



Intel® Active Management Technology Validation Design Guide

Version 3.0.1, December 2006

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppels or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

The API and software may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Copyright © 2005, 2006, 2007 Intel Corporation.

* Third party other names and brands may be claimed as the property of others.

1 Introduction

This document is intended for ISV Validation Architects/Designers that are building infrastructure and test plans to validate an application that interacts with Intel® Active Management Technology (Intel® AMT) devices.

The Intel AMT manageability features covered in this document are grouped into the following functional categories: Setup and Configuration, ISV Storage, Non-Volatile (NV) Log/Event management, Remote Management, Hardware Asset Management, and Redirection.

Applications running on remote platforms may access Intel AMT features even when the operating system on the local platform is inoperable, provided that:

- the Intel AMT enabled platform is connected to a power source,
- the platform is in a power state that allows AMT to work,
- the Intel AMT enabled platform is connected to a network, and
- the network is configured according to Intel AMT requirements.

For example, the features can still be used when the platform is in standby, hibernate, or power-down modes.

1.1 Setup and Configuration Overview

Setup and Configuration is the procedure used to populate an Intel AMT device with usernames, passwords, network parameters, Transport Layer Security (TLS) certificates and keys necessary for encrypted communications and Kerberos security credentials. The setup and configuration capability in an ISV application allows the IT department to configure the Intel AMT system for later management by the ISV application.

An Intel AMT enabled device has two modes of operation: Enterprise mode and Small Business (SMB) mode. Platforms with Intel AMT Release 2.0 or greater can be configured to run in legacy mode and are then setup like AMT Release 1.0 machines.

An Original Equipment Manufacturer (OEM) sets the appropriate mode value in their factory when they build the flash image. The Enterprise Setup and Configuration mode is designed to serve the needs of large enterprises. When supported with the proper network infrastructure services, this mode can provide automated one-touch Setup and Configuration for Intel AMT-enabled platforms.

For Intel AMT Release 1.0 machines and machines with Intel AMT Release 2.0 or greater running in legacy mode, it is strongly recommended setup and configuration be performed in an isolated network. The ISV should also test setup and configuration in a normal network environment, with and without TLS, with and without TLS mutual authentication, and with and without Kerberos authentication to validate proper functionality. Note that TLS mutual authentication and Kerberos authentication are supported only by platforms with Intel AMT Release 2.0 or greater.

1.2 ISV Storage Overview

Intel AMT provides management applications the ability to store critical data in a non-volatile (Flash) data storage area. The data storage area can be accessed in a controlled way by other applications running on the same platform or on a remote platform. Intel AMT will provide local applications access to the data storage area through a local host interface. Remote applications can access to the data storage area through an Out-of-Band (OOB) SOAP-based network interface.

Basic functions of the Storage Manager include:

- Allocate storage space to applications.
- Allow applications to grant other applications write/read permissions on their data (using Permission Groups).
- Require applications to Register.
- Protect the space allocated to one application from other applications.
- Support a set of Security Mechanisms to protect the commands issued by applications.
- Manage the flash memory used by third-party applications.
- Provide access to the flash memory regardless of system state (as long as the system is connected to a power source and to the network).
- Storage Administration: The Intel AMT Storage Administrative Interface enables administrators to reconfigure the global parameters that govern allocation and use of third-party non-volatile storage. The ISV application should allow administrators to manipulate these parameters. In addition, since IT departments are expected to deploy multiple applications that utilize Intel AMT, the application should assume that another application may have changed these storage administration parameters. The application should handle this case gracefully.

1.3 Non-Volatile Log/Event Management Overview

Intel AMT Remote Management provides always-available remote access to event logs of: PC sensors, effectors, and other non-volatile data. In addition, Intel AMT has the ability to generate out-of-band alerts (PETs), based on sensor events.

1.4 Remote Management Overview

Remote management capabilities are used to remotely monitor and control an Intel AMT enabled platform, either for remote diagnosis and repair or as a component of a greater remote management solution.

1.5 HW Asset Manager

Intel AMT will allow remote Management Consoles to query hardware asset information collected from the PC BIOS and store that hardware asset data for access when the system BIOS is inoperative.

1.6 Redirection Overview

Intel AMT allows Management Consoles to receive text and provide keyboard data from and to the managed machine. This capability is referred to as “Serial-over-LAN” or SOL. SOL allows the remote operator to see text that was sent to the serial controller of the managed (Intel AMT-enabled) machine, and to type commands that will be received at the managed machine as if coming from the serial controller.

Further, the Intel AMT enabled device can function as an IDE device on the managed machine, with the actual CD, floppy or CD-image data residing on the Management Console’s machine. This capability is called IDE Redirection and is often used with Serial Over LAN (SOL) capability.

The introduction provided in Chapter 1 is meant to be cursory only. For additional details about Intel AMT, see the Intel AMT Users’ Guide, also provided in this SDK.

2 Scope

The physical Intel AMT enabled device can be tested using the various platforms that support Intel AMT. Please refer to the platform matrix in the user guide for more details. This document describes how validation engineers will use the Intel AMT enabled device.

3 Entities

3.1 Required Tools and Debug Features

The following documents, tools and debug features are needed to understand and validate Independent Software Vendor (ISV) Applications that implement Intel AMT

3.1.1 Documents (found in this SDK)

- Intel AMT Overview
- SDK User Guide
- Validation Design Guide (this document)
- ISV Coexistence Guidelines
- Storage Design Guide
- Network Interface Guide
- Small Business Configuration User Guide
- Developers Guide to the Sample Setup and Configuration Application Guide
- Intel AMT Integration With Active Directory
- Host Information Design Guide
- Redirection Library Design Guide
- System Defense and Agent Presence Overview.

3.1.2 Tools, Applications, Code

- Intel AMT Sample Code provided in the SDK.
- Intel AMT ISV Storage library provided in the SDK.
- Intel AMT redirection library provided in the SDK
- API Test script provided in the SDK.

3.2 Required Hardware / Software / OS Environment

Refer to the Intel AMT User Guide, “System Requirements”, for system requirements.

Validation may be completed using a Software Development Platform (SDP). The Management Console can run any supported operating system, including Windows and Linux.

4 Schematic of Test Environment

The following schematics detail the infrastructure required for validating software applications implementing the Intel AMT API.

4.1 Test Scenarios Overview

In each scenario, the system containing Intel AMT may be a Software Development Platform or other platform containing an Intel AMT device.

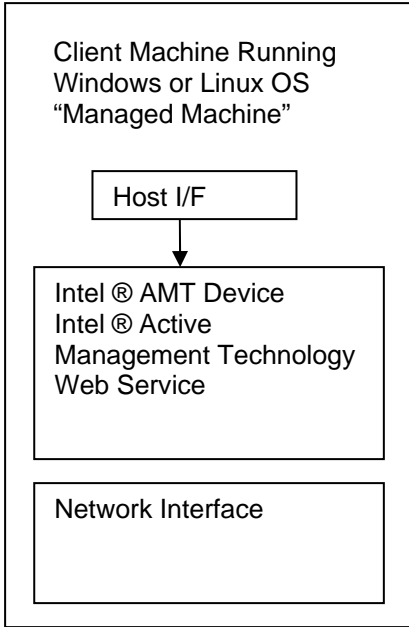
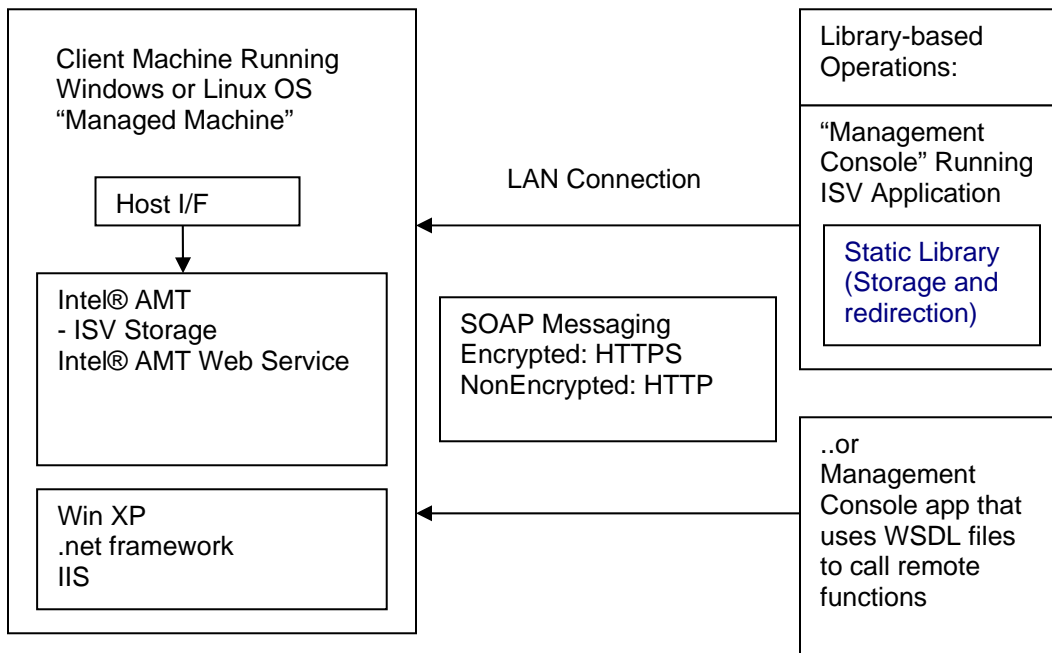


Figure 1: System Diagram including Intel AMT Software Development Platform (SDP)

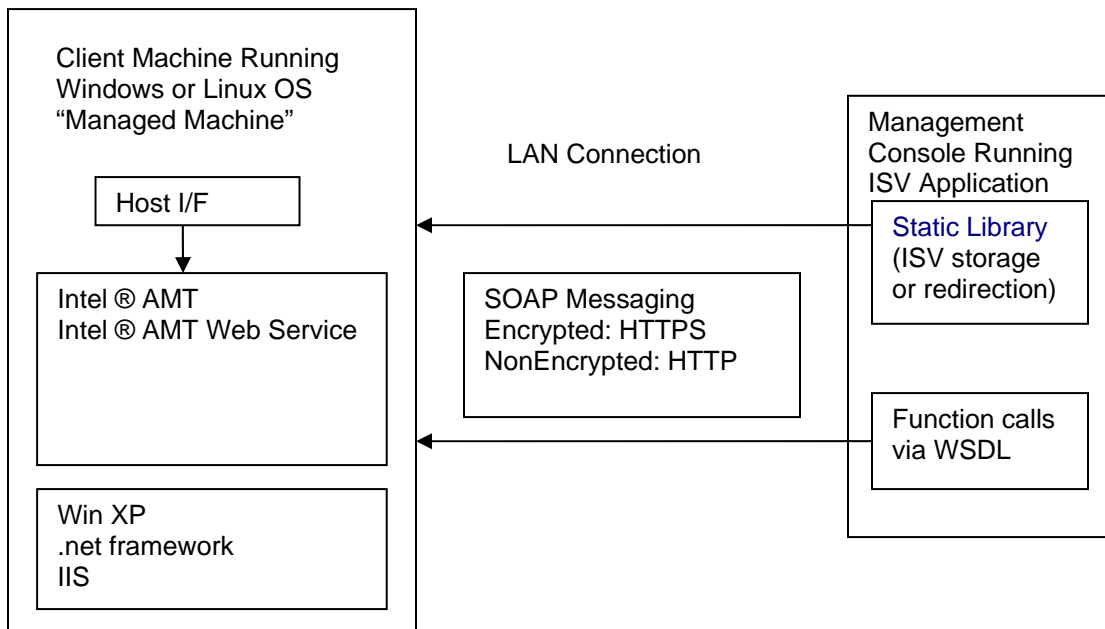
4.2 Test Scenario 1: A Single Management Console running against the system containing Intel AMT

In this scenario, one machine is configured as the Management Console running the ISV Application (for example, the Management Console software), and the second machine is configured as the managed client (called "Managed Machine". Note: the static libraries are the only way to access the storage manager and redirection capabilities, and the WSDL files are the only way to access the setup and configuration, asset, remote control, and event managers.



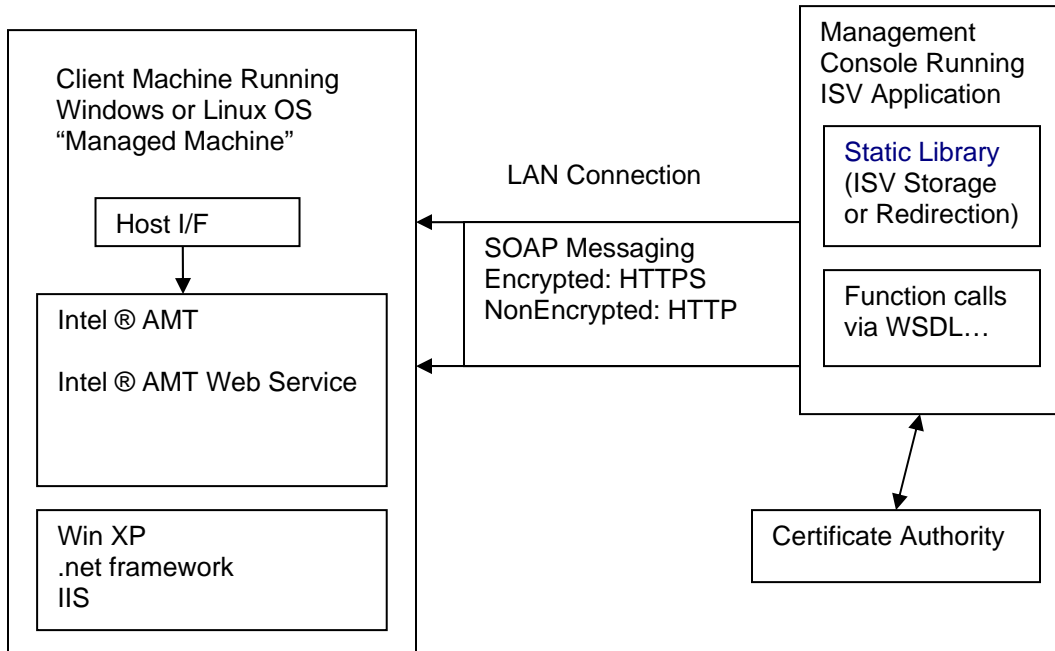
4.3 Test Scenario 2: Multiple Management Consoles running against a system offering Intel AMT

It should be noted that more than one ISV application may access the Intel AMT features at a time. This schematic illustrates that more than one application may be active at a time, and that such situations require validation. The number of concurrent sessions is limited by design. The application should check for related errors (for example, attempt to create a session when the max number has already been reached) and handle them gracefully.



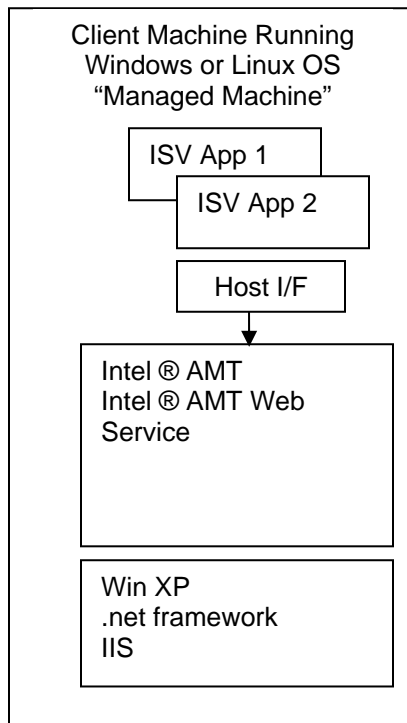
4.4 Test Scenario 3: Validation of Encrypted Operations

Encrypted communication with the Intel AMT device should be tested with devices configured with TLS = enabled option. In some security configurations, this may require the addition of a Certificate Authority subsystem which will validate the certificate presented by the Intel AMT device.



4.5 Test Scenario 4: Stand Alone Validation

ISV Applications can be loaded on the same machine containing an Intel AMT device. Note: this scenario only applies to NV Storage, since that feature is available locally through the KCS driver. Other features are accessible through the network interface as shown in Test Scenarios 1-3.



4.6 Test Scenario 5: Combined validation

Combination of Test Scenario 2 and 4: Multiple Management Consoles running against the Intel AMT platform and Stand-alone Validation. In this scenario, remote Management Consoles access the firmware at the same time that the Intel AMT platform operates in stand-alone mode.

The number of concurrent sessions is limited by design, which reduces the number of combinations that need to be tested. Validation testing should also exceed each of these limits at least once to ensure proper handling of the fault condition.

4.6.1 Concurrent Session Limits

Following is a list of the limits on concurrent sessions to a given Intel AMT-enabled system.

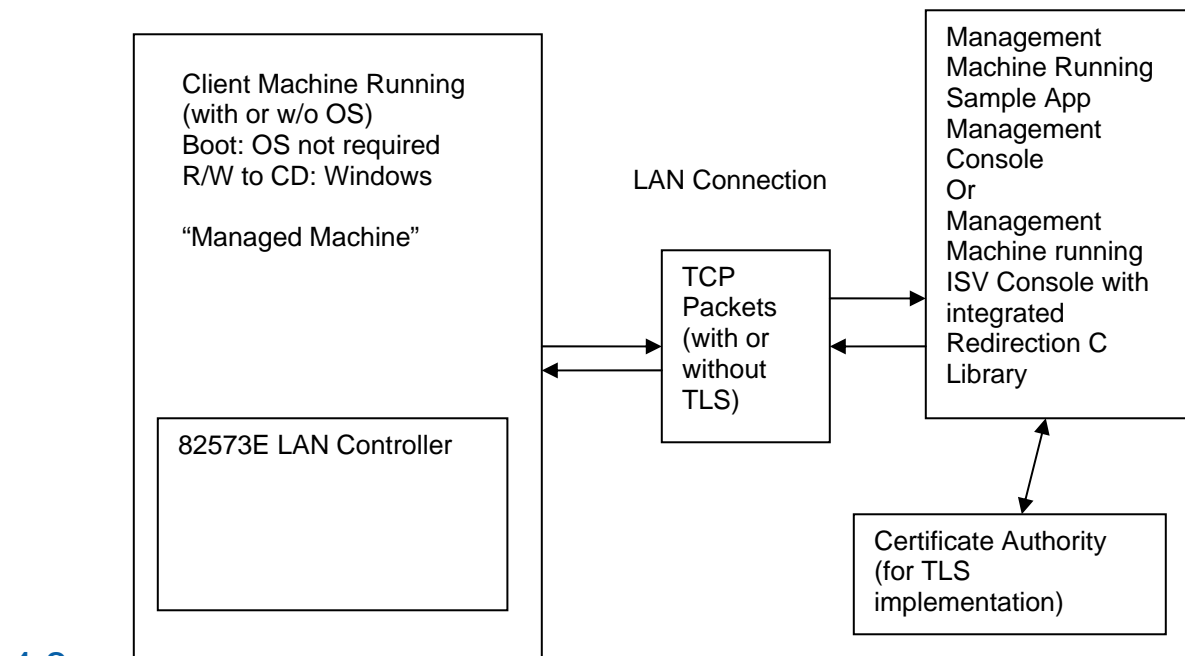
Intel AMT Release 1.0:

- A maximum of two concurrent SOAP connections.
- A maximum of one SOL and one IDER session can be opened concurrently.
- One remote connection to the ISV storage area at a given time.

Intel AMT Release 2.0:

- 18 sockets are available for the Web user interface, SOAP connections, and access to ISV storage. These sockets can be used for both local and remote applications.
- Local connections require two sockets each. Also, there is a limit of eight local connections, using 16 sockets, so that there will always be two available for remote connections.
- There can be multiple connections to ISV storage within the above defined limits.
- In addition to the above, there can also be one SOL session and one IDER session opened concurrently.

4.7 Test Scenario 6: Redirection (SOL / IDE-R)



4.8

4.9 SOAP Interface

It is important to note that the actual contents of the SOAP packets passed between the SDK library and the Intel AMT device consist of binary data that is only meaningful to the SDK and storage device.

4.10 Hardware Limitation

The Intel AMT Storage Design Guide cautions that Flash memory has intrinsic limitations concerning the number of write cycles that can occur before a particular memory location becomes unusable. The number of write cycles possible is determined by each Flash vendor's implementation, but is generally in the range of ~100K cycles.

During testing Validation engineers should be cognizant of this limitation, and ensure that regression testing does not exceed manufacturer's limits, thus rendering the flash unusable.

Production systems using Intel AMT include a flash erase wear-out monitor that prevents a high frequency of erase operations in an effort to ensure the product lifetime. The storage manager monitors the write operations, and actually allows additional writes after time has passed. Specifically, after each 40 minute period, the number of remaining writes allowed increases by one. If the write monitor indicates that no write cycles are available, the ISV application should handle this message and check again after 40 minutes or longer. As some operations require more than one write, it may be necessary to wait multiple periods of 40 minutes.

5 Setup and Configuration

Setup and configuration is the procedure through which a machine with Intel AMT is populated with usernames, passwords, network parameters, Transport Layer Security (TLS) certificates and keys, and Kerberos security credentials necessary for encrypted communications. It is a good idea for each ISV to provide setup and configuration capability to the IT department, allowing all these parameters to be set.

Intel AMT Device has two modes of operations: Enterprise and Small Business. An OEM sets the appropriate mode value in their factory when they build the flash image. The Enterprise setup and configuration mode is designed to serve the needs of large enterprises. Devices running Intel AMT Release 2.0 or greater can be set work in legacy mode, whereby they operate like exactly like Intel AMT Release 1.0 devices (this statement holds for setup and configuration but is not strictly true for other areas of operation). When supported with the proper network infrastructure services, this mode can provide automated one-touch setup and configuration for Intel AMT platforms as shown in Figure 1.

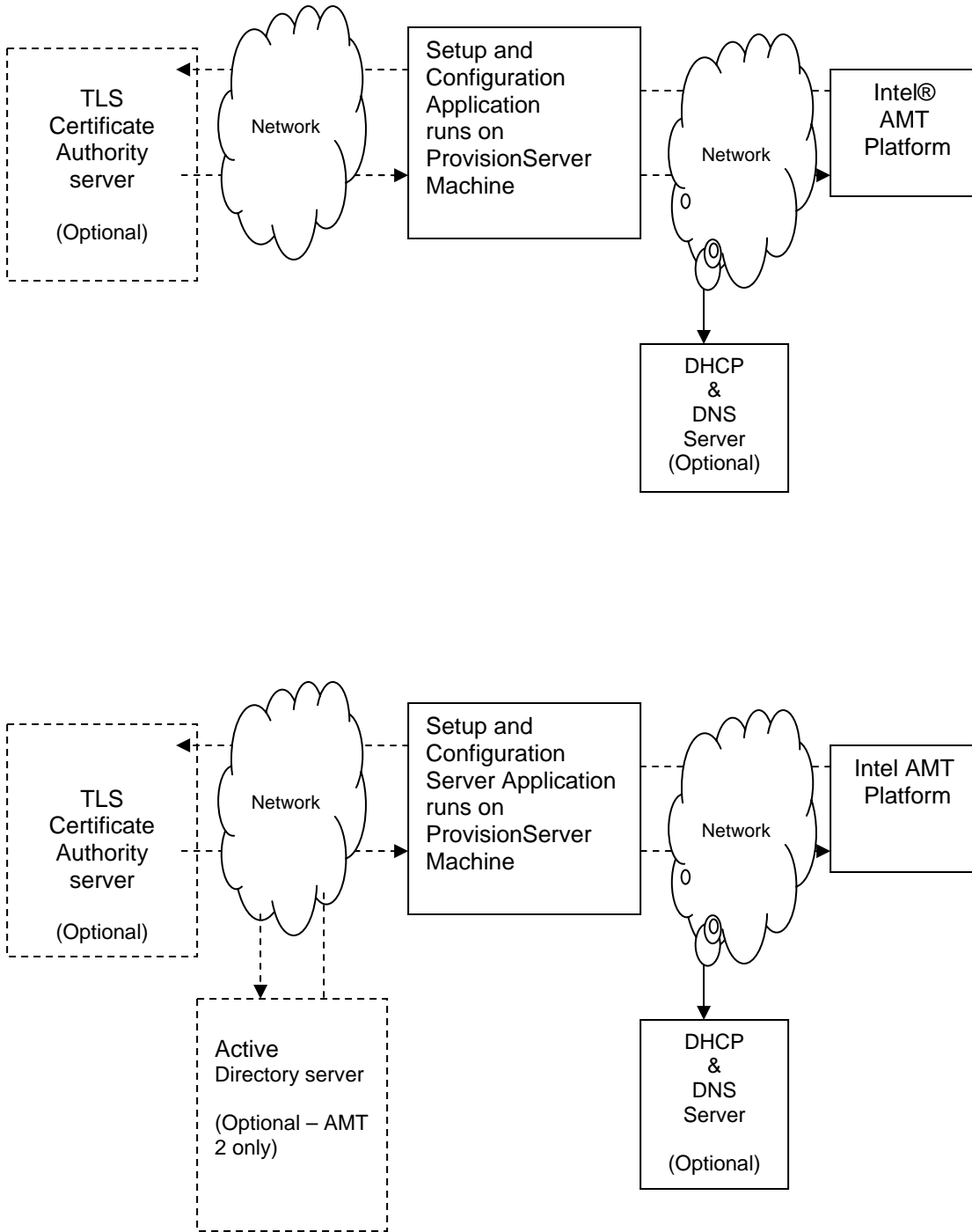


Figure 3: Enterprise Setup and Configuration Flow

For Intel AMT Release 1.0 machines and later releases running in legacy mode, it is strongly recommended that setup and configuration be done on an isolated network. The ISV should also test setup and configuration in a normal network environment, with and without TLS, with and without TLS mutual authentication, and with and without Kerberos authentication to validate proper functionality. Note that TLS mutual authentication and Kerberos authentication are supported only by Intel AMT Release 2.0 or greater.

Please see the Developers Guide to the Sample Setup and Configuration Application and Small Business Setup and Configuration Guide for more details on the setup and configuration process.

After testing setup and configuration, also returning an Intel AMT device to Factory Setup Mode to ensure expected behavior and ensure the application behaves properly if it attempts to communicate with an unconfigured system.

5.1 Security and Encryption

Install a Certificate Authority service, configure Intel AMT to function over TLS and repeat all tests. Also test application behavior with incorrect credentials to ensure the application handles the error correctly.

6 ISV Storage

This section contains several suggested tests for ensuring proper functioning of the Intel AMT features in the management applications using ISV Storage functions. It is limited by the fact that each ISV will implement the technology in its own way.

6.1 Validation Categories – ISV Storage

6.1.1 Basic Aliveness

Run APITest -v to check that the SDK is functional and installed with the correct version. See the Intel AMT Storage Design Guide (provided in SDK) for additional information. The API Reference Chapter contains all the APIs related to this function.

6.1.2 Writing Blocks / Reading Blocks

Test cases should be created that write information to a storage block using Application A. Ensure that Application A can read the block's information. Be sure that the first instance adds the permissions group filter before terminating. Terminate Application A and launch it from another machine with a different UUID. The first instance should add a generic application filter, thereby granting permissions to all applications sharing the same names, regardless of UUID. Ensure that the block can be written to and read from.

Add Application A and B to the permissions group for a particular block. Ensure that the same information can be read both by Application A and B. Modify the information using Application B. Ensure that the correct information is read by both applications. Experiment with variations of local versus remote access to the block.

Stress and Load Test cases should be written that write large amounts of information, up to or exceeding permissible limits (see Intel AMT Storage Design Guide for those limitations).

See the Intel AMT Storage Design Guide, API Reference Chapter, Data Storage section for the related APIs.

6.1.3 Locking / Unlocking

Application A locks a block. During this time, Application B (which has permission to access the block) attempts both a read and a write, but should be prevented from doing so. When Application A unlocks its block, Application B should be able to read and write to that block. Applications should respond appropriately while being “locked out”.

See the Intel AMT Storage Design Guide, API Reference Chapter, Data Storage section for the related APIs.

See the Intel AMT Storage Design Guide, API Reference Chapter, Data Storage section for the related APIs.

6.1.4 Building / Reading / Updating Permissions Groups

Depending upon the ISV-specific implementation, test cases should be written to ensure that Permissions Groups and Permissions Members are correctly managed by the application. Note: the un-supported CLI provided with the SDK may be useful in co-validating permissions. Also try using the application and vendor name filters as members of a permission group, ensure that failure cases are handles by the application.

See the Intel AMT Storage Design Guide, API Reference Chapter, Permissions section for the related APIs.

6.1.5 Removing and Unregistering the application

Refer to the design documentation for the maximum number of open sessions permissible, and the timeout duration before inactive sessions are de-registered automatically.

Create test cases that ensure that applications un-register and free-up all memory when complete.

Create test cases that ensure that the application(s) do not expect to open sessions in excess of the maximum number permitted, and that handle related errors gracefully.

See the Intel AMT Storage Design Guide, API Reference Chapter, Initialization and Registration section for the related APIs.

6.1.6 Erase-cycles

As mentioned above, Flash memory contains intrinsic limitations on the number of erase cycles possible before the flash sector becomes unusable. Validate that the application does not perform a larger number of storage operations than necessary in order to achieve the required functionality. Ensure that the application detects when the flash device is reaching its erase cycle limit, using the GetBlockWriteEraseLimit API and responds accordingly.

6.1.7 Negative Testing and Error Trapping

Ensure that applications deal with the various error states returned by the API. Ensure that the ISV application checks the return status of API functions and reacts accordingly. For example, Attempt to exceed the maximum limit of simultaneously registered applications and ensure that the ISV application reacts appropriately.

Attempt to access an ISV Storage block after Deleting that block (as with the ISVS_RemoveApplication), with the RemoveApplication CLI (included in the SDK) or with ISVS_DeallocateBlock command

Attempt to use `ISVS_AddPermissionsGroupMembers` to add members exceeding the maximum specified in the SDK.

Attempt to access a storage block without the application appearing in the permissions group for that block

Ensure that session timeout is properly handled. After inactivity an application needs to reregister in order to continue to access storage blocks. The user can call `ISVS_GetTimeoutValues` in order to learn the timeout value of a locked block.

Ensure that locked blocks are properly dealt with after the timeout has expired. The SDK permits the deallocation of a block by its owner even when it is locked by a second application accessing it. Deallocate a locked block and verify that the second application properly processes the status code: `PT_STATUS_BLOCK_DOES_NOT_EXIST`.

7 Network Interface

7.1 Network Environments

Tests should be completed in various network and system settings, including static and dynamic IP addresses. DHCP enabled or disabled, VLAN on or off. Many of these settings are configured in pre-Setup and Configuration.

See the Intel AMT Network Interface Guide, API Reference Chapter, for detailed APIs for relevant features.

7.2 Security

7.2.1 User Access Lists

The primary purpose of the first set of Intel AMT security functions are to create a list of authorized users who can access an Intel AMT device, managing User IDs and Passwords. They essentially determine which system administrators are permitted read/write access to the Intel AMT Device. Intel AMT Functions implemented in ISV Applications relevant in this area are located in the Network Interface Guide, API Reference chapter, Security Administration Interface section.

ISV Validation efforts should ensure that these functions have been implemented properly by adding, changing and deleting users and/or passwords, and attempting to access Intel AMT functions for Asset Management, Event management and Remote control. They should also handle the case in which an IT representative may have changed these entries without the ISV application knowledge. For example, if the IT representative removed the application name or enterprise name from the list. Or, if the access control list is already full when the ISV attempts to add an entry. Also ensure the application handles incorrect credentials correctly.

7.2.2 Enabling TLS

ISV validation should ensure that the application implementing Intel AMT is able to enable encrypted (TLS) operation and non-encrypted (non-TLS) operations. Validation should include Setup and Configuration of the Intel AMT device with TLS enabled and making sure it accepts only encrypted communication (e.g. access through https only while rejecting access through http).

Intel AMT Functions implemented in ISV Applications relevant in this area are in the Intel AMT SDK User Guide, Network Interface Guide Chapter, API Reference, Security Administration Interface section.

7.2.3 Proper integration with Authentication/Security Certificates

Validate that the ISV Application is able to function when certificates are installed. Also check that the application functions correctly if the certificate is deemed invalid by the OS.

7.2.4 Removing Intel AMT Settings

ISVs should validate that they have correctly implemented the Intel AMT feature for removing all data and settings from the Intel AMT device, as for example must occur when a computer is being sold or discarded.

The relevant command, Unprovision, is found in the Intel AMT SDK User Guide, Network Interface Guide, API Reference Chapter, Security Administration Interface section.

The Unprovision command resets the Intel AMT device to default factory settings, except for the FPAcl entries which are left intact. The device will need to be re-configured after issuing this command.

7.3 Remote Control

ISV Validation should ensure that the application correctly obtains managed-client remote control capabilities, and is able to perform remote operations such as power-up, power-cycle, etc. In addition, when the Intel AMT enabled device is installed, validate that the ISV application can cause a boot with ASF Options (PXE boot, etc.).

See the Intel AMT Network Interface Guide, API Reference Chapter, Remote Control section for the related APIs.

7.4 Non-Volatile Log / Event Manager

General Description

The NV Log / Event manager functions permit the ISV application to set up filters on the Intel AMT device using the AddEventFilter and the UpdateEventFilter functions that determine the action that Intel AMT takes when platform events occur.

Events are typified by the FilterConfiguration structure (see the Network Interface Guide) and include such attributes as Event Type and severity. The four possible Intel AMT actions are:

- log the event to the non-volatile Event Log
- create a PET (Platform Event Trap) message and send it to subscribing consoles
- both 1 and 2 above (both log the event in non-volatile memory and create and send a PET)
- None of the above – no PET message and no log message.

See the Intel AMT Network Interface Guide, API Reference Chapter, Event Manager Interface section for the related APIs.

7.4.1 Application Flow and Validation Steps

If no filter is in place for a given event, that event is disregarded, and no action is taken by Intel AMT. The basic flow for working with and validating filters is:

- Build filters according to the field or fields of interest.
- Build subscriptions with the IP address of the PET collector console and associate them to a filter.
- Create events matching/not matching the filter

Create Events:

- Using actual environmental events on an Intel AMT platform such as chassis-intrusion or other sensor events, a link up event, or a password attack.

Then, based on Filters / subscriptions, validate that a matching event has:

- Been sent as a PET (SNMP packet) and received by PET collectors.
- Been logged in the event log.
- both 1 and 2 above (where appropriate)
- No PET and no event log entry.

During validation with the Intel AMT enabled device present, the system must be configured with:

- “Legacy” sensors
- An SMBus connected to one or more sensors and to the Intel AMT enabled device.

Please see the *Network Interface Guide* in the SDK for details about the Event manager functions.

7.5 Asset Management

Intel AMT provides the ability to read Asset management information recorded in BIOS. In order to save time, eliminate redundancies, and reduce network activity, the information is transferred to Intel AMT in binary format and must be parsed by the ISV application.

Validation tasks include ensuring that the Asset information has been received and interpreted correctly by the application. Repeat steps for differently configured machines from different vendors.

See the Intel AMT Network Interface Guide, API Reference Chapter, Hardware Asset Interface section for the related APIs.

7.6 Redirection (Serial-over-LAN “SOL” / IDE-Redirection “IDE-R)

The Intel AMT SDK contains a SOL/IDE-R sample app which can be used by validation engineers to view the generic functioning of the SOL/IDE-R features. The SDK also includes a SOL/IDE-R Library used by ISV developers to implement these features in ISV Applications and Management Consoles.

The most common development issue which requires validation attention relates to releasing resources and ensuring that memory leaks do not occur. Validation engineers should verify that the SOL or IDE-R session is closed in a timely fashion after utilization. This is easily validated since only one SOL session and only one IDE-R session can be opened against a given client at one time. If the session is left open, a second Management Console instance will receive a client busy return code, and will not succeed in opening a session. System Memory should be monitored at validation time to ensure proper use of the library by noting that progressive use of the IDE-R and SOL features does not lead to memory leaks.

Client Management: Ensure that clients are enumerated correctly, and that the list of clients can be added to, purged, etc, and that the Management Console database reflects the information received from the Redirection SDK.

See the Intel AMT Network Redirection Library Design Guide, C Library Chapter, Library API section for the related APIs.

7.6.1 SOL

Unlike with IDE-R, the Management Console has complete control over redirection. It decides when the client will send or receive text. Ensure that the Management Console operation does not slow, corrupt or block desired transfers of text within its logic. This can be validated by noting that the flow of text to the client is smooth and rapid, as would be expected in normal, local operation of a client, and that text appears (when desired) on the Management Console in the same manner as would be seen on the client.

7.6.2 IDE-R

Use Model: The Management Console opens a session, and then either the client manually initiates remote boot or remote file access (Read only for CD / CD Images, RW for Floppy), or the Management Console agent on the client automatically initiates remote boot or access. Finally, the Management Console deactivates the device and closes the session.

Intended behavior is that the Management Console will not leave the redirected drive active, except when remote booting / file access is desired. Ensure that the user does not “see” an active remote drive except under the desired circumstances (accomplished in the SDK with “IMR_IDERSetDeviceState”.

Please see the Redirection Design Guide for a categorized list of functions. The functions are organized by General Library Usage, Client Management, SOL, and IDE-R.

7.6.3 System Defense and Agent Presence

System Defense and Agent Presence are security toolsets built-in into Intel AMT. The System Defense and Agent Presence toolsets are targeted at closing two gaps in the IDS (Intrusion Detection Systems) methods currently employed by IT:

- A time window between the identification of an OS/Agent vulnerability and completed deployment of corresponding patch.
- End-user tampering with IDS agents.

The System Defense toolset enables a Management Console application to define and enforce network security policies. System Defense policies consist of sets of networking packet transmit and receive filters. System Defense policies are placed on an Intel AMT device by a Management Console application. Once a System Defense policy is activated, the Intel AMT device inspects each incoming and outgoing packet and performs the necessary action specified in the policy.

The Agent Presence toolset enables Management Console applications to configure Intel AMT devices to monitor for the presence of software agents (e.g. Anti-Virus, Firewalls, etc.) running on the Intel AMT system platform. The Management Console application configures the Intel AMT device with timers set to detect when the software agent initializes and periodically transmits "heartbeat" signals. If any of the timers expire, Agent Presence will perform an action. Possible actions include one or all of the following:

- Activate a pre-programmed System Defense policy which contains Network Isolation filters
- Send an SNMP PET alert
- Log the event to the local Intel AMT event log

Validation tasks include ensuring that the System Defense filters and polices can be defined and activate and have the expected results and that agents can be registered and report in using the local agent presence interface and the expected actions are triggered. Repeat steps for differently configured machines from different vendors.

See the Intel AMT Network Interface Guide, API Reference Chapter, System Defense and Agent Presence sections for the relevant APIs, see provided sample code for usage examples.

8 Use Cases

Use cases describe the expected scenarios in which and end user (IT professional) would use the Intel AMT platform. It is recommended to test as many of these as apply to the ISV software functionality. Following are high level descriptions of the expected use cases:

1. Using an asset management app that supports Intel AMT, the IT professional discovers and inventories all Intel AMT-based platforms remotely, down-the-wire. Intel AMT makes that possible via OOB remote access to the platform's persistent, tamper-resistant asset ID. The IT professional can compare the remotely scanned asset IDs against the asset inventory database kept in the third-party management app. That allows validation of the stored asset data.
2. The IT professional uses a third-party asset management app that supports Intel AMT to discover platforms, and their firmware-resident software information, remotely down-the-wire (DTW), regardless of operating system or power state. Intel AMT makes that possible via OOB remote access to the platform's persistent, tamper-resistant asset ID and firmware-resident software information. The firmware-resident information can include software lists (e.g., anti-virus update information) and a database key to the third-party asset management database (e.g., to match the discovered platform to corresponding software inventory information). By gathering this information accurately, quickly and remotely, the enterprise can more efficiently and effectively manage its software licenses, and optimize its maintenance/service contracts. In addition, the inventory information enables the IT department to better manage software updates.
3. The IT professional uses a third-party asset management app that supports Intel AMT to discover platforms and their firmware-resident FRU information, remotely down-the-wire, regardless of operating system or power state. Intel AMT makes that possible via OOB remote access to the platform's persistent, tamper-resistant asset ID and firmware-resident FRU information (as populated by the BIOS during the last successful boot). By gathering this inventory information accurately, quickly and remotely, IT can more efficiently and effectively detect configuration drift, and manage its FRU inventories, recalls and warranties.
4. Event from users system received on console, console evaluates event to determine if an alert to help desk is needed. User may call help desk with problem symptoms, too. Help desk diagnoses problem down-the-wire using Intel AMT SOL/IDE-R remote boot capability. Help desk resolves problem down-the-wire.
5. An event may be received on console denoting a dead FRU and the console determines if there should be an alert, User may also call the help desk to report problem. Help desk diagnoses problem OOB and down-the-wire using Intel AMT SOL/IDE-R remote boot capability.
6. Third-party anti-virus apps that support Intel AMT scan platforms down-the-wire, regardless of their operating system or power state, to discover and update down-rev virus signature files and anti-virus engines.
7. The third-party apps can access firmware resident software logs in each platform, even before the platform boots, to determine if updates are needed
 - a. Intel AMT can boot a platform and then third-party apps can deliver and install the updates
 - b. Intel AMT can then turn off the system