CHAPTER 5

Raising Visibility for Trust: The Role of Attestation

Facts are stubborn things; and whatever may be our wishes, our inclinations, or the dictates of our passions, they cannot alter the state of facts and evidence.

—John Adams

Up to this point, our book has discussed the platform requirements and implementation mechanics for Intel TXT as a function of setting up the server infrastructure. This chapter will now turn to look at the critical capability of how to collect platform trust information for use in more far-reaching operational use models. As the previous chapter mentioned briefly, the capability that makes this happen is *attestation*. Historically and practically, attestation services have been the missing piece of the trusted computing puzzle. This chapter will discuss in more detail what attestation means, how it relates to Intel TXT, the role attestation plays in the Intel TXT use models, and how Intel works with the ecosystem of third-party software and service providers to enable this capability for delivering meaningful and compelling solutions.

Attestation: What It Means

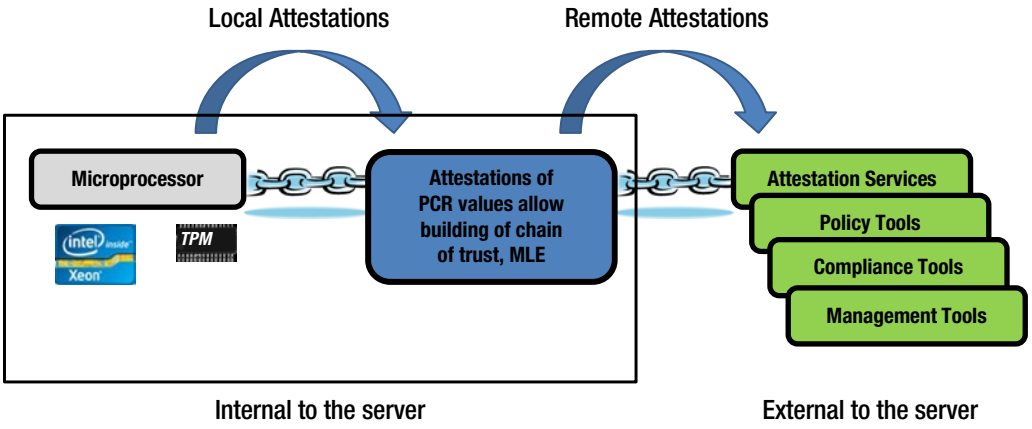
While we will discuss how attestation is a critical component of Intel TXT and its use models, it helps to be reminded that there are actually a number of different attestations at play. They all have some common attributes, because in general, attestations are all about providing some sort of evidence or proof of some platform operation, value, or process.

For a computer security researcher interested in developing trusted computing architectures and technologies, a thorough evaluation and discussion of each definition might be interesting. However, a top-level definition of attestation that matters most to an IT professional who wants to build a more secure datacenter and cloud using technologies such as Intel TXT should suffice. The Trusted Computing Group (TCG) defines attestation as follows:

"The process of vouching for the accuracy of information. External entities can attest to shielded locations, protected capabilities, and Roots of Trust. A platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a platform. Both forms of attestation require reliable evidence of the attesting entity."[1]

The evidence and proofs of the platform values are vital in at least two dimensions of attestation (local and remote). And as discussed in Chapter 4, these attestations and proofs are quite complementary. Figure 5-1 shows a simplified process model for attestation in the context of platforms and use models.

[1] http://www.trustedcomputinggroup.org/developers/glossary/.



Source: Intel Corporation

Figure 5-1. *Attestations help extend the chain of trust and enable use models beyond the platform*

Local attestations occur within the system. This is the mechanism in which the PCR values are sealed/unsealed by verified modules to establish the chain of trust from the processor-based root of trust. The attested assertions of trust between launch components as described in the launch process in Chapters 2 and 3 are evaluated and used in the context of launch control policy (LCP) through the Intel TXT–enabled boot process. Local attestations occur internally to the server during the crucial initial interactions between ACMs, the TPM, launch code modules, and policy indices. The result of local attestations processes is a platform that is trusted, and capable of UNSEALing secrets from the TPM or determining if it is untrusted.

As Chapter 4 indicated, *remote attestations* provide platform trust values to outside entities. In a nutshell, remote attestation is the method we use to securely expose the results of our trusted launch process and the platform chain of trust to the world—thus providing the "pump" to fuel our operationalized use models for Intel TXT. A server can't assert its own integrity or location. So we need attestation services that will allow us to gather and authenticate platform trust information for use by tools such as virtualization or systems management consoles, security policy tools, and so on. As we will discuss in more detail in subsequent chapters, we can use this trust information to control and report on physical, virtual, and cloud infrastructures and workloads more effectively as it provides greater insights into the state of the platform. Since the local attestations, chain of trust, LCP, and the other platform internals have already been discussed in depth, the rest of this chapter will focus on remote attestation as the enabler to maximize the value for Intel TXT use models.

Attestation Service Components

As you will soon see, attestation provides the fuel that drives Intel TXT use models beyond basic trusted launch of a platform into more scalable, flexible, and operationally valuable assets in the IT manager's security portfolio. Figure 5-2 provides a view of the role that attestation plays in the general abstract system architecture model. It also provides a relatively simple framework to discuss attestation from a component and role perspective. Even looking at it from this general architectural view, the critical central role really becomes clear.

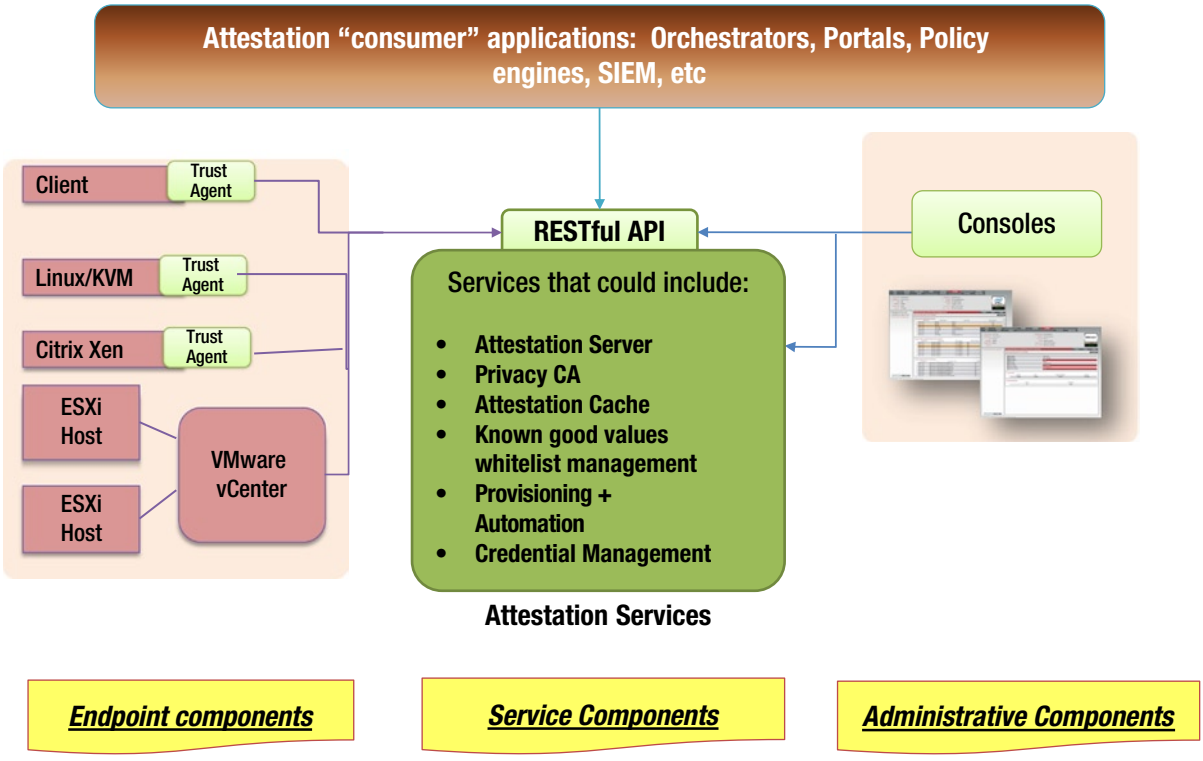


Figure 5-2. *An overview of an attestation service conceptual architecture*

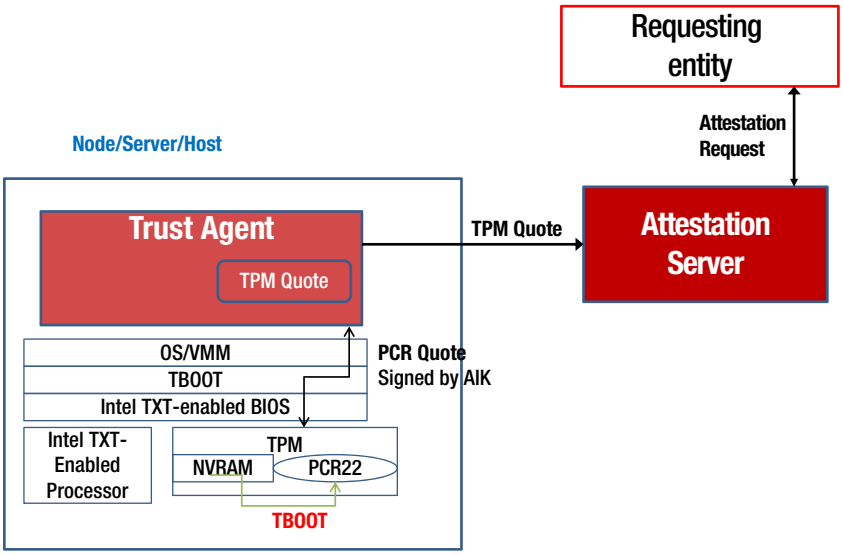
From Figure 5-2, we can see that an attestation infrastructure will interface with a number of key components in the IT infrastructure. It touches endpoint devices (which could be traditional clients such as desktops or laptops) as well as server endpoints. The attestation service will communicate to these endpoints through APIs and/or agents on the devices (depending on the architecture) to gather trust status information. The attestation service will communicate with the administrative components such as management consoles to provide status information. And the attestation service will provide attestation results and information to other applications (such as self-service or ordering portals, security policy tools, cloud orchestration applications, etc.) in the infrastructure for the enablement of key use models for trusted computing.

Endpoint, Service, and Administrative Components

It is now time to take a brief review of the roles and architectures of the endpoint components, service components, and administrative components shown at the bottom of Figure 5-2.

Endpoint component capabilities and roles are primarily related to providing the mechanism for *gathering* platform launch status, as well as other potentially useful platform data (such as geotag/asset-tag data, which will be described in Chapter 7) from the TPM, and packaging it for reporting into the attestation server/service. Data from the endpoint will be accessed from the PRCs and NVRAM using TCG-standard TPM Quote mechanics, and packaged

using XML schema. In most host operating environments—including Linux and the commonly used related open-source hypervisors such as KVM or Xen, a trust agent will be resident on the host to provide this capability. Note that some hypervisor environments (such as VMware vSphere) provide their own native capabilities for providing platform trust information into the attestation infrastructure. Figure 5-3 shows the process flow for a trust agent capturing platform data and delivering it to the attestation service.



Source: Intel Corporation

Figure 5-3. *Trust agent roles in the attestation architecture*

In this scenario, an external entity (perhaps an administrator at a console or one of the "attestation consumer" applications referenced at the top of Figure 5-2) seeks information about the trustworthiness of the endpoint. The trust agent can work with the platform operating system or hypervisor to issue a TPM Quote from the platform TPMs. (Note that in some cases, the hypervisor or operating system can answer this request natively—without a trust agent). Only a trusted operating system can unseal the hash values within the TPM and issue this response back to the attestation server in the form of a signed TPM Quote. From here, the attestation server can process the quote and provide information back to the requesting entity.

Attestation Service Component Capabilities

The attestation service component capabilities and their role are rather broad. In their simplest form, the primary purpose of the service component is to evaluate and provide platform trust status assertions to other entities for use in trusted pools, compliance, and other key use models. As such, they receive requests for information about a platform, and generate requests to the host to collect the information (hash values) about its trustability, location, and so forth, as described in the previous section. They have facilities to validate the information source (via keys) and compare the reported values (which previously arrived in the SAML assertion of the TPM Quote) against whitelists of known good expected hash values. Of course, this means such services must also possess facilities for building and maintaining whitelists. It must also have a mechanism for packaging responses to the original requests for information about the platform—typically via RESTful APIs. There are often desirable related services such as tools for doing bulk attestations, caching, and so forth, which provide value in terms of enhancing the usability, scalability, and so forth, of an attestation service in a high-volume datacenter or cloud implementation.

It is critical to note that just as authenticity is important for building the initial chain of trust and assertions against the elements of the LCP in local attestation, it is equally crucial that the attested information must be verifiable as authentic to the reporting entity for uses in remote attestation models. Attestation services must provide such facilities for trust to be usable by outside entities. For this reason, you will see security functions such as signatures, certificates, SSL and TLS, SAML, and more, as key capabilities of a core attestation service component. They complement traditional access control mechanisms to help provide tamper resistance, integrity, and authenticity to the evidence generated from the root of trust through the operating and reporting infrastructure in the attestation process. This is especially valuable when one remembers that this attestation infrastructure environment spans not only potentially scores of systems, but virtual physical and even off-premise, third-party cloud platforms. These security capabilities allow IT and security professionals to have confidence in the reporting infrastructure so that these tools can be used for audit, compliance, and forensics work.

Administrative Component Capabilities

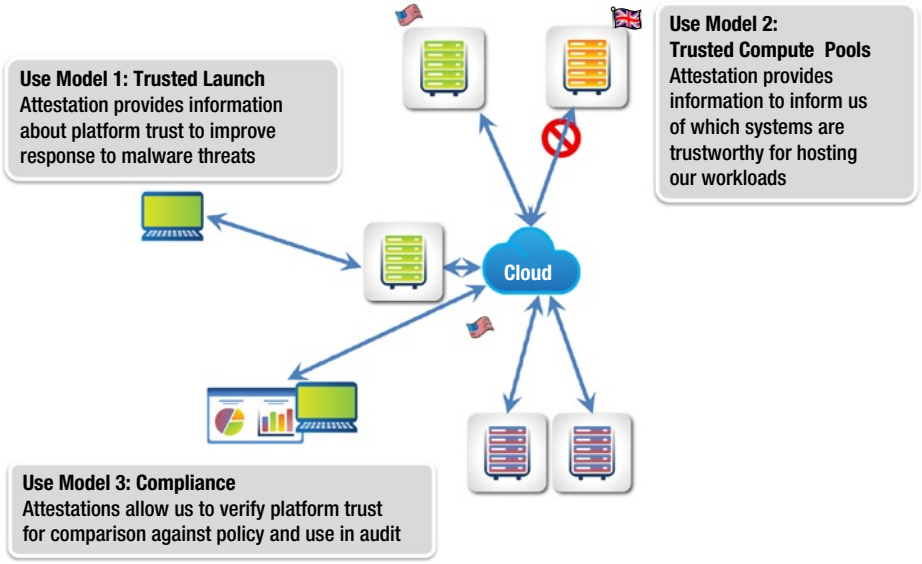
The administrative component capabilities shown in the Figure 5-2 overview are fairly basic, and include a portal for the service administrator that provides a way to consolidate, monitor, and display trust status (often using HTML5 tools). This portal provides the administrative front end for the tools used to configure the attestation service, build and manage the whitelist database, and so forth.

Attestation in the Intel TXT Use Models

The preceding sections have described some of the more critical components of attestation services and outlined how these services work to pull launch status and other key information such as geotags from the platform in a secure manner. This capability is foundational and essential to the operation of all the interesting use models for Intel TXT. In short,

- Attestation is needed to securely query a platform to get platform trust and geotag/asset tag information from the platform(s).
- Attestation services provide a manageable whitelist of known good platform hashes—and evaluate trust and location/asset information against this whitelist.
- Attestation services provide trust, location, and information to other enterprise and cloud management and security tools for enforcement (e.g., controlling virtual machine migrations based on defined trust policy) and reporting (e.g., monitoring compliance to policies).

As a result, it can be expected that there would be many attestation activities occurring during the execution and operation of the key trusted launch (determining if one happened on a target platform), trusted pools (providing platform trust attributes for use in enforcing workload migrations), and compliance (providing platform trust attribute data for comparison to expectations and policy) use models. Figure 5-4 shows the use models and general path of attestations for each.



Source: Intel Corporation

Figure 5-4. *Attestations are critical in all Intel TXT use models*

These use models will be described in more detail in Chapter 7, but it is easy to see that attestation really does provide essential services to allow these use models to function. Therefore, it is equally essential that the attestation mechanism has broad accessibility in the infrastructure, as it needs to

- Accept requests for trust status from portals, SIEM, GRC tools, and other entities across an enterprise or between cloud infrastructures.
- Initiate attestation requests from a service to a platform hypervisor, OS, or trust agent (and get host values from TPM) across an array of hosts in, possibly, many dispersed locations. This request may alternately go to the virtualization management layer (for example, vCenter) in VMware implementations.
- Accept and verify TPM Quote responses from the platform against the expected known good list (whitelist).
- Provide trust assertions about the platform via RESTful APIs to the inquiring entities: portals, SIEM, policy tools, GRC, and so forth, for use (as shown in Figure 5-2).

In short, the attestation service provides the mechanism to get trust information securely out of the platform hardware, verify it, and provide it to applications that help IT reduce malware, gain operational control, and meet audit needs.

With all these responsibilities, and all these related touch points, where attestation sits (at least generally) in the architecture is useful to know. Figure 5-5 provides a general architecture perspective to understand where attestation services fit in relation to platforms, cloud infrastructure, management, and security tools.

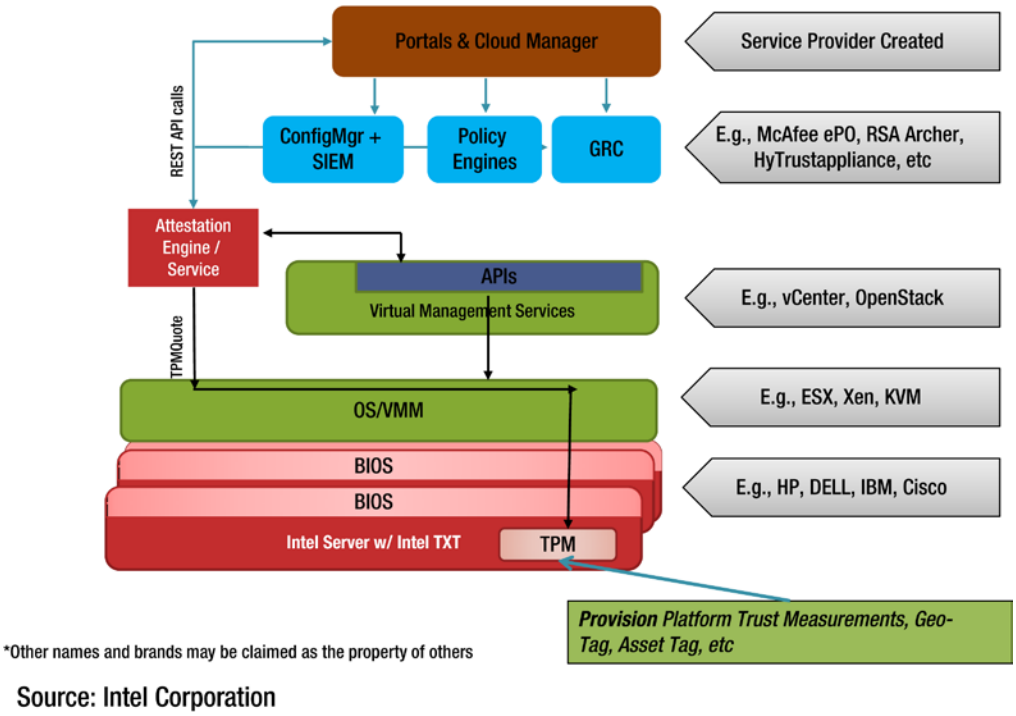


Figure 5-5. *Trusted pools attestation conceptual architecture*

Figure 5-5 shows us that attestation services can sit in any number of places—with the core criteria being that it has accessibility to the relevant platforms (the providers of its information) and the dependent consumers of its information (security applications, portals, virtualization management systems, etc.). So in many ways, it can sit anywhere (as long as accessibility can be granted), which means that attestation services could be a cloud service, on-premise appliances, or integrated into any of a number of layers in our stack (though it is shown as a discrete service here). In a similar vein, attestation services could be placed everywhere. Therefore, many layers and applications (both company-owned or provided as a service) can provide their own attestation capabilities, or such capabilities could be federated among a number of applications or service providers. Aspects of how these implementation and integration choices are being made by ISVs and service providers today and how they are likely to evolve will be discussed in greater detail in Chapter 6.

Enabling the Market with Attestation

We mentioned earlier that there was a void in the commercial market for attestation capabilities. In trusted computing components such as TPMs that have shipped by the hundreds of millions over the past few years, saw activation rates that were low and the use models that were in use were relatively contained within the platform—such as SEALing encryption keys for trusted operating systems. There were a few efforts at remote attestation technologies, perhaps the most notable being the OpenPTS project from IBM. But these have been somewhat basic approaches and saw limited implementation at best.

It seems the technology was caught in a classic catch-22: there was limited demand for developers to build solutions because the market had not shown compelling use models. And it was hard to envision compelling use models with no technology to stimulate the solution set. With Intel TXT technology shipping in volume and gaining ecosystem support, and armed with some powerful use models driven by security concerns and the changing dynamics of virtualized and cloud datacenters, the time and opportunity were right for change.

To provide that catalyst for change to the market, Intel undertook two projects to help enable the market to deliver our chosen use models: the OpenAttestation project (often referred to as OAT) and a project code-named Mt. Wilson technology. We'll review each briefly.

OpenAttestation

OpenAttestation is an Intel-maintained open-source project that is a software development kit (SDK) for managing host integrity verification using TCG-defined remote attestation protocols. The project includes code that was developed by the National Information Assurance Research Lab (NIARL) of the US National Security Agency—an agency that has a long history of involvement in developing security and trusted computing technologies. The project is available for use and contributions by all; it is hosted on the GitHub repository, which is commonly used by open-source projects, at `https://github.com/OpenAttestation/OpenAttestation.git`.

Intel expects the primary audience of collaborators, customers, and developers for OpenAttestation to be comprised of cloud service providers and enterprise security and management tools providers. The vendors that have a strong focus on building services and tools based on primarily open-source platforms and technologies, such as Xen or KVM hypervisors and OpenStack, have shown the greatest interest. Generally, this interest is driven by cost (free!), accessibility (GitHub), flexibility (source code, BSD license model), license model, and technology affinity (compatibility with tools and skills).

Of course, it has to do something useful to drive interest, and it does. Some of the key features and services of OAT (as of this writing in 2013) include:

- Supports major Linux-hosted operating systems (including Ubuntu, SUSE, Red Hat Enterprise Linux) and the associated hypervisors (Xen, KVM)
- Java-based privacy certificate authority (Privacy CA) and appraiser for authenticating platform quotes
- Java-based host agent that accesses the platform TPM through the well-established open-source TSS (also known as TrouSerS) trusted computing software stack
- RESTful-based simple Query API with added support for Tomcat two-way SSL/TLS for securing the Query APIs
- Reference CLI Curl scripts for API access
- Basic whitelist service and API, with whitelist management capabilities for building and maintaining whitelists of approved system configuration values
- Reference web portal/GUI implementation to speed interface development
- PCR-based report schema and policy rules with historical PCR data tracking/comparison capabilities
- Flexible access control to the attestation server with hooks for ISVs or service providers to implement custom access control mechanisms

Of course, this list of capabilities will grow over time as Intel and others in the open-source community and elsewhere extend the project to meet new needs. But to help visualize how these capabilities come together, Figure 5-6 provides a graphical overview of the OpenAttestation project components and architecture.

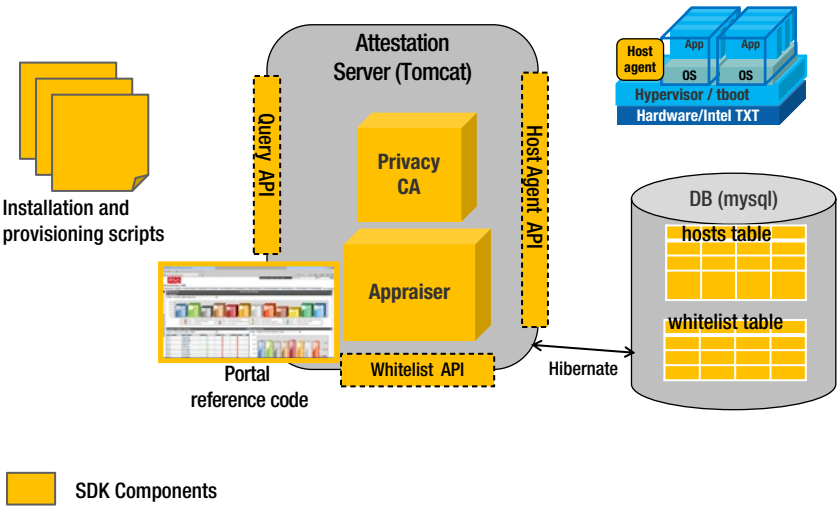


Figure 5-6. *OpenAttestation SDK architecture overview*

Hopefully, it is of little surprise that you will note many commonalities in the OAT project components and services and the general attestation architecture shown in Figure 5-2. As it was Intel's intent to enable the solution for the target use models, it indeed covers the endpoint (for example, the host agents), the attestation service (for example, Privacy CA, Query APIs, and whitelist model), and the administrative components (for example, portal and GUI reference code) of the general architecture.

Mt. Wilson

Mt. Wilson technology is the other software project that Intel has applied resources toward in an effort to stimulate the market and assure that solutions to enable use models are commercially available. Structurally, there are many similarities between the two projects, so the authors will spare you yet another visual representation. But it is useful to note that Mt. Wilson is different from OAT in a few key ways:

- Generally more feature rich—including trust agents for platforms not currently supported by OAT
- Developed by Intel (though some open-source components are used)
- Offered in binary (not source code) form
- Offered under an Intel license to a limited number of selected solution providers

Intel chose to undertake the Mt. Wilson program to accelerate the development of solutions. This model allowed Intel to develop more of the components needed to fully enable the use model, and then to share that technology strategically with the third-party ecosystem partners that were most aligned or vital in the marketplace. In short, it was all about solving the technology availability problem for the market. It was Intel's belief that providing a relatively turnkey, binary code base brought significant value for a market that was relatively new to trusted computing. The key allies in using Mt. Wilson are a small number of critical virtualization and security management tools providers, systems integrators and cloud service providers. A key difference is that many of these early partners are more focused on heterogeneous environments or on non-open-source platforms such as Citrix XenServer, VMware vSphere, or others.

The natural next question is usually, "Should I choose Mt. Wilson or OpenAttestation?" The flippant answer that many of us involved in enabling these use models in the industry is, "Who cares?" But the honest answer is that it makes the most sense to let the solution and the software/service provider's needs drive the choice. Ideally, it is best for the ecosystem to select the solution that meets your target market needs or environment, and which suits your business model and skills. Beyond that, it really does not matter much, because the end user or IT administrator customer will likely never know which technology is under the hood of the solution.

That answer might not satisfy all, so we can revert back to our original intent. It is playing out that Mt. Wilson strategically enables key ecosystem partners to use a turnkey package to deeply and more quickly bring solutions to market), whereas OpenAttestation will likely play a broader role over time because it is much more accessible and flexible. But some confusion may be expected, and we have to recognize that software projects and solution needs in a rapidly evolving market can drive the need to change plans quickly. This would be disruptive to the ecosystem. To minimize this issue, Intel has assured that API compatibility between the projects is maintained, and that over time, there will be more common code where there are equivalent services. With these plans, the barrier for an ecosystem provider to change out the core attestation plumbing can be dramatically reduced, and provides them with significant flexibility as their needs change.

How to Get Attestation

Intel's work with the ecosystem, as well as the efforts of others, is resulting in growing success in driving the market for solutions. The best news on this front is that other vendors are also working on attestation technologies now as well, with offerings likely from operating system, cloud, security software, and other types of vendors. Once again, from Intel's perspective, who "owns" or develops the technology is more or less irrelevant—as long as it meets the needs of the solutions and use models. In the end, innovation and customer choice is always the best scenario for customers—and ideally, the technology itself is open and transparent to the IT administrator.

How "transparent" attestation technology becomes will largely be a function of the manner in which it is delivered. Some of the possible models include:

- Dedicated virtual appliances
- Dedicated physical appliances
- Integrated as a function in security application software
- Integrated in cloud management software
- Integrated in virtualization management software
- Offered as a component of a cloud service offering
- Integrated as a "Security as a Service" (also known as SecaaS) offering

In the authors' views, some of these options are less likely than others—and certainly some will come to market faster than others. This last statement we can make with surety from our positions of helping to enable the technology with ecosystem solutions providers. From this vantage point, we can see that some of the first solutions to market will be in the classes of those integrated in security software tools, and cloud and virtualization management. Cloud service offerings that utilize attestation to facilitate trusted pools and compliance use models will also be among the first to market. In each of these cases, the attestation function will largely be "hidden" behind security policy or cloud service creation, management, and enforcement tools.

While this is more conjecture than hard experience on the authors' part, note that other compelling options such as delivery as a component of Security as a Service offerings may indeed ultimately also play a bigger, long-term role in the market. And none of these delivery models is inherently wrong—though in our view, some will do less to lower cost and complexity, and therefore will likely see less broad adoption. Further discussion of where we expect to see attestation services deployed into the software ecosystem is provided in Chapter 6.