GLOBALPLATFORM®

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

The Trusted Execution Environment:

Delivering Enhanced Security at a Lower Cost to the Mobile Market

White Paper

June 2015, revised from February 2011

secretariat@globalplatform.org | www.globalplatform.org |

Table of Contents

About GlobalPlatform

GlobalPlatform defines and develops specifications to facilitate the secure deployment and management of multiple embedded applications on secure chip technology. Its standardized infrastructure empowers service providers to develop services once and deploy across different markets, devices and channels. GlobalPlatform's security and privacy parameters enable dynamic combinations of secure and non-secure services from multiple providers on the same device, providing a foundation for market convergence and innovative new cross-sector partnerships.

GlobalPlatform is *the* international industry standard for trusted end-to-end secure deployment and management solutions. The technology's widespread global adoption across finance, mobile/telecom, government, healthcare, retail and transit sectors delivers cost and time-to-market efficiencies to all. GlobalPlatform supports the long-term interoperability and scalability of application deployment and management through its secure chip technology open compliance program.

As a non-profit, member-driven association, GlobalPlatform has cross-market representation from all continents. 130+ members contribute to technical committees and market-led task forces. For more information on GlobalPlatform membership visit www.globalplatform.org.

Publication Acknowledgements

GlobalPlatform wishes to offer special thanks to the members of the Trusted Execution Environment Roadmap Working Group and their respective organizations for their involvement in developing this white paper.

Contributors include the following:

Full Members:

Jan Eichholz – Giesecke & Devrient
Marcus Streets – Good Technology
Janne Hirvimies – Huawei
Sampo Sovio – Huawei
Pierre Quentin – Ingenico
Cyrille Ngalle – Inside Secure
Christian Schwarz – Nagravision
Cedric Colnot – NXP
Sebastian Hans – Oracle
Jean-Philippe Galvan – Samsung
Sanjeev Verma – Samsung
Mike Parsel – Sprint
Rob Brown – Trustonic
Don Felton – Trustonic
Young Choi – Verizon
Hans van Tilburg – Visa

Public Entity Members:

Paul Waller – CESG
Jon Rolf – U.S. Department of Defense

GlobalPlatform Team Members:

Kevin Gillick – Executive Director of GlobalPlatform
Gil Bernabeu – GlobalPlatform Technical Director
Alliances Management – Operations Secretariat

Intended Audience & Companion Documents

This document is intended as a high-level introduction to the Trusted Execution Environment, and this paper is a wholesale revision of a 2011 whitepaper published on the same topic. While it contains several use cases in an effort to introduce the TEE to stakeholders across the industry, it is not intended to be a technical document. GlobalPlatform envisions publishing two complementary documents in 2015 aimed at addressing technical topics—a Technical Whitepaper to introduce the architecture and components/building blocks of a TEE, and a Device Assurance Whitepaper to discuss the TEE Compliance and Certification programs.

Executive Summary

This document introduces the Trusted Execution Environment (TEE) and explores the benefits of implementation.

As the mobile and consumer markets for connected devices mature and expand, an increasing number of security concerns demand attention. With consumers using their devices for a variety of "lifestyle" applications, there is a proliferation of security needs that result from the use of an open environment, notably for mobile devices. Content protection, corporate applications, connectivity, financial transactions, and more exacerbate these security concerns, which are relevant to all participants in the value chain and not just to the consumer. Content owners, service providers, banks, mobile network operators, OS and application developers, device manufacturers, platform providers, and silicon vendors are all key stakeholders in this market—and thus have a vested interest in seeing proper security implemented.

Balancing the needs of openness and security is a difficult problem to solve. Today's devices must not only meet both the functional and security requirements of various stakeholders, but also allow for "opening up" the device. This openness is expected by both commercial and development stakeholders, as well as device end users: everyone wants to be able to integrate the latest available software, download applications, and customize look and feel, but without being exposed to security risks such as privacy invasion, device intrusion or asset stealing.

Doing so introduces complex features and software that are ultimately impractical or impossible to manage from a security perspective. Overall the challenges lie in the following:

1) Relaxing device control rules (or constraints) to allow for openness while guaranteeing security for all stakeholders,

2) Providing a reliable execution environment and accessible APIs for security functions that are implemented in hardware or unmodifiable software,

3) Shrinking the Trusted Computing Base (TCB) and associated complexity for security functionality,

4) Reducing fragmentation of security function APIs to allow most applications to execute on most devices, and

5) Shortening the duration of security evaluation to match device and software lifecycles.

The Trusted Execution Environment (TEE) offers the best route to meeting these security objectives and simultaneously addressing the needs of key stakeholders. The TEE is an isolated execution environment that runs alongside a Rich OS and hosts trusted services offered to that rich environment. The TEE offers an execution space that provides a higher level of security than a Rich OS and delivers security that is sufficient for most applications. In this way, the TEE offers an exceptional balance by allowing for greater security than a Rich OS environment without the constraints of other methods, such as a Secure Element (SE). However, it should be noted that the

TEE and SE do naturally complement each other to provide a best-of-breed solution for devices requiring both security models.

With such a broad number of actors and use cases that benefit from the TEE, standardization in this area brings many benefits to the industry: better interoperability, greater assurance, quicker time-to-market, and lower costs. Crucially, it allows service providers to develop a Trusted Application (TA) once yet deploy it across all device types, regardless of other apps that are present on the TEE. This portability of service addresses compatibility and scalability issues encountered in many multi-channel, multi-device, and multi-app deployments.

GlobalPlatform's history of introducing specifications for what are now billions of Secure Elements worldwide makes it a logical choice to drive TEE standards, which it has done since the introduction of the TEE Client API 1.0 specification in July 2010. GlobalPlatform continues working on specifications for a variety of TEE APIs and Protection Profiles, and its work will be further supported by the launch of a TEE Security Evaluation Secretariat in 2015.

Specifications that are available today include the TEE Client API, TEE Internal Core API, TEE DEBUG API, TEE Trusted UI API, and TEE SE API, all of which are available for download on the GlobalPlatform website. As of this document's publication date, additional GlobalPlatform TEE standards are still in development and are expected to be published in the near future: TEE Administration Framework API 1.0, TEE Trusted UI API 1.1, TEE SE API 1.1, and TEE Sockets API 1.0.

Given that GlobalPlatform represents a broad ecosystem of stakeholders, the resulting specifications are sure to satisfy most necessary market requirements.

INTRODUCTION: The Increasing Need for Security with Connected Devices

Our lives are increasingly dependent on smart connected devices: we use them to conduct business, maintain social relationships, make purchases, and enjoy media content. Despite the obvious benefits, these devices have huge amounts of code and data that are susceptible to attacks from hackers; furthermore, the sheer number of applications that are easily available for download represent an even larger opportunity for fraudsters.

Similarly, automotive and home devices are becoming increasingly connected and providing more capabilities than their original core functionality.[1] Consumers are increasingly using their devices in new ways—organizing a trip from one's TV, streaming music while driving, and using a smartphone to pay for petrol. These expanded practices create new security vulnerabilities, which in turn emphasize the need for mechanisms that allow trusted parties to have access to applications without granting hackers the same opportunity.

The increased security needs are ultimately driven by new characteristics of consumers' smart connected devices; a few such features, and the related security concerns, are as follows:

- **Use of an Open Environment**: New devices are generally built with operating systems that provide an open environment. A key benefit of this is that users can add applications at any time, often with little concern as to the impact to the stability and security of the device. An open environment, however, exposes devices to an expanding variety of attacks. Device manufacturers want to take advantage of such Operating Systems but need to be in control of how the software that runs on the device behaves.

- **Authentication:** The traditional method of authenticating a user involves requesting a username and password. This is increasingly being deemed inadequate because consumers use weak passwords or reuse existing ones, and hackers are increasingly able to gain access to accounts. Because application or service providers often have stores of personally identifiable and sensitive information on their servers, such hacks make headlines, upset consumers, and undermine business confidence. Accordingly, there is a need for improved authentication mechanisms that protect consumers while allowing application developers flexibility.

- **Privacy:** Devices store increasing amounts of personal information (such as contacts, messages, photos, and video clips) and even sensitive data (including credentials, passwords, medical data, etc.). To prevent exposure of this information in the event of loss, theft, malware, or another negative event, sufficient security is needed to store, process, and distribute such personal data.

---

[1] The increasing trend for devices to be connected to the Internet and one another, known as the *Internet of Things*, is discussed in a separate GlobalPlatform whitepaper published in May 2014. Visit http://www.globalplatform.org/documents/whitepapers/IoT_public_whitepaper_v1.0.pdf.

- **Content Protection**: Today's increasingly open devices offer high definition (HD) video playback and streaming, mobile TV broadcast reception, and console-quality 3D games. They even serve as content gateway devices, thereby replacing traditional set-top boxes (STBs) or consoles. As a result, a device's playback capabilities become less important, while the security needs increase. It thus becomes necessary not just to protect the Full HD or Ultra HD content on a mobile device, but also to protect for instances where a device forwards the content to a TV set.

  All of this functionality requires content protection, Digital Rights Management (DRM), or Conditional Access (CA) services to protect high-value HD content. The DRM and CA schemes are often associated with content management and protection models, such as Content Management License Administrator (CMLA) or Content Protection for Recordable Media (CPRM), which favor hardware-strengthened content key protection[2] and ultimately content protection.

- **Corporate Data Access**: Enterprise company IT professionals are often wary of enabling access to their internal networks, fearing that the devices could carry malware, be stolen, or create attacks from within the internal network when used outside of company premises. Therefore, IT departments frequently establish green-lists and red-lists of devices based on their security capabilities. They are also concerned by the always-on nature of these devices and the enforcement of password protection and device locking when not actively in use.

- **Financial Risks**: Financial transactions on connected devices (especially mobile devices) are becoming more common. These include ticketing, remote payment, proximity payment, and financial e-transactions. And, it is increasingly possible to use a mobile device to make purchases at retail locations. Furthermore, there are an increasing number of use cases where the mobile device becomes the point of sale (POS) terminal, particularly for highly mobile points of sale.

All of these factors present security concerns that must be addressed in the market, and one way of resolving these issues is to provide a small, isolated execution environment for sensitive assets and code that would allow service providers and Original Equipment Manufacturers (OEMs) to improve the user experience while reducing fraud. The GlobalPlatform Trusted Execution Environment (TEE) effectively addresses all of these concerns.

[2] Whereas content key protection can be fully implemented within the hardware (assuming co-design efforts with software), it is difficult to fully implement content protection within the hardware.

SECTION 1: Defining and Understanding the Trusted Execution Environment

Addressing the security concerns discussed in Introduction is most easily accomplished via the Trusted Execution Environment (TEE). The TEE is an isolated execution environment that runs alongside the Rich OS and hosts trusted services offered to that rich environment. The TEE has a Protection Profile that can be used as a basis for security evaluation, and such independent assessments can provide assurance to those providing high-value services. The TEE addresses many of the primary security concerns by reinforcing the following:

- Confidentiality (e.g. protecting the keys and DRM code in a content protection scheme);
- Authenticity and protected execution of authenticated code (e.g. ensuring that the user is who s/he says s/he is and that a payment application is the one that the user provisioned);
- Privacy (e.g. protecting all user data, whether stored on a device or when in transit between devices); and
- System integrity (e.g. by allowing multiple power cycles and firmware updates).

The TEE accomplishes these things by preventing access to its hardware and software security resources from the Rich OS and its applications, yet offering access to such resources through well-defined APIs. Figure 1 shows the architecture of the TEE.

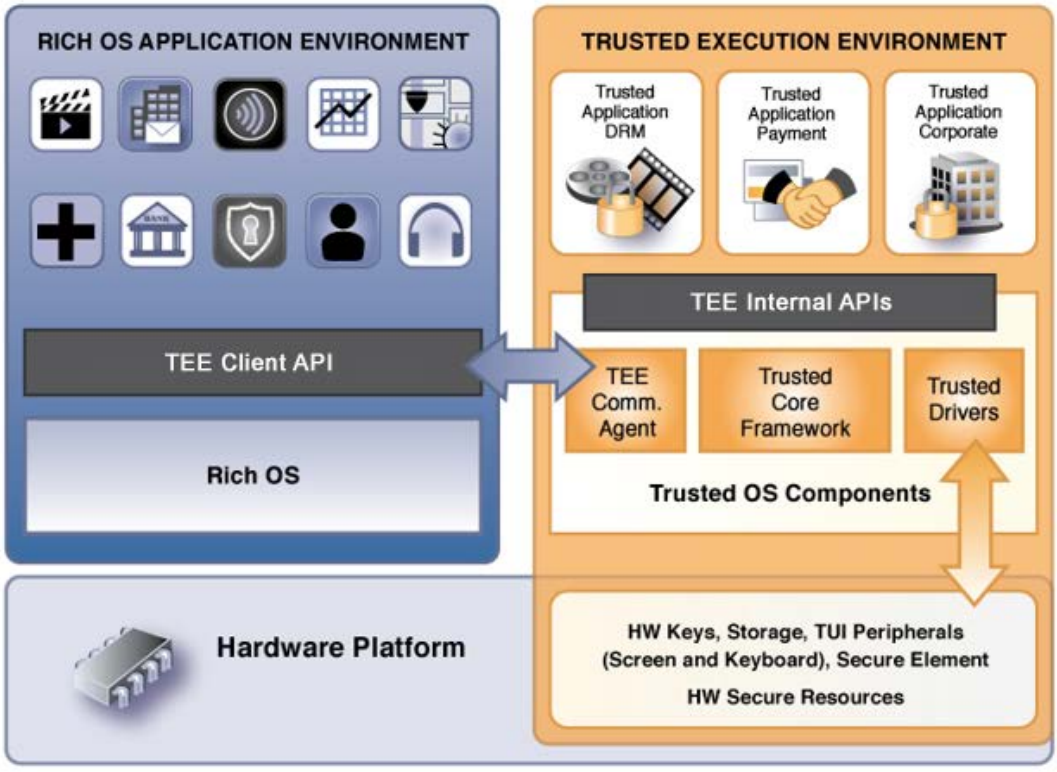

Figure 1 : Architecture of the TEE

As Figure 1 illustrates, the TEE offers safe execution of authorized security software, known as **Trusted Applications (TAs)**; it also enforces protection, confidentiality, integrity, and access rights of the resources and data belonging to those Trusted Applications. In order to guarantee the root of trust of the TEE, the TEE is authenticated and then isolated from the rest of the Rich OS during the secure boot process.

Inside the TEE, each Trusted Application is independent from the others, and a Trusted Application cannot perform unauthorized access to security resources from another Trusted Application. Trusted Applications can originate from different application providers, and it is expected that TEE standardization will enable a large ecosystem of Trusted Application providers, and therefore reduce fragmentation.

Trusted Applications are given controlled access to security resources and services via the **TEE Internal APIs**, which are currently being standardized in an ongoing program by GlobalPlatform. Such resources and services can include cryptography, secure storage, secure clock, Trusted User Interface (TUI), Secure Element (SE) interfaces, and more. In the near future they could additionally also include client Internet Protocol (IP) Sockets, key injection and management, enhancements to TUI (including fingerprint biometry), and improved SE interfaces.

As defined by GlobalPlatform, a TEE will undergo a qualification process, which will include functional testing (compliance) and security evaluation testing (certification). This certification is based on Common Criteria methodology, and a TEE Protection Profile has been validated by a laboratory and approved by a Certification Body.

The following GlobalPlatform TEE standards have been published and are now publicly available:

- **TEE Client API** is a communication interface designed to enable a Client application running in the Rich OS to access and exchange data with a Trusted Application running inside a Trusted Execution Environment.

- **TEE Internal Core API** is a programming interface designed to enable a Trusted Application running inside a Trusted Execution Environment to perform the general operations of a security application. Support is provided for Cryptography, Secure Storage, communication, and general tasks, such as timekeeping and memory management.

- **TEE DEBUG API** provides a simple, but standardized, debug methodology. Standardization allows development devices to provide debug modes suitable for compliance and development testing, and at the same time remove these in a controlled and verifiable manner from production devices.

- **TEE Trusted UI API** currently allows a Trusted Application to display text and graphics while asking the user to perform an action ranging from navigation to entry of an associated PIN- or Password-backed ID. The TA can provide graphic assurance as to the security state of the display, and this is backed by further user assurance measures from the TEE.

- **TEE SE API** removes the need for the TA writer to communicate with SE applets via a Rich Execution Environment[3] (REE) resident Client Application. Now the TA can directly open communication with SE readers and then SE applets on attached secure elements such as a UICC, Secure Digital Card (SD-Card), embedded Secure Element (eSE), etc.

These specifications can be downloaded from the GlobalPlatform website.

As of this document's publication date (April 2015), the following GlobalPlatform TEE standards are still in development and are expected to be published later in 2015:

- **TEE Administration Framework API 1.0** will bring standardization to today's TEE remote administration systems. It will provide online and offline methods to manage the TEE, its Security Domains and Trusted Applications, and the associated secure data. Symmetric and Asymmetric cryptography will be supported.

- **TEE Trusted UI API 1.1**, while retaining strict isolation from REE and other TAs, will add lower-level interfaces that allow TAs more control and TA developers more flexibility in their designs.

- **TEE SE API 1.1** will take the existing interface and add GlobalPlatform Secure Channel Protocol interfaces, along with SE applet discovery methods.

- **TEE Sockets API 1.0** will remove the need for the TA writer to communicate with remote servers via an REE resident Client Application. The TA will be able to open client UDP or TCP sockets, and receive secure communications over Transport Layer Security (TLS).

Once available these documents will be made available on the GlobalPlatform website.

1.1. Leveraging the TEE for Service Deployment

A Trusted Application is the means for a Service Provider to deploy secure services on a device that supports a TEE. The TA is executed in a secure manner in the TEE and relies on the TEE's Internal APIs. Among other services, the TEE supports key management, key storage, secure storage of data, and cryptographic operations. Nevertheless, all of the operations of a given service need not be executed in the Trusted Application that is located in the TEE. Thanks to a distributed architecture, the Rich OS can execute part of the functionalities.

As a result, an application that leverages the TEE is partitioned into two parts: one is executed in the Rich OS while the other (the Trusted Application) runs in the TEE. Using the TEE Internal APIs ensures that the Trusted Application is portable across a wide variety of GlobalPlatform-compliant devices, therefore reducing fragmentation.

[3] A Rich Execution Environment is an environment that is provided and governed by a Rich OS, potentially in conjunction with other supporting operating systems; it is outside of the TEE, and as such, both this environment and the applications running on it are considered untrusted.

1.2. Evolving Service Administration via the TEE

Today's standard way for deploying secure services that rely on TEE technology is for the device manufacturer to do so when the TEE is being integrated into the device. However, this model needs to be changed so that several parties can provision and download Trusted Applications.

GlobalPlatform is in the process of creating a new GlobalPlatform TEE Administration Framework that will enable both local and remote administration of the TEE.

These will support both on-line and off-line management activities, supply methodologies for installation, updates and removal of security domains, Trusted Applications in those domains, and personalization data associated with either the Security Domains or the Trusted Applications. To cater to the widest range of technologies, both symmetric and asymmetric cryptography may be used to authenticate and bring confidentiality to these operations.

1.3. A Summary of TEE Benefits

The TEE is a unique environment that is capable of increasing the security and assurance level of services and applications requiring security, including the following:

- **User Authentication**: Through its Trusted User Interface feature, the TEE makes it possible to securely collect a user's password or PIN code that will then be verified locally, on a remote server, or within a Secure Element. This trusted user authentication can be used to verify a cardholder for payment, confirm a user's identification to a corporate server, attest to a user's rights with a content server, and more.
- **Trusted Processing and Isolation**: Any processing that needs to be executed on a device can be isolated from any untrusted software attack by being run in the TEE; this is possible while still leveraging any of the device's resources. Examples include processing a payment, decrypting premium content, reviewing corporate data, and more.
- **Transaction Validation**: Through its Trusted User Interface, the TEE makes it possible to ensure that the information displayed accurately portrays the application's request—as opposed to displaying misinformation offered by a rogue application. This is useful for a variety of functions, whether validating payment, protecting a corporate document, or other.
- **Abstraction of Usage of Secure Resources**: By using TEE APIs, application developers can easily leverage the complex security functions available from a device's hardware instead of using less safe software functions. Such hardware security resources include hardware cryptography accelerators, Secure Elements, biometric equipment, key materials handling, secure clock, and more.
- **Certification**: Trusted certification is only achievable through standardization of the TEE; an appropriate evaluation scheme improves stakeholder confidence that the security-dependent applications are running on a trusted platform (comprised of the TEE and its underlying hardware) that has been deeply evaluated and certified.

SECTION 2: Understanding the TEE Vis-à-Vis the SE and REE

As noted in the prior section, a Rich Execution Environment (REE) is governed by the Rich OS and resides *outside of the TEE*. As such, it is important to underscore that applications running on an REE are not trusted in the same manner as are those that run on either the TEE or a traditional Secure Element (SE). The TEE and SE, by contrast, are *highly complementary* to one another.

The TEE is designed to work in collaboration with an SE, particularly through the SE API, Security Channel Protocols (and particularly SCP11), and SE Access Control. The TEE provides a framework for security within the device, offering a layer of security between a typical Rich OS and a typical SE. Both the SE and TEE enable secure deployment of Trusted Applications.

However, in a broad sense, the two differ in that an SE is characterized by physical isolation, exceptionally strong security certification, slower performance and data speeds, limited storage space, and indirect access (through either a TEE or Rich OS) to the user interface. The TEE shares many of the same characteristics of the Rich OS, with isolation being dependent on each TEE implementation, certification being focused on both security and short duration, and direct confidential/integrity-based access to the user interface.

Accordingly, either an SE or TEE (or both) could be used for deployment of Trusted Applications. A simplified view of the tradeoff between the two would be to favor a TEE for deployments with strong security requirements and a desire for a faster, easier, and more attractive user experience. An SE, by contrast, would be favored when the highest security is required at the expense of implementation ease and the user experience. The following are a few examples of different implementations, as well as considerations that could influence the choice of an SE or TEE:

- Although the SE brings a higher level of security to execute mobile financial transactions, not all transactions actually require that level of risk mitigation. The need for security depends on the type of the operation, the amount of the transaction, and/or the user's profile and history. A TEE may provide sufficiently robust security while enabling more flexibility with implementations.

- Enterprise networking can be appropriately protected with authentication and encryption that can be provided by the TEE while offering a level of performance comparable to the Rich OS. Additional protection, if required, could be brought by storing and processing credentials in a SE.

- The TEE is an ideal environment to host DRM agents that protect content or applications downloaded from an app store.

- Collaboration between TEE and SE could be an ideal approach to mobile financial implementations—perhaps leveraging an SE for payment functionality and a TEE for couponing and loyalty programs.

So, if we understand a Rich OS to be a rich environment that is vulnerable to both software and physical attacks, and an SE being resilient to physical attacks but somewhat constrained in execution processing capabilities, the TEE serves as an ideal balance between Rich OS performance and SE security, and a companion to both.

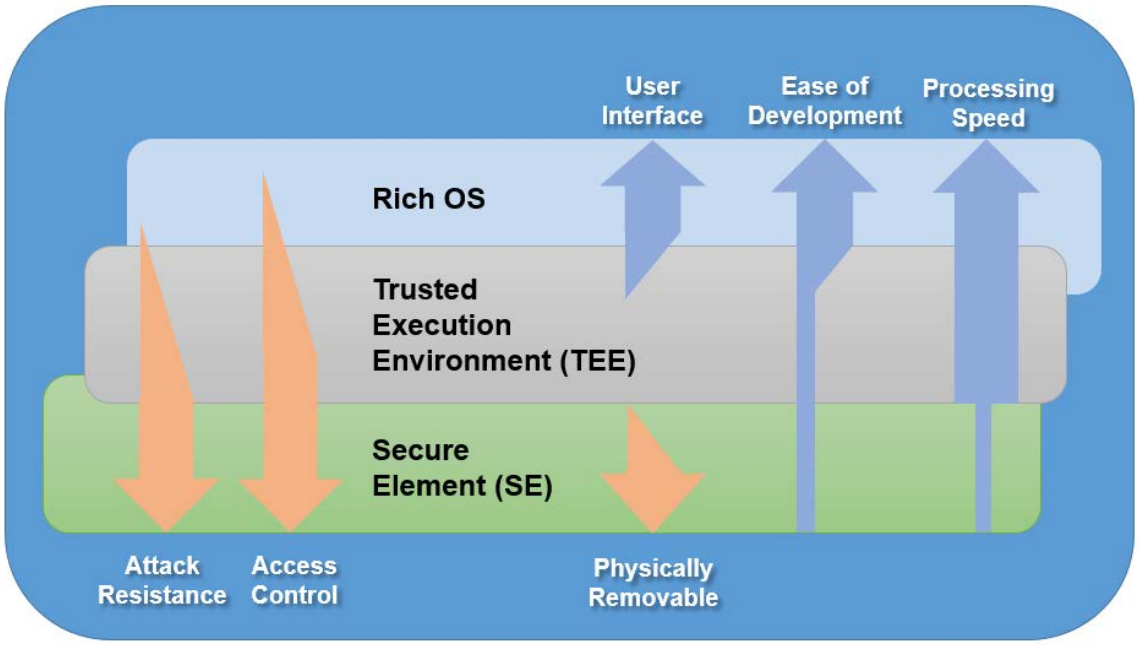

Figure 2 : Rich OS, TEE and SE Positioning

Figure 2 represents the security and usability characteristics in particular environments—a Rich OS, TEE, and SE. However, the capabilities indicated are not of the same strength on the whole range of a particular implementation of an environment. The diagram reflects this through the given width and height of the arrows.

In very general terms, the TEE offers an execution space that provides a higher level of security than a Rich OS; though not as secure as an SE, the security offered by the TEE is sufficient for most applications. Moreover, the TEE provides a more powerful processing speed capability and greater accessible memory space than an SE (these are, in fact, quite similar to that of a Rich OS).

Because the TEE supports more user interface capabilities and peripheral connections than an SE, it allows development of security applications that enable a rich user experience comprising a complex UI targeted for secure input and display. In addition, since the TEE is isolated from the Rich OS environments (as a result of software and/or hardware partitioning), it combines the best of the Rich OS functionality while maintaining security characteristics typically associated with an SE. In particular, the TEE is able to reduce the consequences of software attacks occurring in the Rich OS (e.g. OS rooting, jailbreaking, malware, etc.).

Ultimately, what this discussion makes obvious is that security is a compromise between the cost of the protection and the cost of the attack. Embedded within this high-level conclusion are several other driving factors:

- The inconvenience to the user

- The cost of training and supporting the user
- The direct and indirect value of the asset being protected
- The cost to attack the asset in other manners
- The awareness of the attackers that there is an asset to be attacked

Figure 3 below illustrates the security positioning for the TEE as compared to a Rich OS alone or an SE.

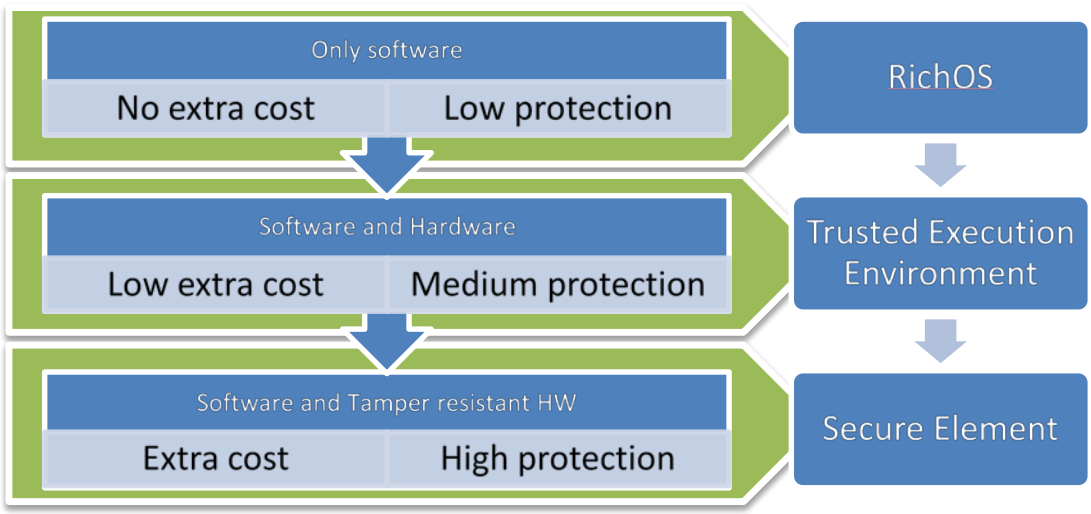

Figure 3 : Security vs. Investment

SECTION 3: Different Perspectives on TEE Security

The previous sections focus on the general security characteristics of the TEE and the general benefits to stakeholders of deploying TAs. However, it must be acknowledged that each market vertical—as well as each stakeholder within those verticals—is different. While the TEE will generally perform the same functions across different markets, what it enables and how it impacts market deployments will differ.

The following two sections are not intended to be comprehensive in their market coverage, but they nonetheless illustrate how the enhanced security enabled by the TEE impacts various markets and the actors within.

3.1. Security Perspectives across Different Markets

In addition to the fact that increased *functionality* drives security requirements, each industry has unique requirements that further increase security needs. Consider the following markets as a few examples:

- **Mobile**: In the mobile market, security concerns are tied to several factors, not the least of which being the sheer number of stakeholders involved in device and application delivery.

  Security concerns are tightly intertwined with the way that the distribution model has changed in the mobile market. Consider that, a decade ago, most mobile handsets were sold via Mobile Network Operator (MNO) distribution channels and that MNOs had more latitude to "lock" devices. Today's market involves a considerably more complicated system of inventory management, and consumer demand is driving the need to "unlock" phones for use across MNO networks and/or to make remote updates to devices.

  Thus, security concerns span the MNO, the makers of the dominant OS platforms, application developers, software vendors, and the consumer. Standardization of certain building blocks—Internal API, Client API, Biometric API, and others—help define a path toward improving security. A framework (such as a GlobalPlatform-certified TEE) that guarantees a minimum baseline for platform security would allow all stakeholders to make updates to devices and applications while minimizing threats to consumers.

- **Professional/Government:** Enterprises and governments are recognizing that the traditional working day and structure have changed; employees have a need to work in different places and times, with different devices, and with access to different information.

  Whereas IT departments have historically mitigated many risks by requiring employees to work only on company-issued computers or mobile devices, today's employees have a need to connect to enterprise systems while away from the office and on any number of devices. Managing risks in these environments is critical, and the only way to avoid compromising on cost or usability is to provide improved security technology across the market.

  There is an expectation that computing platforms should be 'secure by default.' Such a broad-based confidence in security requires standardization that ensures a consistent level of security functions across a wide range of devices.

- **Internet of Things (IoT)**: The IoT is an increasing trend for devices to be connected to the Internet. Such devices span types and industries, from mobile phones, to sensors in automobiles, to measuring devices in home energy meters, and more. The potential impact of such cross-device, cross-industry activity is enormous, but there are no consistent requirements across deployments in terms of security.

  A first step toward achieving security in the IoT is establishing a secure environment within an IoT device in order to store and manage sensitive data (e.g. cryptographic keys) and perform sensitive operations (e.g. authentication or cryptographic operations). A GlobalPlatform-certified TEE would be a logical and effective approach to achieving this security.

- **Corporate Use Case:** A Trusted Application can support the deployment of Corporate Services. In cases where the Corporate Services rely on a VPN, credentials need to be securely stored in the memory managed by the TEE. Additionally, sensitive documents managed in the framework of office tools should be stored securely in the same kind of memory. One possible architecture that would provide a solution would be to have a Trusted Application, stored and executed in the TEE, which would be in charge of VPN channel establishment and access control to sensitive documents. Those Trusted functions, implemented in the Trusted Application, would be accessed thanks to a TEE Client API running on the Rich OS.

  Using the GlobalPlatform TEE Administration Framework, such a Trusted Application could be remotely deployed and personalized through a cloud-based Trusted Service Manager on behalf of the corporate services, to specific devices, or more directly through physically authorizing tokens inserted into the target device.

3.2. Security Perspective from Different Actors

Just as each market has different dynamics, the *actors* within markets have unique security and business requirements. While such actors will vary across markets, consider the following major players' security concerns:

Content and Service Providers: To protect their premium content from theft, content owners and rights holders have traditionally made use of content protection, Digital Rights Management (DRM), or Conditional Access (CA). As consumers have increased their consumption of video, music, and other media across multiple devices, content and service providers have lost much of the ability to control the full device so as to ensure complete content security. Content and service providers want to ensure that their content is secure, but they need to do so while enabling consumers to get access to content as easily as possible.

As such, content and service providers are contending with a proliferation of devices and distribution channels—all the while needing to ensure privacy; strong authentication; quality of service, even with limited network connectivity; protection of value-added content; deployment of services in an open environment; compliance with security regulations, and more. In this

context, a controlled and Trusted Execution Environment becomes a key demand from content and service providers to protect their businesses.

They must do so, however, while ceding device control to the end user. In turn, they must make use of the trust mechanisms made available on a particular device and ensure that content remains secure nonetheless.

Mobile Network Operators: MNOs are reliable partners for service deployments on smartphones. The use of a UICC (usually owned and managed by the MNO) is relevant for deploying some services, but some applications exceed the resource capabilities of the UICC. For these instances, there is a need to have a higher level of security than what the Rich OS offers in order to protect their assets, their partners' assets, and their customers' information.

For MNOs, the TEE delivers a higher level of security than what Rich OS offers and higher performance than what a Secure Element (SE) typically offers. In essence, the TEE ensures a high level of trust between the device, the network, the edge, and the cloud, thereby improving the ability of a MNO to enhance services for Root Detection, SIM-Lock, Anti-Tethering, Mobile Wallet, Mobile as PoS, Data Protection, Mobile Device Management, Application Security, Content Protection, Device Wipes, and Anti-Malware Protection.

MNOs also require secure management and deployment of applications to minimize security risks, as well as consistencies in secure service development and deployment through standards so as to avoid fragmentation.

OS and Platform Providers: OS Providers have a constant need for regular updates, even at the critical level of the device "root of trust." Furthermore, there is a need to support a growing number of secure applications across different markets. However, these desires are in a continual race with hackers that wish to break the platform entirely. The costs of this security battle have increased as applications have hundreds of variations per product; one approach to resolving this has involved proprietary hardware solutions, but such customization has increased the time required to launch new products and is thus less than ideal.

A standardized TEE would reduce the need for such customized hardware initiatives. Instead, hardware vendors could use their own platforms while porting essential security features, allowing them to compete on performance and allow solutions to more rapidly come to market. Standardization will allow for healthy competition between hardware vendors without the loss of platform security features at the software level. The end result for OS and Platform Providers is that the transparency afforded by this standardized approach will improve trust among service providers and governments—across a variety of operating systems and platforms.

Application Software Developers: Fragmented security frameworks and proprietary APIs complicate the job that application developers do. By contrast, an approach that enabled commonly-deployed security frameworks and associated software and APIs would improve efficiencies and security.

Device Manufacturers: With mobile devices handling increasing amounts of personal and sensitive information (personal and corporate data, including user, corporate and device credentials), there is increasing demand for a robust security model built into the device itself to protect, isolate, and manage data and credentials. The objective is to increase the trustworthiness of the device itself, which will improve the user experience for consumers and service providers alike. And, a more secure device is required to meet the needs of other stakeholders important to the device manufacturer: MNOs, service providers, application developers, enterprises, and legislators.

Use of standardized GlobalPlatform APIs allows new applications to be easily developed and deployed by third party developers. It also facilitates remote installation and management. With a broader number of applications developed via such APIs, the "trusted" nature of the TEE increases: service providers and stakeholders across the ecosystem can have an increased level of confidence in GlobalPlatform-based implementations.

Hardware and Silicon Vendors: The TEE enables hardware-backed security for key hardware and software assets and resources. TEE standardization allows vendors to differentiate the value-added security features of their platforms while maintaining a level of interoperability that allows broad ecosystems to develop across market verticals. TEE Certification guarantees that the various hardware platforms meet the necessary level of security that the TEE requires.

Though different actors contribute different pieces of the value chain, it is a fundamental best practice that security protection is provided end-to-end since security is only as strong as the weakest link of the end-to-end solution. Confidence and trust are paramount to the adoption and growth in the handset market and mobile services.

Regardless of the actor being discussed, and regardless of the market being addressed, one thing is certain: security is critical. As is alluded to above, the TEE provides a path to resolving these security needs while still enabling the key performance that is required.

To better illustrate the TEE benefits, the next section will explore a number of use cases—each with distinct needs that can be met by the TEE.

SECTION 4: Detailed Use Cases

The three previous sections define the TEE, differentiate it from alternatives, and explore how different actors benefit from the security offered by a TEE. While this section is by no means exhaustive in its examination of TEE implementation examples, it seeks to illustrate particular use cases so as to better understand how a TEE can address major concerns within such use cases.

4.1. Mobile Payments

Mobile payments technologies are rapidly evolving, with payment systems seeking to extend their trusted payment applications via both existing and new technologies. Their hope is to improve the value proposition for issuers, merchants, and the end user alike.

Consider Host-based Card Emulation (HCE) as one example. HCE enables mobile applications to emulate a smartcard without the need for a Secure Element. The payment applet is implemented as a Trusted Application and installed within the TEE on the consumer device. During a transaction, the mobile application interacts with the trusted payment application and then communicates via the consumer device's contactless communication channel (e.g., NFC and Bluetooth) to pass the payment data to the POS terminal. Instead of personalizing credentials on an SE, they can be secured with the trusted payment application within the consumer's device.

The TEE can also augment an SE by delivering security services for functions that do not require the security protection that an SE offers. The TEE can set up a secure communication channel with the SE (embedded SE, UICC, or SmartSD), thereby protecting the confidentiality and integrity of the messages exchanged. In this scenario, the SE protects critical security assets such as long-term cryptographic keys, and performs security critical functions such as providing limited use keys and information to the TEE. The TEE can make use of these dynamic keys, credentials, and tokens to perform single-use payment transactions that prevent fraud by preventing re-use. This arrangement allows for use of fast cryptographic functions to securely process limited use information: On-device verification (e.g., PIN, biometrics) can be securely provided, processed, and displayed. Other advantages include higher space memory size and data transfer.

Mobile payments require secure interaction between the end user and the consumer device. For example, the user validates sensitive information displayed on-screen and enters sensitive information (such as a passcode) via keyboard. This is problematic on an untrusted consumer device with an unsecure interface: sensitive information could be exposed via malware or key or logging devices. The TEE protects sensitive information by providing features such as a Trusted UI that allows for secure input and display, as well as a Trusted Internal Core API for secure storage, fast cryptographic processing, and secure communications.

These capabilities reduce the risk of passcode logging and allow transaction, logs, and statement information to be securely displayed. Trusted PIN entry is especially important for use cases such as high-value transactions that require verification by the trusted payment application. Furthermore, trusted on-device biometrics (such as a

fingerprint) could be leveraged to maximize user convenience or improve transaction speeds, which is important for several applications, including transit.

The TEE further allows for secure remote management and communication via a protected channel across the Internet or contactless interface, which allows for Over-the-Air (OTA) TEE or SE application loading, provisioning, updates, and/or secure distribution of sensitive data and key credentials. The Administration Framework standardizes TEE remote administration systems; it further provides online and offline methods to manage the TEE, its Security Domains and Trusted Applications, and the associated secure data.

Finally, payment schemes can rely on TEE security certification to provide stakeholder confidence that trusted applications are running on a trusted platform that has been independently evaluated and certified.

4.2. The Enterprise & 'Bring Your Own Device'

Modern enterprises are caught between two conflicting forces—security versus mobility. There is an increasing awareness of the need for data security, not just in traditionally regulated industries (such as health care and banking), but also in the wider commercial world. Recent examples, such as the late-2014 hack of Sony's systems, have shown that it not just banking or personal information that is being sought, but also information once considered trivial—such as internal email—that can damage corporate reputations if leaked.[4]

At the same time, however, enterprises need employees to be more mobile. Gone are the days when a meeting's decisions are typed up and later circulated via memo: today's enterprises need employees to have real-time access to key information wherever they are.

This is further complicated by the move to a 'Bring Your Own Device' model, where employees choose which devices they prefer and use. Enterprises benefit from this model in several ways, including reduced upfront deployment costs (since the employee pays for the device) and increased employee accessibility since enterprise applications run around-the-clock on employees' devices. Employees benefit because they get their choice of devices, do not have to carry more than one device, and have increased workplace flexibility regarding their physical location.

These benefits are not without costs, however. The BYOD model means that enterprise applications must be deployable across a heterogeneous set of devices—devices that are ever-evolving and with an ever-expanding set of applications. And, practically speaking, testing enterprise applications across all conceivable devices is infeasible.

The BYOD model also imposes a different ownership over the device: when an enterprise deploys devices to employees, it can rigorously test its application and

[4] While controversy surrounds those supposedly responsible for this attack, the scale of the breach and the reputational damage inflicted underscore how relevant security concerns are for industries outside of banking and health care. As one example, see http://www.cnet.com/news/13-revelations-from-the-sony-hack/, accessed 11 February 2015.

determine which types of applications are or are not allowable on the device. It can deploy Mobile Device Management (MDM) if desired. With the employee as owner, the enterprise must accept the employee's autonomy and right to install far-reaching applications—from efficiency tools, to social media applications, to games, and more.

Third, the BYOD model raises bidirectional security concerns between the enterprise and its employees. Employees, no doubt, have personal information that they would not wish to share with their employer—from family photos, to banking or health care information, to an email from another prospective employer. Thus, while the enterprise wishes to protect its information from rogue attacks, it must accept that employees will wish to safeguard information from their employer.

All of these concerns increase security requirements for both users who want to install their own, private applications and enterprise IT departments concerned with preventing corporate information from being hacked. Traditional enterprise approaches to securing applications and data, such as (MDM), cause friction with users, who in turn find ways to work round the security in order to get their jobs done. An enterprise's objective, then, should be to securely deliver data to workers without encouraging them to subvert policy—something that is enabled by leveraging a TEE.

The only solution is to have widely adopted standards and compliance testing to prove that devices are in compliance. This gives confidence that an application tested on one compliant device will perform similarly on any other compliant device.

In this multi-modal world, the ability of the TEE to manage separate security domains and ensure data separation between them is vital. The GlobalPlatform TEE maintains separation between running Trusted Applications as well as between TAs and the REE. The Trusted Application Framework enables different Security Domains to be controlled by different off-device entities. It provides for different permissions for different operations. Consider, for example, that the MNO may own a device that it is leasing to a consumer. As the owner, it would retain the right to install trusted applications within its own Security Domain, as well as the right to factory-reset the device, thereby deleting all other Security Domains and trusted applications. This would allow the MNO to pass the device to a new customer. However, these rights would *not* give the MNO the right to install or remove individual trusted applications that reside in other Security Domains.

A TEE implementation can also ensure that data stored on the device is protected in the event that the device is lost or stolen. Consider that offline attacks on data can often crack passwords, but a TEE implementation can prevent this by keeping the encryption keys within the trusted environment and limiting the rate at which passwords can be guessed. Employee credentials can also be stored safely in the TEE Trusted Storage, or even in an embedded Secure Element for further tamper resistance of corporate IDs.

A TEE can also ensure a reliable communication channel between the mobile device and the corporate infrastructure. To be properly secure, the corporate server needs to be certain that it is communicating with the correct device. By keeping a device's cryptographic identity inside the TEE and then establishing a tunnel to the TEE itself—for example using the TEE Socket's API—the security of the communication channel can be ensured.

Finally, consider that the TEE also protects data on the device from malware. The Trusted User Interface enables users to enter their PINs or passwords on a screen that is under the control of the TEE and therefore kept well away from any malware that might have accidentally been installed. Future releases of the TUI will increase its functionality, and it may become practical to display some documents, which would ensure that the most sensitive data never leaves the TEE.

4.3. Content Protection: Media

The media industry is complex and encompasses a large number of stakeholders, each of which possesses different expertise and has different business objectives. Consider just a few of the stakeholders impacted by content protection efforts:

- **Content Rights Holder:** This is an organization with exclusive rights to content and/or to the related rights of producers, performers, and broadcasters. Examples include Hollywood Studios, Broadcasters of Live content, and more.

- **Content Service Provider:** This is an organization that distributes content via the Internet, DVDs, CD-ROMs, or other software-based products. Examples include Cable, Satellite, and Internet Protocol Television (IPTV) Broadcasters; Internet streaming broadcasters; and Mobile Network Service Providers.

- **Content Aggregator:** This is an organization that combines content from various sources and makes it available to customers. Examples include Google and Facebook.

- **DRM Content Protection Solution Provider:** This is a vendor that provides the complete device and server-end specifications defining how to protect distributed content. Examples include OMA, Intertrust, Google (Widevine), and Microsoft (PlayReady).

- **OEM and SoC Platform Providers:** An Original Equipment Manufacturer (OEM) is an organization that makes devices from SoC (System-On-Chip) component parts acquired from SoC Platform Providers.

Content Rights Holders establish rules to define which security features they require of Content Service Providers. Historically, Content Rights Holders have typically agreed to allow their content (up to 720p quality typically) to be distributed on Consumer Electronic devices; as long as DRM was implemented by using software-based obfuscation and protection techniques, Content Rights Holders were typically comfortable with software solutions running on the Rich OS. As users demand even higher quality video on these devices (including Full HD and Ultra HD), Content Rights Holders have increased their security requirements on Consumer Electronic devices. They increasingly want to prevent content from being shared on the Rich OS environment. This is where the TEE provides an ideal solution.

Stakeholders within the media ecosystem can leverage a TEE on devices to implement secure keys and secure storage, as well as perform security-sensitive operations like cryptographic decryption in an environment where the data path is secured. This enables an enhanced secure execution environment expected by the content owners for Full HD and Ultra HD content.

However, TEEs currently on the market are fragmented, with various sets of proprietary APIs, and this makes scaling video services difficult. The GlobalPlatform TEE brings the standardization that would clear this API fragmentation, thus easing deployment reducing costs. The result of all this would be an increased level of content protection while retaining a premium user experience and easing service deployment.

Within the media ecosystem, there are countless "mini use cases" where the TEE can aid in content protection while preserving the user experience. Consider the following examples:

- **Playback of premium content from a broadcast network:** In this use case, a Service Provider broadcasts premium content that is protected with a protection system provided by a Conditional Access System (CAS) provider and rendered on a display device (such as a set-top box or TV). The Service Provider provides the protection mechanism on the display device that enables secure distribution and rendering of the content. This is achieved via a TEE, through either pre-integrated or downloaded TEE Trusted Applications (TAs).

- **Playback of premium content using streaming technologies:** A consumer or mobile device can receive content provided directly from a Service Provider's servers. This content is delivered using streaming technology and protected via TEE TAs that have implemented the necessary DRM scheme for decrypting and rendering encrypted streaming content.

- **Playback of stored premium content:** A consumer or mobile device can store a Service Provider's content suggestions regarding what is available for download. While content could be protected using a DRM system, it may be unwrapped using license keys that are stored in the TEE's Trusted Storage. Depending on the business model, the license keys may expire and could thus be checked against the TEE Secure Clock.

- **Playback of premium content using an application Over-The-Top:** This is the instance where an end-user selects an application that requires leveraging DRM different from what is compatible with the device. With a TEE, the application would begin downloading the new DRM's TA after confirming that the device's certification level is suitable for content being accessed.

- **Storage of premium content received in a Personal Video Recorder connected to the local network:** A Personal Video Recorder (PVR) on a local network can temporarily store content for delayed playback. The exchange of the content from one device to another is protected using an approved copy protection mechanism. A TEE is able to check the rights associated with the content to enable the content to be distributed to other devices.

- **Transmission of premium content from one device to another display device:** A user may want to watch content stored on a primary device on a secondary display. This includes, for example, streaming content from a living room set-top box or PVR to a TV in a bedroom. In this use case, the secure video path includes TEE TAs that implement secure protocols, such as High-bandwidth Digital Content Protection or Digital Transmission Content Protection (HDCP or DTCP, respectively).

- **Playback of premium content for which the protection system is no longer available on the device:** After a user changes devices or Service Providers, s/he may still want to watch the previously owned or purchased content, but the playback may fail because the DRM from the previous device is no longer available. In this instance, the TEE TA can confirm the device's certification level and subsequently download the required DRM scheme.

- **Playback of different premium content, each protected by different protection systems:** Whether at home, in a car, or elsewhere, users may wish to playback content on different screens, and different content may leverage different protection mechanisms. During content rendering, these protection applications run simultaneously. The TEE ensures isolation so that assets from one protection system are not be available to the other protection system.

In closing, while there are several different use cases for content protection, the TEE can ensure proper access without disrupting the user experience. It should be noted that, in all of the above use cases, the TEE can also aid in authentication when required: should user authentication and validation be required, the TEE Trusted UI can ensure protection against illegitimate access.

4.4. Governmental Use Cases

There are obviously countless potential examples for leveraging a TEE across various governmental agencies. For the purpose of this document, we would like to explore one general type of implementation that applies across multiple governments (eID smart cards), as well as a unique look into a specific government's TEE implementation—that of the United States Department of Defense.

4.4.1. Securing Governmental eID Solutions

An electronic identity smart card (eID)[5] is a government-issued identification that aims to provide citizens with higher levels of security and authentication. Applications vary by issuer, but they include identity verification at points of entry, for government services, or to satisfy other authorities' requirements. eID issuance has dramatically increased during the last decade.

Proponents tout eIDs as beneficial to both citizens and governments because of their enhanced security features and convenience. Convenience is the primary driver behind many new applications, including online eGovernment services, as well as services available via mobile devices.

From a security perspective, it is important to note that eIDs are based on Secure Elements, which means that the data stored within the eID itself (such as the user's identity, keys, and certificate information used for authentication and authorization) is highly protected. However, there are other sensitive user assets (such as the user's PIN and service information), that are not stored within the Secure Element. The TEE is ideally suited to protect these assets; Table 1 illustrates how different types of data are protected within an eID that also implements a TEE.

[5] eID is sometimes referenced as "EIC," meaning "Electronic Identity Card." While this paper uses the former acronym, they are the same.

Asset	Level of required Protection	Entity hosting the asset
User Data (e.g. Name, ID)	Strong (AVA VAN 5)	eID Token
Keys and certificates	Strong (AVA VAN 5)	eID Token
2nd Factor (e.g. PIN, biometrics)	**Enhanced** (AVA VAN 3)	**TEE-based Terminal**
eService Data	**Enhanced** (AVA VAN 3)	**TEE Trusted UI**

Table 1 - User assets in the context of eGovernment

As Table 1 shows, a user's most sensitive information is protected by the eID. However, to access this information for authentication and identification, the cardholder is typically asked to present a PIN code, which needs to be kept confidential. In PC-based environments, the PIN can be secured by using smart card terminals equipped with a keypad and display. However, in mobile device-based scenarios, such as those where the NFC (Near Field Communication) interface is used to communicate with the eID, using an external card terminal is often not feasible. In this case, the TEE can effectively serve as the smart card terminal by providing the necessary functionality as part of a dedicated Trusted Application. See Figure 4.

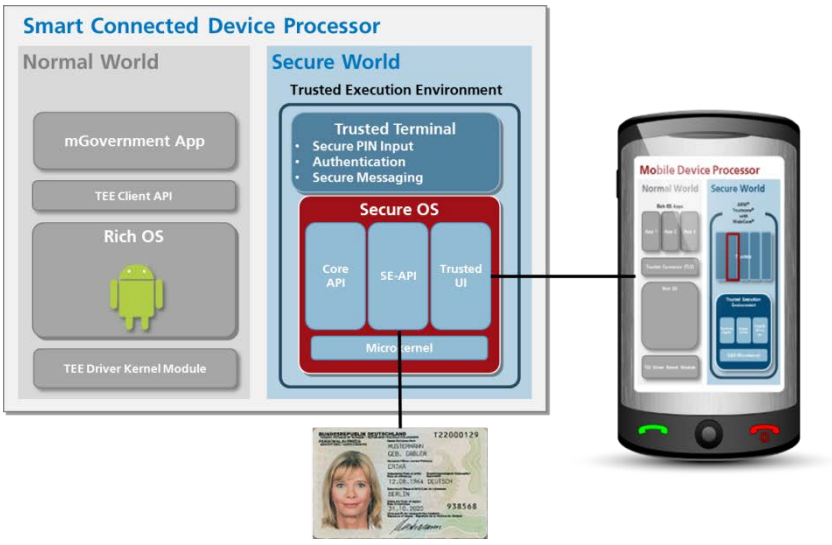

Figure 4 - Using the TEE as a trusted eID terminal

It is important to note that most eID solutions use a secure messaging communications channel to secure the contactless communication channel. For example, the German eID uses the Password Authenticated Connection Establishment (PACE) protocol to secure the contactless channel and to gain user consent.

In such an implementation, the TEE serves the following security functions (with applicable GlobalPlatform Specifications referenced in parentheses):

- eID components can be installed in the TEE, in a trusted way (Administration Framework)
- The cardholder can securely input his/her PIN (Trusted UI)
- Important user information can be retrieved from the eID and securely displayed to the user (Trusted UI)
- The authentication protocol can be securely executed, and messaging keys can be securely derived (TEE SE API, Internal Core API)
- The secure messaging scheme can be executed (TEE SE API, Internal Core API)

The TEE provides the best way for eID implementations to ensure that all user information is protected, including user inputs (such as PINs) and data recalled from the Secure Element. By contrast, eID implementations that do not leverage a TEE are potentially exposed in these areas.

4.4.2. U.S. Department of Defense: High-Security Use

The basic DoD usage model[6] involves employee access to enterprise capabilities from a mobile device. Although various government agencies are leveraging enterprise owned devices, most mobile devices are very limited on the services provided and are restricted by locked down policies. While frequently called "Corporately Owned, Personally Enabled" (COPE), aside from this usage taking place with governmental employees, the requirements and concerns are largely similar to what was discussed in Section 4.2 for the Enterprise.

What is different, however, are uses cases involving high-security use. Within the DoD, this often involves an enterprise-owned device with intentionally-limited network connectivity, tightly-controlled configuration, and limited software inventory that is appropriate for specialized, high-security use cases. For example, the device may not be permitted connectivity to any external peripherals. It may only be able to communicate via its WiFi or cellular radios with the enterprise-run network, which may not even permit connectivity to the Internet. Use of the device may entail compliance with policies that are more restrictive than those in any general-purpose use case, yet may mitigate risks to highly sensitive information.

In addition to these known, current use cases, there are other use cases that will increasingly benefit from TEE implementations. Two examples are as follows:

- **Government Official Data Sharing:** There is a need to share highly sensitive data with specific users in times of national crisis, but this is a challenge with current mobile solutions. What is required is a way to control when and how

[6] Current government use cases are best described by the NIAP Generated Use Cases in the Mobile Device Protection Profile. The set of security requirements called out by the government are listed in the Mobile Device Fundamentals Protection Profile on the NIAP web site at https://www.niap-ccevs.org/pp/pp_md_v2.0.pdf.

long the data is available. The TEE, in conjunction with an SE, can be instrumental in identifying the device and individual, storing user and device credentials, storing keys necessary to protect the transmission and storage of critical information, allow viewing of the data, and ensure controlled removal of the data from the device.

- **Use of Untrusted Apps on Trusted Mobile Device:** The government would like to use commercially developed apps at minimal risk. However, the current app development and deployment model cannot prevent malicious code from being introduced onto the device. While the U.S. government has established security requirements for application testing, the cost to test and validate applications has prevented application vendors from doing this. An alternative is to host critical apps in isolated application environments.

In both the general and high-security use cases described above, a TEE can be used in conjunction with an SE to provide core functions and validate needed security controls. These common functions include the following:

- Device Integrity
- Secure ID/Credential storage
- Secure ID validation
- Crypto Services
    - Secure Key Storage
    - Signature generation and validation
    - Data Encryption
- VPN Services
- Secure Document/Data Viewer
- Trusted UI

Beyond these general functions, the TEE can help to support other use cases and functionality that are important to the DoD:

- **Unclassified—For Official Use Only (U//FOUO) Device Identity:** (U//FOUO) Ensure that only authorized devices are allowed to connect to the network. As such, the identity of the device must be verified. Using a key as identity, which is protected in the device's hardware (TEE/SE), will ensure that the identity of the device cannot be spoofed or stolen.

- **Secure Key Storage:** In order to prevent unauthorized access and theft of secrets on a mobile device, a hardware-based (TEE/SE) keystore is required.

- **Data Protection:** Data that is stored on a device or is transmitted should be protected. The data must be protected when the device is not being used (sleep state) or is turned off.

  Crytpo Erase: There are several situations in which the information on a device should be fully erased. These situations include but are not limited to re-

purposing of the device, disposing of the device, and theft. Consider that there are three scenarios in which a wipe may need to be initiated: local, remote and time-based. A local signal would be triggered by the user if the user senses a possible dangerous situation. A remote signal would be sent from the mobile device manager in such cases as loss of the device or when a device tries to connect but is not in a known good state. The last case, a time-based signal, would be initiated if the device does not connect to the enterprise within a specific amount of time.

However, a full wipe of information in non-volatile storage is not only time consuming, but also difficult to accomplish on mobile devices. And yet, if the device is protected by an encryption key that is stored in hardware, then removing or deleting the encryption key (cryptographic erase) will make the data unreadable. Thus, removal of this encryption key effectively accomplishes the "wipe" desired.

- **Device Health:** Before a device is allowed trusted access to the enterprise, the enterprise needs to ensure that the device is in a known good state (sometimes referred to as being in a "healthy" state). If a device is not healthy it may introduce backdoors into the enterprise and result in a loss of important data. A trusted boot should attest not merely to the early stages of booting of the kernel of the operating system, but also to the healthy presence of software that will continue monitoring the state of the full system's health.

The DoD has other areas of interest as well including Proximity locking, device monitoring for malicious behavior, traffic limiting to prevent data loss, application whitelisting, and process isolation.

SECTION 5: Why Standardize the TEE (proprietary vs. standard)?

Standardization of the TEE is key to both avoid fragmentation of APIs and protect differentiation. Fragmentation would lead to the proliferation of non-compatible, proprietary security features, applications, and management systems platforms. This fragmentation would in turn lead to the following:

- Higher costs to develop or change applications/solutions when creating or adapting to proprietary platforms
- The need for very specialized skills
- Extended time-to-market due to longer development times and potential integration issues

Standardization, by contrast, enables simplified and unified implementation, limits complexities, and improves interoperability between stakeholders. Furthermore, standardization enables a large ecosystem to thrive and blossom, allowing for multiple business partners and, because it ensures long-term stability and survivability, protects investment in a way that proprietary solutions cannot. It also defines a basis for evaluating and comparing different solutions. Lastly, standardization creates a foundation for a certification process.

Created in 1999 to standardize smart card infrastructure, GlobalPlatform card specifications are now embedded in more than 5 billion Secure Elements. As a recognized standards body, GlobalPlatform represents the full ecosystem, including chip manufacturers, IP providers, software developers, OEMs, network operators, service providers, certification laboratories, and more.

Following its OMTP and TCG standardization efforts, GlobalPlatform's Device Committee delivered the TEE Client API 1.0 specification in July 2010. The Committee is now actively working on the specification for the TEE Internal API, as well as higher-level functional APIs for the TEE Client API.

SECTION 6: Conclusion

There are today increasing security concerns resulting from wide usage of mobile, consumer, enterprise, and wearable devices, and the TEE offers the market a solution that addresses many of these concerns without imposing an undue burden on applications.

The TEE is an isolated execution environment that runs alongside the Rich OS and provides security services to that rich environment. This is accomplished while protecting and isolating access to hardware and software security resources from the Rich OS and its applications.

The TEE protects the assets that fall between a Secure Element and Rich OS. It provides robust, hardware-backed, scalable-consistent, OS-independent security. Furthermore, it offers device features and performance that cannot be delivered by a Secure Element.

As discussed throughout this document, the TEE is applicable across a variety of industries and use cases. Also noted, to maximize this value standardization is required, and to this end, GlobalPlatform continues its work to publish specifications that will enable this broad-scale value to be achieved throughout the market.

APPENDIX A: Abbreviations

Abbreviation	Meaning
CA	Conditional Access
CAS	Conditional Access System
CC	Common Criteria
CDVCM	Consumer Device Cardholder Verification Method
CMLA	Content Management License Administrator
CPRM	Content Protection for Recordable Media
DoD	Department of Defense
DRM	Digital Rights Management
DTCP	Digital Transmission Content Protection
EAL	Evaluation Assurance Level
eID	Electronic Identity
eSE	Embedded Secure Element
HCE	Host-Card Emulation
HD	High-Definition
HLOS	High-Level Operating System
HDCP	High-bandwidth Digital Content Protection
IoT	Internet of Things
IP	Internet Protocol
IPsec	Internet Protocol Security
IPTV	Internet Protocol Television
MDM	Mobile Device Management
MNO	Mobile Network Operator
MFS	Mobile Financial Services
NFC	Near Field Communication
OEM	Original Equipment Manufacturer
OMTP	Open Mobile Terminal Platform
OS	Operating System
OTA	Over-the-Air
OTP	One-Time-Password
PACE	Password Authenticated Connection Establishment

Abbreviation	Meaning
POS	Point Of Sale
PVR	Personal Video Recorder
REE	Rich Execution Environment
ROM	Read-Only Memory
SE	Secure Element
SIM	Secure Interface Module
SoC	System-on-Chip
SSL	Secure Socket Layer
STB	Set-Top Box
TA	Trusted Application
TCB	Trusted Computing Base
TCP	Transmission Control Protocol
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TSM	Trusted Service Manager
TUI	Trusted User Interface
U//FOUO	Unclassified—For Official Use Only
UDP	User Datagram Protocol
UI	User Interface
UICC	Universal Integrated Circuit Card
VPN	Virtual Private Network

APPENDIX B: Definitions

Rich OS:

A High-Level Operating System (HLOS) environment with a rich capability set; further, it allows consumers to download and run applications. Android™, Linux®, Symbian OS™, and Microsoft® Windows® Phone 7 are examples of a Rich OS.

Secure Element (SE):

A tamper-proof combination of hardware, software and protocols capable of embedding smart card-grade applications. Typical implementations include UICC, embedded SE, and removable memory cards.

Trusted Execution Environment (TEE):

An isolated execution environment that runs alongside the Rich OS. The TEE provides security services to that rich environment and isolates access to its hardware and software security resources from the Rich OS and its applications.

APPENDIX C: Comparing Rich OS, TEE, and SE

The table below summarizes security and computational facilities offered by typical implementations of the three environments, in order to identify their fundamental differences.

	Rich OS	TEE	SE
Application download controlled by	User choice.	Authorization process.	Authorization process.
Application code	Typically un-validated and uncertified.	Typically validated and certified before authorization, and in authorization checked on loading.	Typically validated and certified before authorization, and in authorization checked on loading.
Isolation	Limited by Rich OS capabilities – some Rich OS may provide a sandbox model (e.g. Java VM) or support virtualization	The TEE is separate from the Rich OS – the depth isolation relies on the strength of the TEE implementation	Isolated physically – runs a separate OS (e.g. JavaCard, STIP, etc.)
Certification	Uncertified	Certified	Strongly certified
OS Kernel, driver and library code	Created for flexibility and speed	Created for security and speed	Created for security
	Rich API set	Limited API set	Very limited API set
	Typically large RAM size	Typically medium RAM size	Typically small RAM size
Confidential and integrity of access to user interface devices (Keyboard, screen, audio I/O)	Within the limits of the Rich OS capability	Confidentiality- and integrity-bounded by the TEE (the TEE can have access to user interfaces which are isolated from the Rich OS)	Only indirectly, and so bounded by delegation from an external enabler such as Rich OS or TEE.
CPU speed	GHz range	Hundreds MHz to GHz range	Few to 20 MHz range
Cores	1->4	1 master	1
RAM size	16MB->1GB+	64KB to many MB secure	A few 10's of KB
RAM speed	64 bits @ 200Mhz -> 800Mhz	64 bits @ 200Mhz -> 800Mhz	32 bits @ 5Mhz (limited by power)
FLASH size	1GB-> 32GB + (incl SD cards, etc.)	shared with Rich OS – each Trusted	64KB-> 1 MB

		Application may have its own secure storage	
Data Transfers with Rich OS	Very fast	Very fast	Slow
Protection against unauthorized software attack, including software making "illegal" use of hardware on the device	Confidentiality internally protected by non-certified OS	Confidentiality Protected vs. Rich OS and device hardware. Internally protected by certified OS	Confidentiality Protected vs. external software and device hardware. Internally protected by certified OS
	Limited integrity protection during boot (Typically Kernel only)	Integrity Protected vs. Rich OS and device hardware. Internally protected by certified OS	Integrity Protected vs. external software and device hardware. Internally protected by certified OS (and other mechanisms?)
Protection against external hardware attack	no protections against attacks limited anti-rollback protection	Protection depending on TEE implementation mechanisms and hardware features of hosting platform	Strong protection for SE but not for hosting device
Protection guarding the device, i.e. preventing the device from being unlocked or flashed with unauthorized software	Optional secure boot	TEE mandates secure boot	Often trivially removable from the device by user or attacker. Any linkage with the device can only be as strong as the security guarding the weakest part of that link (typically on the device and typically the weak point being the Rich OS or TEE)

APPENDIX D: Table of Figures