrypting drives could utilize Intel ME service interfaces. Audit, key recovery, user authentication, and remote administration could be extended to include these devices.

Intel AT-d is more than simply an instance of storage controller encryption. It is a point of control for enterprise DAR protection that is flexible enough to accommodate the demands of an increasingly mobile workforce.

Future generations of Intel AT-d may provide some of the building blocks for a Dynamic Virtual Client (DVC) compute model in which data are streamed from a central storage service and cached locally. Intel AT-d could be used to encrypt both local cache and remote network-attached storage.

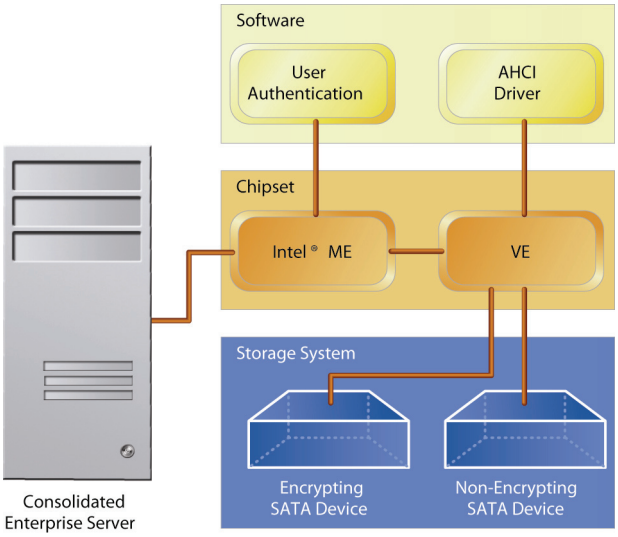

Figure 21: Consolidated Intel® AT-d services

Conclusion

To return to my original fire metaphor, international, national, and local privacy advocacy are the "building codes" that need to be enforced for the safety of information systems in an increasingly interconnected electronic world. Mandates for DAR protection extend to requirements for, not only encryption, but also auditing, authentication, and authorization, and these requirements are fast becoming core capabilities for most computing platforms.

Storage controller encryption, as provided by Intel AT-d, appears to provide the right mix of inexpensive support services for enterprise IT departments to operate. The platform capabilities of Intel ME, Virtualization Engine (VE), and embedded cryptographic engines provide a rich yet protected environment for DAR encryption.

As computing models evolve, platform-based data protection mechanisms can evolve to offer the same low-cost, yet reliable and secure, benefits as those currently provided by Intel AT-d. These benefits will keep private and sensitive data protected and so in turn offer protection to businesses and individuals alike.

Acknowledgements

I acknowledge Vince Von Bokern for providing technical review and feedback. I also thank Victoria Moore, Venkat Gokulrangan, Duncan Glendenning, Steve Deutsch, Dave Hines, Selim Aissi, Alberto Martinez, Srinivas Vuppula, Craig Owen, Jeremy McCormick, Stefan Richards, Dave Singh and many others who have helped in the definition of Intel

AT-d. Lastly, I thank Steve Grobman who has championed Intel AT-d from its inception.

References

[1] "Health Insurance Portability and Accountability Act of 1996." Public Law 104-191. 104th Congress. At http://www.cms.hhs.gov/HIPAAGenInfo/Downloads/HIPAALaw.pdf

[2] "H.R. 3763." 107th Congress of the United States of America at the 2nd Session, January 23rd, 2002. http://fl1.findlaw.com/news.findlaw.com/hdocs/docs/gwbush/sarbanesoxley072302.pdf

[3] "NRS 597.970." At http://www.leg.state.nv.us/Nrs/NRS-597.html#NRS597Sec970

[4] "Directive 95/46/EC of the European Parliament and of the Council." 24 October 1995. At http://ec.europa.eu/justice_home/fsj/privacy/law/index_en.htm

[5] "Act on the Protection of Personal Information." Law No.57, 2003. At http://www5.cao.go.jp/seikatsu/kojin/foreign/act.pdf

[6] Bruce Schneier. Applied Cryptography, 2nd Edition. John Wiley & Sons. New York, 1996.

[7] Vincent Zimmer, Michael Rothman, and Robert Hale. Beyond BIOS. Intel Press, 2006.

[8] http://en.wikipedia.org/wiki/LRW_mode#LRW

[9] http://en.wikipedia.org/wiki/CTR_mode#Counter_.28CTR.29

[10] Jason Garman. Kerberos The Definitive Guide. O'Reilly Media, Inc., Sebastopol, CA, USA. August, 2003.

[11] Pearson et al. Trusted Computing Platforms TCPA Technology in Context. Prentice-Hall PTR. Upper Saddle River, New Jersey, 07458. 2003.

[12] http://en.wikipedia.org/wiki/Cold_boot_attack

Author Biography

Ned Smith is a platform architect for Intel® Anti-Theft Technology focusing on data protection. During his 19-year career he helped define several platform technologies including Intel® Trusted Execution Technology, Endpoint access control in Intel® Active Management Technology, and the Trusted Platform Module (TPM). He chairs the Infrastructure Working Group in the Trusted Computing Group. His security background includes firewalls, PKI, cryptography; TCSEC (orange book) trusted systems, and trusted networking. Ned is the holder of 3 patents with more than 45 pending. He received his M.Sc. degree in computer science from Portland State University and a B.Sc. degree in computer science from Brigham Young University.

Innovating Above and Beyond Standards

Kevin Cline, Business Client Group, Intel Corporation
Lynda Grindstaff, Business Client Group, Intel Corporation
Steve Grobman, Business Client Group, Intel Corporation
Yasser Rasheed, Business Client Group, Intel Corporation

Keywords: Intel® vPro™ technology, Intel® AMT, DASH, WS-MAN

Abstract

By the end of 2008, Intel will have shipped three platform generations of Intel® Core™2 processors enabled with Intel® vPro™ technology, offering unique energy-efficient performance, built-in manageability, and proactive security features targeted at information technology (IT) organizations for large, as well as small- and medium-sized enterprises. Intel vPro technology solutions, offered in strong collaboration with ecosystem independent software vendors (ISVs), have demonstrated significant return on investment through the reduction of total cost of ownership (TCO), attributed primarily to the unique capabilities in platforms enabled with Intel vPro technology.

In this article we describe the open manageability framework offered by platforms enabled with Intel vPro technology. We begin with the history of Intel's client platform innovations above and beyond standards in the context of Intel vPro technology and provide examples of how standards and innovations are integrated to produce some of Intel's best products and technological advances. We also describe Intel's role in co-founding and driving the relevant standards existing in the market today, such as Web Services Management (WS-MAN) and Desktop Management Taskforce/Desktop and Mobile Architecture for System Hardware (DMTF/DASH). We then provide an in-depth description of the manageability architecture, including details of the platform capabilities and technology ingredients, as well as the open interfaces that ISV products use for managing and controlling these platform capabilities. Finally, we discuss future ecosystem development opportunities for ISVs, based on the open interfaces run on Intel vPro technology. The target audience for this article is system integrators, IT professionals, and ISVs.

Introduction

Intel cooperates with other vendors in the industry to accelerate technology adoption by driving industry specifications and by innovating above and beyond these standard specifications. This balanced mix allows for the timely deployment of value-added functionality, followed by the broadest possible industry support for the necessary infrastructure to deliver the next level of innovations. This model for ongoing innovation is based on evolving standards, and as we will show, is an integral part of Intel's manageability philosophy. From this perspective, one can clearly see why innovators are often the primary drivers of standards.

Innovators could avoid standards, but ultimately it would cost them a lot more to provide a steadily increasing variety of end-customer-focused value as quickly and easily as they can by adhering to, and promoting, standards. Similarly, without innovations on top of standards, there would be very little need for standards, since innovations are almost always a reflection of end-user added value. In reality the best products and services require a balance of both.

In this article, we outline the history of Intel's client platform innovations above and beyond standards in the context of Intel vPro technology and provide examples of how standards and innovations are integrated to produce some of Intel's best products and technological advances.

How It All Began

In the late 1980s analysts such as the Gartner Group popularized the term total cost of ownership (TCO). TCO reflected the cost to support a personal computer (PC) over the lifetime of that PC, and it

was found to be a much higher cost consideration than the cost of the system itself (over $10,000 in support costs versus an initial PC cost of roughly $3,000 at that time).

The high cost of supporting a PC was a daunting challenge for IT organizations, and they had to weigh this cost carefully against the obvious benefits in personal productivity that PCs introduced.

Through the 1990s, Intel created the Wired for Management (WfM) initiative, and in the WfM 2.0 baseline, delivered related technologies, such as the Advanced Configuration and Power Interface (ACPI), Wake on LAN (WOL), the Desktop Management Interface (DMI), and the Preboot eXecution Environment (PXE), as a set of tools to address the PC TCO reduction objective. However, IT organizations still had no widespread means of ensuring interoperability amongst multi-vendor solutions to manage systems remotely in their own environment. The Desktop Management Taskforce (DMTF) was formed to fill this need. This taskforce helps drive and establish manageability specifications and related initiatives within the industry. Early on, Intel saw the need for steadily evolving management specifications within the industry and led the formation of the DMTF[1].

Intel has taken on leadership roles throughout the life of the DMTF, acting as the founding chair of the board in the early years of the DMTF, contributing the first conformance tool for DMI, and creating and acting as chair for the Pre-Operating System (Pre-OS) Working Group. At the time of writing, Intel holds both the positions of Vice President of Interoperability and Co-lead for the Systems Management Forum, where compliance is being defined today. Intel has also recently made key contributions to the DMTF infrastructure by submitting Intelligent Platform Management Interface (IPMI) and Web Services Management (WS-MAN) technologies for use in DMTF specifications and technologies [2].

While the DMTF was putting the Common Information Model (CIM) and the DMI in place, Intel and IBM Corporation collaborated on the next level of client innovations in the development of Alert-on-LAN technologies that were integrated into IBM client platforms and into Intel chipsets. Alert-on-LAN delivered early innovation in the Out-of-Band (OOB) management space, and it was a subset of what would later become the alerting aspects of Intel® Active Management Technology (Intel® AMT).

As IT professionals, end users, and other industry players showed interest in Alert-on-LAN-based capabilities, Intel hosted a meeting with a number of independent hardware vendors (IHVs) to discuss and investigate the right level of specification around Alert-on-LAN. The outcome of that meeting was a draft charter for the Pre-OS Working Group in DMTF, and it formed the core of what would later be ratified, published, and implemented as the Alert Specification Format (ASF).

Intel® vPro™ Technology is Born

While Intel played a significant role in driving ASF through DMTF to its final 2.13 revision, with its editing and authoring contributions, Intel's innovative efforts mainly focused on the ecosystem extensions to the Alert on LAN hardware support. Intel added more features and a broad ISV ecosystem that was based on the Simple Object Access Protocol (SOAP). These provided IT organizations with timely solution-level options across a variety of their existing console vendors. The result was that IT organizations could realize interoperability value immediately, without waiting for future standards-based compliance tools and programs.

As part of Intel's vision for helping to reduce IT organizations' TCO, Intel vPro technology was born. This new technology included Intel AMT, Intel® Trusted Execution Technology (Intel® TXT), and Intel® Virtualization Technology (Intel® VT). Intel vPro technology is designed to support a seamless transition to these new specifications as they are finalized as well as provide additional benefits in the meantime. In 2006, Intel released its first platform, enabled with Intel vPro technology, that supported SOAP-based capabilities and the ASF specification.

As the industry accepted and embraced these new specifications, it was clear that additional work was needed in the area of system hardware. In 2007, the DMTF published the Desktop and Mobile Architecture for System Hardware (DASH) specification which, according to the DMTF, is "a suite of specifications which standardize the manageability interfaces for mobile and desktop hardware. The DASH suite of specifications defines the external interfaces for management in the form of protocols and profiles for representing mobile and desktop hardware[2] [1]."

[1] The DMTF was later renamed the Distributed Management Task Force as it broadened its focus to include all enterprises.

[2] DMTF website: www.dmtf.org

Intel® vPro™ Technology Goes Above and Beyond Standards

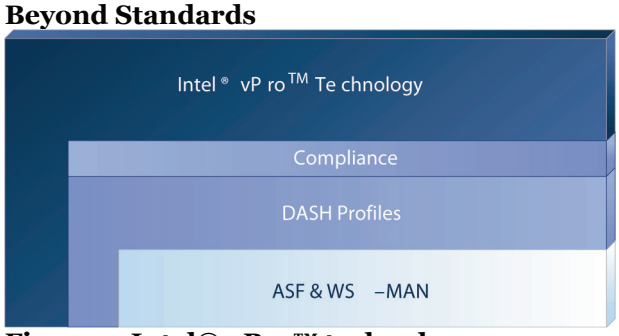

Figure 1: Intel® vPro™ technology innovation on top of standard specifications
Source: Intel Corporation, 2008

In 2007, Intel released its first DASH-capable platform enabled with Intel vPro technology. This platform integrated the pre-standard DASH 1.0 specifications available at the time, in addition to other innovative manageability and security capabilities. Examples of these innovative features include remote diagnostics/repair/configuration, enhanced system defense filters, 802.1x and Cisco SDN pre-OS support. More information on features and use cases in Intel vPro technology can be found at http://www.intel.com/technology/vpro/.

Having identified the relevance of Web services to remote management, Intel and Microsoft Corporation collaborated on what would later be called WS-MAN, which eventually became the de facto Web services transport for not just DASH, but for all DMTF initiatives and technologies. WS-MAN is essentially replacing ASF as the standards-based remote communications mechanism for remote client platform management [2].

As illustrated in Figure 1, Intel vPro technology goes above and beyond the industry specifications to provide added innovative manageability capabilities in addition to supporting the required standards. One misunderstanding in the industry is that when a standard is in place, a solution provider can only deliver a solution that is standards-based and nothing more. That is not the case, as specifications do not dictate the complete list of capabilities available to a customer; rather, they enforce the minimum infrastructure elements and basic features necessary to facilitate interoperability between vendor implementations. DASH, for example, provides a standards-based protocol that forms the basis for external interfaces to discover capabilities and to interact with a platform, but it does not provide a description of how features are implemented. Platforms enabled with Intel vPro technology not only meet the needs of the IT professional by integrating a common set of specifications such as DASH, but they go above and beyond these by allowing IT organizations to achieve a lower TCO with additional manageability features that work across multi-vendor solutions and are enabled for the most relevant enterprise security and management consoles in the market.

The Architecture of Platforms Enabled with Intel® vPro™ Technology

We just gave you a historical perspective on innovation and standards as they relate to Intel vPro technology. We now move on in this section to describe the fundamental building blocks, hardware hooks, and software interfaces that enable platforms running Intel vPro technology to deliver this unique combination of energy-efficient performance, proactive security, and built-in manageability features. As we will see in the next sections, ISVs can innovate using these building blocks and deliver IT solutions that can achieve the desired TCO reduction goals for IT organizations in large and small enterprises.

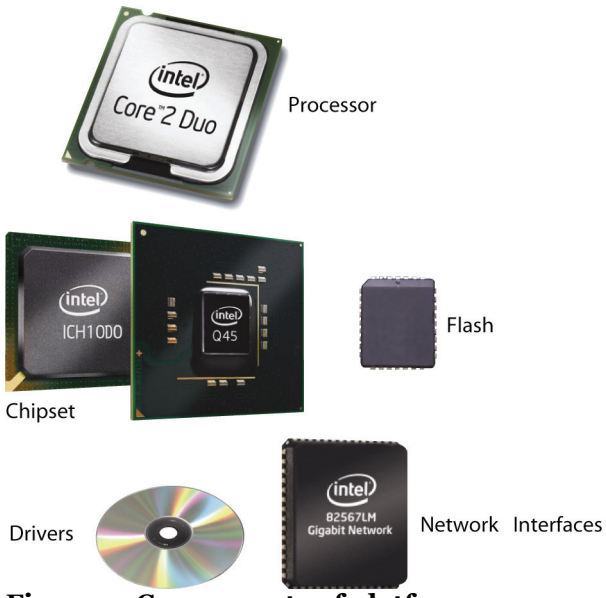

Figure 2: Components of platform architecture, enabled with Intel® vPro™ technology (circa 2008)
Source: Intel Corporation, 2008

Figure 2 shows the various architecture components of a platform, enabled with Intel vPro technology, circa 2008. For the purposes of this article, we focus on the components that play an active role in providing built-in manageability features, namely the Intel Management Engine (Intel ME), the

nonvolatile flash memory, BIOS extensions, and the network controller extensions.

Intel® Management Engine
The Intel ME is an embedded microcontroller (integrated in Intel chipsets) running a lightweight microkernel OS that provides a low-power, OOB execution engine for management services. At system initialization, the Intel ME loads its code from system flash memory. This allows it to be up and running before the main OS is started. For runtime data storage, the Intel ME has access to a protected area of system memory at runtime (in addition to a small amount of on-chip cache memory for faster and more efficient processing). A fundamental feature of the Intel ME is that its power states are independent of the host OS power states. This allows it to be up and running when the CPU and many other components of the system are in deeper sleep states.

As a result, the Intel ME can be a fully-functioning component as soon as power is applied to the system. This allows it to respond to OOB commands from the IT management console without having to wake up the rest of the system, thereby reducing power consumption significantly. This opens the door for a large number of innovative, low-power, secure OOB usages that result in a significant reduction in TCO.

Network Controller Manageability Extensions
In order for the Intel ME to access the network while the host OS is absent, the Intel ME needs direct access to the network interfaces. The current network architecture allows for the Intel ME manageability services to share the IP address of the host OS, by using specific and dedicated transport level (Level 4) ports to distinguish manageability traffic from regular host traffic.

In addition, network controllers on platforms running Intel vPro technology are equipped with quintuple network filters to facilitate network management functions by redirecting traffic to either the host OS or the Intel ME, based on port numbers. The quintuple filters also allow for innovative features, such as programmable circuit breakers, which allow an IT administrator to disconnect a PC from the network yet still have secure remote access to it to diagnose it, patch it, and reestablish connectivity, once the PC is ready for general use.

Nonvolatile Flash Memory
A partition of the system flash memory is typically carved out for use by the Intel ME. This partition is protected and hidden from the host OS to ensure integrity and confidentiality of the information stored in it. At boot time, the Intel ME loads its firmware image from flash into system memory and starts running, independent of the host OS. The Intel ME flash partition also provides nonvolatile storage for ISV applications. This allows ISV applications to store critical information (that is, license information, asset inventory, and so on) that can be accessible out of band by the IT console, even if the host OS is in a sleep state.

Intel® Management Engine BIOS Extensions (Intel® MEBX)
The Intel® Management Engine BIOS Extensions (Intel® MEBX) are used for a variety of purposes. For example, Intel MEBX initialize Intel AMT functions, and they can also be used to reset Intel AMT to its initial factory default state. Intel MEBX also capture platform hardware configuration information and store it in nonvolatile memory so that Intel AMT can make the information available out of band.

With this brief description of the various hardware architecture building blocks for platforms with Intel vPro technology, we describe below the various software interfaces that allow ISV agents and IT consoles to interact with these platforms in various power states.

Open Manageability Architecture on Platforms Running Intel® vPro™ Technology

This section provides a detailed description of the external interfaces for the manageability services that use Intel vPro technology. This includes the network interfaces for a remote console to interact with a client running Intel vPro technology, as well as the host-based interface for the host OS (and applications) to interact with the underlying embedded manageability services running on the Intel ME.

Host-Based Interfaces
Platforms enabled with Intel vPro technology offer two host-level interfaces for the underlying embedded manageability services: a driver-level interface and the Host Embedded Controller Interface (HECI). They also offer a user-level local management service (LMS), as depicted in Figure 3.

Host Embedded Controller Interface
The HECI is a bidirectional bus that allows the host OS to communicate directly with the Intel ME,

exchanging system management information and events. HECI enables the host OS (by loading the HECI device driver) to control other devices, such as on-board fan controllers, Wake-on-LAN, power supply devices, and so on. HECI is the primary software interface between the host OS and the Intel ME. Other interfaces, such as the LMS, build on the HECI infrastructure, to provide a more programmer-friendly interface that applications can use.

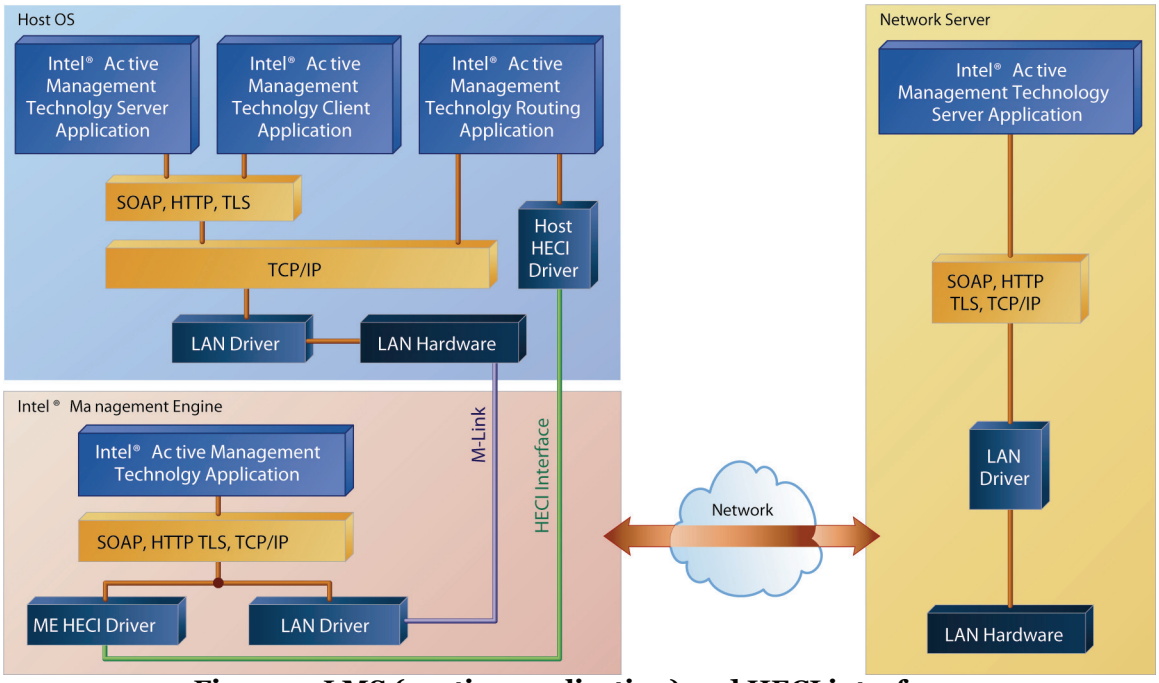

Figure 3: LMS (routing application) and HECI interfaces
Source: Intel Corporation, 2008

Local Manageability Service

The LMS is a user mode service that runs as part of the host OS. LMS exposes Intel AMT functionality through standard interfaces, such as general-info interface, firmware update interface, and local agent-presence interface. LMS listens for the request directed to the Intel AMT local host, and when an application sends a SOAP/HTTP message addressed to the local host, the LMS intercepts the request and routes the request to the Intel ME interface via the HECI driver.

External Interfaces

Starting in 2007, with the second generation of platforms enabled with Intel vPro technology, external interfaces for new management services running Intel vPro technology were all based on the WS-MAN industry standard. In the past, a non-WS-MAN interface called Intel® Active Management Technology network interface was the only external interface. Platforms running Intel vPro technology also offered DASH-compliant interfaces that are defined and available for specific Intel vPro technology features. This is not the case for all features, since DASH supports a subset of Intel vPro technology features only.

The Intel® Active Management Technology Network Interface

Intel AMT network interface is a SOAP-based interface between a remote management server and Intel AMT. The interface allows an external network host, such as an enterprise management console, to access Intel AMT features. Intel AMT network interface describes the network programmatic interface that allows an external network host to do the following:

Perform administrative operations.
Access the event log and manipulate the event filters.
Power off, power on, reboot, or wake up the PC.
Read hardware asset management information.
Read/write unformatted data to the public area of the nonvolatile store.

In the first generation of platforms enabled with Intel vPro technology, Intel AMT network interface was the only interface available. The WS-MAN/CIM-based interface (see next section) was introduced in the second generation of platforms in 2007. Intel AMT network interface is still supported on newer platforms running Intel vPro technology for backward compatibility. Intel will support multiple generations of the interface with reasonable overlap to ensure backward compatibility. Developers are highly encouraged to switch over to WS-MAN and CIM-based interfaces going forward.

External Interfaces Based on WS-MAN/Common Information Model

WS-MAN is a SOAP-based industry standard protocol designed for Web service-based remote management of systems, such as desktops, notebooks, servers, and other IT-related infrastructure. WS-MAN defines basic operations that can be performed on resources; it does not define what an actual resource looks like [2]. This is the role of the CIM, an object-oriented modeling language that can be used to define the actual structure of resources, including their properties and methods. The model or definition of a specific resource is known as a CIM class, where CIM profiles are collections of classes needed to implement a specific management feature. The description of the CIM schema that Intel AMT is based on can be found at the following DTMF website:
http://www.dmtf.org/standards/cim/cim_schema_v211/

DASH-Compliant External Interfaces

DASH interfaces also build on WS-MAN and CIM. They assume and depend on the existence of profiles, which roughly correspond to programming interfaces found in software development kits, in cases where standards do not currently exist. These profiles describe base functions, usages, and the relevant portions of the CIM model to focus on for a base set of capabilities.

DASH profiles are organized in functional areas (such as hardware inventory), but describe only interfaces. They do not define or prescribe how to develop a feature and are not the same thing as features.

ISV Ecosystem Value, Innovation Opportunities, and the Promise of Interoperability

In this section we briefly look at how the Intel vPro technology ecosystem extends the platform value proposition described in previous sections. We then investigate the differences between ecosystem interoperability and specification compliance from the perspective of IT organizations.

Ecosystem Value: Transforming Features to Solutions

Intel vPro technology ecosystem partners provide a critical ingredient to the overall Intel vPro technology value proposition to IT organizations. The WS-MAN standard is used to deliver end-to-end solutions for IT organizations in areas, such as manageability and security, in the following ways.

An Intel vPro technology hardware inventory OOB capability can be remotely accessed by using WS-MAN, and it can be exposed on an inventory or support screen in the current version of the most common enterprise management consoles such as Altiris* or LANDesk*. In this way, IT organizations can realize how a platform feature of Intel vPro technology, once integrated into their systems, can offer them more value than that offered by a variety of console and software offerings they are already using.

Interoperability and Compliance

Once industry standard specifications are completed, the value for IT organizations is only partially realized. For true interoperability to be achieved, which is the primary benefit of standards for IT organizations, compliance programs and tools need to be developed and vetted in the field. While Intel is a primary driver of the DASH Compliance Program, we realize that it will take broader participation from hardware and software vendors and a vetting of the tools themselves before this interoperability goal can be realized by end-user IT organizations.

Intel's experience with developing and validating an ISV ecosystem is not only a proxy for the inevitable goals of specification compliance, it allows IT organizations to take advantage literally of the exact level of vendor selection and interoperable goals that vendors hope to realize from DASH and related industry specifications. Solution-level interoperability is possible today, due to the broad adoption of Intel vPro technology across all the original equipment manufacturers (OEMs) and in conjunction with all the primary enterprise management and security ISVs.

Innovating Beyond and on Top of Standards

As mentioned in the introduction to this article, there is a common misconception that standards

dictate commoditization of capabilities and prevent rapid innovation in the products that implement the industry specifications. As long as these specifications are implemented at the appropriate level, this is not the case. Innovation can happen on top of industry specifications. For example, Web-based capabilities on the Internet use well-defined standards for Hypertext Transfer Protocol (HTTP) and Transmission Control Protocol/Internet Protocol (TCP/IP) to ensure that communication can seamlessly occur between a wide range of browsers and servers; still, innovative applications and capabilities are continually being delivered to the market. On the Web, industry specifications are not only used for basic communications, but also for authentication and encryption, as well as being used in other functional areas. Additionally, there are sets of optional specifications (either formal or informal) that developers can choose to use. For example, Asynchronous JavaScript and XML (AJAX) can be used for advanced rendering and "smash-ups" if a Web application developer chooses to take advantage of its features. However, if an application developer determines that these capabilities do not meet its needs, the developer is free to build an independent capability. The application is therefore not limited by AJAX and the developer can use any method to develop the content.

The evolution and philosophy of manageability specifications are very similar to what we see with Internet and Web-based capabilities. There must be a balanced approach such that the foundational capabilities are implemented in a uniform manner, without constraining the pace or availability of innovative capabilities developed by independent hardware and software vendors. Additionally, it is reasonable for parts of the industry specifications to be optional, so that these parts can be used at the discretion of the solution innovator.

DASH and WS-MAN are the industry specifications that deliver this balanced approach to manageability. Intel AMT was built using these specifications, ones that deliver baseline core capabilities. These core capabilities would be common in most implementations of a manageability system. Intel AMT also solves problems facing the market by innovating on top of the standards and delivering timely value and variety in an interoperable fashion. For example, the implementation of the hardware inventory feature uses both WS-MAN for definition of the communication protocol, and a DASH profile to define the behavior of the communication exchange between clients and servers (management consoles). This very straightforward feature allows for a well-defined mechanism to determine what a hardware platform is, and what its basic hardware inventory is, regardless of the platform type (it even works for non-computer devices, such as networking or embedded equipment). Conversely, on a feature that is highly specific to the Intel PC platform, such as Intel® Anti-Theft Technology (Intel® AT), Intel is able to implement highly innovative manageability features on top of the WS-MAN standard by defining a custom interface. For example, a custom interface in Intel AT is an "out-of-band unlock" interface. This capability allows a management console to provide appropriate credentials to an unattended encrypted machine, enabling it to self-boot in order for a security patch or other management function to be executed.

A Developer's Point of View

Given that the underlying data model for manageability with Intel AMT is based on an object-oriented schema; it maps well to object-oriented languages such as C#*, Java*, and even object-oriented scripting languages, such as VBScript*. However, there is nothing preventing low-level programming languages from participating in Intel AMT, WS-MAN operations, by way of low-level TCP/IP or HTTP requests. There is no pragmatic reason to do this: it adds significant complexity to development; however, for illustrative purposes, we show an example of how a raw HTTP request would be structured to issue a WS-MAN command [3].

For example, building the following request and submitting it with HTTP (Figure 4) would generate a "pull" action to gather data about "CIM_AssociatedPowerManagementService." This would correlate to both the current and most recently requested power state for the machine that is being queried. In this example, the current power state (Figure 5) is 8, which implies that the machine is currently powered off.

```
POST /wsman HT    TP/1.1
Content-Type: application/soap+xml;charset=UT     F-8
User-Agent: Direct Generated
Host: 192.168.0.157:16992
Content-Length: 11  39
Connection: Keep-Alive
Authorization: Digest
username="admin",realm="Digest:0FD1B83E2C63BE137C4DCDDDBD232F1EE35",nonce="0900d55b4d5a6760cd5c35e4401960
4b",uri="/wsman",cnonce="0416214bba514074a4b7f152f29bb0fd",nc=00000001,response="51718af          f9a6d3ba3f91019196102bbe0"
,qop="auth"



<s:Envelope xmlns:s="http://www  .w3.org/2003/05/soap-envelope" xmlns:a="http://schemas.x      mlsoap.org/ws/2004/08/addressing"
xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/enumeration" xmlns:w="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
 <s:Header>
  <a:To>http://192.168.0.157:16992/wsman</a:To>
  <w:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_AssociatedPowerManagementService</w:ResourceURI>
  <a:ReplyTo>
   <a:Address s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</a:Address>
  </a:ReplyTo>
  <a:Action s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull</a:Action>
  <w:MaxEnvelopeSize s:mustUnderstand="true">153600</w:MaxEnvelopeSize>
  <a:MessageID>uuid:8711     95C4-6BAF-4327-8353-F1BB1209B5F6</a:MessageID>
  <w:Locale xml:lang="en-US" s:mustUnderstand="false" />
  <w:OperationTimeout>PT   60.000S</w:OperationTimeout>
 </s:Header>
 <s:Body>
  <n:Pull>
   <f:EnumerationContext xmlns:f="http://schemas.xmlsoap.org/ws/2004/09/enumeration">03000000-0000-0000-000          0-
000000000000</f:EnumerationContext>
    <n:MaxElements>20</n:MaxElements>
  </n:Pull>
 </s:Body>
</s:Envelope>
```

Figure 4: Raw HTTP request. Source: Intel Corporation, 2008

Note: This is a display of HTML code and not HTML code to be executed. It's also an image of the code, which makes it unexecutable.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=UT     F-8
Date: Sun, 12 Oct 2008 15:47:24 GMT
Cache-Control: no-cache
Expires: Thu,    26 Oct 1995 00:00:00 GMT
Tra nsfer-Encoding: chunked
Server: Intel(R)    Active Management   Tec hnology 5.0.2


59e

<?xml version="1.0" encoding="UTF-8"?><a:Envelope xmlns:a="http://www       .w3.org/2003/05/soap-envelope"
xmlns:b="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:c="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:d="http://schemas.xmlsoap.org/ws/2005/    02/trust" xmlns:e="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:f="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
xmlns:g="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AssociatedPowerManagementService">
 <a:Header>
   <b:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</b:To>
   <b:RelatesTo>uuid:871   195C4-6BAF-4327-8353-F1BB1209B5F6</b:RelatesTo>
   <b:Action a:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse</b:Action>
   <b:MessageID>uuid:00000000-8086-8086-8086-000000000008</b:MessageID>
   <c:ResourceURI>http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_AssociatedPowerManagementService</c:ResourceURI>
 </a:Header>
 <a:Body>
   <f:PullResponse>
     <f:Items>
       <g:CIM_AssociatedPowerManagementService>
         <g:PowerState>8</g:PowerState>
         <g:RequestedPowerState>1</g:RequestedPowerState>
         <g:ServiceProvided>
           <b:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</b:Address>
           <b:ReferenceParameters>
             <c:ResourceURI>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_PowerManagementService</c:ResourceURI>
             <c:SelectorSet>
               <c:Selector Name="Name">
                 Intel(r)  AMT Power Management Service

                 25d

               </c:Selector>
               <c:Selector Name="CreationClassName">CIM_PowerManagementService</c:Selector>
               <c:Selector Name="SystemName">Intel(r)      AMT</c:Selector>
               <c:Selector Name="SystemCreationClassName">CIM_ComputerSystem</c:Selector>
             </c:SelectorSet>
           </b:ReferenceParameters>
         </g:ServiceProvided>
         <g:UserOfService>
           <b:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</b:Address>
           <b:ReferenceParameters>
             <c:ResourceURI>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem</c:ResourceURI>
             <c:SelectorSet>
               <c:Selector Name="Name">ManagedSystem</c:Selector>
               <c:Selector Name="CreationClassName">

                 0d6

                 CIM_ComputerSystem
               </c:Selector>
             </c:SelectorSet>
           </b:ReferenceParameters>
         </g:UserOfService>
       </g:CIM_AssociatedPowerManagementService>
     </f:Items>
     <f:EndOfSequence></f:EndOfSequence>
   </f:PullResponse>
 </a:Body>
</a:Envelope>
```

Figure 5: Raw HTTP response. Source: Intel Corporation, 2008

Note: This is a display of HTML code and not HTML code to be executed. It's also an image of the code, which makes it unexecutable.

If operating at an HTTP level, one big challenge, from a developer's perspective, is that a large amount of structure needs to be defined accurately in a WS-MAN request to comprehend and support the current CIM schema referred to by DASH standards. Similarly, parsing a WS-MAN response for the data that are needed can be quite complex as well, especially for those developers who are less familiar or experienced with CIM. Fortunately, a wide range of classes are available that can help a developer interact with WS-MAN and Intel AMT without subjecting the developer to the complexity that exists at the CIM/XML content level.

One implementation of WS-MAN is Microsoft's WinRM* [4] implementation. WinRM encapsulates most of the complexity of WS-MAN interactions into a COM implementation that can be used for a wide variety of languages. For example, the following piece of code shows how to read the power state of the machine from C# language by using WinRM.

```
static public WSManClient _WinRM = new
WSManClient("192.168.0.157", "admin",
"passwordgoeshere", false);
static void Main(string[] args)
{
ArrayList enumSystemPowerstate =
_WinRM.Enumerate(typeof(CIM_AssociatedP
owerManagementServiceType));
    for (IEnumerator enumObj =
enumSystemPowerstate.GetEnumerator();
enumObj.MoveNext(); )
    {

CIM_AssociatedPowerManagementServiceTyp
e cur =
(CIM_AssociatedPowerManagementServiceTy
pe)enumObj.Current;

Console.WriteLine("SystemPowerstate is:
{0}\n", cur.PowerState);
    }
}
```

Since WinRM is COM-based, it is relatively easier to use than a scripting language such as VBScript. To get information about the power state of the machine (the raw XML result), the following simple script can be used.

```
Dim WSMan
Dim Session, Options

Set WSMan = CreateObject(
"WSMAN.Automation" )
iFlags = WSMan.SessionFlagUseDigest Or
_

WSMan.SessionFlagCredUsernamePassword
Or _
        WSMan.SessionFlagUTF8

Set Options =
Wsman.CreateConnectionOptions
Options.Username = "admin"
Options.Password = "P@ssw0rd"

Set Session =
WSMan.CreateSession("http://10.19.68.21
8:16992/wsman", _
                iFlags, Options)

strResource =
"http://schemas.dmtf.org/wbem/wscim/1/c
im-
schema/2/CIM_AssociatedPowerManagementS
ervice"

Set objResultSet = Session.Enumerate(
strResource)
Wscript.Echo objResultSet.ReadItem
```

Looking Ahead

As we look to the future, platforms running Intel vPro technology will continue to deliver innovative solutions above and beyond standards, while at the same time evolving the standards themselves for faster adoption and future innovation. At the time of writing, Intel is actively working with the DMTF to help define future DASH 1.2 specifications. During this time, Intel vPro technology innovations will continue to deliver business optimizations for IT organizations, in the areas of services management, power management, and security and virtualization management. Platforms enabled with Intel vPro technology will provide additional management capabilities, thus further reducing IT organization's TCO, as well as addressing new business process needs and requirements.

References

[1] "DASH." At www.dmtf.org

[2] "WS-MAN." At www.dmtf.org

[3] "Intel AMT Software Developers Kit." At http://softwarecommunity.intel.com/communities/manageability

[4] "WinRM." At http://msdn.microsoft.com/en-us/library/aa384426(VS.85).aspx

[5] For additional information on Intel vPro technology, please visit the Intel vPro technology Expert Center: http://communities.intel.com/community/vproexpert and http://www.intel.com/technology/vpro/index.htm.

Acknowledgements

We thank our reviewers and contributors: Stephanie St. Denis, Dori Eldar, Sarah Flanagan, Josh Hilliker, Craig Owen, Tom Quillin, Christie Rice, and Jeff Tripp for their invaluable comments and insights.

Authors' Biographies

Kevin Cline is a 14-year Intel veteran with experience in specification design, applications design and support, ecosystem development, as well as planning and strategic marketing across all enterprise platform segments. Specifically, Kevin has led the company in defining and integrating management and security technologies into Intel communications and client and server product lines—beginning with the solutions based on the Wired for Management hardware baseline and continuing with ecosystem and ISV solutions today. He is now focused on driving initiatives, standards, and ecosystem development for emerging technologies. He is Intel's board representative in the DMTF. He has an undergraduate degree in computer engineering from California State University, Sacramento and an M.B.A. from University of California, Davis.

Lynda Grindstaff leads the business client strategy for platforms enabled with Intel® vPro™ technology, a strategy that improves IT's ability to manage and secure PCs. An expert in her field, she has won several awards, including the Intel Achievement Award, the Intel Software Quality Award, and the Society of Women Engineers Emerging Leader Award. Her tenure at Intel spans more than a decade and includes a patent, system software development, chipset validation, and management of a global technical marketing team based in the United States and India. A valued industry conference speaker, Lynda holds a Bachelor of Science degree in computer science from the California State University, Sacramento. She remains active in community outreach programs.

Steve Grobman leads the team that defines Intel's future business client strategy, architecture, and roadmap. Steve was instrumental in developing the architecture for Intel® vPro™ technology, Intel's first digital office professional platform. Previous positions in his 14-year Intel career include architecture, engineering, and management positions in Intel's Desktop Platform and Information Technology divisions. Steve is a principal engineer, has two patents and 22 pending, and has written two programming books. He received his bachelor of science degree in computer science from North Carolina State University.

Yasser Rasheed is the lead Digital Office Solutions Architect in the Digital Enterprise Group at Intel Corporation, driving architecture definition for next-generation business-client platforms. Yasser has been with Intel since 2000, and he has led various R&D projects on platform partitioning, advanced firmware technology, as well as wired and wireless multimedia networking in the home. Yasser played an instrumental role in establishing Intel's early vision for Digital Home technologies. He was a co-chair of the UPnP Forum's Audio Video working group, established the UPnP Forum's QoS working group, and co-authored a number of specifications and publications in the area of wired and wireless multimedia networking in the home. Yasser holds a B.Sc. degree in electrical engineering from Cairo University, Egypt, M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Toronto, Canada, and an M.B.A. degree from the University of Oregon.

Table of Acronyms

Acronym	Meaning	Description
ACPI	Advanced Configuration and Power Interface	Open standard that defines common interfaces for hardware recognition, motherboard, device configuration, and power management.
AJAX	Asynchronous JavaScript and XML	Group of interrelated Web development techniques used for creating interactive Web applications or rich-Internet applications.
Intel® AMT	Intel® Active Management Technology	Technology enabling remote access and management of networked computer systems even when they lack a functioning OS, hard drive, or are turned off.
ASF	Alert Specification Format	Specification defining standards-based interfaces with which vendors selling alerting or corrective-action offerings can implement products to ensure interoperability.
Intel® AT	Intel® Anti-Theft Technology	Security technology that enables encryption and theft deterrence capabilities. Also integrated with Intel® AMT.
CIM	Common Information Model	Conceptual schema that defines how the managed elements of an IT organization's environment are represented as a common set of objects, and the relationships between those objects.
DASH	Desktop and Mobile Architecture for System Hardware	A suite of specifications using WS-MAN that delivers standards-based Web services management for desktop and mobile client systems.
DMI	Desktop Management Interface	Standard framework for managing and tracking components in a desktop, notebook, or server computer.
DMTF	Distributed Management Taskforce	Standards organization that develops and maintains standards for systems management of IT environments in enterprises and on the Internet.
HECI	Host Embedded Controller Interface	Software interface that establishes communication between the host OS and the Intel® ME of the Intel® AMT subsystem.
HTTP	Hypertext Transfer Protocol	Communications protocol for transfer of information over the Internet used for retrieving inter-linked text documents.
HW	Hardware	A general term that refers to the physical components of a computer system, as in computer hardware.
IHV	Independent Hardware Vendor	Companies specializing in making or selling hardware.
IP	Internet Protocol	Protocol used for communicating data across a packet-switched internetwork.
ISV	Independent Software Vendor	Companies specializing in making or selling software.
IT	Information Technology	The study, design, development, implementation, support, or management of computer-based information systems.

LAN	Local Area Network	Computer network with small geographic range typically having higher data-transfer rates than wide-area networks.
LMS	Local Manageability Service	A user-mode service running locally in the host OS that redirects requests to the Intel® AMT local host, to the Intel® ME interface, via the HECI driver.
Intel® ME	Intel® Management Engine	Embedded microcontroller running a microkernel OS that provides low-power, out-of-band execution for manageability services.
OEM	Original Equipment Manufacturer	A company that uses a component made by a second company in its own product or sells the product of a second company under its own brand name.
OOB	Out of Band	Communications that occur outside of a previously established communications method or channel.
OS	Operating System	A master program that controls a computer's basic functions and allows other programs to access the computer's resources such as disk drive, printer, keyboard, and screen.
PXE	Pre-boot Execution Environment	An environment to boot computers by using network interfaces independently of available data storage devices or installed operating systems.
SOAP	Simple Object Access Protocol	A protocol for exchanging XML-based messages over computer networks and for providing the foundation layer for the Web services protocol stack.
SW	Software	The programs that tell the computer what to do.
TCO	Total Cost of Ownership	Financial estimate of direct and indirect costs associated with owning and operating a PC.
TCP	Transmission Control Protocol	Internet protocol providing ordered delivery of a stream of bytes from one program on one computer to another program on another computer.
WfM	Wired for Management	HW-based system allowing a new computer without software to be manipulated by a master computer with access to its hard disk on which it can paste and install programs.
WOL	Wake-on-LAN	Networking standard that allows a PC to be turned on or woken out of sleep state remotely by a network message.
WSMAN	Web Services Management	SOAP-based protocol for management of servers, devices, applications, and more.
XML	Extensible Markup Language	General-purpose specification used to encode documents and serialize data for sharing by information systems over the Internet.

Configuring Intel® Active Management Technology

Dori Eldar, Digital Enterprise Group, Intel Corporation
Arvind Kumar, Digital Enterprise Group, Intel Corporation
Purushottam Goel, Digital Enterprise Group, Intel Corporation

Keywords: Intel® AMT, computer system deployment, down-the-wire configuration, TLS, x509 certificates, bare-metal configuration, management console suite, hardware-based manageability, DMTF's DASH initiative

Abstract

Security and manageability are two primary pillars of Intel® Active Management Technology (Intel® AMT). The hardware-based manageability aspects of Intel AMT provide unprecedented capabilities to information technology (IT) administrators to effectively manage the thousands of platforms in their enterprises. A huge ecosystem of independent software vendors (ISVs) and system vendors has built up over the last few years to enable these use models. Each customer has different needs. Intel AMT therefore provides building blocks for a wide range of configuration settings with regard to security levels and integration, and ISVs have built their mechanisms on top of these building blocks.

In this article, we explain the configuration process for Intel AMT. We start with an overview of the general configuration needs and the challenges involved, such as security of the configuration process and ease of configuration. We go on to explain the various configuration scenarios and use models that are supported, such as where the system will be configured (system manufacturer, IT, end-user), time of configuration (before installing the operating system (OS), or post-OS configuration), and automated versus manual configuration.

We then move on to elaborate on the three primary configuration mechanisms for Intel AMT: Web-based quick configuration; pre-shared, key-based configuration; and remote configuration. Web-based configuration is primarily for small businesses and requires no additional infrastructure. Pre-shared key and remote configuration are options for large and medium-sized businesses, and we explain the protocols for these configurations in detail.

Introduction

Most hardware or software products need some kind of initialization or setup before becoming operational. Personal computers (PCs) based on Intel® vPro™ technology are no different in this regard. The IT administrator of these PCs needs to properly configure them for the IT environment in order to derive maximum benefit from deploying this technology. Intel would like computers enabled with Intel vPro technology to be the systems of choice for most businesses worldwide; to this end, a wide choice of options and tools have been made available to set up and configure Intel vPro technology, catering to a wide range of businesses. Our goal in this article is to explore the most dominant IT deployment scenarios, describe how configuration procedures for computers enabled with Intel vPro technology are integrated into the existing IT procedures in an enterprise1, and delve into protocol details in those areas that are of interest to our readers. The actual entity in the computer's hardware that is configured within the computer is the Intel Active Management Technology subsystem (Intel AMT).

We formally define the Intel AMT configuration process as the initial procedure of setting up an out-of-the-box Intel AMT client system with all the necessary configuration options required for it to be operable and managed remotely by the governing IT organization.

Background and Problem Description

Client System Deployment Strategies for IT

The Intel AMT configuration procedures should be seamlessly integrated into the existing client system deployment model of the IT organization. There are

[1] For the purposes of this article, we use *enterprise* to mean both a large business and a medium-sized business.

probably as many different deployment procedures as there are businesses. Nonetheless, in the case of Intel AMT configuration, we have identified two major deployment strategies that call for distinct Intel AMT configuration methods:

1. *Direct shipment*. Client systems are shipped directly from the system manufacturer to the enterprise end-user.

2. *IT staging-area setup*. Systems undergo an initial setup by an IT technician prior to shipment to the end-user.

As well as the aforementioned methods, some system manufacturers offer their customers the option of purchasing customized PC client systems. Typically, this is done in order to simplify the deployment process: for example, by pre-loading the enterprise's operating system (OS) image.

Intel AMT configuration can be incorporated into the two types of deployment models we mention. Furthermore, Intel is providing the system manufacturer with the capability to pre-configure various Intel AMT settings, which we explore in subsequent sections of this article.

Security
Before the advent of Intel AMT, manageability technologies such as the Alert Specification Format (ASF) [1] and Wired for Management (WfM) [2] were configured through the host OS. For example, configuration of Intel's ASF management controller was performed through a Windows Management Instrumentation* (WMI) provider [3], on the local Microsoft Windows* OS, which could be used by any software application to configure ASF, specify ASF policies, and designate remote management servers. From a security perspective, malware applications, such as computer viruses and Trojan horses operating on a PC client OS, could exploit the capabilities provided by ASF. However, since these capabilities are limited to alerts and remote power-up/power-down operations, the consequences of such misuse were typically low, as was the overall vulnerability of the system. Intel AMT offers much stronger protections to the enterprise IT group, including the capability to boot systems from a remotely-situated media and to share data between local and remote software agents. With this in mind and with the ongoing rise in PC client malware vulnerability incidents [4], Intel AMT requires a more resilient configuration method that can withstand such malware. The Intel configuration methodology aims to establish a trusted and secure channel between the device-given Intel AMT instance and the authoritative enterprise's management server, such as the Microsoft System Center Configuration Manager* (SCCM), thereby reducing the likelihood of a malware attack and lowering the overall exposure of Intel AMT to such attacks.

Scalability
Intel AMT can be used by businesses of any size from Fortune 500 enterprises or other large businesses, to small- and medium-sized businesses. We consider a business to be small if it has less than 100 employees; a medium business would have 100 to 1000 employees; and any business with over 1000 employees is a large business. Each of these types of businesses may have a different set of requirements for how they would like to configure Intel AMT. A cost-effective deployment solution might not be the same for all: it will differ depending on the size of the business. A simple, manual option would best meet the needs of a small business, while a fully-automated, yet infrastructure-dependent, solution would best meet the needs of an enterprise. Intel AMT offers both configuration options.

Overview of Intel® AMT Configuration Process

Configuration Process for Small Businesses
Intel AMT is designed to allow small businesses to configure and utilize it without depending on third-party software. However, since the configuration process requires manual operation on each system with Intel AMT, it is not scalable beyond the needs of a small business. The initial configuration of Intel AMT is performed through a specialized BIOS module, available on systems with Intel AMT, called the Intel® Management Engine BIOS Extension (Intel® MEBX). An administrator uses the Intel MEBX screens, such as the one presented in Figure 1, to enable Intel AMT to configure a password, and possibly specify the network settings required for network connectivity.

From this point onward Intel AMT is accessible over a Local Area Network (LAN), and further configuration or use can take place through a set of Web pages that Intel AMT exposes, such as the one presented in Figure 2.

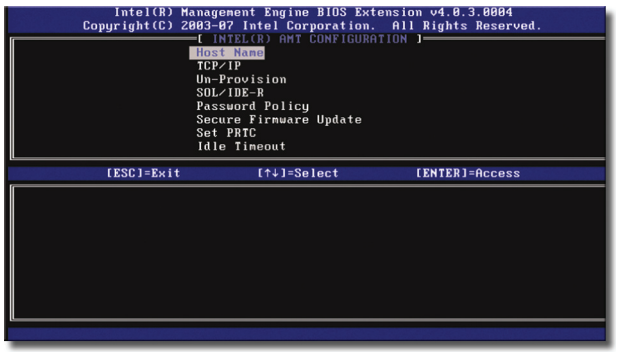

Figure 1: Intel® MEBX, Intel® AMT configuration screen example
Source: Intel Corporation, 2008

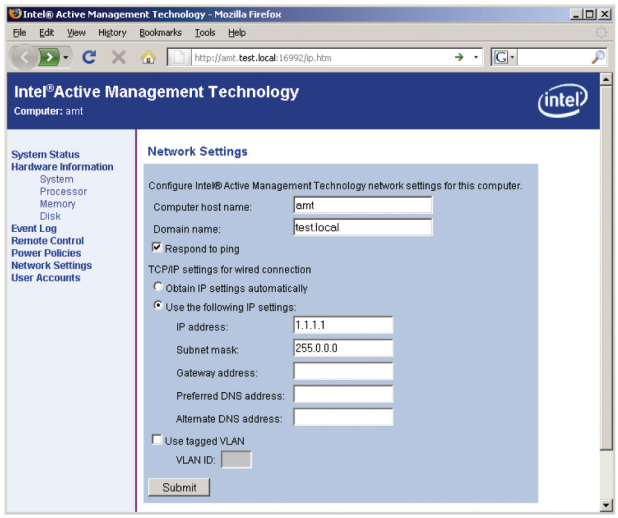

Figure 2: Intel® AMT Web page example

Configuration Process for Enterprises

For enterprises, Intel AMT configuration is performed automatically and remotely, by using a configuration server. The configuration server is integrated into ISV management suites, such as Symantec Notification Server*, LANDesk Management Suite*, and the Microsoft SCCM*. The configuration server establishes a secure connection with Intel AMT and then downloads the configuration data into Intel AMT. The protocols for setting up the secure connection are described in subsequent sections.

Intel® AMT Configuration Methods

Configuration of Intel AMT in an enterprise is fundamentally based on two available protocols. These are the Transport Layer Security (TLS)[5] protocol, based on the Pre-Shared Key (PSK) method; and the TLS protocol, based on the Asymmetric Key method (or remote configuration). Certain attributes and properties of these protocols can be adjusted to achieve varying levels of security and configurability.

TLS Configuration Protocol Based on the Pre-Shared Key Method

This protocol is based on the TLS-PSK Request For Comment (RFC)[6]. This RFC specifies a mechanism by which two parties can establish a secure channel of communication with one another.

In the case of Intel AMT, the two parties interested in setting up a secure communication channel are the Intel AMT and the Intel AMT configuration server. The starting assumption of the TLS-PSK protocol is that both parties must already share a secret. The PSK method offers stringent security that meets the needs of the most security-conscious enterprises; however, this method requires distribution of the PSK and that comes at a price.

Remote Configuration Protocol Based on the TLS Asymmetric Key Method

This protocol is also based on the TLS standard. This TLS standard specifies a protocol by which two parties can set up a secure channel of communication with one another, by using Rivest-Shamir-Adleman (RSA) key pairs established by each of them, a priori. There is no need for the two parties to pre-share any secret, as is the case when using the PSK protocol. This is the biggest advantage of this mechanism, that is, we do not have to devise mechanisms to share secrets, as in the PSK case.

Configuring Enterprise Data

Once a secure and trusted session is established between the configuration server and Intel AMT, the configuration server can push down the enterprise-specific information, enabling Intel AMT systems to be operational on the enterprise network. Intel AMT leverages the Common Information Model (CIM) of the Distributed Management Taskforce (DMTF) [7] to represent the various configuration settings that are communicated, by using DMTF's Web-Services for Management (WS-MAN)[8] protocol. Intel AMT supports DMTF's Desktop and Mobile Architecture for System Hardware (DASH) initiative [9], which aims to standardize the manageability of desktop and mobile systems. Certain configuration properties of Intel AMT utilize DMTF's management profiles mandated by DASH. For example, local user-account management and authorization are based on DMTF's Simple Identity Management [10] and Role Based Authorization [11] profiles.

The Intel AMT Software Development Kit (SDK) [12] provides the complete list of supported management profiles and the complementary CIM-based data model available for ISVs and IT.

Asymmetric Key Method (Remote Configuration) Detailed Flow

Preparation

In this flow, Intel AMT acts as a TLS server, and the configuration server acts as the TLS client. Since we depend on a mutually-authenticated TLS, both parties require private and public RSA key certificates. Intel AMT generates a 2048-bit modulus-based RSA key pair early in its lifetime, creates a self-signed X.509v3 [13] certificate for this key pair, and stores the certificate and private key in the non-volatile flash memory associated with Intel AMT. The enterprise IT administrator obtains a TLS certificate from an Intel AMT partner Certification Authority (CA). Intel has partnered with some of the leading CAs to streamline the process for obtaining certificates for Intel AMT configuration. The Intel AMT firmware image comes pre-configured with cryptographic hashes of the root certificates of these partner CAs (VeriSign*, Comodo*, GoDaddy*, and Starfield*). Since these certificate hashes are part of the firmware image, they provide a pre-configured root of trust for verifying configuration server certificates. The only reason why we chose to store a hash of the root certificate and not the entire root certificate itself is to save storage space on the nonvolatile flash memory. A full certificate could run into a few kilobytes of space, whereas a Secure Hash Algorithm (SHA-1) [14] of the certificate requires just 20 bytes of storage.

Having done the preparation, we now delve into the actual mechanics of the configuration protocol. In the scenario we present, initial Intel AMT configuration takes place at any given time, assuming the host OS is already operational. We define this scenario as the delayed configuration scenario. Delayed configuration caters to both the direct shipment and the IT staging area deployment strategies, mentioned earlier in the "Client System Deployment Strategies for IT" section of this article.

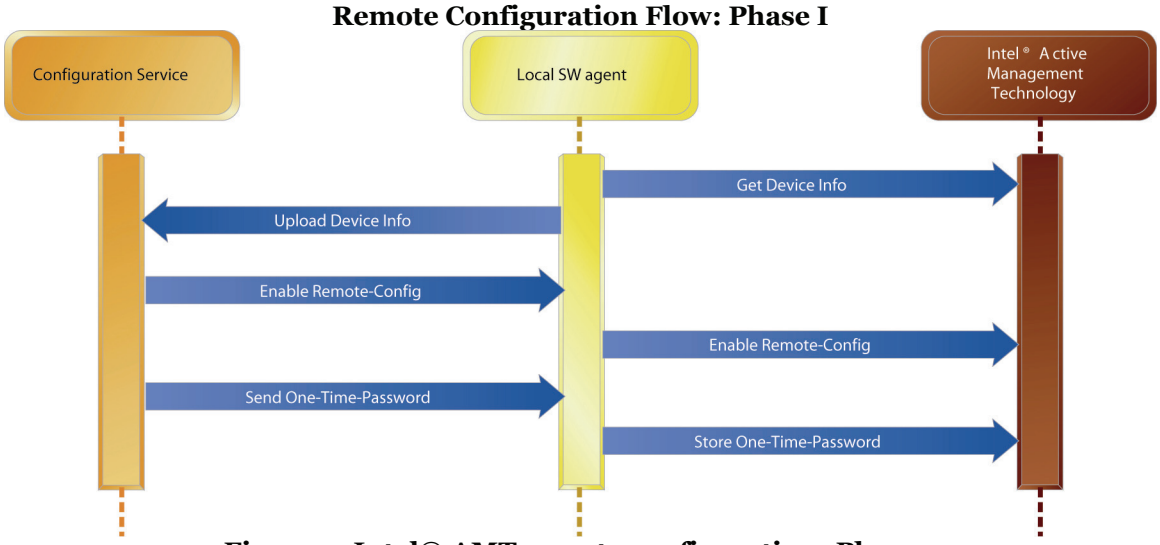

Figure 3: Intel® AMT remote configuration–Phase 1
Source: Intel Corporation, 2008

Figure 3 outlines the initial phase of the flow. To start with, Intel AMT is at the un-provisioned state: its network interface is disabled, effectively disabling any remote configuration attempts; the local host to the A communication channel, known as the Intel® Management Engine Interface (Intel® MEI), is enabled. A local host software agent can detect the state of Intel AMT, by using the Intel MEI, and then can upload device information to the configuration server. The information includes the firmware version, the installed root certificate hashes, and whether the device is configured to operate in PSK or Asymmetric Key provisioning mode. In this scenario we assume it is the latter.

Next, the configuration server instructs the agent to enable remote configuration of Intel AMT, and it provides the agent with a One-Time-Password (OTP). The role of the OTP will become clearer when we explore Phase 2 of the protocol. The agent then sends the "enable remote configuration" and the "store One-Time-Password" commands to Intel AMT. Once Intel AMT enables remote configuration, it enters Phase 2 of the protocol, in which the

configuration server communicates directly with Intel AMT via the Out-Of-Band (OOB) channel. Phase 2 is described next.

Remote Configuration Flow: Phase 2

Figure 4: Intel® AMT Remote Configuration–Phase 2
Source: Intel Corporation, 2008

In Phase 2 of the configuration flow, the configuration server and Intel AMT establish a TLS session between them (see Figure 4). This phase commences when the configuration server opens a TLS session with the Intel AMT device. As part of the TLS handshake, Intel AMT sends its self-signed certificate. In Step 1 in Figure 4, the configuration server skips the verification of the self-signed certificate. However, the server requires the public key embedded in it to successfully complete the TLS handshake with Intel AMT. Next, the configuration server sends the entire TLS certificate chain (including the root certificate) to Intel AMT. In Step 2 Intel AMT validates the configuration server, based on the information provided in the certificate chain. This process entails validating three main areas:

- *Certificate chain validation*. Intel AMT verifies the entire certificate chain, first by verifying that the root certificate is a trusted one. This is done by calculating its hash and comparing the hash with the ones stored in the firmware image. Subsequently, Intel AMT verifies the rest of the certificates in the certificate chain, following the validation procedures described in Section 6.1 of RFC 5280 [15].
- *Configuration server identity*. The identity of the configuration server is provided in the Certificate Subject's Common-Name (CN) field. To verify the authenticity of the Fully-Qualified-Domain Name (FQDN) presented in the configuration server certificate, Intel AMT obtains the name, assigned by the Domain Name System (DNS) of the network it is in, by querying the Dynamic Host Configuration Protocol (DHCP) server for DHCP Option 15. Per the DHCP options RFC [16], Option 15 specifies the domain name the client should use when resolving hostnames, via the DNS.
- *Certificate usages*. Intel AMT first confirms that the certificate includes both "TLS Client" and "TLS Server" roles. This guarantees that the CA issuing the certificate has verified the certificate applicant is associated with the name appearing in the CN field, as is the case for standard TLS certificates used on the Internet. Next, Intel AMT checks a particular qualifier in the certificate, specifically indicating this certificate is meant for Intel AMT configuration. When submitting a request to a partner CA, the IT administrator must explicitly require this qualifier, as Intel AMT will not accept any certificate that does not have this qualifier. The qualifier can be one of two types:
    - A well-defined usage Object Identifier (OID). This OID is defined and registered as an OID to denote it as an Intel AMT configuration certificate. The OID value is 2.16.840.1.113741.1.2.3.
    - A well-known string in the Subject's Organization Unit (OU) field. The string is "Intel AMT configuration server certificate."

If the aforementioned validation steps are successful, the parties complete the TLS handshake. Finally, the configuration server requests the OTP value, mediated through the host agent in Phase 1, and compares this value with the locally-stored OTP value. If the two values are identical, the configuration server has successfully authenticated the Intel AMT device.

At this point in the protocol, both the configuration server and Intel AMT have established a mutually-authenticated and confidential channel through which subsequent configuration commands can flow.

Note that the protocol does not require any manual configuration of the Intel AMT system (no one has to touch the machine), which is a key requirement for many enterprises. Still mal-configuration is quite difficult as it would require an attacker to gain access to the Intel AMT device in two ways: (1) intrude into the local OS, and (2) control the enterprise DHCP infrastructure.

Configuration Hardening Through a One-Touch Operation

Intel AMT offers additional means to harden the protocol, as some enterprises have IT policies that they need to comply with, and the remote configuration protocol does not fit fully within the constraints of those policies. For such customers, the security of the configuration process can be further strengthened so long as these customers are willing to touch the Intel AMT machines and configure some parameters into the Intel AMT subsystem. The mechanisms to configure these parameters via a one-touch operation are these:

- *Manual*. IT technicians can type in configuration parameters through the Intel MEBX user interface.
- *Semi-automated*. IT technicians can use an ISV application to create a configuration information file and store it on a USB thumb-drive. A technician can then repetitively plug the USB thumb-drive into a machine with Intel AMT. Once the machine is powered-up and the Intel MEBX logic executes, Intel MEBX detects the presence of the thumb-drive with configuration information. Intel MEBX then applies the settings to the Intel AMT system. The USB configuration file specification is provided in the Intel AMT SDK. The USB thumb-drive method offers increased convenience and scalability relative to the manual method. Typically, this method can be used for small to medium businesses, or in an IT environment in which PC client deployment is performed in an IT staging area. However, this automated method is not cost-effective if direct shipment to the end-user is involved.
- *Negotiating with a system manufacturer to configure the parameters at the time of manufacturing*. This method is more likely to be used when dealing with a large enterprise customer that orders a fairly substantial number of machines and negotiates with the manufacturer to configure the settings on the computers on the manufacturing line. This saves the customer from having to configure the settings into the machines. We do not go into the economics of this method in this article.

Once the IT has chosen one of the above mechanisms, the following parameters related to the Asymmetric Key configuration are available:

- *Use of an enterprise root CA*. Suppose an enterprise policy disallows trusting a commercial CA for root of trust; company policy only allows for the enterprise's own root of trust. In this case, the enterprise IT administrator can configure Intel AMT with the designated enterprise root certificate hash and disable all existing pre-configured partner CA roots. The configuration server will then have to use a certificate issued by this enterprise CA, instead of any other CA. This mechanism completely cuts off an attacker from being able to attack the configuration process of Intel AMT, because the attacker needs to have access to the CA infrastructure of the enterprise, which is usually a very heavily guarded asset.
- *Specify a trusted pre-configured DNS suffix*. A domain suffix can be configured into the non-volatile flash memory of the Intel AMT subsystem (by using one of the aforementioned one-touch operations). If this domain suffix is configured into the nonvolatile flash memory, then Intel AMT does not use DHCP Option 15 to learn the network's domain suffix. Instead it uses the configured domain suffix validating the FQDN presented in the configuration server certificate. This eliminates vulnerabilities that are due to a compromised DHCP infrastructure. It also allows configuration in environments where the DHCP infrastructure does not support Option 15.

Configuration Audit Record

Once the device with Intel AMT establishes trust with the configuration server, it creates a configuration audit record, recording the configuration TLS certificate details and additional parameters. This record is subsequently locked down

to prevent any further modifications, but it is still available for being read via the local Intel ME Interface as well as through the Intel AMT WS-MAN interface. Since the record is read-only it allows policy enforcement applications to detect occurrences of unauthorized configuration and use of Intel AMT systems.

Pre-Shared Key Protocol: Detailed Flow

The starting assumption of the TLS-PSK protocol is that both parties must already share a secret. In our context, we call this shared secret the Provisioning Passphrase (PPS). There is also an associated identifier with each PPS called the Provisioning ID (PID). The PID and PPS are strings of characters comprising capital letters A-Z and numbers 0-9. The PID is 8 characters long and the PPS is 32 characters long. The PPS offers up to 125 bits of entropy.

Sharing of PID and PPS between the configuration server and the Intel AMT subsystem is achieved by any of the one-touch methods we described previously in the context of the Asymmetric Key flow: manual configuration through Intel MEBX or loading of a configuration file with PPS and PID information onto a USB thumb-drive. Sharing can also be done by the system manufacturer.

In these last few sections we describe how symmetric PSK-based configuration works for machines with Intel AMT.

Pre-Shared Key Provisioning Flow

Just as in the asymmetric configuration model, described in the previous section, there's an initial phase that involves a communication between a software agent and the configuration server, that is similar to the one described for the Asymmetric Key method. The main difference is that there is no need to pass an OTP value between the two parties, since the shared secret key (PPS) is used by the configuration server to authenticate the device with Intel AMT, eliminating the need for an OTP value. The second phase is also straightforward, as it involves a standard TLS-PSK handshake, whereby Intel AMT acts as the TLS server, and the configuration server acts as the TLS client. Per the TLS-PSK RFC, Intel AMT sends the PID as the "PSK_Identity_hint" value within the TLS handshake, allowing the configuration server to locate the matching PPS value and use it in the communication.

Note on Security

It is more secure if every machine is configured with a unique PID/PPS pair; however, in some scenarios, multiple systems are configured with the same PID/PPS pair. This method reduces the number of PID/PPS pairs to be managed and can in this way it be seen as more convenient; however, it is less secure. If, for example, an attacker breaks into the hardware of one machine in that group of machines and acquires the PPS value, then the security of all the machines is compromised.

Bare-Metal Configuration

The scenarios demonstrated in the previous section were called delayed configuration scenarios, and the assumption is that the Intel AMT configuration process takes place once the host OS is already deployed. Recall that Phase 1 for the two configuration methods required a software agent to enable the network interface of the Intel AMT system and provide discovery information back to the configuration server.

Bare-metal configuration is another configuration capability of Intel AMT that allows configuration prior to OS installation. In fact one key usage of bare-metal configuration is to push down an OS installation or image to the platform by using Intel AMT remote boot operations. Naturally, this step can only take place after Intel AMT has been configured.

Both PSK and Asymmetric Key methods can be utilized for bare-metal configuration. Intel provided system manufacturers the capability to designate in manufacturing a "bare-metal timer", typically limited to 24 hours, in which Intel AMT enables its network interface. This allows a configuration server to configure a device without the need for the software agent trigger, required in Phase I of the delayed configuration model. Bare-metal configuration is enabled from the initial boot of the system and for the accumulated system up-time duration, specified by the bare-metal timer. After this duration, Intel AMT disables its network interface. To configure Intel AMT from this point onward, the delayed configuration method must be used. Following is a sequence diagram depicting Phase 1 for bare-metal configuration. In order to use bare-metal configuration, an alias for the configuration server address is registered on the relevant DNS servers in the enterprise. The reason for this will become clear when we describe the protocol details.

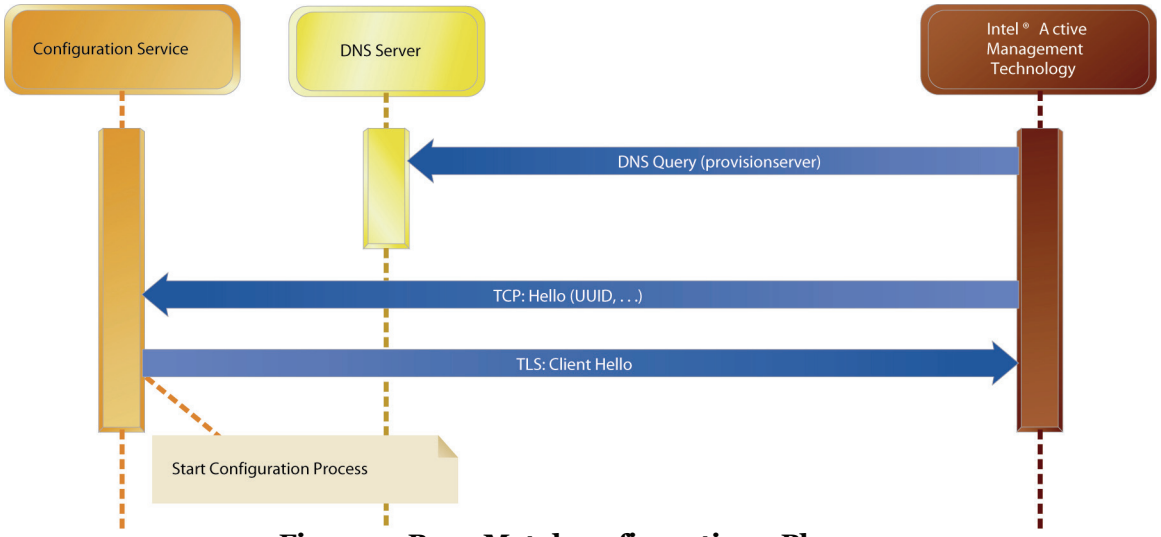

Figure 5: Bare-Metal configuration—Phase 1
Source: Intel Corporation, 2008

Figure 5 illustrates Phase 1 of bare-metal configuration. During the bare-metal time window, Intel AMT tries to acquire a DHCP IP address, detect the DNS server address, and use that address to query for the designated configuration server. Intel AMT uses a concatenation of a predefined host name, "provisionserver" and the DNS suffix it has learned. If, for example, the DNS suffix returned by the DHCP server is foo.com, Intel AMT tries to resolve both "provisionserver" and provisionserver.foo.com entries. If it succeeds, Intel AMT sends a notification to the configuration server, depicted in Figure 5 as a "Hello" message. The Hello message is TCP-based and provides the configuration server with the platform's Universally Unique Identifier (UUID) [17] and additional information that can assist in completing the configuration process. Note that the message does not carry any authentication information. As long as the bare-metal time window has not elapsed, the configuration server can complete the configuration process (by using either the PSK or the Asymmetric Key method). Note that if the Asymmetric Key method is used, then the configuration server has no way to validate the authenticity of its peer.

Conclusion

In this article we discussed the various scenarios that are supported for configuring Intel AMT, the various mechanisms and protocols available for configuration of Intel AMT, and we outlined the various options and parameters that can be adjusted to make the tradeoffs between security, cost, and convenience. One of the aspects of Intel AMT configuration that we want to reiterate here is that Intel AMT offers a wide selection of configuration options, catering to almost every type of customer, ranging from a small home business to a Fortune 500 enterprise. At one end of the spectrum, it is possible to configure Intel AMT in a matter of minutes, and get it up and running on a test machine. At the other end of the spectrum, it is possible to configure a vast array of machines with Intel AMT in a large enterprise without even physically touching those machines once; moreover, they can be configured in such a way that the process is trusted and secure, and not vulnerable to being attacked or snooped by malware or other prying eyes.

Acknowledgements

We thank Tsippy Mendelson, Moshe Valenci, Randy Templeton, Gareth Bevan, Michael Navon, and Gal Alkon for reviewing this article.

References

[1] DMTF. "Alert Standard Format Specification (ASF)." DSP0136. April 2003.

[2] Intel Corporation. "Wired for Management Baseline, Version 2.0." December 1998.

[3] Dell, Inc. "ASF Administrator's Guide." May 2002.

[4] R. Richardson. "CSI Computer Crime and Security Survey." 2008.

[5] Dierks, T. and E. Rescorla. "The Transport Layer Security (TLS) Protocol." Version 1.1. RFC 4346. April 2006.

[6] Eronen, P. and H. Tschofenig. "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)." RFC 4279. December 2005.

[7] DMTF. “Common Information Model (CIM) Infrastructure Specification.” DSP0004, Version 2.3. October 2005.

[8] DMTF. “Web Services for Management (WS-Management) Specification.” February 2008.

[9] DMTF. “DASH Implementation Requirements, Revision 1.0.0b.” October 2007.

[10] DMTF. “Simple Identity Management Profile, DSP1034, Version 1.0.0.” July 2008.

[11] DMTF. “Role Based Authorization Profile.” DSP1039, Version 1.0.0a.” October 2006.

[12] “Intel AMT Software Development Kit.” At http://software.intel.com/en-us/articles/download-the-latest-intel-amt-software-development-kit-sdk

[13] “ITU-T Recommendation X.509 (2005) | ISO/IEC 9594-8:2005.” Information Technology-Open Systems Interconnection-The Directory: Public-key and attribute certificate frameworks.

[14] NIST FIPS PUB 180-2. "Secure Hash Standard." National Institute of Standards and Technology. U.S. Department of Commerce. August 2001.

[15] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL).” RFC 5280, May 2008.

[16] S. Alexander and R. Droms. “DHCP Options and BOOTP Vendor Extensions.” RFC 2132, March 1997.

[17] DMTF. “System Management BIOS (SMBIOS) Reference Specification, DSP0134, Version 2.6.” September 2008.

Authors' Biographies

Purushottam Goel graduated from the University of Roorkee, India (now called the Indian Institute of Technology) with a bachelor’s degree in electrical engineering, and later pursued his master’s degree from the computer engineering department of BITS, Pilani, India. He was at the Bangalore R&D Center of Novell, Inc. from 1996 to 2000, working on various projects, most notably on the security components of the NetWare* operating system. Subsequently, he tried his luck in a couple of startups, before joining Intel in 2002. Purushottam is one of the key architects of Intel® AMT. He is responsible for all security features and aspects of the product, including the design of the provisioning and setup mechanism of Intel AMT. His email is purushottam.goel@intel.com.

Dori Eldar joined Intel in 1999 and is currently working as a software architect in Intel’s Digital Enterprise Group at the Israeli Design Center, Jerusalem. His current work relates to manageability technologies: he holds four patents in this area and more pending. He received a B.S. and M.S. degree in mathematics from the Hebrew University in 1996 and 1999, respectively. His email is dori.eldar@intel.com.

Arvind Kumar is Principal Engineer and Chief Manageability Architect at Intel. During his 14 years at Intel, Arvind has worked on different manageability products, including server, blades, and desktop and mobile systems; he is currently driving long-term common manageability architecture. Arvind represents Intel in DMTF and has been active in industry manageability initiatives such as CDM, SMASH, and DASH. His email is arvind.kumar@intel.com.

Remote System Repair Using Intel® vPro™ Technology

Venkat R Gokulrangan, Mobility Group, Intel Corporation
Ram Chary, Mobility Group, Intel Corporation
Dominic Fulginiti, Digital Enterprise Group, Intel Corporation
Ansuya Negi, Mobility Group, Intel Corporation
Hema Tahilramani, Mobility Group, Intel Corporation
Jeong Yoon, Mobility Group, Intel Corporation

Keywords: Intel® AMT, Active Directory, boot, diagnostics, discover, fast call for help, heal, manageability, PC maintenance, protect, remote, repair

Abstract

Client platform diagnosis and repair of hardware and software components are key post-deployment aspects of a typical enterprise IT shop or an Original Equipment Manufacturer (OEM) PC support cycle. Even though most support issues in client systems are remotely resolved by using applications running in the client operating system (OS), a certain percentage of IT support calls have traditionally required technicians to make desk-side visits to resolve the problems. As Mobile platforms become more ubiquitous, the ability to provide support can be more challenging due to the on-the-go nature of these clients. When these clients are not on-site and they experience problems, such as the inability to boot the user's OS, sluggish performance, or a malfunctioning network driver, having the ability to remotely and securely diagnose and remediate problems reduces the support costs for IT organizations and OEMs.

In this article, we discuss key scenarios for remote platforms in such inoperative states and describe remediation solutions for platforms with Intel® vPro™ technology. We will also discuss certain performance issues related to the protocols laid out in the solutions and will outline how service-oriented solution architecture can utilize Intel vPro technology to provide such services for the small to medium business markets.

Introduction

Problem Summary and Motivation

Enterprise IT personnel have to ensure that client platforms located on premise or in extended offices can function effectively and efficiently with minimum interruption of service to end users. The operational state of a system depends on fully functioning hardware and software components. When a component fails, IT experts utilize software agents in the operating system (OS) to initiate remediation. Typically, IT shops use off-the-shelf software to perform such tasks, but when such actions do not fix the problem, IT technicians have to make desk-side visits to resolve the problem. Situations such as the inability to boot the user's OS, sluggish performance of an OS, or malfunctioning network drivers usually warrant desk-side visits.

As enterprises transition to using mobile systems, such as laptop or notebook computers that increase worker productivity inside and outside the enterprise, the ability to support these platforms when they are remotely located, poses a considerable challenge.

There are two solutions currently utilized to address this challenge of supporting clients remotely. However, both of these solutions have limitations. The two solutions and their limitations are:

- Embed a diagnostic/spare-tire OS, such as Windows PE* or Windows XP*, in an unused disk partition in a local storage device. While original equipment manufacturers (OEMs) deploy this solution, IT shops usually replace the hard disk content with a custom image tailored

for their own employees. Further, adding another OS such as an XP-embedded (XPE*) one, which is the most suitable OS for downtime productivity, usually involves significant additional capital cost per client as well as increased maintenance costs.

- Download the diagnostic/spare-tire OS from IT's remote console to the client on a demand basis. While this is feasible, there is no well-defined mechanism to search, locate, and manage a client, or to query the operational state or the platform integrity over the Internet on demand.

Desktop and mobile platforms with Intel vPro technology and Intel® Active Management Technology (Intel® AMT) provide new features to better support PCs remotely and remove some limitations of current solutions, reduce the need for desk-side visits, and reduce operational costs. Intel Fast Call for Help, IDE-Redirection (IDE-R), and platform Remote Access that uses Serial-over-LAN (SOL) are features of Intel AMT that provide the infrastructure to support platforms remotely while providing robustness and availability of the solutions even when the OS is not present.

While Intel vPro technology addresses many of the challenges of supporting platforms remotely, it is important to note these embedded technologies operate in very constrained environments. These may include environments where the power is low, and where limited static and dynamic memory is available. Such environments could potentially undercut some performance capabilities of the embedded OS.

In this article, we propose a two-stage solution architecture: Remote System Repair (RSR). In this solution, the features of Intel vPro technology, Fast Call for Help, and IDE-R are used together to perform platform diagnostics and repair. While Fast Call for Help enables a secure connection to the management console, IDE-R and SOL are used to provide tools to heal the client remotely.

Organization of This Article

This article is organized as follows. We first look at two applicable problem scenarios and describe why existing in-enterprise solutions cannot be used to repair on-the-go mobile clients. We then discuss the ingredients of Intel vPro technology that are used in the remote heal solution and describe three use cases to showcase this solution at work. We also provide a sample solution sequence that uses Microsoft MS-DOS*, and show how someone can construct a solution to solve remote repair problems. We move on to outline the performance challenges encountered and how we resolved them with Intel vPro technology. Finally, we extend the solution to service-based heal usages in constrained deployment environments that could be delivered to small and medium size businesses (SMBs) using a subscription or pay-per-use model. We wrap up by looking at the future of the RSR model.

Problem Description

Discover and Heal

As notebook and laptop platforms become more ubiquitous in the realm of enterprise computing, the chances of them being outside the physical premises of the enterprise and requiring support at some point in their life cycle are high. It is critical, therefore, to be able to manage and support these remote platforms to maintain end user productivity and reduce support costs. The manageability and support of these platforms can be divided into two distinct phases: discover and heal. [1]

The discover phase involves the process of managing the location and identity of the platform as well as managing asset-related information. Typically, this involves a corporate server actively looking for registered clients to ensure that the server tracks appropriate hardware and software assets to protect the client while it is outside the enterprise and to protect the corporate infrastructure from an infected client. Enterprise technology, such as Active Directory, is typically used to locate the client during the discover phase. The challenge for discovery occurs when the platform is outside the firewall and not in a healthy state.

The heal phase includes the process of re-actively managing the health of the system, including restoring the operational health of a platform to a well-known, operable state.

In this article, we take a closer look at one of the reactive solutions and show how a console can remediate a system that is enabled with Intel vPro technology. We present two scenarios in which problems are identified, and we show how Intel vPro technology can help solve these problems.

Scenario 1: Non-Booting Client System

Information technology (IT) departments often encounter a situation in which a platform will not even boot up to the normal OS. Possible reasons for failing to boot up include hardware issues such as a memory bank being defective, incorrect arguments programmed in hardware or software, a missing OS file, corrupt boot drivers, invalid registry entries, and

so on. In most of these situations, the normal process would be for the customer support person to walk the customer through the diagnostic procedure manually over the phone, without the support person being able to see any of the end-user actions. What if a remote capability existed that could view the screen remotely and exercise control over the client? This would help the support person to resolve the issue quickly, without end-user support.

Scenario 2: A Software Disk Unlock for Disk Encryption Software

In this scenario, the host OS does not boot but this time it is not due to any component failure. Rather, it is due to the fact that the user forgets the password that is required by platform authentication software and therefore the platform cannot be booted. The escrowed key, which is backed up by enterprise infrastructure, must be re-presented to the host software to resolve this issue, when other local authentication recovery schemes do not help resolve the problem.

We now look at how RSR solution can help solve these problems by using features of Intel vPro technology.

Remote Heal Usage Requirements
When clients are on the go or they are in geographically distant places, the remote healing of those platforms depends on two important characteristics:

- Robustness of the client network connections, including performance, reliability, and latency.
- Establishment of a secure connection to the platform to facilitate remediation.

Without a secure, robust connection, the advantages provided by Intel vPro technology are diminished.

Prior to 2008, Intel vPro technology provided the capability to remediate a client remotely by using IDE-R, but the client had to be inside the firewall of an enterprise network. Beginning with new desktop and notebooks with Intel vPro technology introduced in the second half of 2008, IT organizations can utilize a new feature in Intel AMT, Fast Call for Help, to resolve the issue of managing extended and home offices.

Criteria of a Successful Remote Heal Solution
In designing the RSR solution, our goal was to allow a remote console to locate a client system, inside or outside the enterprise, in a secure manner. In addition, we expected that the solution meet end-user quality levels, while providing a low total cost of ownership (TCO). The performance levels of an embedded heal solution should be comparable to those of a host OS in a similar usage situation. Specifically, any visualization and interactive behavior experienced by the customer support staff remotely should not differ substantially from the experience of operating the system directly.

RSR Architecture Overview

RSR is a succinct solution architecture using two key technological ingredients present in platforms with Intel vPro technology, Fast Call for Help and IDE-Redirect (IDE-R).

Fast Call for Help
Fast Call for Help allows platforms outside the enterprise firewall to trigger a connection to the Internet with a gateway enabled by Intel vPro technology, as shown in Figure 1. The platform user presses a special button or key combination to initiate the platform connection to a remote gateway enabled with Intel vPro technology. Upon establishing a successful connection, a standardized event, such as a Platform Event Trap (PET) or WS_Event is propagated to the enterprise network to register the client, either in the enterprise or in the remediation network. Full details of this technology are provided in [2].

Once this secure connection to the enterprise is established by the platform firmware, all of the traffic originating from the client can be routed through this connection. Any console can potentially connect to this remote platform and subsequently trigger the remediation process.

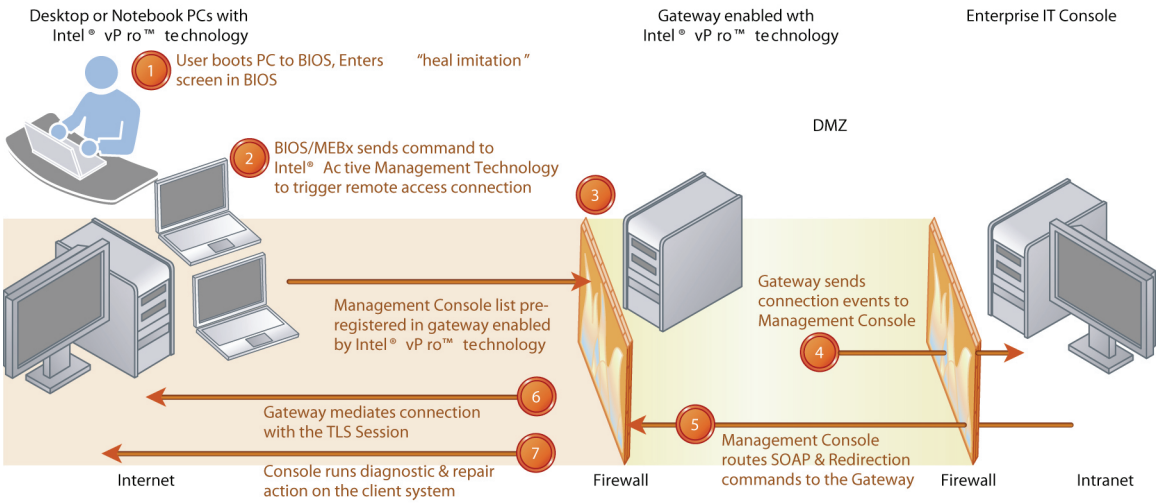

Figure 1: Intel Fast Call for Help
Source: Intel Corporation, 2008

IDE- Redirection

The IDE-R feature in Intel vPro technology platforms provides the capability to boot a platform to a remote CD/DVD drive or to a bootable image file present in a server. Basically IDE-R creates the illusion that a remote CD drive or an ISO image is on a local drive. Intel vPro technology internally intercepts all read/write requests and transports them over a pre-established, secure, network connection set up just for this purpose. This gives the platform an opportunity to boot over the network by using IDE-R, and upon successful boot, it provides access to a remote CD/DVD drive. Usually, images, such as a diagnostic OS, are used for remote heal.

RSR Solution Outline

By using Fast-call for Help and IDE-R, a two-stage RSR solution can be created with appropriate sequencing.

The first stage involves booting the platform to a small OS as follows:

1. Instruct the end-user to press a key and initiate a secure Fast-call for Help session.

2. Upon successful connection, the console should take control of the system with user acceptance and remotely redirect the client to boot to a small OS, such as MS-DOS, from the server.

In the second stage, utilities from stage one will help copy the diagnostic OS image or the spare-tire OS image, from the remote drive, to a storage device in the client. Once the image is present locally, the client can be booted to it as many times as necessary until it is fully functional, without having to download the image every time.

RSR Use Cases

Case 1: Re-use Local Diagnostics and Repair

Many OEMs build in some basic hardware diagnostic routines into their BIOS. Since BIOS setup usually runs in text mode (or can be run in text mode), it can be remotely controlled via SOL. Using SOL, IT support personnel can boot the system into BIOS and perform diagnostics remotely to determine if major hardware components are functioning properly. A typical example of this use case is to verify if the hard disk is functional by remotely performing hard-drive diagnostics from BIOS.

Some OEMs also install a diagnostic OS in a separate partition in the hard disk drive (HDD). Such an OS can be booted when the main OS is inoperable, by pressing a special key during boot. Typically a diagnostic OS is a trimmed-down version of the main OS, and therefore it comes with many of the same system tools as the main OS, tools that can be used to diagnose and repair problems. With IDE-R, additional tools and updated drivers can be downloaded at runtime and patched into the main OS. To remotely display and control this OS, some type of remote keyboard/video/mouse (KVM) capability is needed, since the OS typically runs in graphical mode, and SOL is effective only in text mode. Alternatively, an OEM can build software-based remote-control applications, such as KVM redirection software, into the diagnostic OS for systems without hardware KVM features. This is shown in Figure 2.

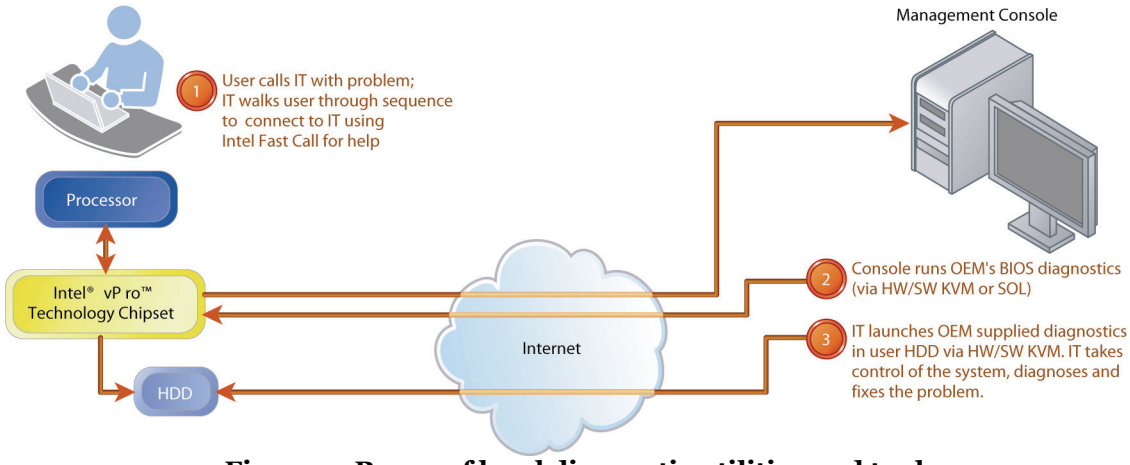

Figure 2: Reuse of local diagnostic utilities and tools

Case 2: Download and Store Diagnostic OS Using a Two-stage Boot Process

If the system does not have a resident diagnostic OS, IT support personnel can remotely download or install one at runtime. IDE-R and SOL can be used to boot a small footprint OS, such as MS-DOS or a trimmed-down Linux* OS. Once the small foot-print OS is running, then a full-featured diagnostic OS can be downloaded to the user's system. It can be installed on a Universal Serial Bus (USB) flash drive or the main HDD, if the drive and its file system are in a healthy state. The system can then be booted into the diagnostic OS for interactive diagnostics and repair. If the diagnostic OS runs in text mode, SOL is all that is needed to remotely control this OS. But, if it runs in graphical mode, a remote KVM (either hardware or software) capability is needed. This use case is shown in Figure 3.

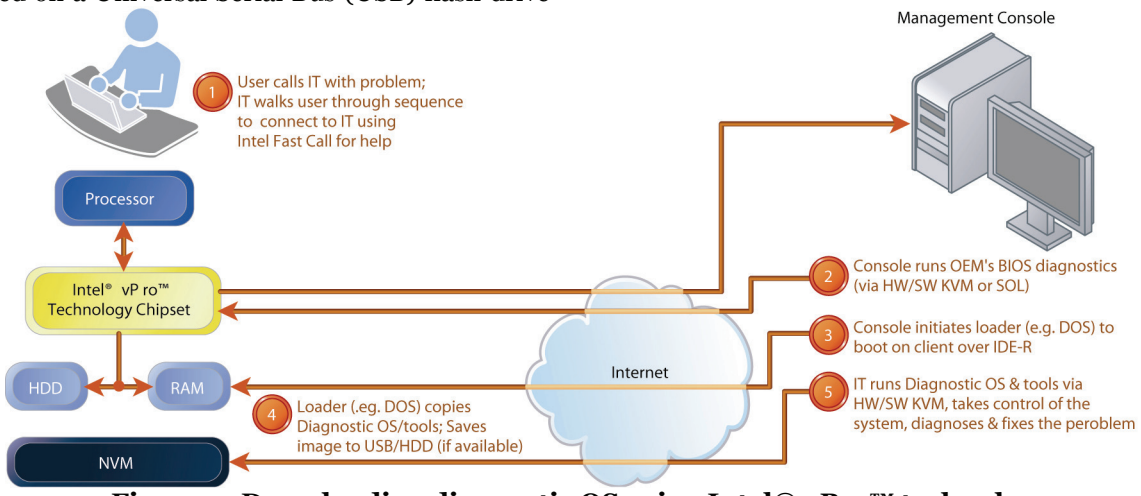

Figure 3: Downloading diagnostic OS using Intel® vPro™ technology

Case 3: Download and Store a Spare-Tire OS on Local Resources

When all remote diagnostics fail to repair the problem, IT support personnel can remotely download and install a spare-tire OS on the user's system. A spare-tire OS is a locally resident OS that allows a user to regain basic functionality of the system on a temporary basis, until the system can be permanently repaired. A spare-tire OS can be a version of Windows* (such as Windows XP Embedded or WinPE* 2.0), or any other OS that can be downloaded and stored as a ready-to-run disk image–such as a file in International Standards Organization archival format (ISO). It can be installed on a USB flash drive or the main HDD if the OS is in a healthy state. From the spare-tire OS, the

user can access data in the main partition of HDD, run essential office productivity applications, access the Internet, and so on.

In addition, for use cases 2 and 3, there are tools available to download large images in a robust manner—especially for unreliable network connections. This can simply be achieved by downloading the image in chunks and resuming downloads from the point where the connection was broken.

This procedure makes the system usable in a relatively short time and allows the user the option of delaying the repair until later. This can be particularly useful if the user is away from the office but needs a computer to access his/her data: the repair of the user's system can be put off so it doesn't interfere with the user's productivity. The spare-tire OS download use case is shown in Figure 4.

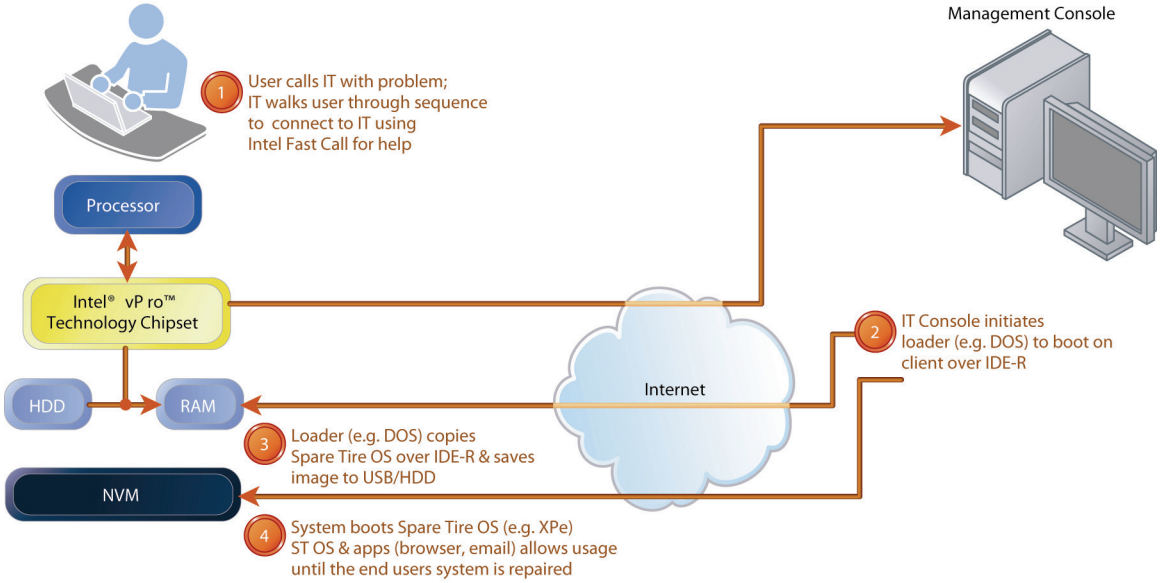

Figure 4: Downloading spare-tire OS using redirected storage

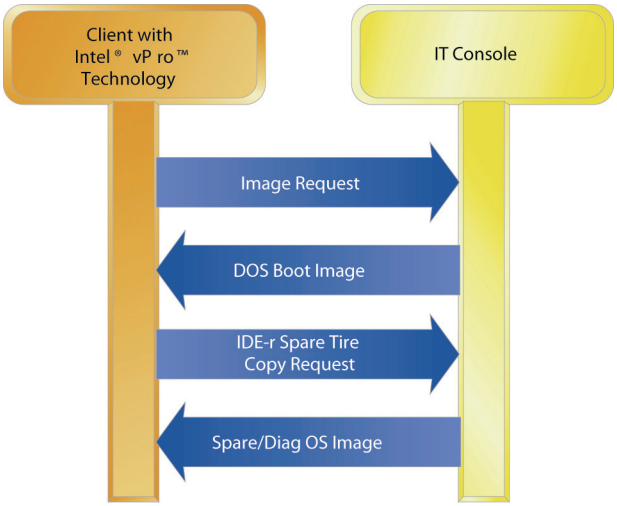

Figure 5: DOS-based, two-stage RSR solutions

Sample Implementation of RSR

We have implemented use cases 2 and 3 in Figure 4 by using a two-stage solution based on a diagnostic OS. This method was chosen due to its small footprint, but also because it can use the IDE-R capabilities of Intel vPro technology. The sample two-stage DOS-based RSR solution relies on the native BIOS USB stack for storage purposes. The overall flow of this two-stage solution is shown in Figure 5.

Software Disk Unlock Scenario—Solution

With RSR in place, we can create a solution for the software disk unlock scenario presented earlier. If a software-based disk encryption solution is used and it has to be unlocked for any reason, the remote console can boot the system to a remote CD image by using Fast-call for Help and IDE-R. Doing so allows an IT shop's disk recovery solution code to be run from the remote location and present the authentication credentials to the verifying application. By using SOL/keyboard replication, the remote console can provide the master key or the escrowed key so that the disk can be unlocked and the system subsequently booted.

Remote Heal Solution Challenges

As we continue to make technological advancements, the usability of such advanced features is a key concern. These range from simple ease-of-use to preventing an unauthorized person from gaining access to corporate data.

Usability Challenges

Triggers to Initiate Heal Usage

Currently, the trigger to establish the Fast-call for Help connection to a remote gateway enabled by Intel vPro technology is initiated in software—usually controlled by the BIOS or the host OS. However, the trigger cannot be initiated in some unknown OS or BIOS state, such as when the OS ends up in an inoperable state. For BIOS-based triggers, the user has to be instructed to interact with BIOS screen.

IT Integration Challenges for Remote Services

Currently, when an IT worker (customer) hits the special key for Fast-call for Help to trigger the remote heal usage, the connection event itself is trapped at a remote IT server, waiting for a corresponding event to be generated to register the client in the enterprise Active Directory. Specifically, if the customer has already triggered this remote access and subsequently talks to a remote console operator over a phone, there is no easy way to associate the operator's remote console being used by the service provider to the existing remote connection event. To compare this to a phone-based support infrastructure it would be like having an automated call management tool route waiting phone calls to a servicing agent or a queue until service personnel are available. The interface of the remote access server or service into an existing remote console infrastructure is critical for successful customer interaction. Specifically, the following integration aspects need to be covered in the overall service delivery:

- Direct mapping of a remote access event from a specific client platform to a customer support console where all heal actions associated with the client platform are performed.

- Ability to trap or forward an existing remote Fast Call for Help connection to the remote console, in case the customer does not make contact over a phone but instead initiates a client connection by using Fast Call for Help.

- Active Directory for swift integration of a remotely connected client event such as PET/WS-Event via Fast Call for Help to the enterprise Active Directory or Name Resolution service.

- Quality of service metric to monitor and improve the clients that are serviced by Fast Call for Help technology.

The resolution of such new challenges is critical for the successful integration of this remote heal usage in the enterprise.

Deployment Issues of a Remote Access Gateway

Deploying a RSR solution utilizing Fast Call for Help requires the deployment of an Intel vPro-enabled gateway to provide the client authentication and access service. This may be a standalone gateway or integrated into an existing gateway residing in an enterprise DMZ for normal IT usage. If deployed as a standalone gateway this adds the need to maintain potentially another gateway in the enterprise DMZ.

Performance issues in Delivering Remote Heal Usage

High-Latency Networks

The IDE-R protocol was implemented and designed to work within the enterprise, that is, on high-bandwidth, low-latency networks. Further, the embedded processor (EP) operates in a constrained memory environment. It limits the number of buffers that are posted to WiFi/Ethernet* Network Interface Cards (NICs) and limits the socket buffers in the Intel Management Engine (Intel ME) network firmware—just enough to meet targets in low-latency networks. However, as the latency increases, the buffers needed to hide the latency are insufficient to meet these targets, thereby compromising network performance.

In the remote system repair scenario, the platform connects to the Internet by using DSL, or cable or fiber optic networks. These are usually limited in bandwidth and have higher latencies. In this environment, as network latencies increase, the performance drops much faster—especially in typical Internet speeds and latency ranges. This performance drop was observed in the context of both Transmission Control Protocol/Internet Protocol (TCP/IP) and IDE-R protocol stacks.

In addition, all application layer protocols such as IDE-R are multiplexed over the Intel AMT Port Forwarding (APF) protocol, used by Fast Call for Help technology in a Transport Layer Security (TLS) session. So, these application-level flow controls are

also layered on top of APF's own flow control. As a result, performance issues are accentuated when the additional protocol layer, with its own data-trunk protocol, is introduced. This limits the efficiency of payload transfer.

The Impact of High Latency Network Performance on RSR Solution

Table 1 below demonstrates the impact of a high latency network on the RSR solution. In this table, we show a comparison of the IDE-R application with and without optimization on an APF connection.

For the purposes of measurement and tuning, we used the runtime performance of IDE-R by remotely mounting a CD image and copying large files from the server to the client, enabled with Intel vPro technology. We compared how well the files copied over APF connections versus non-APF connections. The Internet latency and bandwidth were emulated by using a NISTNet* Linux* router sitting between the client system and the remote access gateway.

Table 1: IDE-R application performance without optimization on an APF connection

Bandwidth: 2 Mbps Symmetric (250 KB/s)
File Size: 16 MB

	Round-trip Time (RTT)		
	0 ms	40 ms	80 ms
Without optimization IDE-R Copy Throughput (KB/s)	185	34	18

To provide real world context, without optimization, transferring a copy of a 100MB ISO file would take approximately 95 minutes with limited success.

Tackling the High Latency Network Issue

In an effort to improve the RSR performance, we began our analysis of the end-to-end IDE-R copy performance by using the layered application architecture. We started at the bottom of the stack and moved up, while ensuring the corresponding peer in the server side was equally capable of delivering the performance. Figure 6 shows the application stack.

We first focused on the TCP/IP stack implementation in the Intel ME, and then analyzed IDE-R, since IDE-R can simply be transported on TCP/IP. The APF layer (c and d) is a transparent layer that can be added or removed as part of the overall stack–giving us the flexibility to fine-tune IDE-R, independent of APF. Subsequently, we introduced the Fast Call for Help or APF layer over TLS, to understand the impact of TLS and the multiplexing protocol (APF).

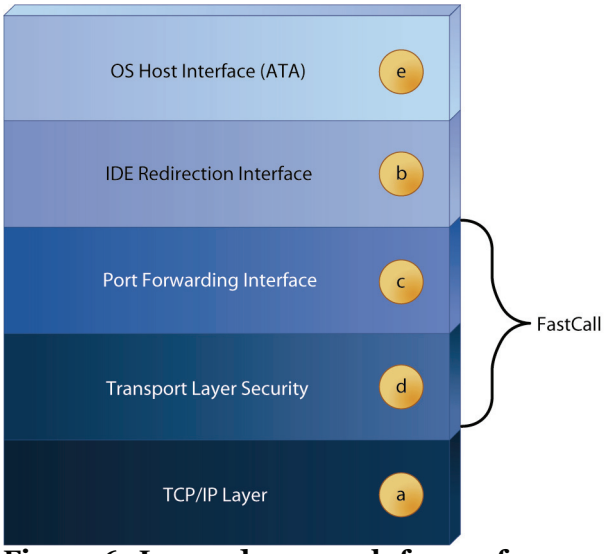

Figure 6: Layered approach for performance analysis

The Improved Performance of Intel® vPro™ Technology Architecture

TCP/IP stack in the Intel® Management Engine (Intel® ME)

In order to ensure the highest performance from the TCP/IP stack implementation, we instrumented it by adding added a special FTP module; we designed the module to discard packets once FTP packets are received. We found that the TCP was advertising a small receiver window (8K) over the wire by using tools such as Wireshark*. We fixed this by increasing the size of the receiver window, as well as increasing the socket receive queue to 64k. The TCP/IP stack then started performing at the levels of Windows XP*. For a 2Mbps symmetric link, 32KB buffers or a maximum TCP receive window are sufficient. With the fixes, we concluded that the TCP/IP stack performance was sufficient for on-the-go usage.

IDE-R on regular networks

Since IDE-R is a TCP application and has its own buffering scheme, we focused on the sizes of the IDE-R buffer, since we knew that at higher latencies, larger buffer sizes or more data were needed to fill the end-to-end network pipe. Applying the same thinking to IDE-R, we increased the redirection buffer sizes to 64K to measure any improvements in performance. For 80ms RTT on a 2Mbps symmetric link, with 64K buffers, we measured an IDE-R file copy throughput of 178KB/s, as opposed to 18KB/s under the same network conditions without the optimizations.

IDE-R with Fast Call for Help
Since layer (a) and (b) in Figure 6 were fixed by increasing the buffer size, we introduced the Fast Call for Help layer with APF and TLS encryption above TCP/IP. Fast Call for Help was simply a multiplexing and encryption layer, and since encryption was done in hardware, we expected no difference in performance levels for IDE-R with the introduction of APF and TLS.

However, our measurements indicated low performance at un-buffered levels. Subsequent wire-level debugging of packet traces indicated smaller receiver window size and delayed acknowledgements from the Intel ME firmware. These issues were resolved by turning on the TCP_NODELAY option and increasing the receiver window size appropriately. The new performance measurements are shown in Table 2.

APF Protocol and implementation analysis
Even with the changes discussed so far, we observed that on a 250KB/s link at 80ms RTT, FTP throughput was 230KB/s and IDE-R throughput was 178KB/s. Conversely, IDE-R over Fast Call for Help throughput was only 110-120 KB/s. Further investigation led us to take a closer look at an aspect of the APF protocol itself.

Table 2: Optimized IDE-R application performance

Bandwidth: 2 Mbps Symmetric (250 KB/s)
File Size: 16 MB

	Round-trip Time (RTT)		
	0 ms	40 ms	80 ms
With Optimization IDE-R Copy Throughput (KB/s)	195	158	124

APF has its own flow control layered on top of TCP's flow control in order to ensure fairness to multiple TCP connections sharing the Fast Call for Help tunnel. With data from APF being sent in 4380-byte chunks to the TCP, efforts are underway to increase this to 64K for increased throughput.

To ensure that TLS encryption was not limiting the performance, we removed the TLS layer and found little difference in performance. This was in line with our expectation that encryption was offloaded to the hardware engine.

Host Interface (ATA) and end-to-end performance
Even after completing the performance optimizations at all lower levels, the maximum observed throughput at that target of 80ms was still at 120 KB/s. This was just a little more than 50 percent of host-based FTP transfer rates.

As a result, we focused on the topmost layer of the application stack. The host IDE/ATA stack issues a 64KB request for data to the underlying CD drive. In the case of IDE-R, these data are fetched from the remote drive by using the IDE-R messages. This sequence can be established as follows:

1. The host OS issues a 64KB request to firmware. These requests are standard requests issued by operating systems such as Windows XP* and Windows Vista*.

2. The Firmware sends a 64KB request via an IDE-R message to the remote server spanning the entire latency.

3. The remote gateway starts the 64KB transfer only after "t/2" msec if "t" is the end-to-end RTT.

4. The Firmware receives and copies 64KB of data to the host; however, the first packet of the 64KB is received "t/2" msec after Step 3 is completed.

5. The host initiates the next request as in Step 1.

Therefore, in the case of high-latency networks, after completion of one ATA request, no data were transported until the server (console) receives the next request for 64 KB of data (one way delay). In a one-second interval, if we need to transport three 64 KB chunks of data to at least meet a 192 KB/sec target, we need to incur this additional RTT overhead three times, resulting in a substantial loss of nearly 20 percent, prior to issuing the next request.

As we analyzed each layer of the stack and implemented the optimization in firmware, we were able to observe the improvement in network performance substantially from 18 KB/s to 124 KB/sec in an Internet environment with 80-msec (RTT) latency. To provide real-world context for the optimization, transferring a copy of a 100MB ISO file would take approximately take 14 minutes.

Future Improvements for RSR
Since the RSR usage is primarily end-to-end image copying, we also determined it would help if we could pre-fetch the next data chunk without waiting for the host OS to issue the request. In order to

optimize the request for an Internet-based connection, we want to pre-fetch at least one second's worth of data, to keep filling the end-to-end pipe. Currently, efforts are underway to target the next-generation platforms with Intel vPro technology to incorporate IDE-R data pre-fetching.

RSR Usages Deployments in SMB

Deploying RSR solutions in small to medium businesses (SMB) presents another set of unique challenges. Typically, small to medium businesses do not have the resources to deploy an elaborate infrastructure to manage their clients. To address these SMB challenges, Intel is working on extending the RSR solutions to SMB.

In 2008, new desktop platforms with Intel vPro technology introduce support for Intel® Remote PC Assist Service. Intel Remote PC Assist Service provides:

1. An Intel service infrastructure, the Intel Remote PC Assist Service, for online centralized activation and connection to platforms with Intel vPro technology.

2. PC firmware, BIOS, and software extensions to support interaction with the Intel Remote PC Assist Service.

3. A Software Developers Kit (SDK) for interacting remotely with PCs that have the technology for, and have opted into, the Intel Remote PC Assist Service.

The overall Intel Remote PC Assist Service infrastructure is fully encapsulated in Internet-based data centers that can dispatch connections made between the service provider and the end-customers in the cloud. This concept is illustrated in Figure 7.

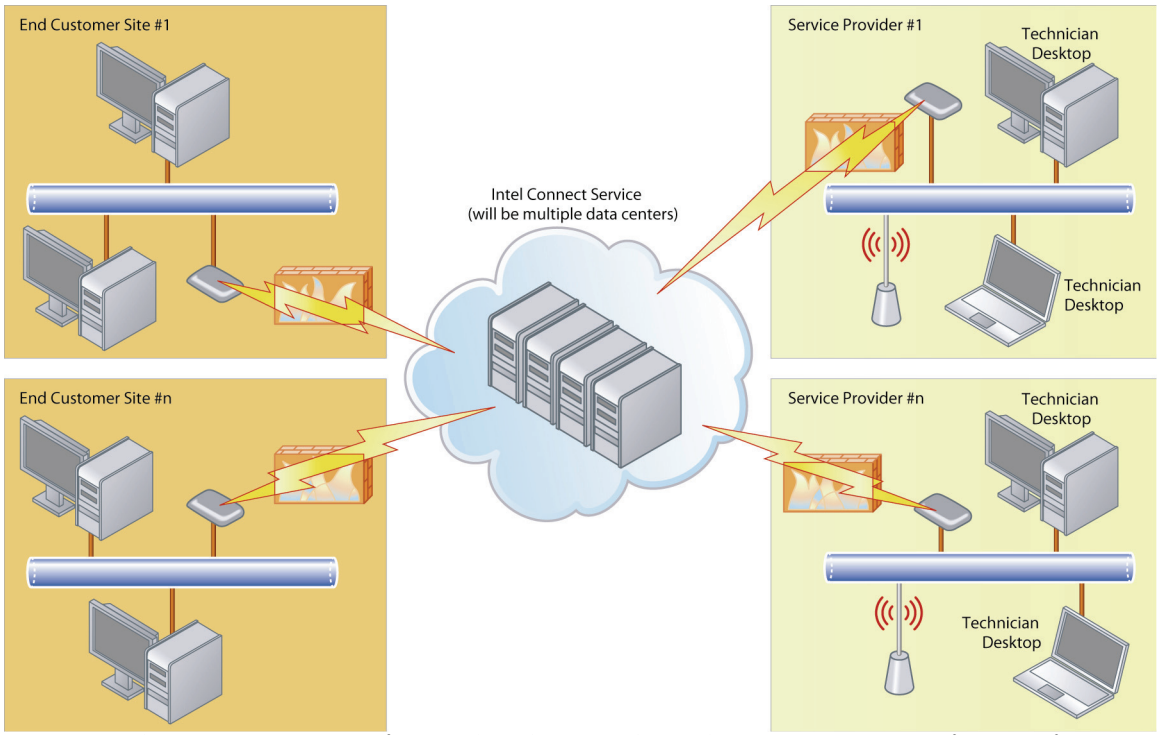

Figure 7: Intel® Remote PC Assist Technology and Intel® Remote PC Assist Service network diagram

The Intel Remote PC Assist Service supports three types of assistance:

1. **One-Time Assistance.** Typically known as "Break/Fix Service", one-time assistance ends when the relationship between the user and the service provider ends, after the support session.

2. **Reactive Enrollment.** This is also typically known as "Break/Fix Service." The user enters into a more permanent relationship with the service provider, but the user still always initiates the request for help.

3. **Proactive Enrollment**. This is typically used by "Managed Services." The user enters into a more permanent relationship with the service provider, and the user gives the service provider permission to access the PC on demand, even when the user is not present.

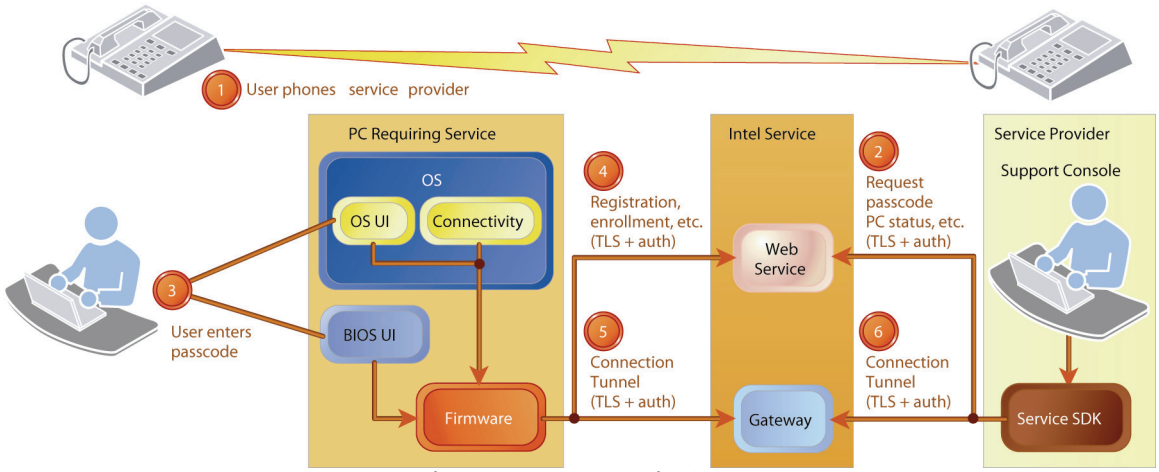

Figure 8: Break-Fix flow

In Figure 8, we show the overall flow as well as the different elements comprising the architecture. When a remote heal usage is initiated due to an inoperable PC, the following actions are taken to get the service activated and to receive remote assistance for the heal scenario to begin:

1. The PC end user contacts a service provider over the phone.

2. The service provider requests a passcode from the Intel Remote PC Assist Service.

3. The service provider verbally guides the user into the UI screens (which include a legal/privacy opt-in step) and the user is given the passcode by the service provider (which was requested from the Intel Remote PC Assist Service). The user is requested to enter the passcode and hit the commit button to complete the connection.

4. The firmware embedded in the user's PC connects to the Intel Remote PC Assist Service over a secure session by using an FQDN and root certificate contained in the firmware from the factory (part of the signed image) and then the PC uses the passcode to perform a one-time registration operation and then associates with the specific service provider.
5. The firmware then locates an available gateway server in the Intel Remote PC Assist Service; it then connects and waits for the service provider to complete the connection to form a remote access session.

6. The service provider's support application connects to the same gateway server and completes the connection to form a remote access session. The user's PC can now be remotely accessed using normal Intel vPro technology protocols as if the support application and PC are on the same LAN.

A subtle point to note is that the user initiates the conversation and therefore prevents unsolicited connections. This is because the security and privacy of the PC end user is paramount, regardless of whether it is a small business or a consumer.

This "break/fix" flow is most relevant to the main topic of this article, that is, remote heal. A number of other variations to the flow also exist, for example, a proactive service flow. In such a sequence, the user could be provided with an e-mail or a paper form of details (that is, a passcode) to enroll in the proactive service, after which point the service provider's automatic tools would immediately be able to monitor the user's PC. The user does not necessarily need to talk to a human technician on the phone.

The following are example activities that can be performed over this "instantly activated" Fast call for Help tunnel:

1. Boot from remote image by using IDE-R for Remote Heal usages.

2. Reboot the system and boot directly to BIOS.

3. Query asset inventory, 3PDS, Agent Presence and Circuit Breaker—all manageability features.

The Intel Remote PC Assist Service is capable of delivering a robust environment with the following characteristics:

- Secure and private. The service does not store or use any end-customer's or service provider's private data.

- Round-the-clock support and monitoring.

- The ability to be load balanced across multiple data centers in multiple regions (1 to start), using Akamai* to load balance on the main FQDN known by the firmware.

- Load balancing for redundancy also within each data center, with many servers for each component (for example, Web service, database, gateways), as well as internal and external networks (multiple Internet service providers, for redundancy).

Future Direction for RSR Usages

Remote Streaming Architecture for RSR Usages

Instead of downloading images and storing them locally, the concept of streaming a minimal OS to memory and subsequently streaming all other applications on a needs basis is gaining ground. The same architecture can also be extended to Spare-tire OS cases that have IT worker streaming applications and data on a demand basis only. This provides some flexibility for IT-worker productivity as workers do not have to download the entire data set. Existing architectures would require all application and data to be downloaded locally before processing them. Emerging client-streaming architectures, when combined with Web services, provide an easy way to enable both subscription-based services as well as pay-per-usage services as described earlier. Platforms with Intel vPro technology are being enabled with features that can easily adapt to such streaming environments.

Summary

The RSR model of Heal usages provides enterprises with the opportunity to create solutions by using Intel vPro technology features such as Fast call for Help and IDE-redirection. The solutions can be deployed to help IT personnel diagnose and remediate client systems by downloading diagnostic software in a secure manner. When none of the diagnostic mechanisms work, the remote administrator can download a spare-tire OS image and store in a platform storage device and allow the IT worker to boot from the image. In this article, we presented a systematic approach and techniques to optimize the download performance over Internet at higher latencies. By optimizing the TCP/IP stack as well as IDE-R and APF protocols significantly, we were able to increase the end-to-end network performance by almost a factor of seven. As a result, with the reduced download times, IT personnel will be able to efficiently manage remote platforms with Intel vPro technology. Finally, we presented a solution to the SMB market with limited infrastructure. Intel Remote PC Assist technology enables service providers, platform capabilities, and the infrastructure to offer heal services remotely to SMBs. IT shops and service providers, as well as SMB owners should take advantage of Intel vPro technology remote heal services to create technological solutions and deploy them to increase their productivity during system downtime.

Acknowledgements

The authors acknowledge the firmware team and the Remote Access Software Delivery team. In particular we thank Tal Shustak whose input has been very valuable in setting the direction for the performance tuning efforts. Sincere thanks also go to John Vicente who helped guide us in the writing of this article. We also thank Ulhas Warrier and David Akerson for their critical reviews of the content as well as their guidance on the structure of this article.

References

[1] "Intel vPro Technology." At http://www.intel.com/technology/manage/iamt/303749.pdf

[2] "Intel® Fast Call for Help." At http://software.intel.com/en-us/articles/fast-call-for-help-overview

[3] "Intel® Remote PC Assist Technology." At http://www.intel.com/technology/product/remotepcassist.htm

Authors' Biographies

Venkat R Gokulrangan is a staff architect in Intel's Mobile Platforms Group and has focused on Client Manageability and Security Solutions with special emphasis on enterprise deployment of

systems with Intel vPro technology. While at Intel Labs, he initiated the work on IPv6 for residential gateways. Venkat has designed and implemented data sub-systems such as 802.1D and voice subsystems such as VoDSL software. He has an M.S. degree in computer science and engineering from the University of Michigan, Ann Arbor. His e-mail is venkat.r.gokulrangan at intel.com.

Ram Chary manages the Mobile vPro Manageability & Security Architecture team in Intel's Mobile Platform Group. With over 20 years at Intel, Ram has led Intel's Energy Efficient Platforms research (focused on power management technologies for mobile & server platforms) in Intel's Corporate Technology Group, developed processor micro-code, real-time kernels, networking products, audio/video conferencing stacks, and managed multiple product development and research efforts at Intel. Ram holds six patents and received the Intel Achievement Award for his work on PC power management. He has also coauthored a book titled *Building the Power Efficient PC*. His e-mail is ram.chary at intel.com.

Dominic Fulginiti has been at Intel for over 12 years. He is currently a staff architect in the Business Client Group and the lead architect for the Intel Remote PC Assist Service and related pieces of the Intel Remote PC Assist Technology. He holds B.S. and M.S. degrees in computer engineering from Boston University. In his 12-year career at Intel he has held positions in various IT-related areas. He was a developer and designer of authentication management systems; he was involved in datacenter rack server management solutions; and he worked on embedded management features for servers and clients.

Ansuya Negi is a software engineer in Intel's Mobile Platform Group. Her current focus is on the impact that manageability and security components in the platform have on performance and power. She has published many papers on wireless networking. She received her M.S. degree from Oregon State University and her Ph.D. degree from Portland State University. Her email is Ansuya.negi at intel.com.

Hema Tahilramani is a senior software engineer in the Mobile Platforms Group at Intel. She is currently working on architectural issues, performance analysis, and optimization related to manageability and security features of notebook computers using Intel Centrino with vPro technology. Hema received her Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute in 2002. Her technical interests include system power and performance analysis, wireless communications, and networking protocols. Her e-mail address is hema.tahilramani at intel.com.

Jeong Yoon is a senior software engineer in the Mobile Products Group. He has been with Intel for 17 years and has worked in various areas including firmware/BIOS, kernel-mode drivers, network protocol, and user-mode applications. His e-mail is jeong.yoon at intel.com.

Mobile Manageability in Low-Power and Operating-System-Absent States

Michael Berger, Business Client Group, Intel Corporation

Keywords: Intel® vPro™ technology, Intel® Management Engine (Intel® ME), manageability, mobile manageability, mobile platforms, Sx states, low-power states, Wake On LAN, WoL, ASF, dynamic IP, DHCP, WLAN, 802.1x, client-initiated, remote connectivity, AC, DC, battery, Embedded Controller, EC.

Abstract

Before the advent of Intel® vPro™ technology, there was no complete solution for managing mobile computers (laptops) remotely while the computer was not powered on or was without a functional operating system (OS). With the deployment of Intel vPro technology, however, a comprehensive enterprise-focused mobile manageability capability in low-power states and OS-absent states became available for the first time. In this article we examine the evolution of remote manageability of computers in low-power states, and we show how Intel vPro technology comes full circle in addressing the shortcomings of pre-existing technologies for remote management of mobile computers.

A Note on Terminology

This article makes frequent use of the term "platform." This term is used when we refer to the insides of a computer—the aggregate of all the computer's physical components, their states, and their behaviors. The term "computer" is used when the emphasis is on the end-user's or Information Technology (IT) department's usage of the box that encases the platform. In certain cases, these terms are interchangeable.

Introduction

The concept of remote manageability has been around for many years now.

Historically, it was only possible to manage computers when they were on, that is, in the state that the Advanced Configuration and Power Interface (ACPI) denotes as the S0 state.

With the advent of Wake On LAN, however, computers in low-power states could be woken up and managed remotely. (Low-power states comprise ACPI S3, S4, and S5, referred to collectively as Sx states).

The next innovation in manageability was Alert Standard Format (ASF). This manageability standard allowed for the management of computers in Sx states. ASF, however, is limited in that it was developed for stationary computers: it does not provide a complete solution for managing mobile computers.

A mobile computer differs from a stationary desktop computer in a few respects. First, a mobile platform cannot have a static IP address in most cases; instead it must rely on a Dynamic Host Configuration Protocol (DHCP), so it needs to actively obtain a DHCP lease. Second, mobile computers rely heavily on Wireless LAN (WLAN/802.11), which means that for the WLAN link to be active, the mobile platform may have to re-associate with the WLAN Access Point (AP) after link loss. Third, WLAN link security (and some LAN link security) relies on 802.1x, which means that the platform must re-establish its authenticity at certain times in order to avoid being shut out of the network. Fourth, even if a mobile computer belongs to a corporate enterprise, in many cases it is taken home by the end user, and the home environment is often situated behind a Network Address Translation (NAT) mechanism. This means that managing that computer requires that the platform initiate the connection. Last but not least, a mobile computer is very likely to be connected at certain times to limited battery (DC) power instead of to a virtually unlimited AC power source: in this case, managing that computer remotely will take its toll on battery power.

In this article, we demonstrate how Intel® vPro™ technology overcomes the shortcomings of ASF in dealing with these challenges of managing laptop computers both in Sx states and in the absence of an operating system (OS) in the S0 state.

We first present an overview of the benefits of being able to efficiently manage computer assets remotely; that is, we present a value proposition for remote management. We then look at specific manageability solutions prior to the deployment of Intel vPro technology. We follow that with a discussion of the differences between desktop and laptop computers, and the shortcomings of management solutions for handling the mobile computing world prior to the advent of Intel vPro technology. Finally, we end with a discussion of the actual solutions employed by Intel vPro technology to handle the world of mobile computing. Readers who are less interested in the historical perspective and problem statement may want to skip to the final section of this article, though an understanding of the historical perspective helps in the overall understanding of Intel vPro technology's solution to manage IT assets remotely in the world of mobile computing.

Manageability's Value Proposition

A manageable computer allows for IT service providers to perform asset management operations remotely. Basic asset management operations include retrieval of information and control of a computer's state or data. Armed with this information, the IT service provider can fix a computer, ensure a computer's optimum health, or track a computer's hardware and software assets—all remotely.

Up until five to ten years ago, corporations owned desktop computers for the most part. Therefore, researchers and developers of client manageability standards and tools concentrated on desktop computers, especially manageability in low-power states. An Original Equipment Manufacturer (OEM) who wished to add a manageability capability to mobile platforms could attempt to do so, but only in a "copy-paste" manner. In other words, an OEM could take manageability-supporting hardware and firmware originally developed for desktop platforms, plug them into a mobile platform, and hope that they would work well. As we will demonstrate in this article, in most cases, such hopes could not be realized.

Managing a computer remotely reduces the Total Cost of Ownership (TCO) for corporate computers. TCO includes the cost of hardware and software components. TCO also includes these less tangible costs:

- Expenses and wages paid to IT engineers for the time they spend walking, driving, or even flying to a malfunctioning computer.
- Wages paid to the support personnel at IT call centers.
- Costs involved in lost productivity of employees while a computer is disabled or malfunctioning and awaiting the intervention of an IT engineer.

The History of Manageability

The ACPI standard[1] defines various power states of computers. These are termed "S-States."

For the purposes of manageability, at first there was only the S0 state. In other words, a computer could only be managed remotely when it was in an S0 state. In the ACPI S3 (sleep), S4 (hibernate), and S5 (off) states—known collectively as Sx states—there was no support for manageability. As such, all that was required on the managed computer was software running on the main CPU, either on top of the computer's OS or as a part of its BIOS, which was able to respond to remote requests.

Over time, it became very clear that what was needed was the capability of managing a computer remotely when it was in a state other than S0, especially since computers spend a significant amount of their time in states other than S0, that is, in Sx power states.

Wake on LAN (WoL)

The first attempt at managing computers in Sx states remotely comprised the following. If the platform is in an Sx state, and IT personnel want to manage it, they need some sort of hook to allow them to pull the platform out of the Sx state and into the S0 state. From that point, the computer can be managed by IT personnel remotely.

That hook was "Wake on LAN" (WoL). WoL was developed as part of Intel's "Wired for Management" initiative, for which specification version 2.0 was published in December 1998 0. WoL required that a LAN (wired network) adapter be powered in Sx states, and that some basic hardware logic be included in the LAN adapter that could assert a hardware signal that would wake up the platform and put it into the S0 state. The WoL standard defined a "magic packet". A WoL-supporting LAN adapter receiving a magic packet in an Sx state

would interpret the packet as a request from the network to wake up the platform to the S0 state. Other packets could also be used for that purpose.

Subsequently, WoL was extended to other types of network adapters. For example, Wireless LAN (WLAN) adapters could support a similar capability, termed "Wake on WLAN" (WoWLAN).

Note that with WoL and its derivatives, the actual manageability logic could still be run on the main CPU (that is, in software).

Alert Standard Format

The next improvement in remote management of computers was Alert Standard Format (ASF), which reduced the impact of manageability operations on the platform as a whole. In other words, instead of the whole computer having to be turned on (as with WoL) in order to perform a small routine operation, an additional auxiliary processor could be added, with the capability of operating even while the rest of the platform remained in an Sx state.

In parallel, a standard for manageability support in S0 and Sx states, known as ASF, was developed. The first version of this specification, 1.03, was released in June 2001, and the next major version, 2.0, was released in April 2003 [3]. While a discussion of the ASF specification is outside the scope of this article, it is important to mention a certain limitation of this standard, which is written into the specification itself (in version 2.0):

"After a change to the system's hardware configuration, the OS-present environment is used to configure the ASF alert-sending device with information that is not known or easily determinable within the OS-absent environment, e.g., management console and local system TCP/IP addresses."

Again, to remind the reader, this limitation was written with desktop computers in mind, which were client manageability's main usage model until a few years ago. In fact, for mobile computers, the problem is much bigger than just a case of a change in hardware configuration. Additional usecases, which are more relevant to mobile computers, would soon come to light.

Manageability Differences Between Desktop and Mobile Computers

Let us look now at the relevant differences between desktop and mobile computers:

- Desktop computers are stationary, while mobile computers are not. Or, translating this to IP-speak: desktop platforms can operate fine with static IP addresses; mobile platforms cannot. Because of roaming, mobile platforms require frequent changes to their IP addresses. In other words, they require dynamic IP addresses that are obtained by means of DHCP leases. It should be noted that the use of DHCP leases has become more and more common even for desktop computers; however, at the time that the ASF specification was written, this factor may not have been taken into account (refer to the limitation of the ASF standard quoted in the Alert Standard Format section of this article).

- Desktop computers mostly use wired LAN, while mobile computers often use WLAN (wireless LAN—as defined by IEEE's 802.11 suite of protocols [4]). With a wired LAN, a physical cable is connected to the LAN network port. Determining the existence or nonexistence of network access is simple: if there is no network access through the cable, then there is no network access—period. In contrast, with a WLAN, determining the existence or nonexistence of network access is more difficult. It requires scanning for a suitable wireless Access Point (AP) within range. Loss of network access through one AP, or movement of the computer out of an AP's range, requires that the scan for networks be repeated.

- The usage of WLAN also has an impact on security. As the security of wired LAN is inherently better than the security of WLAN (due to the physical properties of the transmission), there is a demand for WLAN networks to compensate for the reduced physical security by adding a link-layer security protocol. (In standard communication models, the link layer is the layer immediately above the physical layer.) A common protocol used for this purpose is 802.1x [5]. Thus, mobile computers that typically operate in WLAN networks would also typically have to operate in 802.1x networks. In contrast, for wired LAN networks, 802.1x is not as prevalent as it is in WLAN networks. Since desktop computers typically operate only in wired LAN networks, support for 802.1x is not as strong a requirement for these types of computers.

- Desktop computers are left on the corporate premises; mobile computers are moved around within the corporation and/or they are brought

to locations outside the corporation and to employees' homes. Outside the corporate premises, the computer may be located in a network whose protection mechanisms (firewalls or NAT) are not controlled by the corporate IT department.

- When desktop computers are powered, it is always from a virtually unlimited AC power source. Mobile computers may also be powered from an AC power source; however, they may also be powered from a limited battery (DC) power source.

Manageability's Handling of Mobile Characteristics Before the Advent of Intel® vPro™ Technology

In this section we examine how each of the differences between mobile and desktop computers, mentioned in the previous section, could be handled without Intel vPro technology, according to the spirit of the ASF specification.

Dynamic IP

In a dynamic IP environment, when the DHCP lease expires and the computer is in an S0, OS-present state, the manageability logic could depend on some logic within the host OS to both renew the DHCP lease when needed and to push the provided IP down to the manageability logic.

However, in an S0, OS-absent state (for example, when the OS cannot boot), the DHCP lease could not be renewed in this way or else it would require BIOS support.

In an Sx state, the renewal operation would require that the manageability logic turn on the computer and boot the OS, in order to receive a new IP address from the host OS, and then return the computer to an Sx state.

Re-Associating with WLAN Access Points

When the computer is in an S0, OS-present state and the computer moves between WLAN networks (that is, loses connection to one AP but is in range of another AP), the manageability logic could depend on some logic within the host OS (for example, the host's WLAN Connection Manager) to renew the association when needed, and push the associated AP's profile down to the manageability logic.

However, in an S0, OS-absent state (for example, when the OS cannot boot), the association could not be renewed in this way.

In an Sx state, the re-association operation would require that the manageability logic turn on the computer and boot the OS, in order to receive a new WLAN profile from the host OS, and then return the computer to an Sx state.

802.1x Networks

802.1x networks are aware of the computers that are granted access on the network. Periodically, 802.1x switches may send a challenge to each computer, requiring the platform to provide an authentication response. If a computer fails to answer the challenge, it is excluded from the network.

If a computer receives an 802.1x challenge while it is in an S0, OS-present state, the manageability logic could depend on some logic within the host OS (for example, 802.1x supplicant) to respond to the authentication challenge when needed, or to actively request a challenge in case of link loss and renewal (as a challenge could have been sent while the platform's link was down).

However, in an S0, OS-absent state (for example, when the OS cannot boot), this response mechanism would not work.

In an Sx state, in order to authenticate the computer with the 802.1x network, the authentication initiation operation would require that the manageability logic turn on the computer and boot the OS, in order to authenticate the computer with the 802.1x network, and then return the computer to an Sx state.

Computers Behind Security-Enabled Networks Outside the Control of IT Departments

When a computer is located in some security-enabled network (for example, one with a firewall or NAT) and the computer is in an S0, OS-present state, the manageability logic could depend on some logic within the host OS to initiate a client connection that would traverse the secured network, allowing the IT's management console to respond and access the computer.

However, in an S0, OS-absent state (for example, when the OS cannot boot), a client connection could not be initiated.

In an Sx state, in order to open up a communication channel, the client connection would require that the manageability logic turn on the computer and boot the OS, in order to open up a communications channel, and then return the computer to an Sx state.

Switching Between AC and DC Power Sources

Original Equipment Manufacturers (OEMs) of mobile computers that contain Intel vPro technology generally accept manageability's value in S0 states, even when the mobile computer is operating on DC power (that is, S0/DC state). In contrast, as of 2008, OEMs do not consider manageability to be valuable enough in Sx/DC states to warrant the extra drain on battery life. In addition, OEMs are concerned that IT personnel may explicitly decide to wake the computer remotely, bring it up from an Sx/DC state to an S0/DC state while the computer and its hard drive are tilted at an angle, or while the computer is located within a thermally constrained location (both cases being relevant to when the computer is located inside an end-user's carry bag). Still, manageability does operate on Sx/AC states, just as it does on desktop computers.

Another point to consider is that manageability draws power from more than just the processor running the manageability program. In terms of power consumption, additional components are turned on when manageability is active, such as all manageability-supporting network adapters on the system.

The implication of this is that the manageability logic and all its associated components need to be switched on in Sx/AC states and switched off in Sx/DC states, even on direct Sx/AC to Sx/DC and Sx/DC to Sx/AC transitions.

In the past, the manageability components offered no support for receiving such dynamic powering decisions in these transitions:

- If the components were powered on in an Sx/AC state, they would remain on after they were transitioned to an Sx/DC state (wasting precious battery power).

- Conversely, if the components were powered off in an Sx/DC state, they would remain off after they were transitioned to an Sx/AC state (causing a denial of service for manageability in Sx/AC states).

There existed two possible methods to provide support for such dynamic decisions:

- The OEM's platform logic could receive autonomic decisions about turning off or turning on the power to manageability's logic and components. However, that could be problematic, because the OEM's platform logic does not have direct knowledge of the IT department's policy in powering the manageability components. (It is possible that an IT department may decide not to employ manageability in Sx states at all in order to save power.) In addition, in Sx states, the OEM's platform logic has a limited view of why components are powered in a certain way. For example, if an OEM's platform logic sees a LAN adapter that is powered, it may not know if the LAN adapter is on because of manageability's powering requirements or because of other requirements of the host.

- Alternatively, dedicated hardware logic could wake up the platform and boot the OS on every transition from Sx/AC to Sx/DC and from Sx/DC to Sx/AC, just so the OS operating logic could power up or power down the manageability logic and associated components on the subsequent return to Sx/DC or Sx/AC states, respectively.

Why Relying on the Host OS to Handle Mobile Characteristics is Problematic

While technically feasible, reliance on a transition to an S0 state and waking the host OS to handle the characteristics of the mobile world poses a few problems:

- Using the host OS assumes that a working OS exists in the computer. However, one of the most valuable uses of manageability is that of remediation: an OS is broken and a new OS needs to be installed. If manageability cannot be used in this scenario because of loss of connectivity, manageability's value to a corporation is greatly reduced.

- Frequently waking a computer poses the risk of creating unexpected side-effects. For example, waking a laptop while it is physically docked may sometimes cause the laptop's hardware and BIOS to engage in a logical docking operation, even if the end-user meant to physically undock the laptop, or actually performed a software-undock of the laptop.

- When a laptop is woken, it may not be in its usual position. For example, a laptop may be in a tilted position, and powering it up may cause the hard disk drive's heads to start spinning (an action that may damage the integrity of the data on the hard disk drive).

- Waking the OS from an Sx state when the platform is operating on a DC power source, as

mentioned earlier in this article, will waste battery power (note that waking the OS itself takes time).

In general, all of these scenarios are examples of why a smaller manageability footprint on the system is a desirable direction, and why WoL evolved to ASF in the first place.

How Intel® vPro™ Technology Handles Mobile Characteristics

With the advent of Intel vPro technology, all of the drawbacks of previous manageability solutions that we just discussed were addressed. Intel vPro technology was able to accomplish this by using a mix of hardware, firmware, and software solutions.

Before continuing in our exploration of Intel vPro technology, it is worth mentioning a couple of important facts:

- As of 2006 (first mobile platform–2007), the core logic of Intel vPro technology has been run on firmware on a dedicated auxiliary processor residing in Intel's chipsets. The firmware running on that processor is known as the Intel® Management Engine, or Intel® ME for short. A diagram showing the general architecture of a system running Intel vPro technology is depicted in Figure 1.

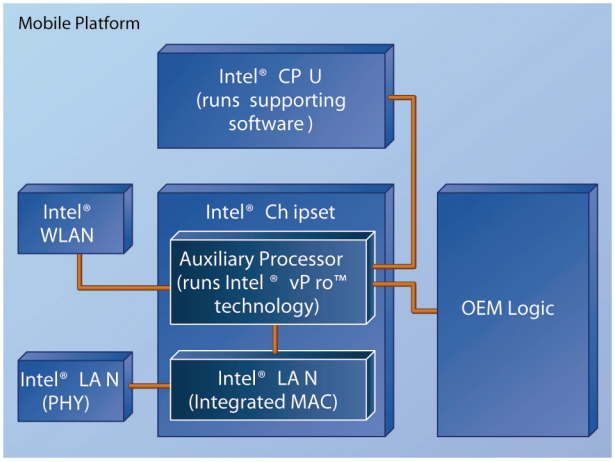

Figure 1: General architecture of a system supporting Intel® vPro™ technology
Source: Intel Corporation, 2008

- To save power, the Intel ME has a mode in which it shuts off completely, even though from a functional perspective it should remain accessible to the network. This is relevant to Sx/AC states only. The main reason ME shuts off is so that the platform complies with power-related regulations, such as Energy Star* [6]. Going forward in this article, we refer therefore to the Intel ME power state in such cases as "M-Off with wakes." Note that "M-Off with wakes" is not relevant to Sx/DC states, in which the Intel ME is required to shut off and remain inaccessible.

The Intel® vPro™ Technology Solution for Dynamic IP

The Intel vPro technology solution for dynamic IP networks (see "Dynamic IP" section) is to use a combination of hardware, firmware, and software.

It should be noted that as of 2008, Intel ME firmware on mobile computers operates only on an IP address that is shared with the host OS. Thus, in an OS-present state, it is the host OS stack that is responsible for acquiring and maintaining a DHCP lease. Even so, Intel ME firmware tracks the acquisition of DHCP leases by the host OS, by using dedicated hardware filters that are located within the network adapter (for Intel LAN), or by relying on software messages from the driver located in the host OS (for Intel WLAN). This tracking system is depicted in Figure 2.

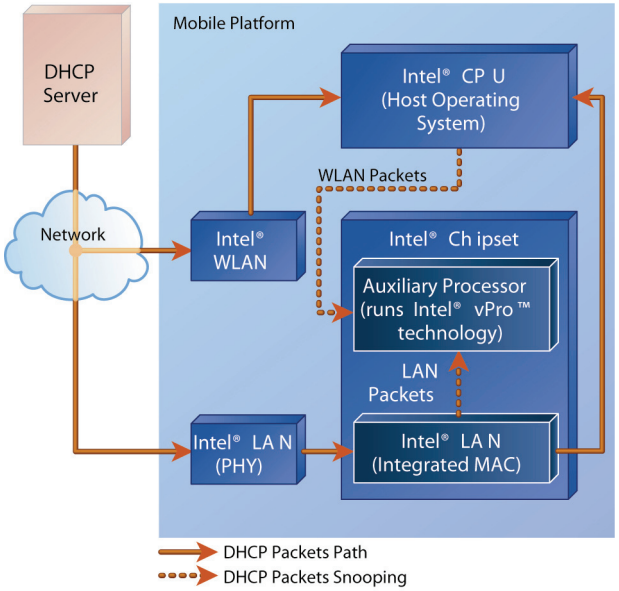

Figure 2: DHCP packets and their tracking by Intel® ME firmware in OS-present state
Source: Intel Corporation, 2008

In an OS-absent state (for example, Sx states), it is Intel ME firmware that directly maintains DHCP leases. This is shown in Figure 3.

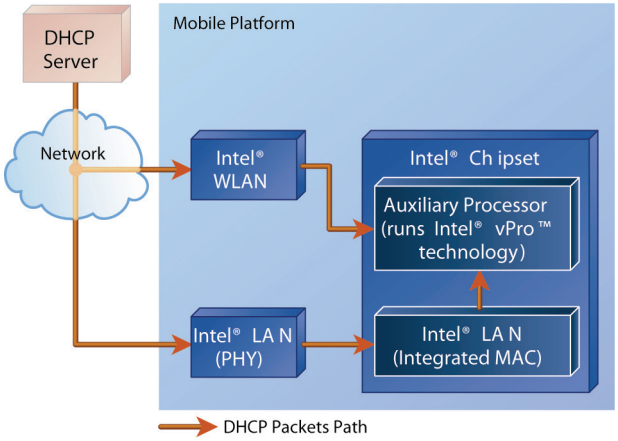

Figure 3: DHCP packets and their tracking by Intel® ME firmware in OS-absent states (including Sx states)
Source: Intel Corporation, 2008

In an OS-absent state, whether the OS software or the Intel ME firmware previously acquired the latest DHCP lease, Intel ME firmware takes note of when the lease will expire.

When the lease expires in an OS-absent state, Intel ME firmware identifies this occurrence by receiving a timer interrupt or even by waking up (in case it is in an "M-Off with wakes" power state). This operation triggers the Intel ME firmware to autonomously renew its lease, without the need to wake the host OS. In such a way, Intel vPro technology can continuously maintain its IP presence on the network, even in Sx states.

The Intel® vPro™ Technology Solution for Re-Associating with WLAN Access Points

The Intel vPro technology solution for re-associating with the WLAN AP (see the section "Re-Associating with WLAN Access Points") is to use a combination of the Intel ME firmware and the firmware that is located within the Intel WLAN device.

After a transition to an OS-absent state (including Sx states), the Intel ME firmware starts communicating directly with the Intel WLAN device by using a dedicated link, and it is able to establish an association with a WLAN AP. Following that, it is the firmware running on the WLAN device that actually maintains the association with the AP. Once the Intel WLAN firmware identifies that the association was lost, it sends a message to the Intel ME firmware, and this generates an interrupt to the Intel ME firmware (or generates a wake, in case the Intel ME firmware is in an "M-Off with wakes" power state). This triggers the Intel ME firmware to autonomously re-associate with a new WLAN AP. Thus, Intel ME firmware can maintain and re-establish associations with WLAN APs in Sx states without having to wake the host OS.

The Intel® vPro™ Technology Solution for 802.1x Networks

The Intel vPro technology solution for 802.1x networks (see the section "802.1x Networks") is to use a combination of hardware, firmware, and software.

Intel vPro technology handles the issue of 802.1x networks similar to how it handles dynamic IP networks; that is, it allows the host OS logic to retain control of 802.1x authentication in OS-present states, and it uses Intel ME firmware to take control of 802.1x authentication in OS-absent states (including Sx states). One important difference, however, from Intel vPro technology's handling of dynamic IP networks is that in the case of 802.1x networks, Intel ME firmware does not require hardware or software assistance in tracking the content of 802.1x-related traffic: it requires software or hardware assistance only in order to track the existence of such traffic (in order to determine whether it should take control of the 802.1x authentication logic or not). In order for Intel vPro technology to be able to control the 802.1x logic in OS-absent states, the IT user needs to pre-configure 802.1x credentials during the Intel vPro technology provisioning process.

Intel vPro technology contains its own 802.1x supplicant logic. That logic initiates 802.1x authentication on switching to an OS-absent state, on receiving an 802.1x challenge, on the expiration of a 802.1x authentication period, and on re-establishment of the network link (to deal with cases where a challenge was issued during link down). In the latter case, the hardware is involved: a link-up event triggers an interrupt or a wake to the Intel ME firmware (in case it is in an "M-Off with wakes" power state).

In all these cases, Intel ME firmware performs 802.1x re-authentication on its own without having to wake the host OS.

The Intel® vPro™ Technology Solution for Computers Behind Security-Enabled Networks Outside the Control of IT Departments

The Intel vPro technology solution for security-enabled networks outside of the IT department's control (see the section "Computers Behind Security-Enabled Networks Outside the Control of IT

Departments") is to use a combination of firmware and software.

The method by which Intel vPro technology deals with mobile computers that leave the corporate premises is shown in Figure 4.

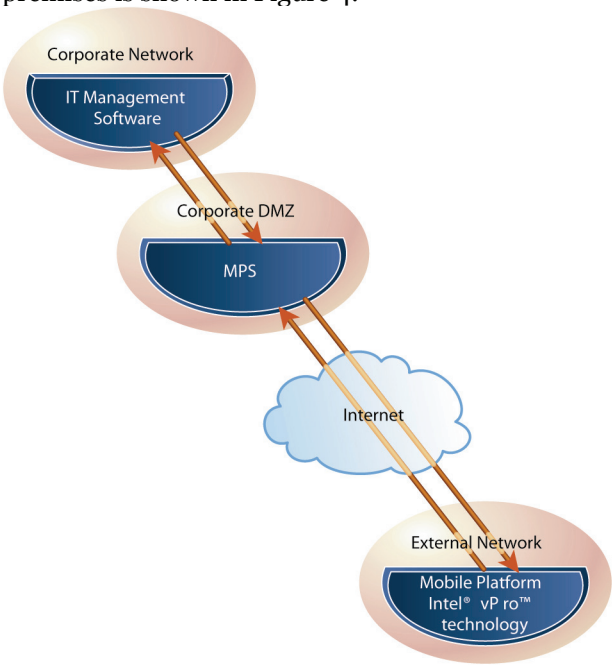

Figure 4: Intel® vPro™ technology architecture for remote connectivity (client-initiated remote access)
Source: Intel Corporation, 2008

Intel ME firmware contains logic that, on various triggers, establishes a connection to the corporate network. For Sx states, the only trigger for establishing a connection is a periodic timer. However, the connection may also be established in an S0 state by a user-initiated request for assistance, and then be maintained on entering Sx states.

The connection is not established directly with the IT user's management console software, but instead is established with a Manageability Presence Server (MPS). The MPS is located in the corporate network's "de-militarized zone" (or DMZ—a sub-network that exposes this and other external services to the Internet). In its turn, the MPS alerts the internal IT management software that a computer, enabled with Intel vPro technology, is now connected and available for any commands that the IT user wants to send to it.

This combination of firmware and software (running the MPS) is what allows computers running Intel vPro technology on remote networks to be available for IT commands even in Sx states, circumventing any need to wake the host's OS and establish a Virtual Private Network (VPN) connection.

The Intel® vPro™ Technology Solution for Switching Between AC and DC Power Sources

The Intel vPro technology solution for switching between an AC power source and a DC power source (see the section "Switching Between AC and DC Power Sources") is to use a combination of Intel hardware, Intel ME firmware, OEM hardware, and OEM firmware (running as part of a dedicated Embedded Controller (EC), developed by OEMs).

As mentioned earlier, the Intel vPro technology core logic runs as part of the Intel ME firmware, which resides within Intel's chipsets. Intel's ME firmware can only read and set pins that are inputs to or outputs from the chipset, respectively. On the other hand, consider these points:

- The logic for reading the power source state is a platform logic, external to the chipset (either within the OEM's EC or directly on the OEM's motherboard).

- The logic for shutting off the power to most of the chipset is a platform logic, external to the chipset (either within the OEM's EC or directly on the OEM's motherboard).

Because of these two points, we have the following situation:

- A hardware input is required from the OEM's EC or motherboard into the chipset to indicate the current power source to Intel ME firmware. Intel ME firmware can use that input in Sx states for determining when to shut itself down or power itself up (the latter requiring hardware assistance), and when to indicate to the OEM's EC/motherboard that power to the chipset is no longer required.

- A hardware output from the chipset to the OEM's EC or motherboard is required, so that Intel ME firmware can indicate to the OEM's EC firmware/motherboard logic when it may shut down the power to the chipset.

The hardware connections and resultant firmware requirements have been defined by Intel to the OEMs in an ME-EC interface specification. A hardware depiction of that interface is summarized in Figure 5.

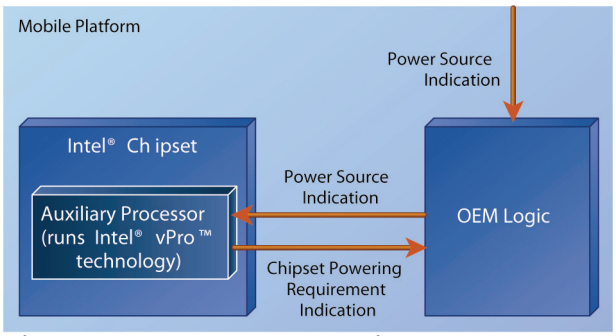

Figure 5: Intel® ME and chipset's hardware interface with the OEM's embedded controller/motherboard logic
Source: Intel Corporation, 2008

The resultant overall platform behavior, due to the interface, is summarized in Table 1.

Table 1: Intel® ME Firmware Behavior in Low-power States
Source: Intel Corporation, 2008

Power State and Source	Intel® ME Firmware Behavior
S3/AC, S4/AC, S5/AC	If Intel ME firmware needs to remain active or go into an "M-Off with wakes" mode, it knows to inform the OEM's EC/motherboard that power to the chipset needs to remain on. Otherwise, it knows to inform the OEM's EC/motherboard that it may power down the chipset.
S3/DC	Intel ME firmware knows to shut itself down.
S4/DC, S5/DC	Intel ME firmware knows to shut itself down and to inform the OEM's EC/motherboard that it may power down the chipset.
Power cycle resets, i.e. S0 to S5 to S0 transitions (AC or DC)	Intel ME firmware knows to inform the OEM's EC/motherboard that power to the chipset needs to remain on so that the power cycle can be completed back to S0.

Conclusion

As this article shows, remote manageability of mobile computers in low-power states (and OS-absent states) requires a few adaptations of the manageability logic from the desktop computer environment to the mobile computer environment. Only by combining hardware, firmware, and software means, and by handling mobile connectivity issues on various levels (from the platform level, through the local networking level, to the remote networking level), did Intel vPro technology succeed in introducing a manageability solution that can operate in Sx and OS-absent states in a mobile world, without requiring redundant wake-ups of the managed system.

References

[1] Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd., Toshiba Corporation. "Advanced Configuration and Power Interface Specification." Revision 3.0a. December 2005. At http://www.acpi.info/

[2] Intel Corporation. "Wired for Management Baseline." Version 2.0. December 1998. At http://www.intel.com/design/archives/wfm/index.htm

[3] "DMTF. Alert Standard Format (ASF) Specification." Version 1.0, June 2001 and Version 2.0, April 2003. At http://www.dmtf.org/standards/asf/

[4] "IEEE 802.11TM Wireless Local Area Networks." At http://www.ieee802.org/11/

[5] "IEEE 802.1x–Port-Based Network Access Control." At http://www.ieee802.org/1/pages/802.1x.html

[6] "Energy Star Program Requirements for Computers." Version 4.0. At http://www.energystar.gov/index.cfm?fuseaction=find_a_product.ShowProductGroup&pgw_code=CO

Author Biography

Michael Berger is an architect for manageability, working in Intel's Israel Design Center in Jerusalem, Israel. Michael joined Intel in August 2001. Prior to his work on Intel Management Engine (Intel ME), his roles included software development for Intel's Linux ASF GUI and agent and firmware development (both ASF and non-ASF) for Intel Ethernet Controllers (LAN), specifically the 82571EB, 82572EI, 82563EB, and 82564EB products.

Michael's first role on Intel ME was as an architect for Intel ME on Intel's 2007 mobile platform, codename Santa Rosa (the first Intel ME project on

mobile platforms). His main focus was on areas of the Intel ME "Kernel": power management, link management, interaction with the OEMs' Embedded Controller (EC) on the mobile platforms, and mobile manageability.
For Intel's 2008 mobile platform, codename Montevina, Michael was co-architect for the Intel ME Wake on WLAN feature.

Currently, Michael's efforts are focused on Intel's next mobile platform, codename Calpella, where he is the product architect for Intel Active Management Technology, version 6.0, as well as the technology architect for some of the mobile-related features.

Michael has a B.Sc. degree in computer science and international relations, an M.Sc. degree in computer science, and an MBA with specialization in marketing and finance–all from the Hebrew University in Jerusalem, Israel. Michael is happily married to Irena!

Power Efficiency and Sustainable Information Technology

Glen Anderson, Channel Platforms Group, Intel Corporation
Philip J. Corriveau, Channel Platforms Group, Intel Corporation
Doug DeVetter, Information Technology, Intel Corporation
Frank Engelman, Information Technology, Intel Corporation
Subhashini Ganapathy, Channel Platforms Group, Intel Corporation
Robert F. Reed, Digital Enterprise Group, Intel Corporation
Alan Ross, Information Technology, Intel Corporation

Keywords: Intel® vPro™ technology, energy, green, sustainable, IT, information technology, enterprise, client PC, consumption, Intel® AMT, power, management

Abstract

Energy efficiency in large data centers is already a concern for many businesses because of the high density of energy consumption. Rising energy costs and global environmental concerns have prompted Information Technology (IT) professionals to also take a closer look at client personal computer (PC) energy consumption. Intel researchers interviewed IT professionals about the monitoring and control of client PC energy consumption. This article describes findings from that study and we discuss how Intel® vPro™ technology can be used to address client PC energy efficiency. Intel vPro technology allows a business to manage energy consumption while ensuring that systems can still be awakened in a reliable, secure manner, allowing energy savings while still providing a high level of operational stability. Both internal Intel pilot studies and an external case study show how Intel vPro technology is already being used to address the client PC energy efficiency challenge. Utilizing Intel power-management technologies will pave the way for the active energy-management capabilities that are required to reduce energy consumption in the office computing domain.

Introduction

IT professionals in many businesses have become more aware of the cost of energy consumption. Energy consumption in data centers has long been a concern [1], but more recently, IT managers have realized that the enormous installed base of client personal computers (PCs) offers at least as much opportunity for energy savings as data centers [2]. Since IT organizations are being held more accountable than they have in the past for energy consumption within enterprises [3, 4], IT professionals need methods and tools to monitor and control energy consumption.

According to a 2002 study by Lawrence Berkeley National Laboratory [5], typical PCs can save .064 kW in a sleep state, versus running in idle mode. A 17-inch LCD display can save .033 kW in a sleep state. Using a United States Department of Energy estimate of $.0931 per kWh [6], a company can save $.009 per hour of sleep time for a PC and display together (.097 kW multiplied by $.0931). If a PC remains in a sleep state for 14 hours per work day (this assumes a 10-hour work day for the PC) and 48 hours per weekend, a savings of $55.22 annually per PC can be realized. Across an installed base of thousands, such a savings can be substantial.

Many factors drive increased interest in managing client PC energy consumption, such as cost savings and environmental concerns. However, there are also factors that work against the management of energy consumption, such as perceived additional system management overhead and lack of reward for saving energy.

Intel has many activities in the energy-efficiency domain, including producing increasingly energy-efficient products, collaborating with governments and industry workgroups on energy-efficient products, and active programs across our manufacturing facilities to reduce the impact on the

environment. Intel® vPro™ technology for business PCs can play a key role in managing business client PC energy consumption as its specific features allow IT professionals to address energy efficiency.

In this article we look at the findings from a study of IT professionals who were asked how they would like to manage client PC energy consumption, what would motivate them to manage it more efficiently, and what tools would they need to do the job. We then show how Intel vPro technology can be used in businesses to help manage client PC energy consumption, and finally we look at case studies on reduced energy consumption as a result of the implementation of Intel vPro technology.

Factors Influencing IT Professionals

The findings discussed here are based on data collected from semi-structured, on-site interviews with 28 IT professionals in the United States and the United Kingdom during August and September of 2007. We questioned participants about their current practices of energy conservation and about their views on using monitoring technologies to lower the energy consumption of client PCs. The businesses were in a variety of industries, including banking, insurance, manufacturing, government, and education. To be selected for the study, participants had to have indicated during screening that IT was strategically important to their businesses. The US sample included 18 US businesses in Denver and San Diego. The UK sample comprised 10 businesses in London. About an equal number of participants were selected from small (30-99 managed PCs), medium (100-499 managed PCs), and large businesses (1000 or more managed PCs).

Cost Savings

The primary driver for energy efficiency in for-profit companies is to save money. In conversations with IT professionals, we found that they often cited savings within 10 to 15 percent as being meaningful for client PCs. We found a general trend in the interviews that indicated that the larger the business, the more likely the IT personnel believed a system for client PC energy efficiency would be worthwhile. When IT professionals were responsible for thousands of client PCs, they said that multiplying the benefit of saving a relatively small amount of dollars or Euros across thousands of PCs would be something they could show to upper management as constituting a meaningful cost savings.

For small and medium business, IT professionals believed that they could make little financial difference by saving power on relatively few devices. Even if a feature of technology could make policy enforcement and monitoring reliable and easy, the cost savings would not add up to enough to make it worthy of consideration.

In many businesses, we found that it can be difficult to account for energy consumption costs, and IT professionals are generally unaware of the cost of electricity needed to run a PC. IT professionals in some small and medium businesses said they rent office space as part of a package that includes utilities. IT professionals in these situations have difficulty using cost savings as an argument for energy-efficient technology, because the business would not save the money directly. Moreover, the savings involved would likely not be enough to justify renegotiating leases with landlords. Even in large enterprises, since electricity consumption is part of operational expenditures, and IT is part of capital expenditures, it may be difficult for IT organizations to get credit for conserving electricity.

Governmental Regulation

IT professionals in London more often cited possible future governmental requirements as a driver toward client PC energy efficiency, though the topic did occasionally come up with US IT professionals also. In the US, IT professionals sometimes expressed a desire to comply with Energy Star, a US government program that sets standards for the energy efficiency of products. There was a general perception across groups in both the US and London, that, over time, government regulations would compel them toward greater client PC power efficiency.

Looking Green from the Outside

We also found that companies with an outward-facing public relations (PR) strategy and companies with users who could observe their client PCs firsthand often believed there was an advantage in presenting a “green” image to the outside world. Knowing that a green image might influence purchase decisions, IT professionals in these companies thought a visible program of energy efficiency, such as a special logo on the PC, could be financially beneficial because of its good PR. This logo could be from a government agency or some other organization that sets a well-known standard for energy efficiency.

Feeling Green from the Inside

Like people in many other occupations, IT professionals often have a personal commitment to save energy based on their own environmental philosophies. In our study, we found that some IT professionals perceive their company’s energy usage as having a direct impact on the environment, so

they feel that by helping to reduce their company's energy consumption, they are helping to reduce greenhouse gases. Technology that reduces client PC energy consumption, therefore, appealed to these individuals, even if there were no tangible, economic benefits for the business.

IT Usage Requirements for Energy-Efficient Client PCs

Our research found that in order to make an energy-efficient client PC enticing to IT professionals, there were certain requirements and conditions that had to be met:

- Return on investment was top of the list. IT professionals have to do more work with fewer resources all the time, as they are pressured to reduce costs. Any additional tasks, including energy-conservation efforts, must show a return on investment. IT professionals want to enforce simple, basic, power-saving rules in relation to client PCs, such as reliable off or hibernation states at night and limited alerts, mostly just monthly reporting.

- Energy savings cannot hinder perceived performance for users. The PCs need to come out of sleep states quickly, and power-saving policies cannot lower perceived performance.

- Users don't usually give much time or attention to IT professionals: they just want them to keep their PCs running well and not interfere too much with their work. IT professionals often have to persuade users to allow them to install software patches, or they need to get users to adopt certain usage behaviors. None of these things are easily done, so making users aware of how a PC may behave in relation to a power-saving feature is just an additional training issue with which they have to contend, and it is something that draws their attention away from other IT concerns.

- Energy solutions must be integrated into other manageability solutions. Power monitoring should be implemented with enterprise manageability solutions such as those provided by original equipment manufacturers, operating system vendors, and independent software vendors.
- Metrics are important. IT professionals need explicit measures of how much energy they can conserve, and if it's not at least 10-15 percent, they're not interested.

- The focus of our study was desktop PCs, not notebook PCs. Client PC energy conservation has limited relevance to notebook PCs, except to the extent that some notebook PCs are being used as desktop PCs. IT professionals did not want to try to control energy consumption of remote users or users whose computers run on battery power.

- Use cases (that is, proof points) are important to IT personnel. They like to look at what-if scenarios vis-à-vis their energy-saving policies. For example, when X number of PCs shut down or go to sleep at night, the company will save Y amount of money.

In the next section we describe the capabilities of Intel vPro technology that help address the factors that influence the energy-consumption policies of IT organizations.

PC Power Management with Intel® vPro™ Technology

Intel vPro technology offers several ways to impact energy efficiency.

Reliable Wake and Power Up

As we showed in our calculations on energy costs, substantial savings can be realized if desktop PCs are shut down or put to sleep during the non-use hours, typically overnight. This is especially applicable to computers that are used by shift workers in government agencies, financial institutions, or insurance industries, where workers do not typically take their computers home. While businesses could just ask employees to turn their systems off before leaving and then back on when they arrive at work the next day, there are three major problems with this approach.

1. Employees may simply forget to turn their systems off.

2. When employees turn their systems back on when they return to work, they will experience lost productivity while they wait for the system to boot up.

3. The start-up time, or even system performance, may be degraded by system updates or inventories that were not performed during the evening (because the system was off).

By using Intel vPro technology, all of these problems can be resolved. System shutdowns can be scheduled for when employees leave, or they can be delayed if the system is still active. The system can be woken up at a fixed time before the employees return to

work. Finally, the system can be woken up during off hours to perform system updates or inventory; it can then be shut down again when these tasks are complete.

In all these cases, Intel vPro technology is used to perform a secure, robust system power-on that is not based on Wake on LAN (WoL). WoL technology is known to be insecure and difficult to manage. Intel vPro technology can also be used to gracefully shut down the systems by using Windows* Management Instrumentation.

How an application is configured for energy efficiency is now emerging as an important area of consideration. As some IT organizations become more application/service centric and less operating system–centric, management applications can utilize Advanced Configuration and Power Interface (ACPI) capabilities to work directly with the hardware and BIOS to most effectively manage platform energy consumption. The native management capabilities of platforms enabled with Intel vPro technology grant administrators the opportunity to seamlessly integrate manageability with security consoles to help control when and how systems are updated.

Integrated Energy Consumption Data Across the Enterprise

It is imperative that enterprise IT organizations implement monitoring systems to provide a comprehensive view of the energy consumed outside the data center. While the industry focuses on reducing energy consumption within the data center, we recognize the opportunity for reducing energy consumption in the entire office computing environment.

Utilizing Out of Band Capabilities to Maximize Energy Efficiency

Traditional PC management is accomplished through agents installed on the operating system. This means that the agent must be running in order for common manageability tasks to occur. Intel vPro technology provides out of band (OOB) capabilities that allow IT system administrators to access systems, independent of the operating system type or the state. This is important because utilizing Intel vPro technology OOB permits the client PCs to gravitate toward the lowest possible power state and still be available to be queried, inventoried, and patched. This integration of enterprise operations management capabilities is unique in that there is now a capacity to manage notebook PCs remotely, independent of their location. We are in an era in which robust OOB management capabilities exist and can be implemented to provide the enterprise a way to balance productivity, manageability, and energy efficiency.

Mobile PCs with Intel® vPro™ Technology

In addition to the obvious solutions described in this article for desktop PCs, there is a less well-understood Intel vPro technology solution for notebook PCs. Notebook PCs are routinely taken home by employees and therefore they cannot be managed on the internal corporate network. Employees who turn their notebook PCs off at night have experienced significant disruption when first connecting to the corporate network in the morning, especially if they are remotely connecting.
However, notebook PCs enabled with Intel® Centrino® 2 processor technology and with Intel vPro technology also have the ability to be managed outside the corporate network. In this scenario, when employees are finished using their notebook PCs at home, they would turn them off to conserve energy, but not turn off the AC power source. The notebook PC can be configured to wake up on a timed basis and securely call back into the corporate network to process any needed updates and then turn itself off. In this manner, employees can avoid time-consuming updates and be assured that their notebook PC is fully compliant with corporate policies before they connect to the corporate network remotely through the Virtual Private Network (VPN).

BIOS Configuration Elements

Intel vPro technology gives administrators the ability to remotely access BIOS power configuration elements on client PCs to ensure adherence to corporate policies via the ACPI standard. Across the board the adoption of the ACPI standard (in conjunction with the operating system and applications) helps to standardize the power state functionality of the hardware device (for example, what happens when you close a notebook PC). There are also C-states for idle central processing units (CPUs) as well as P-states for active CPUs: these states can control frequency and voltage for more efficient operation. The most popular utilization of the ACPI specifications is the S-state, that is, the sleep state (see Table 1).

Operating System Considerations

Operating system configuration is important for energy efficiency. There is robust support for ACPI power-management functionality in many client operating systems. Intel vPro technology capabilities can be utilized in conjunction with configuration management software to set the appropriate configuration, based on the type and version of operating system software installed on the client PC.

It is important to conduct an appropriate amount of testing before integrating BIOS and operating system, power-management configuration options.

Table 1: Sleep states and their energy usage

S-state	Description	Comment
S1	All system context retained	Lowest energy savings
S2	CPU and cache context lost	Not often used
S3	Memory context saved ("sleep")	Typical balanced policy
S4	Platform context only saved ("hibernate")	Longer resume cycle
S5	No context saved ("soft off")	Best energy savings

Future Enterprise Integration and Client Power Management in IT Organizations

Enabling a comprehensive environment for client PC energy consumption takes a combination of forces: understanding the environment, monitoring energy consumption, and managing towards a common goal. As the capabilities of Intel vPro technology evolve, IT professionals will have a complete set of tools to reach that goal.

ISV Console Within the Enterprise

In order to achieve the results we have described, any additional features of Intel vPro technology that provide client power monitoring and control need to be fully integrated into the ISV consoles. Utilizing that information enables IT personnel to get the maximum value out of their resources and to minimally impact the company's users. Monitoring the performance and activity profile of client PCs enables optimal energy efficiency in the enterprise. We advocate the flexibility to run monitoring and control software as an agent, or as firmware native to the platform, to enable flexible deployment, management, and policy definition. Moving forward we see two major areas of interest to the industry for power management:

1. Monitor alternating current (AC) power used by the client to accurately measure energy consumption and to provide an auditable energy-savings trail.
2. Control the client power states to save money, be environmentally conscious, and reduce the carbon footprint of the industry as a whole.

Estimating Power Consumption: Methodology and Practice

Existing estimation technology is limited to software that estimates power, based on time and system states, and then looks up the power profile of the system logged. There are inherent inaccuracies in this methodology due to the inability of the software to be able to accurately model the users' real behavior: the software measures keyboard or mouse inputs, and the system state to estimate the time the system spent in various power states. Once these measurements are taken, the time and state information is translated into power by performing a calculation on a database of "known" or characterized devices. A worst-case scenario is when IT professionals "characterize" systems and manually input and maintain the data. This method is not only expensive, but it is time-consuming and highly susceptible to obsolescence.

Key drivers for improvements in power management are the accuracy of information collected and the additional hardware costs associated with the collection of that information. Intel is investigating adding centralized sensors to the client PC that are standards-based and have minimal impact on cost.

Desktop PCs, enabled with Intel vPro technology, that utilize the embedded Intel Manageability Engine (Intel ME) can potentially run firmware that can provide a standard Application Programming Interface (API) to provide information, through standard network management protocols, to agents that run in the Capability Operating System (COS) [7]. This firmware would collect data from sensors on the desktop PC and would accurately monitor power usage, taking into account basic faults, such as a fan or temperature sensor failure: the findings would be relayed back to the console agent (see Figure 1).

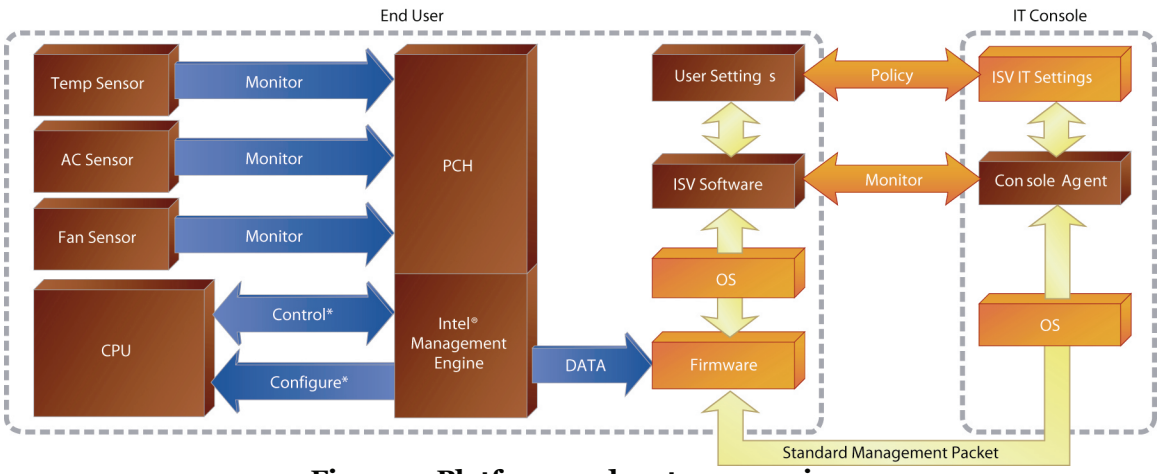

Figure 1: Platform and system overview

The firmware would be flexible enough to enable the end user, via the console, to determine the time between samples and also the time intervals in which data are sent back to the console.

By using software that can access the CPU and the Platform Controller Hub (PCH) registers, the firmware would be able to provide those data to the application software or the console (depending upon which implementation is used, that is, native or agent). By using Intel vPro technology with the embedded security provided by the Intel ME, a platform enabled with this technology, can provide a "tamper resistant" audit path. Also, by using these technology OOB capabilities, any information stored in the Intel ME can be retrieved after hours: the console can wake the client PC, download the information back to the console and then shut down the client PC to its original sleep state.

For client PCs, utilizing existing sensors and standardizing around those make economical sense and expedite industry adoption.

By using Intel vPro technology through the Intel ME that resides in the PCH, it will be possible to monitor and control the platform by using standards-based management console software.

Future Power Management Configuration Settings and Policies

Having accurate and auditable information enables IT managers to gather statistics and characterize the energy used in the enterprise.

Policies can be applied to control the time of day to shut down or power on computers. This is a major power-savings action for a corporation that, as discussed previously, can be accomplished successfully with Intel vPro technology. Other methods can also allow control parameters such as when to power down the hard drive, turn off the display, or suspend the platform. However, such settings do not allow for fine-grain power management of platform components that have to be kept on when the platform is in the ACPI S0 state. It has been determined that even though client PCs in an S0 state are in an idle state for 80 to 90 percent of the time, several platform components are kept in high-power states to meet the service latency requirements of devices and applications in the platform [8].

If the devices and applications could dynamically convey their service latency requirements to the platform based on workload (low latency tolerance when active and high latency tolerance when idle), that could dramatically reduce the energy consumption of the platforms when in an idle state. Such information would allow for future policies that would integrate the local requirements with the remote policy settings to enable substantial power savings without sacrificing performance and reliability.

Enforcing software policy controls locally in each client PC, will lead to significant gains in energy reduction. However, Intel also recognizes systemic issues in existing platforms that cannot be addressed by software: the software cannot service the request to save power fast enough, and moreover, it is not cognizant enough of the entire system state to ensure stable operation. When an operating system or agent issues low-power state commands, they are not instantaneous. Commands take time to traverse the driver stack, and real-time hybrid operation is not possible, due to added system latencies.

A platform is comprised of core logic components, a CPU and PCH, and the devices that connect to these, such as the Universal Serial Bus (USB) devices. These USB devices can be a keyboard or mouse for basic Input/Output (I/O). Other examples of devices are wireless or wired Network Interface Cards (NICs), and discrete graphics cards. Each one of these devices is typically routed through the PCH or CPU (see Figure 2).

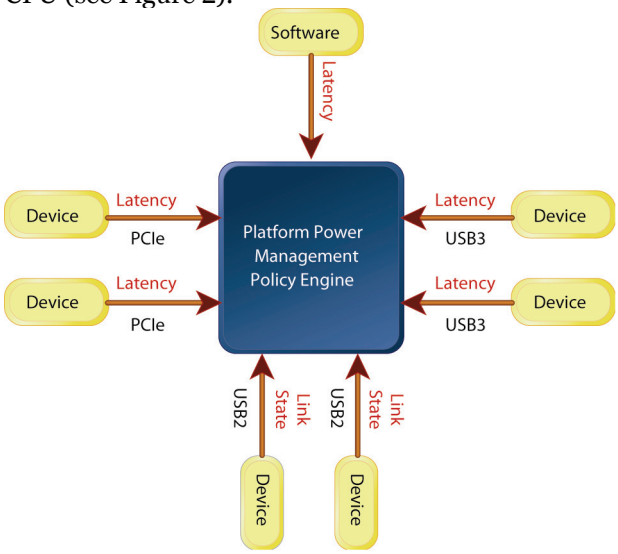

Figure 2: Device latency and power-management concept

Having power control and subsystem logic embedded in the hardware of the platform means the devices can generate power-management messages when they transition from high-power states to low-power states. The platform will act on these messages without having to generate an interrupt that would unnecessarily cause the CPU to use high power to process these messages. In this new method of platform power control, the operating system can provide the guidance and constraints that allow the hardware to make policy decisions to a granularity and latency unattainable by software.

The Role of Energy Efficiency Standards

It is important to continue to drive for adoption of ACPI and other standards and services that expose energy-efficiency capabilities across the client PC. It is also important to provide common methods to connect energy efficiency to other enterprise tasks, such as security and manageability. We have seen a promising evolution of the ACPI standards over the past few years along with the uptake of these standards across multiple OEMs and operating system vendors. Most IT organizations do not consider the notion of integrating power management into their enterprise applications, but the infrastructure to do this has been around for almost a decade with robust capabilities available that can be utilized where possible.

To continue to grow in today's world of diminishing energy resources and higher costs, all computing sectors of the PC industry should work together to reach a goal in which desktop and notebook PCs in an enterprise can use a combination of hardware, firmware, BIOS, and operating system and application configuration elements and policies to maximize the energy efficiency of enterprise client PCs, independent of their location. This concept can then extend beyond the PC and extend to office equipment and additional form-factors over time. For now, however, this is a call for action in the area of enterprise business systems.

Case Studies

In the following section we present two real-world examples of how client PCs enabled with Intel vPro technology have successfully been implemented in the enterprise and how this technology has helped IT organizations manage and save power.

Case Study 1: Intel IT Training Room Environment

The IT training room environment within Intel consists of approximately 1000 desktop PCs in 41 rooms at 30 sites. Prior to the deployment of PCs with Intel vPro technology, the PCs were powered up 24 hours a day to enable after-hours remote maintenance. Remote maintenance consists of asset management, hardware diagnosis, software diagnosis, and patching.

With the deployment of desktop PCs enabled with Intel vPro technology, systems are now powered off when not in use, but they can be woken up remotely for maintenance and can be ready when students arrive for a class. Intel estimates that by shutting down the systems when not in use, their projected savings will be 35 to 45 percent of IT training room PC energy consumption.

In 2006, Intel started a pilot program: it deployed 24 desktop PCs enabled with Intel vPro technology to one room. Based on the success of the pilot, Intel deployed 300 more of these PCs in 7 sites during 2007, replacing older equipment. The deployment of desktop PCs enabled with Intel vPro technology continues in 2009, with plans to deploy about 600 more in the training room environment.

Partially based on the success of our deployment of computers enabled with Intel vPro technology in IT

training rooms, we have begun widespread Intel ME activation of desktop and notebook PCs enabled with Intel vPro technology throughout the enterprise.

Case Study 2: National Law School of India
The focus of this study was to observe the reactions, behaviors, and attitudes toward the Intel vPro technology solutions for power management. The research included semi-structured interviews across two cities in India. As one of the emerging markets with a strong economic growth and increasing adoption of IT solutions, India offers a great opportunity to understand the current experience of small and medium businesses in a managed, service-provider space. The Law School building was managed by two IT administrators maintaining 200 PCs. The Law School wanted to move the PC lab to the library, and the library was not in the same building. The reason for the move was that the library was open until midnight whereas the Law School building was closed earlier in the evening. By moving the lab, IT administrators thought they would reduce the amount of electricity they would have to use in the Law Building by having to keep it open until midnight so that students could use the PC lab for longer hours.

The downside to the plan was that the IT administrators would then have to physically walk to the library to manage the PCs. So after trying different solutions such as streaming, the IT administrators chose Intel vPro technology so they could remotely control the PCs by using the Intel vPro technology management tool LANDesk*. This allowed them to shut systems down remotely and to wake them up in time for the students in the morning.

Summary

As energy costs and environmental concerns rise, IT professionals in many organizations want to increase energy efficiency, and they recognize that client PCs provide an opportunity to improve the cost structure of their departments. IT professionals are motivated by personal and professional reasons to conserve energy, but energy-saving solutions have to fit with other goals of the business, such as improving IT and end-user productivity. Computers enabled with Intel vPro technology provide an infrastructure that allows IT professionals to take practical steps to ensure energy efficiency, such as allowing greater control over power states during remote manageability. This technology also provides an infrastructure to allow the rest of the ecosystem to improve the monitoring of energy consumption. Intel is taking the initiative to provide its own testing of this technology in real-world applications, while understanding the challenges and successes of customers. Intel will continue to work with the ecosystem and customers to explore new ways to both monitor and improve the energy efficiency of client PCs.

References

[1] Koomey, Jonathan. "Worldwide electricity used in data centers." Environmental Research Letters. Vol. 3, no. 034008. September 23, 2008. At http://www.iop.org/EJ/abstract/1748-9326/3/3/034008/

[2] Creative Commons. “SMART 2020: Enabling the low carbon economy in the information age.” The Climate Group on behalf of the Global eSustainability initiative. 2008. At http://www.theclimategroup.org/assets/resources/publications/Smart2020Report_lo_res.pdf

[3] Ted Samson. “Green IT: Leadership at the top can help avoid turning your energy-saving initiative into a political power struggle.” InfoWorld, June 6, 2008.

[4] http://www.cio.com.au/index.php/id;1955002171;pp;2

[5] Lawrence Berkeley National Laboratory, University of California. "Energy Use and Power Levels in New Monitors and Personal Computers." Paper LBNL-48581. July 23, 2002. At http://repositories.cdlib.org/lbnl/LBNL-48581

[6] Energy Information Administration. “U.S. Department of Energy; Electric Power Monthly. October 2006, Table 5.3, 2006 Year-to-Date Average Cost per kWh." At http://www.eia.doe.gov/cneaf/electricity/epm/epm.pdf. “Average Residential Monthly Use.” EIA Electricity Quick Stats, Yearly average use calculated from monthly average use (908 kWh). At http://www.eia.doe.gov/cneaf/electricity/esr/table5.xls

[7] “DMTF Document Number: DSP0226.” Version: 1.0.0, February 12, 2008. At http://www.dmtf.org/standards/published_documents/DSP0226_1.0.0.pdf

[8] "Energy Efficient Platforms." Intel Developer Fall, San Francisco, Session ID: EBLS002. 2008.

Authors' Biographies

Glen Anderson has worked in human factors engineering for 17 years and is the manager of the User Research Group. His interests include applied-research methodologies, mobile device design, online help systems, and patenting of user-oriented technology. Glen has a Ph.D. degree in human factors psychology from the University of South Dakota (1993).

Philip J. Corriveau has 20 years of video and research expertise and has worked at the Communications Research Centre, a Canadian Government Research and Development facility, and more recently at Intel Corporation investigating all aspects of user experience and video quality standards for Intel platforms. Phil holds a bachelor's of science honors degree from Carleton University in Ottawa, Canada (1990).

Doug DeVetter has 20 years of IT experience in software development, data architecture, technical program management, and IT research and development. Doug's recent work has focused on PC energy efficiency topics and solid state drives. Doug holds an M.B.A. from California State University, Hayward (1994).

Frank Engelman has 18 years of IT experience in IT research and development. Frank's recent work has focused on implementation of Intel® vPro™ technology both within Intel and for external companies. Frank holds an M.S.E.E. degree from California State University, Sacramento (1974).

Subhashini Ganapathy joined Intel in 2006 and works in the IT Security Application Services group. She currently conducts human-factors-related research for businesses and consumers in the User Experience Research Group. Her research interests include modeling human interactions on complex systems, decision making, information protection, and model-based information technology systems. Subhashini received her Ph.D. degree in human factors engineering from Wright State University, Dayton OH (2006).

Robert F. Reed has 15 years of networking and communication-standards experience in areas such as system design, manufacturing, strategic marketing, and architecture. Rob's recent work at Intel has focused on systems architecture and platform power innovation on Intel's future platforms with Intel vPro technology. Rob holds a bachelor's of engineering honors degree in electronics and communications engineering from Huddersfield University, England (1994).

Alan Ross is a principal engineer and lead enterprise architect for Intel's Information Technology division. Alan has led architecture development across several domains at Intel including data center, client, energy efficiency, and security, and he has driven industry development of technologies, standards, and products. He is currently focused on disruptive computing models, innovative form factors, and energy-efficient architectural initiatives. Alan has 18 patents pending related to the security and manageability of systems and networks. Alan has a B.Sc. degree from Myers College (1998) and is CISSP-ISSAP certified by ISC2.

Enabling Dynamic Virtual Client Computing with Intel® vPro™ Technology

Thomas James, Digital Enterprise Group Intel Corporation
Jason Kennedy, Software & Services Group Intel Corporation
Steve Kremesec, Digital Enterprise Group Intel Corporation
John Vicente, Digital Enterprise Group Intel Corporation

Keywords: Intel® vPro™, Dynamic Virtual Client, DVC computing, virtual client computing, VMM, virtualization, Intel VT, security, Intel TXT, Intel AMT, manageability, Intel ME, DASH, WS-MAN.

Abstract

In this paper we discuss an emerging direction for Intel® vPro™ technology called Dynamic Virtual Client (DVC) computing. DVC addresses the needs of IT organizations and end-users by enabling client computing diversity across alternative computing models. DVC is a client virtualization architecture combining hardware- and software-enabled features based on the foundation elements of Intel® Virtualization Technology (Intel® VT) and on Intel® Active Management Technology (Intel® AMT). This article examines emerging computing usages and identifies key gaps and challenges associated with realizing client virtualization. We expand on the Intel vPro technology that enables DVC and examine the underpinnings associated with Intel's core manageability, security and virtualization technologies. Finally, we present a vision for client virtualization and discuss future DVC solution integration opportunities.

Introduction

Today, an emerging trend is toward an abstraction of the traditional distributed computing model in which computing hardware, operating systems, data, and applications may be redistributed to enable alternative and more efficient forms of business computing. This trend is driven primarily by the need to alleviate IT security concerns and management complexity through increased centralization of data and operational control. This direction poses challenges as well as new solution opportunities for end-user flexibility, a richer user experience, and mobile usages.

The standard client computing models, monolithic thick client stacks (hardware, OS, applications, and data intertwined), and server-based computing (user interface remote presentation to thin clients) have given way over recent years to numerous variations appropriate to a wide range of end-user scenarios and usage patterns. These variations are a reflection of the desire to balance users' needs to be more mobile while maintaining their rich desktop experiences with the lower costs of centralized management and operations. Virtualization technologies [1] today are used to decouple elements of the desktop model and enable simpler management and delivery of an end-user environment, thus:

- reducing the cost of per-environment configuration;
- providing computing environments appropriate to security requirements;
- supporting mobile and remote workforces; and
- lowering helpdesk and ongoing support costs (including disaster recovery, and technology migrations).

As we look forward, there is not only a multiplicity of computing options, but there is also a high likelihood that enterprise customers will use many of these options jointly. Applications will be accessed locally, remotely, and in the cloud. Data will be scattered among secure, on-premises storage, personal devices, and other distributed storage systems. End-users will expect to access all of their information, independent of their computing device or location, as shown in Figure 1.

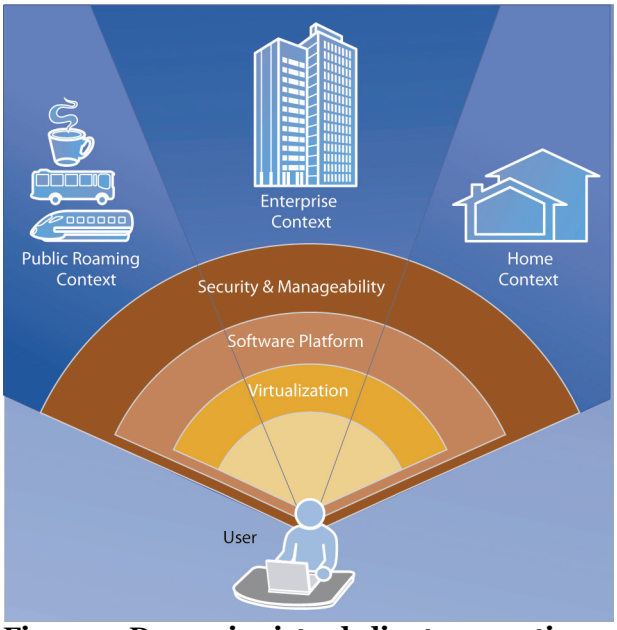

Figure 1: Dynamic virtual client computing: connecting users with their computing environments

Client virtualization technology will play a key role in this future computing environment by helping to bridge the gaps in the current client systems and supporting IT solutions, while facilitating end-user connectivity across these diverse computing environments.

Emerging Corporate Computing Models

When we look at the landscape of client computing, numerous variants of computing usage models and scenarios are emerging within the enterprise and are also blended with consumer usage patterns.

- **Shared computing.** Corporate or business users share PCs, or PC devices are pooled to support a functional group or line-of-business. This scenario is common in task-oriented environments (for example, call centers), traveler stations, development labs, or shared office environments.
- **Flexible and secure roaming.** Corporate mobile workers need access to business applications and data, whether they are operating inside or outside corporate-managed buildings and offices. In such scenarios, end users want the flexibility to compute pervasively. They will need to synchronize business data across alternative client devices and the corporate-managed data centers. At the same time, IT departments must ensure that data is protected when devices are compromised and that users continue to receive service-level support while roaming in public and un-trusted spaces.
- **Customized computing platforms.** Many larger corporations typically support a diverse base of business units, for example, manufacturing, independent design groups, operations, finance, sales and marketing, and so on. Different business units require access to different levels of data or to a core suite of applications. Customizing standard client-platform builds and enabling a centralized image-delivery strategy reduces the total cost of ownership (TCO) and improves overall IT customer satisfaction.
- **Concurrent technology migration and application compatibility.** A challenging activity that a large corporation contends with is technology migration. The implications of introducing or transitioning computing systems to a new OS or driving a major business application redesign can put a significant burden on the supporting IT organization to ensure business continuity and limit TCO exposure. Emerging virtual computing models will facilitate simultaneous technology migration and legacy transition, thus ensuring application compatibility while enabling emerging uses for the end-user.
- **Personal and corporate computing convergence.** While still in its early stages, the trend towards consumerization may introduce alternatives to PC acquisition strategies [2] or enable greater personalization of the end-user's computing environment. Thus, the traditional physical computing environment extends to multiple virtual computing environments, allowing the coexistence and extensibility of both personal computing and corporate computing within the same physical machine.

Challenges

These new usage models present many challenges to providing general solutions to common IT problems such as these:

- Client provisioning and deployment
- Client manageability
- Application compatibility
- Environment and data security
- Mobile and distributed workforces
- Disaster recovery
- Energy management
- License management
- Regulatory compliance

Table 1: Core IT capabilities and flexible user scenarios

Scenarios	Capabilities	User Types	Benefits
HW/SW Isolation	Managing against client virtualization systems Mobile and device aware manageability and security	All users	HW/SW decoupled Reduces IT complexity
Application Isolation	Reduce dependencies on underlying platforms Simplify testing and validation	All users	Full application/OS segmentation Application Availability Reduces IT complexity
Client Manageability	Virtual and physical platform manageability Mobile client virtualization Over-the-air (OTA) provisioning	All users	All users Simplified management. of multi-context environments Reduces IT complexity
Business Continuity	Simple replication of standby environment Automated local/remote session failover for OS image	Financial Services Call Centers	Reduce loss of productivity Simplify continuous solutions
Migration	Migrate while maintaining application compatibility and device support	All users	Reduced time and cost for changing environment Preserve existing application investments
Data Security	Adaptive policy-based solutions for data security, data delivery, or centralization	Government Financial services Onsite contractors	Secure central data ("Top secret zone") User flexibility
Onsite Roaming	Flexible isolation capabilities AAA services Device awareness	Medical Multi-user scenarios Manufacturing	Follow me application, data, settings, images Personal image on primary system
Offsite Roaming	Data security Context-awareness Application continuity and synchronization	Mobile users Sales Consulting	Centralized mobile management. Secure mobile image User flexibility
Dynamic Image	Scaling and managing the creation and updating of stable disk image to broad user populations Separate the management and delivery of disk	Workstations Developers Call centers	Multi-image support

There are numerous capabilities and services that IT organizations are planning to deliver in an efficient manner to a diverse set of users with varying business objectives and personal preferences. For example, how can IT ensure data security and business continuity while supporting a mobile and dispersed workforce? How can IT organizations reduce conflicts and regression testing between key business applications in an environment that provides some flexibility? Finally, how can IT organizations drive faster and more efficient transitions to a new and more diverse set of clients while minimizing costs and downtime? These challenges create a diversity of solution requirements and conflicting priorities. In what follows, we will investigate current client virtualization technologies and explore DVC as an emerging virtual client computing framework to address these diverse challenges.

Client Virtualization

Basic Taxonomy

We now provide a view of the taxonomy of the basic virtualization approaches and their relation to one another, shown in Figure 2.

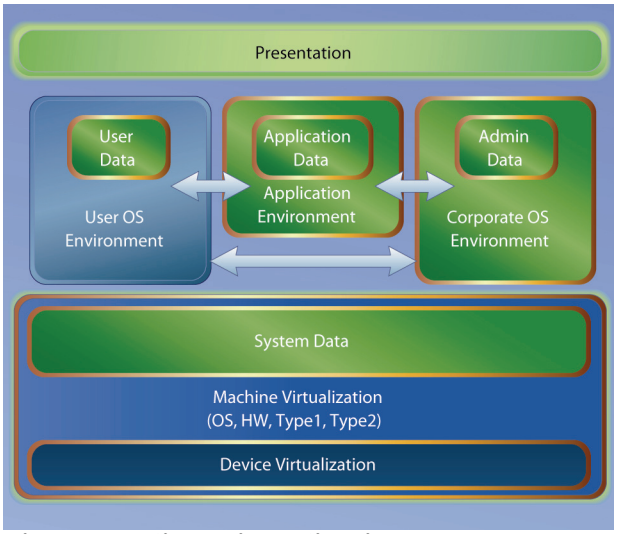

Figure 2: Client virtualization taxonomy

- **Presentation Virtualization** is a client-server architecture for executing applications within a user session that is hosted on a remote server by using a remote presentation protocol to display the session at the client.
- **Application Virtualization** is a client-side technology for executing applications within a protective sandbox designed to isolate and control the interactions of an application with other applications and the underlying OS.
- **Device Virtualization** is a class of technologies or techniques to service operational interactions (for example, discover, control, transfer, interrupt) and enable resource isolation, sharing, and functional extensibility of client devices.
- **Machine Virtualization** is based on a particular type or model of virtualization architecture and associated methodologies to support the abstraction of platform hardware to enable alternative computing models and resource partitioning, and to enable sharing across diverse hosted image types on a physical platform.
- **Desktop Virtualization** is a class of technologies where the entire desktop environment is hosted within a machine virtualization environment, typically accessed via remote desktop protocols.

Overview of Virtualization with Intel® vPro™ Technology

A platform, enabled with Intel vPro technology, is an advanced client platform that allows IT personnel to take advantage of hardware-assisted security and manageability capabilities that enhance a corporation's ability to manage and protect fixed and mobile PCs. With functionality built-in to hardware, Intel vPro technology enables out-of-band (OOB) manageability and down-the-wire security, even when the PC is powered off, unresponsive, or when the host agents are disabled. Along with leading ISV console solutions, Intel vPro technology improves manageability to reduce operational or administrative costs. In addition, hardware-enabled virtualization features in platforms with Intel vPro technology enable more robust, secure, and optimized virtualization usage models.

Platforms running Intel vPro technology support Intel® Virtualization Technology (Intel® VT) for IA-32 Intel® Architecture (Intel® VT-x) [3], Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) [4], Intel® Trusted Execution Technology (Intel® TXT) [5], and Intel® Active Management Technology (Intel® AMT) [6]; all of which can be combined to uniquely deliver next-generation, value-added client usage models. With the use of Intel VT-x and Intel VT-d, hardware acceleration and memory space protection can be achieved. With the use of Intel TXT, a solution can be verified prior to execution to prevent unwanted changes. Many usage models benefit from hardware-accelerated virtualization, hardware protection of memory, measured and verified code before it is executed, and OOB access to the physical platform or

certain virtual machines (VMs). These technologies can be combined to support both industry standard Type 1 (hypervisor-based) virtual machine monitors (VMMs) and Type 2 (OS-hosted) VMMs [7], depending on the supported usage model.

Type 1 VMMs

For increased security and isolation, Type 1 VMMs ensure a sequestered execution environment for each VM. Isolation extends from the hypervisor through the VMs with which the end user interacts. The Intel TXT can ensure a trusted boot process to further reinforce Type 1 VMM, through a hardware-based chain of trust, to secure, launch, and validate the hypervisor and associated VMM components. Jointly, Type 1 VMMs and Intel TXT enable a secure and robust environment from boot to execution.

For Type 1 VMMs, physical machine hardware can be either directly mapped to a VM (that is, pass-through mode) or virtualized. In pass-through mode, the physical device is directly mapped to a single VM. The physical device's driver is loaded in the VM and will physically control the device in this mode; the VM is most typically a Windows XP* or Windows Vista* OS with native device driver support. The benefit of this mode is its uses of the underlying hardware device as it was intended to be used by the developers of the device and device driver. Therefore, this mode offers the best compatibility and performance. Theoretically, the device will function as if there is no virtualization layer, thereby optimizing performance and power management.

However, the main drawback with the pass-through mode is that the device is owned by a single VM and no other VM can access it. For some devices, this may not be a problem and may even be desired (as in the case of a wireless device) but for others, this mode of operation will be problematic. Graphics adapters are a good example of this drawback: if one of the VMs owns the graphics adapter, display by other VMs is problematic or is dependent on the VM that owns the graphics adapter. To put it succinctly, if one VM "owns" the graphics adapter, others cannot display.

If the other VMs have to display, they need to do it through an interface exposed by the owning VM. This could be in the form of a front-end/back-end driver setup (that is, a client/server setup of sorts) or a different technology altogether that has display-back capabilities similar to those of X-Windows*. The former has the added drawback of complexity and the latter has a negative performance impact on video and 3D playback and it may even impact 2D operations. Additionally, this extended device model makes the subsequent (non-owner) VMs dependent on the primary VM, and therefore may introduce other operational issues, if not designed and integrated properly.

Intel VT-d enhances the software-based isolation capabilities of direct-mapped devices to VMs by providing hardware-based memory partitioning, that is, protection domains. Intel VT-d also provides Direct Memory Access (DMA), or interrupt remapping and cache optimizations (for example, remapping structure or translation). These enhancements allow (predominantly) Type 1 I/O-based virtualization usages to improve isolation on the key operational vectors of security, performance, and reliability. The reader can refer to [4] for further details.

The alternative to the pass-through method for hardware assignment is that of virtualization. In this mode, hardware is assigned to a virtualization domain[1], also referred to as the root or parent OS partition. This virtualization domain or root partition, typically exports the device models to subsequent VMs. The root or parent partition has the physical device driver and exports a device model that may or may not match the underlying physical device (that is, an NE2000 Ethernet* card could be exported for a branded gigabit Ethernet adapter). By exporting a generic device, the VMM designer, IT shop, or OEM can better ensure device compatibility with a wide range of operating systems. However, such compatibility comes at the price of features and, potentially, usability. The benefits of exporting a native device model are to ensure full-feature support of the underlying device in VMs and also to ensure the user experience is not affected. However, these benefits are difficult to achieve without investment by the company that designed the native device. Furthermore, there may be performance implications as compared to the use of the pass-through mode.

Type 2 VMMs

Type 2 VMMs run within a host OS, typically Windows XP* or Windows Vista*, but Linux and other operating systems can also be supported. This reduces the isolation benefits that come with the use of Type 1 VMMs, but Type 2 VMMs are easier to adopt because they use the device drivers that are present in the host OS (similar to pass-through mode). A Type 2 VMM essentially turns the host OS into a "functionally analogous" root partition. Alternatively, the primary OS "owns" the devices and the VMM that runs (in Ring 0 mode) within the host

[1] This is DOM0 (domain zero) in Xen [8] parlance

OS accesses the devices through the device drivers or OS APIs. The VMMs will usually emulate or export generic device models as opposed to native ones, but there is no intrinsic reason for this.

In general, the degree of device isolation achieved currently by a Type 1 VMM is superior to a Type 2 VMM. In the Type 2 model, all hosted VMs are exposed to the same issues and constraints as the host OS: if the host is shut down, compromised, or impacted by faulty user invocations, then all VMs are also affected. Alternatively, the Type 1 model has significant benefits. As noted earlier, this model can support devices more easily as the host OS "owns" the device, so any device that has a driver can be utilized by the VMM and made available to all VMs (note that the VMM must be aware these devices and how to export them). There may also be usability benefits with the Type 2 versus Type 1 model. Because all VMs are contained within the host OS they can be exposed to the end-user simply as "applications," thereby blurring the lines of operating systems separation or independence.

A common requirement for both Type 1 and Type 2 models is the efficient sharing of platform device resources by concurrently running virtual machines or system images. While Intel broadly supports this requirement, the applicability of a hardware-based approach for device sharing should be motivated by usage, sharing, and performance requirements. In some usage scenarios, certain devices will not require sharing across multiple VMs, and/or alternatively, software-based methods may be more cost-effective, simpler to employ, and have no user-perceived performance implications. However, in cases of strict security requirements (for example, isolating user host-interface device invocations) or shared isochronous device communications, Intel supports the PCI Express* [9] specifications for hardware-based SRIOV [10] extensions. These enable virtualization and assignment of specific device functions to VMs for device sharing, as required by specific client virtualization usages and device types.

Next-generation virtualization usage models will combine these types of VMMs with streaming technologies to further enable remote management and delivery of OS and application images. The result will be a reduction in operational complexity in software deployment and image management. In the upcoming section, we present an emerging class of client computing usage models, which we refer to as Dynamic Virtual Computing (DVC). DVC embraces the merits of hardware and software virtualization technologies to deliver a flexible and rich computing platform, but integrates the additional Intel vPro technology hardware value propositions (that is, Intel AMT) to enable a more manageable and secure client virtualization platform.

Dynamic Virtual Client Computing

Client virtualization-based technologies have the potential to significantly reduce the need to make costly or ineffective business compromises. Several emerging solutions that take advantage of client virtualization allow the diverse needs of IT organizations, business units, and end-users to be met, simultaneously. These solutions support five key attributes of today's computing world: central management, protected data, on-demand delivery, local compute and graphics, and support for mobility. We call this family of solutions Dynamic Virtual Client computing.

DVC technologies and solutions include application virtualization, OS and application streaming, and an emerging class of solutions based on virtual container models. Centralized management and on-demand delivery go hand in hand by allowing IT to manage applications or OS images centrally, while remotely delivering the user required base image and any sequenced updates as needed by the user.

Data protection is delivered through multiple techniques including central storage and backup, roaming profiles, and client-side data encryption. Local computation and graphics allow users and IT personnel to take advantage of a client-computing capability for a rich and responsive user experience while minimizing data center build out.

The final critical attribute of DVC computing is the support for pervasive mobility, as business units and end users require increased flexibility for travel, day extending, and business continuity. DVC solutions have many advantages over traditional client computing models. With advanced client hardware capabilities, IT can improve the security, management, and delivery of DVC solutions. Some of these capabilities and their applicability to DVC are discussed in the next section.

Traditional Requirements and Emerging Solutions for DVC

Regardless of the DVC model used, management of the physical PC is critical. Being able to inventory both hardware and software regardless of system power or health state is critical for IT groups.

In situations where the PC is unable to boot due to software, BIOS, or hardware issues, remote troubleshooting capabilities such as IDE-R and SOL are extremely valuable, in traditional and DVC compute models. Remote troubleshooting enables IT to run diagnostics and, in many cases, repair systems without having to make a desk-side visit.

Another critical requirement of IT groups is to be able to remotely power on and off PCs. This allows IT personnel to apply updates to applications during off hours thereby minimizing the impact on the user and saving energy by not having to power client systems overnight. Remote power control also reduces peak loads on streaming servers when applying updates to broadly-deployed applications.

Also critical to IT groups is ensuring that the overall PC management solution has the appropriate levels of security in setup, provisioning, and ongoing communication. At the same time, PC management must be well integrated into existing enterprise systems for administration, device, network, and account management. Providing support for device authentication protocols, such as those supported by Cisco* Self-Defending Network (SDN), Microsoft* Network Access Protection (NAP), or 802.1x [11], is good example of such a management solution. Support for device authentication protocols enables client network access for management and troubleshooting functions, even when the host OS is powered down or not functioning.

The client hardware capabilities discussed so far have broad applicability across traditional and DVC models. We move on to discuss a few capabilities that are specific to DVC, or that are unique in the way that they can be applied as part of a DVC solution.

For OS streaming, a critical point in the process is kicking off the OS stream at the initial stages right after the PC power button is pushed. First, the diskless PC needs to have network access. In cases where end-point authentication technologies are implemented (for example, 802.11x) firmware and hardware support to access the network is required. The next challenge is to initiate the boot process securely by executing a small bootloader or bootstrap (<100KB) on the device. Most of the OS streaming solutions utilize PXE (Pre-boot eXecution Environment) to get this bootloader to the client. The challenge this can pose to IT systems is that IT personnel may already be using PXE for another application and they may therefore not allow the use of PXE, because it is a broadcast protocol. Alternatively, IT personnel might not support DHCP on their network, and PXE requires DHCP. An alternative to PXE is to use IDE-R as the mechanism to launch the bootloader. IDE-R service can be triggered by the client by using Intel AMT alerts that are sent during system power cycles.

Another area where hardware capabilities are directly applicable to DVC is virtual containers in which Intel technologies such as Intel® Virtualization Technology (Intel® VT) for IA-32, Intel® 64 and Intel® Architecture (Intel VT-x), Intel® Virtualization Technology for Directed I/O (Intel® VT-d), and Intel® Trusted Execution Technology (Intel® TXT) are utilized. These technologies enable device pass-through, DMA remapping, memory protection, and secure launch to ensure the virtualization layers are not tampered with or modified.

The same mechanisms that allow IT organizations to inventory software from the client can be used to control and modify Access Control lists (ACL) for virtualized applications and for notifying host-based agents that urgent updates to policies/applications are required. Virtualized application solutions often have client-based agents that can add, renew, or remove access to a virtualized application on the client. The ability to write and read from nonvolatile memory on a client, regardless of system state, enables some unique approaches to managing these application ACLs. The nonvolatile memory, also known as third-party data store (3PDS), can be used by both the host-based virtualized application agent and the application console to store application ACLs. This capability enables a console to add, renew, or remove access to an application regardless of system state. Even if the system is turned off, the console can remove or renew access to an application, and once the system is powered on, the host-based (virtualized) agent checks 3PDS and immediately applies the changes. A common example is removing the application from the client. Alternatively, consoles can post flags to 3PDS to notify their agent that a critical update or policy change needs to be applied. This flag is then recognized by the host-based agent and the policy or update is immediately applied when the system wakes. These approaches can save even more power by not requiring the administrator to wake the system to apply the update, but yet ensuring when the system does wake, the ACL, policy change, or the update will be acted upon.

As discussed, DVCs provide a win-win solution for IT organizations, business units, and end-users needs by providing central management, local execution, and support for mobility. The advanced hardware

features in Intel vPro technology are important for both existing and DVC models to ensure cost-effective and secure PC management. In addition, several Intel vPro technology capabilities add value in the areas of performance, management, and security—specifically for the DVC technologies of OS streaming, application virtualization and streaming, and virtual containers.

Virtual Client Computing Vision and Futures

Our vision entails a combined software and hardware solution architecture for client platform virtualization in which end-user empowerment is enabled yet balanced with the delivery of robust IT security and manageability. By enabling a Dynamic Virtual Client computing framework, data types, computing execution models, software delivery strategies, and end-user roaming options can be balanced across alternative client computing scenarios, contexts, and administrative domains. Figure 3 depicts a platform model in which we envision hardware and software solution integration strategies.

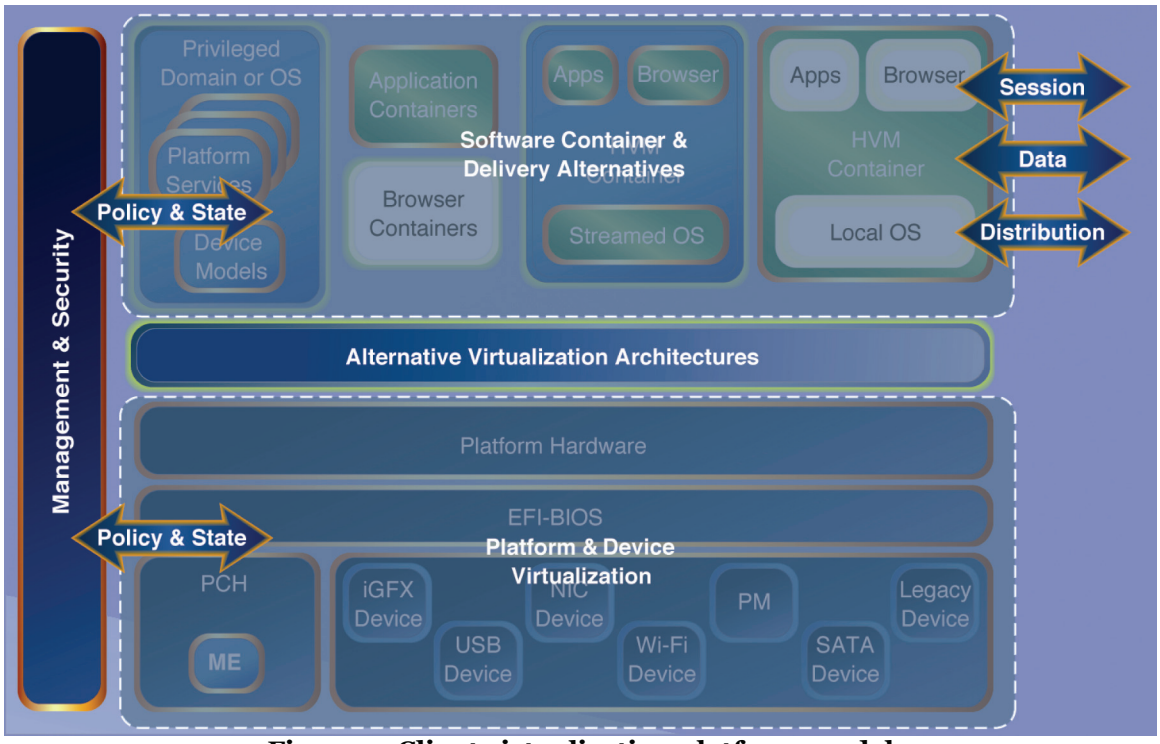

Figure 3: Client virtualization platform model

System Overview

The requirements for DVC computing are consistent with the core CIO needs for compliance, security, reliability, performance, and lower costs—all integrated with enterprise computing infrastructure and service delivery. Alternatively, DVC users will expect flexibility in terms of computing and image customization, mobility, and personalization, as well as assurance that their privacy needs will be met. As reflected in Figure 3, we envision multiple end-user run-time scenarios securely launched and executing on a VMM or software-based virtualization architecture with hardware-based virtualization underpinnings. In supporting an evolutionary client virtualization strategy, we recognize the following transitional vectors for DVC computing:

1. Hardware-assisted software isolation and delivery. The requirement for hardware separation enables alternative software packaging, data management, and delivery strategies, and thus, broader solution choices for IT administrators. Providing an unconstrained software innovation platform, while ensuring operational robustness through hardware, is essential to enabling a flexible computing experience for the user.
2. Usage-driven virtual machine platform architectures. As client virtualization matures and gains broad adoption by the IT community,

we envision greater convergence on deploying alternative VM models. More specifically, we view client provisioning and deployment technologies facilitating alternative platform customization and software delivery strategies. These strategies will be based on functional or business contexts, user usage models, and varying client device acquisition and IT support scenarios.

3. Seamless physical and virtual computing manageability and security. A challenging but necessary shift that must take place to support virtual computing environments. First, management and security solutions must be redirected to comprehend the robust merits of hardware for secure manageability. Second, software must enable flexibility to adapt to contexts and the complexities of managing mixed physical and virtual computing environments.

Software Container and Delivery Models

Supporting the requirement for mixed computing models is an emerging usage scenario. For example, personal-corporate environment or dual OS environments for platform migration will soon be deployed seamlessly and coexist robustly within the same physical machine. Within this context, we expect to see alternative scenarios for image containment or software delivery over existing physical or virtual machine paradigms. Lightweight delivery models can support granular browsers, applications, or service container deployments, and heavier-weighted containers can support local or remote image OS (for example, streaming) deployments. Moreover, we envision a client virtualization framework that comprehends a true dynamic, distributed architecture that adjusts to IT and user constraints across the end-to-end spectrum. Thus, end-user requirements are met for roaming, optimal user experience and computing environment flexibility. At the same time, policies relating to critical IT runtime services and data requirements will be managed dynamically by centralized or decentralized instantiations as required by the IT organization.

Client Virtual Machine Architectures

Within a virtual client computing environment, either Type 1 or Type 2 virtualization is feasible and supported. We will look to integrate common hardware capabilities into both virtualization models, enabling robust virtual container and software delivery models. Matching traditional single-image models for client computing in terms of security, isolation, power and QOS, and reliability in the client virtualization space are challenges that we are tackling jointly via hardware- or software-based solutions. Intel VT enables robust execution of unmodified guest operating systems to run on VMMs enhanced with Intel VT, as discussed in the earlier sections. Enhancements to Intel VT will continue to improve effective (QOS, security) IO device sharing, and will optimize energy and performance to support operating in a variety of DVC computing usage scenarios.

In addition, software delivery or image management systems will need to discover and customize images, based on alternative VMM platform and user execution profiles. Customers may employ a heterogeneous hardware and image deployment strategy that can utilize the inherent advantages of Type 1 or Type 2 virtualization environments and alternative image customization strategies that are based on specific usage scenarios. It is likely that there will be usage scenarios in which both models are supported. Providing a strategy that enables usage-based IT choice for Type 1 or Type 2 models, and more specifically, a strategy that is integrated into a common framework for platform deployment, software delivery, and manageability is key to ensuring broad client virtualization solution opportunities. In such cases, prescriptive guidance on how best to choose or manage the co-existence of mixed virtualization environments will be needed to accelerate deployments and unlock greater business value for the IT industry.

Management and Security

In supporting a client virtualization framework, systems management tools and Intel AMT should be integrated in a fashion that comprehends different virtualization architectures, software delivery models, and platform categories. Supporting a user's computing flexibility and roaming needs, synchronizing and securing data, and provisioning and managing of image (OS, application) granularities and delivery strategies will require hardware-based mechanisms to harden the virtualized platform. Software-based solutions will be needed that support a user's computing flexibility. We envision systems management tool capabilities and platform mechanisms evolving to support dynamic virtual client computing, based on the following requirements:

Supporting existing in-band and out-of-band physical platform management capability and manageability features including alternative software installation or software delivery models, user profile management, and traditional platform management functions.

Transitioning similar platform capability and feature mechanisms across locally-hosted virtual machine and application virtualization environments, including enabling more robust boot procedures and streaming delivery strategies that are fully integrated into the virtualization environment.

Enabling novel management and security solution capabilities, specific to virtualized environments that are partitioned or fully integrated into end-user and corporate constructs. Additionally, dynamic, policy-based provisioning models that is, support for roaming, image and data delivery) for virtual machine environments and software container models should all be flexible enough to serve centralized or decentralized management models.

In partnership with key ISV partners, we will investigate the three major DVC vectors through advancement of hardware and software solution capabilities that support integration of traditional IT provisioning and administrative frameworks into underlying physical and virtual machine manageability constructs.

Summary

Today, an emerging trend for many corporations is the abstraction of the traditional distributed computing model in which computing hardware, OS, data, and applications may be redistributed to enable alternative forms of business computing. This trend is driven primarily by the need to alleviate security and manageability complexity through increased data and image centralization. While the cost and complexity of PC management has been a constant struggle for IT managers, current trends in mobility, software delivery, end-user empowerment, and consumerization present further challenges to the client computing environment. Client virtualization technologies offer a viable path to address the objectives of IT organizations and end-users for policy-based control and flexibility, respectively. In this article, we introduce and expand on the Dynamic Virtual Client computing framework, based on Intel vPro technologies as the IT solution response to address these needs and the associated barriers for alternative computing adoption. Further, we propose a combined hardware and software solution direction and vision that Intel is advancing jointly with ISV partners towards delivering a robust and flexible client virtualization platform.

Acknowledgements

We would like to thank our colleagues, Mike Ferron-Jones, Stuart Schaefer from Microsoft Corporation, Chris Hockett and Paul Hahn from Citrix Corporation for their invaluable insights, comments and assistance.

References

[1] R. Uhlig, Foreword: "Intel® Virtualization Technology: Taking Virtualization Mainstream on Intel® Architecture Platforms." Intel Technology Journal, Volume 10, Issue 03. August 10, 2006.

[2] "BYOC: Bring Your Own Computer—to work." Associated Press. September 25, 2008.

[3] Intel Corp. "Intel Virtualization Technology Specification for the IA-32 Architecture." At http://www.intel.com/technology/virtualization.

[4] Intel Corp. "Intel Virtualization Technology Specification for Directed I/O Specification." At http://www.intel.com/technology/virtualization.

[5] Intel® Trusted Execution Technology http://www.intel.com/technology/security/downloads/arch-overview.pdf.

[6] Intel® Active Management Technology. At http://www.intel.com/technology/platform-technology/intel-amt/index.htm

[7] R. Goldberg. "Survey of Virtual Machine Research." IEEE Computer, pp. 34–45. June 1974.

[8] "Xen Virtual Machine." At http://www.xensource.com

[9] PCI Express* Base Specification 1.1. At http://www.pcisig.com/specifications/pciexpress/base

[10] "Single Root I/O Virtualization and Sharing 1.0 Specification." At http://www.pcisig.com/specifications/iov/single_root/

[11] "IEEE 802.1x-2004 Port Based Network Access Control." At http://www.ieee802.org/1/pages/802.1x-2004.html

Authors' Biographies

Tom James has been at Intel 11 years and currently works as a market development manager in the Digital Office Ecosystem Development Team. For the past two years Tom has been responsible for driving ISV and OEM engagements related to emerging client compute models and prior to that also performed planning and marketing within the Digital Office Platform Division. Before joining DOPD Tom held three diverse roles at Intel, initially working in Intel's manufacturing group with assembly test capital equipment suppliers, in Intel Online Services working with IT hardware and software suppliers, and within Intel IT as a product line manager. Tom holds both B.S. and M.S. degrees in electrical engineering from Brown University. His email is thomas.h.james@intel.com.

Jason Kennedy, senior manager with Intel Corporation's Software and Services Group, is responsible for establishing, developing, and executing Intel's strategy with partners ranging from start-ups through global leaders. These engagements deliver architectural, technical enabling, and marketing synergies that accelerate revenue attainment and market share growth. His career experience includes building a product line from scratch into a business capturing several billion dollars in revenue, as well as other leadership positions in manufacturing, sales, and marketing. His email is jason.kennedy@intel.com.

Steve Kremesec manages the Digital Office Enabling and Execution team, tasked with bringing to market next generation capabilities and usage models for Intel vPro Technology platforms. Steve has been with Intel for 13 years starting out as a systems programmer. Subsequently Steve managed a worldwide team of application engineers leading the first launch of the Intel® Centrino® processor technology-based platforms. Steve holds a Bachelor of Science degree in computer science and is actively pursuing an M.B.A. degree. His email is Steve.v.kremesec@intel.com.

John Vicente is a senior principal engineer in the Digital Office Platform Division. John is a member of the Intel Research Council's Communications Committee and Internet Software patent committee. John was previously the director and chair of the Information Technology Research group at Intel. He is the co-director of the UC Davis Center for Future Information Technology and a member of the UC Davis Computer Science Industrial Advisory Board. John has over 25 years of industrial experience spanning research, architecture, and design of information technology systems. He has authored numerous publications in the field of networking and has patent applications filed in networking and distributed systems. He is currently a Ph.D. candidate at Columbia University's Center for Network Research in New York City. John received his M.S. degree in electrical engineering from the University of Southern California, Los Angeles in 1991 in network systems design and his B.S. degree in electrical engineering from Northeastern University, Boston in 1986. His email is john.vicente@intel.com.

Next-Generation Streaming Clients Based on Intel® vPro™ Technology

Hormuzd Khosravi, Corporate Technology Group, Intel Corporation
Dori Eldar, Digital Enterprise Group, Intel Corporation
Venkat Gokulrangan, Mobility Group, Intel Corporation
Thomas James, Digital Enterprise Group, Intel Corporation
Nancy Sumrall, Mobility Group, Intel Corporation

Keywords: PXE, IEEE 802.1x, IEEE 802.11, OS streaming, Intel® AMT, NAC, OOB manageability, centralized compute model

Abstract

Many enterprises are moving towards a centralized compute model where the operating system (OS), applications, and data are stored on a central server, or on a server farm, and streamed to remote clients for execution. Improvements in flexibility of work environments, availability of data, and business continuity are the factors that motivate enterprise IT departments to consider a central compute model. For example, having the OS and software maintained on a central server simplifies the software licensing and compliance requirements in large enterprises. Security is also a critical consideration for many of these enterprises, but it often poses deployment and interoperability challenges when it has to coexist with other technologies. One such challenge faced by IT organizations is the deployment of OS Streaming (Preboot eXecution Environment—PXE) technology in 802.1x access control networks. In this article we describe how Intel® vPro™ technology provides unique, industry-first solutions that can be used by IT organizations to build the next generation of streaming client platforms that provide the same flexibility without sacrificing security or user experience.

Introduction

To improve management, security, and to reduce operational costs, many enterprises are moving towards centralized compute models in which operating systems, applications, and/or data are stored and managed centrally. The mechanisms for delivery of operating systems and data to remote clients and the end user experience vary greatly: they range from rendering the client user interface in a remote location by using technologies such as Remote Desktop Protocol (RDP) [10], to mechanisms such as streaming the operating system (OS) image to the clients over the network, by using a Preboot eXecution Environment (PXE) [9]. The newer mechanisms, which we call Dynamic Virtual Clients (DVCs), allow IT departments to centrally manage an application or OS image while still allowing it to execute at the client end-point, for a rich and responsive end-user experience.

Along with these emerging compute models, security continues to be a major focus for IT organizations of all sizes. A critical component in delivering a multilayered security strategy is protecting access to the network through end-point access control solutions such as IEEE 802.1x [8]. These solutions allow IT departments to enforce security policies and prevent rogue or unmanaged devices from infiltrating the enterprise's network, thereby minimizing the risk of contamination and data loss or theft. Many IT organizations began to implement stricter network access control mechanisms when they introduced Wireless Local Area Networks (WLANs) [14] into their computing environments. Because a WLAN client does not need to be physically connected to a network, it is important to control its access to the network. These access control protection mechanisms and vendor solutions were subsequently also extended to the wired network infrastructure.

As IT organizations look at new mechanisms for deploying applications and OS images by using client streaming technologies such as PXE [9], they need to be aware that their new technologies have to coexist and inter-operate with other technologies, such as

those used for security and access control, and this coexistence can cause problems.

Over the last three years Intel has focused on embedding security and manageability functions into the hardware and firmware of corporate platforms. These security and management capabilities are included in platforms enabled with Intel® vPro™ technology [25] and include features such as secure access to the computer regardless of its power state or the health of the OS. We call this capability Out of Band (OOB) access, and it helps IT organizations significantly in their efforts to manage and troubleshoot platforms remotely, and to reduce costs and energy consumption. We enable this OOB access and many other capabilities by integrating a microcontroller into our chipsets. This microcontroller is the Intel® Management Engine (Intel® ME) [23], and it runs an embedded firmware stack, Intel® Active Management Technology (Intel® AMT) [21], that supports network connectivity, authentication, power control, and several other security and manageability functions.

In this article we describe how Intel vPro technology provides unique, novel solutions that can be utilized by enterprise IT organizations to build the next generation of streaming client platforms that provide the same flexibility as those of previous generations, without sacrificing security or user experience. Specifically, we describe how the Intel® Embedded Trust Agent [26], which is part of Intel vPro technology, provides a solution for OS streaming or PXE in 802.1x access control networks. These new mechanisms and solutions fall under the umbrella of DVCs, described earlier. Moreover, DVCs allow IT organizations to simplify management processes for updates and patches, improve data security, and deliver a robust solution for end users.

In this article we first present background information on PXE boot, 802.1x authentication protocols, and Network Access Control technologies. We describe why PXE does not work in 802.1x networks, and we examine the problems with existing workarounds. We follow this with a description of the Intel Embedded Trust Agent, and we describe how this agent enables OOB manageability in 802.1x networks. We then describe the architecture and algorithms that support PXE in 802.1x networks that use the Intel Embedded Trust Agent technology. Finally, we present some new challenges and solutions for next-generation streaming clients.

Background and Problem Description

In a centralized compute model, the set of network protocols used to stream and load the OS onto a client (typically diskless) for remote boot is defined by the Preboot eXecution Environment (PXE) standard specification [9]. PXE has been around since the 1990s, and it is widely used in many industries in which diskless, end-user terminals dominate the installed compute base, including banking, finance, healthcare, education, and various other industries and institutions. PXE uses the Dynamic Host Configuration Protocol (DHCP) [18] and the Trivial File Transfer Protocol (TFTP) [19] to transfer the boot-loader (network bootstrap program or NBP) onto the client system, which in turn downloads the complete OS image onto the client from the boot server and boots that image. PXE is typically implemented as an Option ROM (OP-ROM) [9] inside the BIOS [27].

Security is an important consideration for most enterprises. 802.1x is a popular standard that is deployed by many enterprises for providing Layer-2 authentication in their networks. 802.1X is an IEEE standard for LAN, port-based, access control, and it is part of the IEEE 802.1 group of networking protocols. The 802.1x standard provides device authentication mechanisms for clients before they can access the network on a particular LAN port. It can be used for both wired and wireless (802.11) [14] networks, and it is based on the Extensible Authentication Protocol (EAP) framework [1] defined by the Internet Engineering Task Force (IETF). EAP is an authentication framework that relies on different EAP methods, defined in the IETF, to describe authentication protocols. Different EAP methods support both certificate- and password-based authentication. The 802.1x standard is also thought of as the simplest form of Network Access Control (NAC).

NAC defines a set of protocols that is used to secure an enterprise or corporate network before the client accesses the network. It provides mechanisms for the network to evaluate a client device both in terms of access credentials (authentication) and in terms of compliance to corporate IT policies. Cisco* Network Admission Control (C-NAC) [11], Microsoft* Network Access Protection (M-NAP) [12], and Trusted Computing Group's* Trusted Network Connect (TCG-TNC) [13] are all examples of NAC implementations in the industry. Typically, NAC builds on top of 802.1x-EAP methods or other protocols, such as IPSec [16], and it defines extensions for evaluating the clients' compliance with IT policies.

In a typical 802.1x/NAC protocol exchange such as the one shown in Figure 1, a client (herein known as a supplicant or Access Requestor—AR) exchanges data/credentials with an authentication (policy) server, via an authenticator, to seek access to a network. The supplicant or AR is a piece of software running on the client OS that implements the 802.1x/EAP protocol stack; the authenticator (known as a Network Access Device–NAD) is an Ethernet switch or wireless Access Point (AP); and the authentication server is an Authentication Authorization and Accounting (AAA) [17] server that implements the RADIUS [17] protocol to talk with the authenticator. When a client is connected to a switch (authenticator) on a port that is 802.1x enabled, the authenticator sends out an EAP-Request to the client requesting its credentials. The client supplicant sends its authentication credentials (such as username/password or digital certificate) in an EAP-Response message. The switch (authenticator) relays this response back to the authentication server over the RADIUS protocol. The authentication server decides whether the client should be granted network access based on IT policy and the client's credentials/data. The authenticator sends back the results to the switch that enforces the network access policy for the client, based on those results. For example, in the case of an Ethernet switch, the results would indicate which VLAN [20] (Corporate or Guest) the client can access.

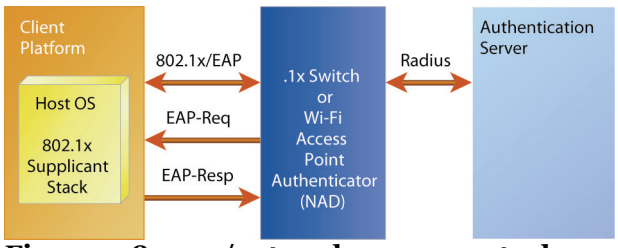

Figure 1: 802.1x/network access control network

Why PXE Fails in 802.1x/NAC Networks

One of the common problems faced by many of these enterprises as they move towards adding security (based on 802.1x/NAC) into their network infrastructure is that this breaks existing OS streaming deployments, that is, PXE. As described earlier, 802.1x networks require that a client (supplicant) authenticate its credentials with the authentication server before it is granted network access. The reason why PXE does not work in 802.1x networks is because it does not have this 802.1x supplicant support or authentication credentials provisioned inside the legacy BIOS. Before platforms with Intel vPro technology were introduced, IT network administrators had to manually set exceptions for PXE in their corporate 802.1x networks. This process is both time-consuming for IT departments, and more importantly, the exceptions also make the corporate network less secure. In the next section of this article, we describe how the Intel Embedded Trust Agent [26] helps support OOB manageability in 802.1x/NAC networks.

Overview of the Intel® Embedded Trust Agent

As described earlier, in order for an end-client to gain network access in 802.1x networks, the client needs to provide its credentials first, and these credentials have to be validated before access is granted. This validation is typically done by the 802.1x supplicant stack running in the OS of the end-client. Furthermore, in the case of NAC networks, additional client posture (C-NAC) or health (M-NAP) information is required up front to ensure compliance with IT policy. An example of the additional posture or health information would be the name, version, and patch level of the OS running on the client. This additional information is provided via other software components running in the client OS. Therefore, in cases where the OS has crashed or the client has not been booted, the client will fail to have network access in 802.1x/NAC networks.

The Intel Embedded Trust Agent that is part of platforms with Intel vPro technology consists of the 802.1x supplicant as well as extensions for Cisco* NAC and Microsoft* NAP embedded in the chipset/ME firmware. It thus enables industry-first OOB or pre-OS manageability in 802.1x/NAC networks. The Intel Embedded Trust Agent is supported for both wired and wireless (Wi-Fi*) networks, and it supports both digital certificate-based and username/password-based authentication. The authentication methods (EAP methods) supported by the Intel Embedded Trust Agent inside Intel AMT firmware are summarized in Table 1.

The Intel vPro technology Remote Configuration method described in [24] can be used to provision authentication credentials inside the Intel Embedded Trust Agent. No user intervention is required for provisioning it. The authentication credentials are stored inside the secure storage area (flash memory) [23] provided by the Intel ME. Table 2 summarizes the support for 802.1x, C-NAC, and M-NAP for different versions of AMT firmware released since 2007.

Table 1: Intel® AMT EAP authentication methods

Intel AMT 802.1x/EAP Methods supported	Intel AMT Cisco* NAC extensions	Intel AMT Microsoft* NAP extensions
EAP-FAST [4]	X	
-EAP-GTC [7](inner)	X	
-MS-CHAPv2 [6](inner)	X	
-EAP-TLS [2](inner)	X	
PEAP [3]		X
-MS-CHAPv2 [6](inner)		X
EAP-TLS [2]		
EAP-GTC [7]		
EAP-TTLS [5]		
-MS-CHAPv2 [6](inner)		

Table 2: A summary of the support for 802.1x, C-NAC, and M-NAP for different versions of Intel® AMT firmware released since 2007

Intel® AMT Version	802.1x support	Cisco* NAC support	802.1x-PXE support	Microsoft * NAP support
2.5	X	X		
2.6	X	X	X	
3.0	X	X		
3.1	X	X		
3.2	X	X	X	
4.0	X	X	X	X
5.0	X	X	X	X

Algorithm for Host OS–Intel® ME 802.1x Synchronization

As described in the Intel AMT Specification [23], the Intel ME and host OS share Layer-2 (Ethernet) and Layer-3 (IP) addressing. Thus, in a typical wired 802.1x-enabled network, the state of the port (closed or open) will apply to both entities. For example, once the 802.1x authentication is successfully completed by the host OS, Intel ME will also be able to use the open port for communication. Furthermore, only a single entity should drive the 802.1x link authentication. The Intel ME policy is to allow the host OS to drive the authentication, as long as the OS is operational. The Intel ME limits its active authentication mode to only those states in which the host OS is nonoperational. The host OS is nonoperational either due to the system being in a low-power state, or due to the fact that the OS malfunctions. We have devised a synchronization algorithm that allows the Intel ME to detect those nonfunctional OS states and to limit its 802.1 x authentications to those states. Figure 2 illustrates the synchronization state machine implemented for the Intel Embedded Trust Agent by platforms enabled with Intel vPro technology.

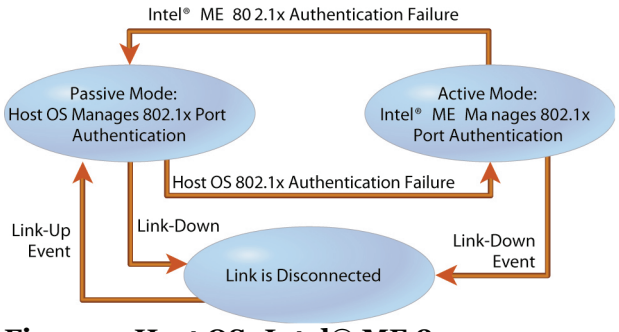

Figure 2: Host OS–Intel® ME 802.1x synchronization state machine

Note that the default mode for the Intel ME, following an initial link-up event, is to allow the host OS to perform authentication. The Intel ME will proactively initiate 802.1x authentication requests (Active Mode) by using the Intel Embedded Trust Agent, only when an 802.1x authentication failure is detected for the host OS. The Intel ME has access to Ethernet packet filters (system defense filters) [23] in the chipset that it uses for exclusively performing 802.1x authentication during its Active mode.

We now examine how the Intel Embedded Trust Agent detects host OS 802.1x authentication failures. As described in the Intel AMT specification [23], the Intel ME has direct access to the platform LAN controller (LOM) and capabilities to record and track Ethernet packets sent or received by the host OS. Platforms enabled with Intel vPro technology use this fundamental capability to track 802.1x protocol messages and specifically to detect EAP success and failure messages sent by the authenticator. Moreover, the transmission of IP traffic (that is, DHCP requests/responses) serves as an indication to the Intel ME that the system is connected to a non-802.1x-enabled network.

Architecture for 802.1x-PXE Technology

As described earlier, PXE fails in 802.1x/NAC networks, because the 802.1x supplicant stack is not present inside legacy BIOS to perform authentication. In this section we describe how we use the Intel Embedded Trust Agent to provide a novel solution to this challenge.

By using the Intel Embedded Trust Agent capability to establish an 802.1x authentication channel without the client OS supplicant, enterprises can now configure Intel vPro brand platforms to complete 802.1 x authentication upon system boot-up to allow PXE boot to take off from an already open 802.1x port. In a PXE boot environment, when these platforms are configured with the 802.1x-PXE-Enable option (via remote configuration [24]), the Intel Embedded Trust Agent/Intel AMT firmware implements the modified 802.1x synchronization state machine that is depicted in Figure 3.

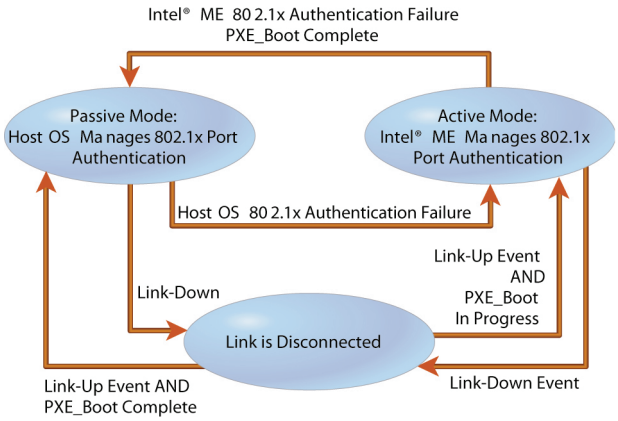

Figure 3: Host OS–Intel® ME 802.1x synchronization state machine for PXE boot

Architecture for 802.1x-PXE Technology

In a PXE boot configuration (802.1x-PXE-Enable option set inside an Intel vPro brand platform), the Intel ME transitions to an active state immediately following system boot-up (boot-in-progress), or immediately following the initial link-up event (during boot-in-progress) as depicted in Figure 3. This is different from the situation depicted in Figure 2, in which the Intel ME only transitioned to the active 802.1x authentication state when the host OS 802.1x authentication failed. This modification to the state machine allows the Intel ME to actively manage the 802.1x authentication during the initial boot-up of the system, so as to allow PXE to use the open 802.1x port to download the OS from the remote server and boot the system. The Intel ME transitions to passive mode when the PXE boot is completed.

Note that, currently, the IT administrator must explicitly configure a system, enabled with Intel vPro technology, for either a PXE boot environment (802.1x-PXE-Enable option) or a non-PXE-boot environment (default behavior), as illustrated by the different initial states within Figures 2 and 3. Also note that although the Intel Embedded Trust Agent is supported for both wired and wireless (802.11) networks, the 802.1x-PXE technology is supported for wired (LAN) interfaces only. Later, we describe our future work that centers on extending this capability for wireless (WLAN) interfaces.

Several options are used by Intel ME to detect the completion of PXE boot (PXE_Boot_Complete flag event in Figure 3). These options are listed in Table 3.

Table 3: PXE boot-completion detection methods

1	Detect 802.1x/EAP packets from Host: PXE boot complete
2	Configurable PXE boot timeout -> 120 seconds (default value)
3	Detect Host OS-Intel ME communication driver (HECI) is up (using a watchdog message)

Protocol Flow for 802.1x-PXE

Figure 4 depicts the protocol flow for enabling PXE inside 802.1x/NAC networks. The protocol flow can be broken down into the following basic steps. (Note that each step can further consist of several protocol exchanges over the wire):

1. The 802.1x-enabled Ethernet switch sends an authentication request (EAP-Request) to the client platform, asking for its credentials.
2. The Intel Embedded Trust Agent sends an authentication response (EAP-Response) to the switch, providing its credentials.
3. The switch passes the authentication credentials onto the AAA (RADIUS) Server for verification and for an IT policy-compliance check.
4. Based on the credentials, the AAA server grants the client platform access to the network. It sends the results of the authentication to the switch that implements the access control.
5. When the client platform has network access, it receives a valid IP address, and the PXE boot agent (inside the BIOS) on the client downloads the OS from the PXE server on the corporate network.
6. Once the OS is streamed onto the client, it starts booting.

When the Intel ME detects that the host OS is up and running, it terminates its 802.1x authentication channel, allowing the OS to authenticate itself.

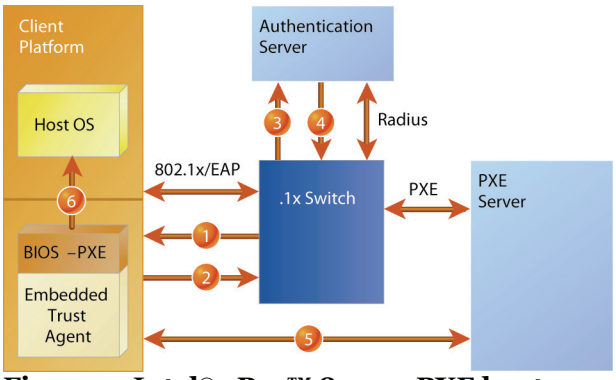

Figure 4: Intel® vPro™ 802.1x PXE boot protocol flow

New Challenges and Solutions for Streaming Clients

With the wide adoption of Wi-Fi (IEEE 802.11) technology in enterprises, many IT organizations are looking for OS streaming solutions in those networks. However, OS streaming (or PXE in Wireless) Wi-Fi networks continue to be a challenge. A wireless PXE solution has never been deployed in enterprises, because of the following significant challenges:

1. Most Wi-Fi network cards do not have sufficient flash memory space to store the PXE OP-ROM and WLAN driver code needed to support wireless PXE. The WLAN driver is relatively large to be part of the BIOS code as well.

2. The secure configuration/provisioning of the Wi-Fi 802.11i Security Profile, to enable a WLAN connection, is not a trivial undertaking.

3. There remain many open issues related to ownership and synchronization of the Wi-Fi communication channel between the host OS and BIOS/firmware in the NIC, similar to the issues described earlier for a wired 802.1x channel (see Figure 2).

We are investigating solutions to these challenges based on Intel vPro technology. Intel AMT provides support for a WLAN stack inside its firmware, including the IEEE 802.11i [15] and 802.1x standards (described earlier) as well as methods for secure provisioning of the platform credentials. One of our options is to utilize the IDE-Redirection (IDE-R) protocol capability that is provided as part of Intel vPro brand platforms. IDE-R allows a remote server to boot a client system off a diagnostic OS for troubleshooting or recovery purposes when its local OS is not booting. IDE-R is a secure protocol [22] that can be used over Wi-Fi networks. Thus, it could be used for OS streaming in wireless networks.

The other option is to utilize the WLAN stack and communication capability in the Intel® Management Engine (Intel® ME) and Intel® AMT firmware from the BIOS/PXE code directly, by using the Host Embedded Controller Interface (HECI) [23], defined for communication between the BIOS/Host and the Intel ME. The PXE specification defines a Universal Network Driver Interface (UNDI) that allows the PXE base code to talk with different kinds of networking devices. We propose using a similar abstraction or proxy for the PXE code to talk with the WLAN NIC, via the Intel ME/AMT firmware. With this approach, the Intel ME would establish an 802.11 session with the wireless AP. It could use the existing Intel Embedded Trust Agent to provide 802.1x/NAC authentication over wireless as well. Once this wireless channel is set up, the PXE code can use this to download the boot-loader and subsequently the OS over that wireless channel and boot off it (see Figure 5). Please note that this capability is a future consideration, and it is not a part of our current Intel vPro technology products.

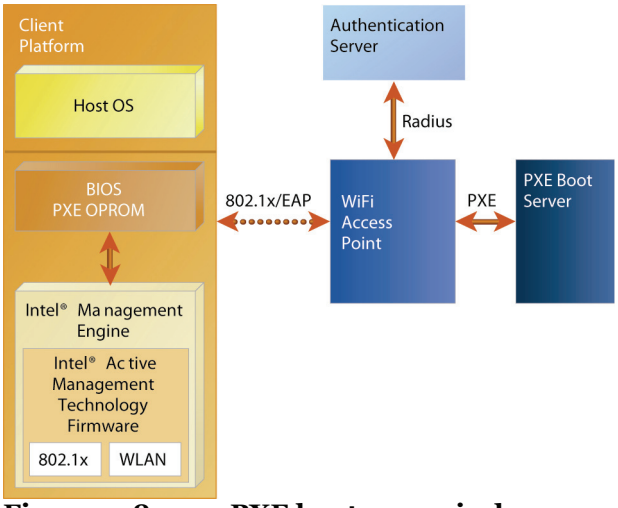

Figure 5: 802.1x PXE boot over wireless (802.11) networks

Summary

In this article, we described some of the motivations for IT organizations to move towards centralized compute models in which the operating systems and applications are streamed onto remote clients. We examined some of the challenges that IT departments face in the deployment of OS streaming/PXE inside secure 802.1x networks. We showed how Intel vPro brand platforms, with the

Intel Embedded Trust Agent, provide an industry-first solution to OOB manageability in 802.1x networks and to addressing these challenges. We also described how Intel vPro technology can be applied to other challenges for next-generation streaming clients, such as support for OS streaming/PXE inside wireless/Wi-Fi networks. By addressing these new usage models and providing novel solutions for next-generation clients, Intel vPro technology is redefining streaming clients for enterprise IT organizations.

Acknowledgments

We acknowledge our reviewers and contributors: Yasser Rasheed, Divya Kolar, Ajith Illendula, Tal Roth, and Yael Yanai. We also thank Cisco and Microsoft for their collaborative efforts on these technologies.

References

[1] RFC 5247. "Extensible Authentication Protocol (EAP) Key Management Framework." B. Aboba, D. Simon, P. Eronen. Eds. August 2008.

[2] RFC 5216. "The EAP-TLS Authentication Protocol." D. Simon, B. Aboba, R. Hurst. Eds. March 2008.

[3] Internet draft. "Protected EAP Protocol, (PEAP–MSCHAP-v2)." A. Palekar, V. Kamath, M. Wodrich. Eds. July 2004.

[4] RFC 4851. "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)." N. Cam-Winget, D. McGrew, J. Salowey, H. Zhou. Eds. May 2007.

[5] RFC 5281. "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)." P. Funk, S. Blake-Wilson. Eds. August 2008.

[6] RFC 2759. "Microsoft PPP CHAP Extensions, Version 2, MSCHAPv2." G. Zorn. Editor. January 2000.

[7] RFC 3748. "Extensible Authentication Protocol (EAP)." B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Eds. June 2004.

[8] "IEEE 802.1X: 802.1X-2004—Port-Based Network Access Control." At http://www.ieee802.org/1/pages/802.1x-2004.html

[9] "PXE 2.1 Spec: The Preboot Execution Environment specification, v2.1." Intel & Systemsoft. At http://www.pix.net/software/pxeboot/archive/pxespec.pdf

[10] "RDP: Microsoft Remote Desktop Protocol." At http://msdn.microsoft.com/en-us/library/aa383015.aspx

[11] "Cisco NAC Network Admission Control." At www.cisco.com/web/go/nac

[12] "Microsoft NAP Network Access Protection." At http://www.microsoft.com/technet/network/nap/napoverview.mspx

[13] "TNC TCG Trusted Network Connect—Trusted Computing Group." At https://www.trustedcomputinggroup.org/groups/network/, http://www.interop.com/archive/pdfs/TCG.pdf

[14] "IEEE 802.11 (Wi-Fi™): Wireless Local Area Networks." At http://www.ieee802.org/11/

[15] "IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements." At http://standards.ieee.org/getieee802/download/802.11i-2004.pdf

[16] RFC 4301. "Security Architecture for the Internet Protocol (IPSec)." S. Kent, K. Seo, Eds. December 2005.

[17] RFC 2865. "Remote Authentication Dial In User Service (RADIUS)." C. Rigney, S. Willens, A. Rubens, W. Simpson. Eds. June 2000.

[18] RFC 2131. "Dynamic Host Configuration Protocol (DHCP)." R. Droms. Ed. March 1997.

[19] RFC 1350. "Trivial File Transfer Protocol (TFTP) version 2." K. Sollins. Ed. July 1992.

[20] "VLAN: Virtual LAN, IEEE 802.1Q." At http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf

[21] "Intel® Active Management Technology (AMT) Overview." At http://softwarecommunity.intel.com/isn/downloads/Manageability/Intel_AMT_Overview.pdf

[22] "Intel® AMT Redirection Library Design Guide." At http://software.intel.com/en-us/articles/intel-active-management-technology-intel-amt-software-development-kit-sdk-documentation

[23] "Architecture Guide: Intel® AMT." At http://software.intel.com/en-us/articles/architecture-guide-intel-active-management-technology

[24] "Intel® AMT Developers Guide to the sample Setup and Configuration Application." At http://software.intel.com/en-us/articles/intel-active-management-technology-intel-amt-software-development-kit-sdk-documentation

[25] "Intel® vPro™ technology." At http://www.intel.com/technology/vpro/index.htm

[26] "Solution Brief: Cisco Security Solutions with Intel® Centrino® 2 with vPro™ Technology and Intel® Core™ 2 processor with vPro™ Technology." At http://www.ciscointelalliance.com/_files/pdf/Intel-Cisco_security_r03.pdf

[27] Jeff Tyson. "How BIOS (Basic Input Output System) Works." At http://computer.howstuffworks.com/bios.htm/printable

Authors' Biographies

Hormuzd Khosravi joined Intel in 1999 and currently works as a software architect in Intel's Corporate Technology Group in Hillsboro, Oregon. His areas of specialization are security, networking, and manageability, and he has been involved with Intel® AMT architecture since 2005. He holds seven patents in this area and has more pending. Hormuzd holds a B.S. degree in electronics engineering from Mumbai University, India and an M.S. degree in computer engineering from Rutgers University, New Jersey. His e-mail address is hormuzd.m.khosravi@intel.com.

Dori Eldar joined Intel in 1999 and currently works as a software architect in Intel's Digital Enterprise Group at the Israeli Design Center, Jerusalem. His current work relates to manageability technologies: he holds four patents in this area and has more pending. He received a B.S. and M.S. degree in mathematics from the Hebrew University in 1996 and 1999, respectively. His e-mail address is dori.eldar@intel.com.

Venkat R Gokulrangan is a staff architect in Intel's Mobile Platforms Group and has focused on client manageability and security solutions with special emphasis on enterprise deployment of systems with Intel® vPro™ technology. While at Intel Labs, he initiated the work on IPv6 for residential gateways. Venkat has designed and implemented data sub-systems such as 802.1D and voice subsystems such as VoDSL software. He has an M.S. degree in computer science and engineering from the University of Michigan, Ann Arbor. His e-mail address is Venkat.r.gokulrangan@intel.com.

Tom James has been at Intel 11 years and currently works as a market development manager in the Digital Office Ecosystem Development Team. For the past two years Tom has been responsible for driving ISV and OEM engagements related to emerging client compute models. Prior to that, he was involved in planning and marketing in the Digital Office Platform Division. Before joining this division, Tom held three diverse roles at Intel, initially working in Intel's manufacturing group with assembly test capital equipment suppliers, then working in Intel Online Services with IT hardware and software suppliers, and finally within Intel IT as a product line manager. Tom holds both B.S. and M.S. degrees in electrical engineering from Brown University. His e-mail address is thomas.h.james@intel.com.

Nancy Sumrall is currently the Intel Anti-theft Program Manager in Intel's Mobile Products Group. Nancy was part of the Digital Office Ecosystem Development Team and was instrumental in the enabling the adoption of Intel® vPro™ technology (Intel® AMT and VT) with major ISVs and OS vendors. Nancy is a 17-year veteran of Intel, and in the past she drove Intel's security initiatives internally as well as in the Trusted Computing Group, in her roles as chair of the Marketing Working Group and on the board of directors. She also drove the IAPC and Energy Star initiative through the PC industry. Nancy received her M.B.A.

from the University of Phoenix in 1990, has a B.Sc. degree in computer science, and holds a B.A. degree from the University of Utah. Her e-mail address is nancy.l.sumrall@intel.com.

Extreme Programming with Intel® vPro™ Technology: Pushing the Limits with Innovative Software

Ylian Saint-Hilaire, Software & Service Group, Intel Corporation

Keywords: Intel® vPro™ technology, Intel® AMT, manageability, out-of-band, innovation, Serial-over-LAN, IDE-Redirect, 3PDS, mesh networking.

Abstract

Intel® Active Management Technology (Intel® AMT) provides a fixed set of out-of-band manageability features that work independently of the operating system (OS) and work even when the computer is sleeping. In this article, we take a look at how to make use of Intel AMT well beyond its intended design, and we examine how these new usages increase the value of deploying and activating Intel® vPro™ technology. Specifically, we discuss creative uses of four Intel AMT functions: Serial-over-LAN, IDE Redirection, Agent Presence, and Third-Party Data Storage. These functions are readily available to a large installed base of Intel vPro technology users. We also show how these creative uses are secured with existing Intel AMT authentication and privacy mechanisms.

Introduction

The Intel® vPro™ brand identifies computers optimized for business, and these computers comprise components that are specifically tailored to the business market. One such component of a computer with Intel vPro technology is Intel AMT, an out-of-band (OOB) management module that is built right into the platform and is used to remotely monitor and fix problems, independent of the operating system (OS). The complete set of Intel AMT features is often divided into three categories: discover, protect, and heal. Most of the well-known and frequently used Intel AMT features have been around since Version 2.0 of Intel vPro technology, which was released in 2006. These features are used by independent software vendors (ISVs) to diagnose and fix computers, thus lowering the total cost of ownership of the platform. For readers just getting to know Intel AMT, an overview of the underlying technology can be found on the Intel Web site at http://www.intel.com/technology/platform-technology/intel-amt/.

To ensure the highest possible security in a business environment, Intel AMT is a signed firmware that can only be updated with another Intel verified firmware. Developers cannot add new features to Intel AMT directly, but, by using the existing functions creatively as software building blocks, developers can build solutions that are well beyond the original intended usage of Intel AMT. These innovative uses increase the value of deploying and activating Intel vPro technology, and expand remote management capabilities to an already large installed base. Table 1 describes four Intel AMT features commonly used today that offer opportunities for innovation. These four Intel AMT features are the focus of this paper.

Extreme Use of Serial-over-LAN

Serial-over-LAN is a virtual communication port that carries data through the network to the administration console, generally at 115 Kb/sec. Intel AMT presents this virtual serial port (that is, a COM: port) to the BIOS and OS, but instead of being connected to a real 9-pin connector, the data are sent over the network to an authorized management console. The intended use of Serial-over-LAN was to allow BIOS vendors to perform text screen redirection in which the BIOS screen could be accessed remotely on a maintenance console. Microsoft Windows* also detects this new serial port, and the appropriate drivers are available from computer manufacturers. Serial-over-LAN is an effective means of communicating with the management console while bypassing the OS network stack. In other words, when data are sent down to the virtual serial port, the Intel AMT network stack sends the data to the management console by way of a Transmission Control Protocol (TCP) connection of its own. Therefore, even if the operating system's network stack is completely disabled, the communication can still take place.

Table 1: Four features of Intel® AMT with opportunities for innovation

Feature	Definition
Serial-over-LAN (SOL)	A virtual communication port that carries data through the network to the administration console. It generally works at 115 Kb/sec. but can be made to work at speeds nearing 1 Mb/sec.
IDE Redirection (IDE-R)	A virtual CD-ROM and floppy device mounted through the network to a computer enabled with Intel® vPro™ technology. Used with remote boot to fix operating-system issues. Compatible software running in the host operating system is monitored by Intel® vPro™ technology, and changes in state can be logged or reported to the administrator.
Agent Presence	Compatible software running in the host operating system is monitored by Intel® vPro™ technology, and changes in state can be logged or reported to the administrator.
Third-Party Data Storage (3PDS)	192 KB of flash memory on the platform can be accessed regardless of the state of the operating system.

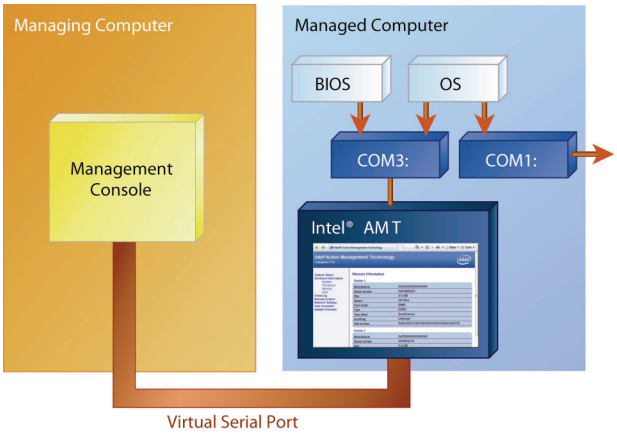

Figure 1: Intel® AMT out-of-band serial port communication (Source: Intel Corporation)

As Figure 1 shows, a user can connect a management console to Intel AMT and use a serial application, such as Putty, on the local computer with Intel AMT. The user types in one screen and the characters are displayed in another screen. Since any binary data can be transported OOB by using Serial-over-LAN, we can build serial agents that are capable of displaying a command prompt to the management console and receiving management commands. When booted into the OS, the management console can communicate to the serial agent, and commands such as start, stop, and list processes can be performed.

Serial-over-LAN also enables binary data to be sent and received simultaneously with the VT100 display and command prompt. A new escape code is used to perform this transaction: this code does not conflict with existing VT100 codes.

A binary request can also be made for the list of processes or device drivers, and a machine-readable response is displayed in a graphical window.

In early 2007, the Developer Tool Kit (DTK) included routing of TCP traffic over the Serial-over-LAN connection, making it possible to perform any TCP connection from the management console to a computer with Intel AMT, even when the network stack on the computer with Intel AMT was completely disabled. An administrator can initiate a remote control session by using Virtual Network Computing (VNC) on a computer with all network drivers disabled. Once the remote control session is open, the administrator can open a command prompt and type IPCONFIG to confirm that no network adapter is enabled. It is also possible to re-enable the driver of the internal Ethernet adapter by using the remote control session that is connected and running on the very same network adapter.

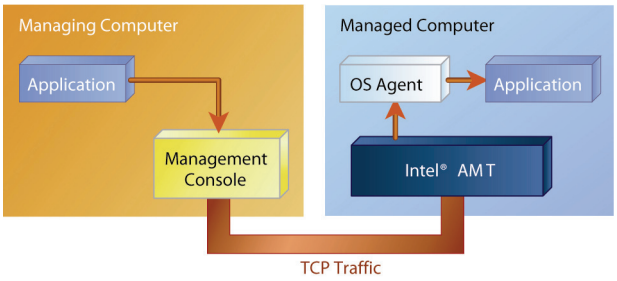

Figure 2: TCP on Serial-over-LAN (Source: Intel Corporation)

As shown in Figure 2, the administrator may connect to management agents running on remote computers, even if the network adapter is disabled or if something else is disrupting the OS network stack. For example, firewall or anti-virus software might not be functioning correctly. Even if Serial-over-LAN allows for bypassing the OS network stack, the computer is still protected because administrators must authenticate their identities, and because of the privacy mechanisms provided by Intel AMT. A Serial-over-LAN connection can only be performed by an authorized administrator.

Extreme Use of IDE Redirection

Let us now shift our focus to the IDE Redirection (IDE-R) feature of Intel AMT. Just as Serial-over-LAN transmits serial port data to and from the administrator, by using IDE-R an administrator can remotely mount a CD-ROM drive on a remote computer with Intel AMT and instruct the computer to boot on that drive. By doing this, an administrator can take over a computer and perform any operation, ranging from a basic boot sector repair to a complete reformatting of the computer with Intel AMT.

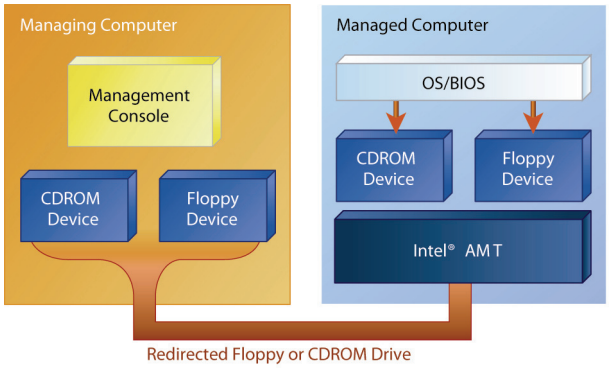

Figure 3: Remote virtual disks using IDE-R (Source: Intel Corporation)

In Figure 3, IDE-R allows an administrator to mount a CD-ROM and floppy remotely, thereby making a set of tools available to the computer with Intel AMT that is having problems. This function works regardless of the state of the OS. IDE-R works by having Intel AMT present a virtual CD-ROM and floppy device to the BIOS and OS. Once this is done, any sector read/write on these devices is redirected over the network to the management console.

A management console enabled with Intel vPro technology can often start IDE-R and reboot a computer all in one step. The console uses IDE-R along with a reboot to boot a diagnostic OS remotely and to fix or reinstall the main OS. This is very useful when the main OS is not booting correctly or needs to be reinstalled. The IDE-R Intel AMT feature can also be used in a different way: when IDE-R is enabled, the local OS can be instructed to rescan installed hardware devices, and by doing this, the OS can find the two new devices, that is, the floppy and CD-ROM. The devices show up with two new drive letters: the CD-ROM device is read-only; the floppy device is read/write. Managers can then use the IDE-R drives that are visible within the OS to send the computer with Intel AMT a large set of patches or new tools within the OS, and the administrator will not have to use the operating system's network stack.

Let us examine one scenario. The administrator starts an IDE-R session and uses a serial agent such as Manageability Outpost to force a rescan of the devices of the computer running Intel vPro technology. The new floppy and CD-ROM devices show up, and the administrator then uses the files on the new devices to patch and fix the local system. This strategy can also be used along with the VNC-over-Serial-over-LAN feature described earlier in this article. The code shown in Figure 4 forces Windows to re-enumerate plug and play devices, thereby allowing the IDE-R devices to show up within Windows Explorer* and other applications:

Intel® HD Boost technology

Intel HD Boost, the combination of SSE4 instructions and the Penryn family of processors' Super Shuffle Engine, can provide large speedups on a wide range of applications. The following instructions in particular can provide significant benefits to video, imaging, and audio applications.

```
[DllImport("cfgmgr32.dll",
ExactSpelling = false, SetLastError =
true)]
static extern int
CM_Locate_DevNode_Ex(ref IntPtr dH,
string ID, uint ulFlg, IntPtr mH);
[DllImport("cfgmgr32.dll",
ExactSpelling = false, SetLastError =
true)]
static extern int
CM_Reenumerate_DevNode_Ex(int dH, uint
ulFlg, IntPtr mH);
private const int
CM_LOCATE_DEVNODE_NORMAL = 0x00000000;

public static int cmdHWRescan()
{
    int resualt = -1;
    IntPtr mHandle = IntPtr.Zero;
    IntPtr deviceRoot = IntPtr.Zero;

    resualt = CM_Locate_DevNode_Ex(ref
deviceRoot, null,
CM_LOCATE_DEVNODE_NORMAL, mHandle);
    if (resualt != CR_SUCCESS) return
resualt;

    resualt =
CM_Reenumerate_DevNode_Ex(deviceRoot.To
Int32(), 0x00000000, mHandle);
    return resualt;
}
```

Figure 4: Sample C# code to force hardware re-enumeration (Source: Intel Corporation)

The MPSADBW and PHMINPOSUW SSE4 instructions can be used to significantly improve motion vector search algorithms (also known as block matching) used in motion estimation for video applications. An Intel whitepaper[2] showcases how to use these two instructions for block matching. The whitepaper reports a 1.6× to 3.8× performance improvement (see (Figure 1)).

One more creative use of the IDE-R feature is that the floppy device is not limited to the 1.44 megabytes of a normal floppy disk. In fact, a floppy image file of up to two gigabytes can be created, making the floppy device much larger than the CD-ROM, which is limited to 700 megabytes. Since the floppy device is both read and write, by using IDE-R and a large .IMG file, an administrator can copy large amounts of data from the Intel AMT system back to the management console at a relatively high speed.

Because Serial-over-LAN is limited to 115 Kb/sec versus the CD-ROM IDE-R speed of 1 times to 4 times greater, IDE-R is more than a powerful Intel AMT feature. When combined with Serial-over-LAN as a control channel, and by using an AMT OOB channel, IDE-R is also a very quick way to transfer data within the OS.

Extreme Uses of Agent Presence

Agent Presence is the third function of Intel AMT that we discuss in this article. Many users first used Intel AMT to query the power state and turn a computer on and off remotely. However, this Intel AMT remote control feature has one important limitation: when a computer is powered on, it can only be abruptly powered off by using Intel AMT. This works fine when the computer is locked up, but if it is not locked up, data can be lost. Intel AMT cannot be used to make a computer go to sleep or hibernate, yet this is a feature that is often required. Since going to sleep and going into hibernation both involve the OS, it is unlikely that Intel AMT will ever be able to perform these operations independent of the OS in the future. In our research, however, we did come up with a partial solution. If a serial agent, such as Manageability Outpost, that is part of the DTK is running, the administrator can send a command to the serial agent instructing it to perform a sleep or hibernate operation. Thus an OOB command is sent to the agent, and the computer can then start saving power. This method works well, but it may be slow when scaling this operation to a large number of computers. Moreover, each agent must have control over the Intel AMT serial port—and only one serial port is available. Here is where a creative use of Agent Presence comes in.

Agent Presence is a feature intended to monitor the running of applications on a computer with the Intel vPro brand. The administrator places a specific Globally Unique Identifier (GUID) and a timeout, 30 seconds for example, into Intel AMT, and local applications signal Intel AMT by using this GUID, about every 15 seconds. If Intel AMT does not receive a signal, also called a heartbeat, in time, it assumes the application is no longer running and notifies the administrator.

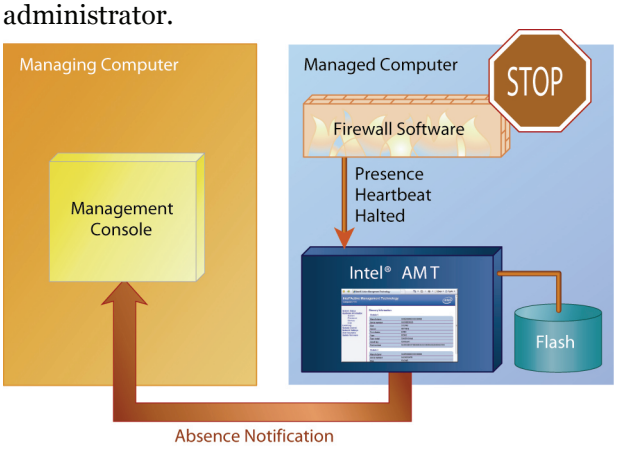

Figure 5: Intel® AMT agent presence and management notification (Source: Intel Corporation)

Video and audio encoding are becoming increasingly important in the world of personal computing. Home-editing of videos and sound recordings are among the popular applications as is standard archiving of DVD material. As shown in Figure 7, the 45nm Intel Core 2 Extreme QX9650 provides a significant boost over previous-generation processors at the same frequency and platform configuration for some of the media-encoding applications. For example, Premiere* Pro CS3 software from Adobe is used to create high-quality visual and editorial effects; it allows users to add color correction, lighting, and other effects such as audio filters and more, with fast, flexible, built-in tools. As shown in Figure 7, the new Qx9650 is 20 percent faster than the Qx6850 in rendering 210 frames to the disk using this Adobe software. Fathom* is an advanced encoding platform product from Inlet Technologies that is used by media companies to encode content for streaming over the Internet or broadcasting over the air. As shown in Figure 7, Intel measures a 23-percent improvement with new 45nm processors for Fathom to transcode 1080i YV12 high-definition video (HDV) to a 1080i VC1 format. Intel measures a 40-percent improvement for Qx9650 over previous-generation

technology for a Pegasys* TMPGenc XPress 4.4 encoder to convert original Variable Bit Rate encoded, 76 second, 29.97fps, 1440×1080 video clips into HDV format MPEG video with 1440×1080 resolution, 29.97fps, and 25000Kb s Constant Bit Rate encoding. Another example is VirtualDub* software, which is a video capture processing utility that uses the DivX* 6.7 software for encoding movies. VirtualDub* 1.7.1 and later with DivX 6.7 are optimized for SSE4 instructions and provide a very noticeable 60-percent performance gain over previous-generation processors that use encoding in SSE2 to convert to the higher-compression DivX format.

In Figure 5 the firewall software is stopped. After a few seconds Intel AMT notices that no signal was received from the firewall: it therefore adds a log entry into flash memory and notifies the administrator. Under normal circumstances, Agent Presence can be a useful tool to ensure computers are compliant with information technology (IT) policies and that they are running proper agent, firewall, and anti-virus software. However, in this enhanced usage, we used Agent Presence to pass a short integer-sized message to an OS agent and to get confirmation of receipt. To use Agent Presence in this way, you have to start by having the agent attempt to signal Intel AMT each 20 seconds by using a GUID that is not set within Intel AMT. Each time the agent attempts to perform the signaling, Intel AMT will report that this GUID does not exist. Under normal operations, the agent would then continue to attempt to signal Intel AMT, by using the GUID that does not exist. In a rare case where the administrator needs to send a short integer-sized message to the Intel vPro brand computer, the management console creates a GUID with a timeout value equal to the message, for example, 201 seconds. In this context, 201 is the number of the short integer message. The next time the agent signals Intel AMT, it will be successful, and the agent will receive a timeout value of 201 indicating what needs to be done. 201 can be designated hibernate, 202, sleep; 203, showdown, and so on. An interesting side effect of this technique is that Intel AMT will report the change in state as soon as the agent signals the correct GUID and obtains the short message. In this way, the administrator is notified of receipt of the short message and can remove the GUID from Intel AMT so the technique can be used again.

In the code shown in Figure 6, we attempt to signal Intel AMT, expecting the signal to fail. If the signal does not fail when it has previously failed, a short message is received from the management console. The GUID used in the sample is the same one used by the DTK.

```
Guid WatchdogNotificationGuid = new
Guid("C0770F68-9174-479b-87A5-
A821FDFEF3C7");
ushort heartbeatTimeout;
AmtCallStatus r =
AgentWatchdogRegister(agentID, out
sessionSequenceNumber, out
heartbeatTimeout);
if ((lastState ==
AmtCallStatus.INVALID_HANDLE ||
    wg.lastState ==
AmtCallStatus.FAILED_WEB_CALL)
    && r == AmtCallStatus.SUCCESS)
{
    // This is a notification, call a
defined event
    if (Notification != null)
Notification(this, heartbeatTimeout);
}
lastState = r;
```

Figure 6: Sample C# notification detection code (Source: Intel Corporation)

This technique allows vendors to create agents, compatible with Intel AMT, that can receive any short signal from an administrative tool that is using Intel AMT OOB, but one that doesn't require the use of the serial port. This creative use of Agent Presence scales better than the solution we described earlier, and its use can benefit software vendors who need to focus on power management.

Extreme Uses of Third-Party Data Storage

We saved the latest and most creative Intel AMT innovation for last. Third-party data storage (3PDS) is 192 KB of platform flash memory that can be read and written, even if a computer is sleeping. Normally when a computer is sleeping, all hard drives and most of the system's memory are off; the 3PDS flash is the only storage space that is still awake and usable. One recreational use of this space could be for a user to save a music file, turn the computer off, and stream the file to a music player while sleeping. However, with only 192 Kb of available space the music would not play for very long. More common usages include storing an installed software list, backup location, or system diagnostic information, useful for computer recovery, into 3PDS. To use 3PDS in this way, a user can build a peer-to-peer file transfer mesh network when a computer with Intel vPro technology is still an active part of the mesh network, even if it is sleeping. Normally, peer-to-

peer networks require all members to be fully powered on, but not in this usage scenario.

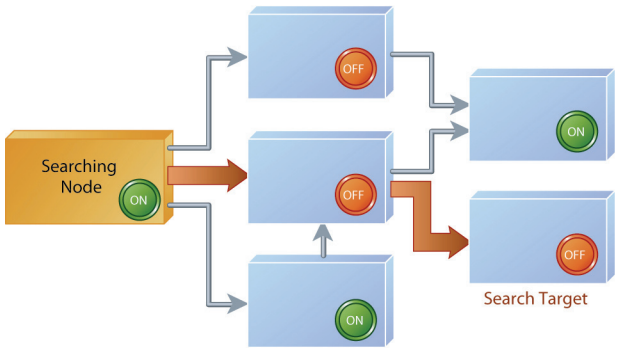

Figure 7: Searching the partially sleeping mesh (Source: Intel Corporation)

Figure 7 shows how Intel AMT can be used to create mesh presence on the network, even when a computer is sleeping. First, mesh agent software is installed on all members of the peer-to-peer network. Each mesh agent stores into 3PDS the list of known mesh network neighbors and also the list of publically available files. Since we have a limited space, only metadata of the files that can be shared publicly are compressed and stored in 3PDS. Each member of the mesh can read other members' lists of neighbors and files, even if the member is reading this information while the computer is sleeping. Software can then be written to discover all of the nodes by iteratively reading the list of peers from each computer. You can also search for a file, and if the file is discovered on a computer, you can wake up the computer to download the file. 3PDS access control allows other members of the mesh to write newly-discovered peer nodes into 3PDS flash memory. To ensure the network is not corrupted by a bad mesh member, locally-checked peers can be separated from remotely-written unverified peers.

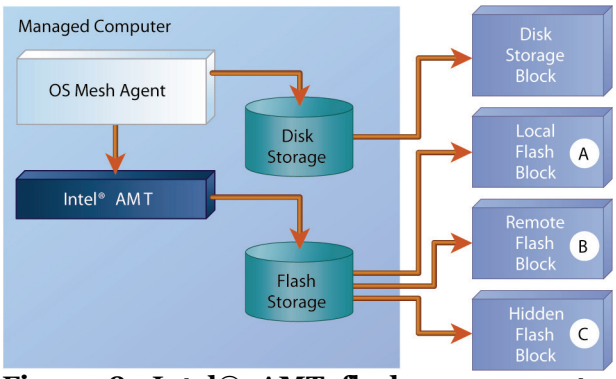

Figure 8: Intel® AMT flash access control (Source: Intel Corporation)

In Figure 8, we separate the flash memory into three distinct blocks: Block A is written by the local agent and is read-only to all other users. This safety measure guarantees that no other nodes tamper with these data. Block B is read/write to everyone and can contain information on new nodes, but it should not be trusted to contain correct information. Block C is visible only to the agent and contains recovery information.

```
// First, add the enterprise entry into
3PDS
AmtEnterprise[] enterprises =
computer.Storage.GetEnterpriseEntries()
;
bool found = false;
foreach (AmtEnterprise enterprise in
enterprises)
{
    if
(enterprise.Name.CompareTo("Intel") ==
0) found = true;
}
if (found == false ||
computer.Storage.AddEnterpriseEntry("In
tel") == null) return false;

// Now, add the mesh application into
prefered partner list
computer.Storage.GetRegisteredApplicati
ons();
AmtStorageAlloc[] allocs =
computer.Storage.GetStorageAllocations(
);
found = false;
foreach (AmtStorageAlloc alloc in
allocs)
{
    if
(alloc.SnrpEntryVendorName.CompareTo("I
ntel") == 0 &&
alloc.SnrpEntryApplicationName.CompareT
o("Mesh") == 0) found = true;
}
if (found == false ||
computer.Storage.AddStorageFpaclEntry("
Intel", "Mesh", 196608) == null) return
false;
```

Figure 9: Initial setup of 3PDS (Source: Intel Corporation)

```
// "localBlock" is a byte[] containing
data to be written

// Log into 3PDS
AmtStorageWrapper storage = new
AmtStorageWrapper(computer, "Intel",
"Intel", "Mesh", new Guid(0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1));
if (storage.Connect() !=
AmtStorageWrapper.AmtStorageStatus.SUCC
ESS) return false;

// Get the list of local blocks &
compute target block size
AmtStorageBlock[] blocks =
storage.GetOwnBlockList();
uint blocksize =
(uint)(localBlock.Length / 4096);
if ((blocksize * 4096) <
localBlock.Length) blocksize += 4096;

// Check if we have an existing block
that is ok
AmtStorageBlock ablock = null;
if (blocks.Length != 1 ||
blocks[0].Size != blocksize)
{
    // Clear all local blocks
    foreach (AmtStorageBlock block in
blocks) block.Remove();

    // Allocate a new block
    storage.AllocateBlock("MeshData",
blocksize, out ablock);

    // Default Permissions

AmtStorageWrapper.AmtStoragePermissions
Group group1, group2;

ablock.AddPermissionGroup(AmtStorageWra
pper.AmtStorageGroupPermission.ReadWrit
e, "ReadWriteGroup", out group1);

ablock.AddPermissionGroup(AmtStorageWra
pper.AmtStorageGroupPermission.ReadOnly
, "ReadOnlyGroup", out group2);

ablock.AddPermissionsGroupMembers(group
1, new uint[] { 0xFFFFFFF1 });
    ablock.SetVisibility(false);
}
else
{
    // Use the existing block
    ablock = blocks[0];
}
```

```
// Write the block
ablock.WriteBlock(localBlock);
```

Figure 10: Locally writing to 3PDS and setting permissions (Source: Intel Corporation)

```
// Get registered Applications
AmtApplication[] apps =
computer.Storage.GetRegisteredApplicati
ons();
foreach (AmtApplication app in apps)
{
    if
(app.EnterpriseName.CompareTo("Intel")
== 0 &&
app.VendorName.CompareTo("Intel") == 0
&&
app.ApplicationName.CompareTo("Mesh")
== 0)
    {
        if (app.UUID.CompareTo(new
Guid(0, 0, 0, 0, 0, 0, 0, 0, 0, 1))
== 0)
        {
            // This is a local block
            AmtStorageBlock[] lblocks =
computer.Storage.GetStorageBlocks(app);
            if (lblocks != null &&
lblocks.Length > 0) localBlock =
lblocks[0].ReadBlock();
        }
        else if (app.UUID.CompareTo(new
Guid(0, 0, 0, 0, 0, 0, 0, 0, 0, 0))
== 0)
        {
            // This is a remote block
            AmtStorageBlock[] rblocks =
computer.Storage.GetStorageBlocks(app);
            if (rblocks != null &&
rblocks.Length > 0) remoteBlock =
rblocks[0].ReadBlock();
        }
    }
}
```

Figure 11: Remotely reading the 3PDS local and remote blocks (Source: Intel Corporation)

This is what Figures 9, 10, and 11 show: a sample setup, local Block A write with permission setup, and remote read of data using the C# DTK stack. Figure 9 shows how 3PDS must be set up before it can ever be used; this setup must be completed remotely. Figure 10 shows data that are written locally into 3PDS Block A, and Figure 11 reads both Block A and Block B from 3PDS.

Another interesting application of this mesh network concept is a new way to provision Intel AMT when each computer is self-provisioning. First, add a TCP reflector into each mesh node. This allows any computer to use the TCP reflector to connect to itself and provision itself. You can then distribute throughout the mesh the public portion of a trusted administrator certificate. Each computer can encrypt its own Intel AMT administrator account password with this administrator certificate and store the encrypted result into its own 3PDS flash memory.

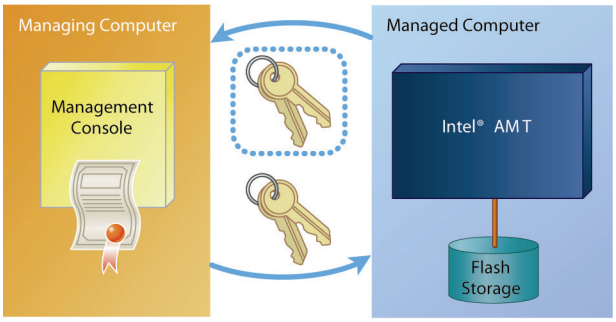

Figure 12: Certificate-protected Intel® AMT password (Source: Intel Corporation)

Figure 12 shows how each member of the mesh network can read the encrypted administrator password of each of the computers in the mesh, but only the administrator with the private key for the certificate can decrypt the Intel AMT administrator password of each computer.

In a model in which each computer administers its own Intel AMT, it is possible for the OS to be wiped out and re-installed, causing the loss of the local copy of the Intel AMT password. To prevent this from happening, 3PDS can be used in one more creative and useful way—that is, by inserting a 3PDS hidden recovery block (Block C in Figure 8). This block is only visible locally, and the mesh agent stores the Intel AMT administrator password and other general mesh information in this recovery block. If the agent is reinstalled after a complete disk replacement and a reinstalled OS, the agent can resume where it previously left off. Such a 3PDS recovery block has many other uses also.

Depending on how computers are meshed, peer-to-peer mesh networks can be scaled to work with small business networks with a few computers as well as large-scale enterprises with thousands of computers. It is worth noting that Intel AMT cannot handle many concurrent connections, so thorough testing is required before peer-to-peer networks can be implemented in larger networks. Many examples of highly scalable mesh networks are deployed on the Internet, but it requires a lot of development and testing for such solutions to work correctly.

Conclusion

While many people at Intel are working on new features for upcoming platforms based on Intel vPro technology, a larger impact can sometimes be felt from users and developers finding innovative new uses for existing and already deployed platforms. This path to innovation is available to everyone inside and outside of Intel; new usages can be deployed quickly, and these usages allow management software differentiation in a competitive market. Whether it is with Intel AMT or with other technology, significant value can be derived from fostering research and innovation on existing as well as on new technology.

Acknowledgements

I thank the technical reviewers of this article for their valuable input. Special thanks go to Sandeep Siroya, Ajay Mungara, and the Intel software community for their many contributions to the Manageability Developer Tool Kit (DTK) over the last two years.

References

[1] "Intel® Active Management Technology." The official Intel Web site. http://www.intel.com/technology/platform-technology/intel-amt/

[2] "Manageability Developer Tool Kit (DTK)." A complete set of freely available Intel® AMT tools and source code. http://www.intel.com/software/amt-dtk

[3] "Intel® AMT Software Developer Kit (SDK)." The official and complete reference for developers of Intel AMT. http://www.intel.com/software/amt-sdk

[4] "Intel® AMT Open Source Drivers and Tools." The official open source site for Intel AMT. http://www.openamt.org/

Author Biography

Ylian Saint-Hilaire is a senior architect at Intel. He is part of a research group within the Intel Software & Services Group (SSG) and currently works on network manageability and services. Recognized as an innovator and public speaker, he was awarded two Intel Achievement Awards for past work enabling the digital home. In his own time, Ylian is a pilot and enjoys flying around the Portland area and traveling to foreign countries.