Project acronym:	**TRESCCA**
Project title:	TRustworthy Embedded systems for Secure Cloud Computing
Project number:	European Commission – 318036
Call identifier:	FP7-ICT-2011.1.4
Start date of project:	01 Oct. 2012 **Duration:** 36 months

Document reference number:	**D1.1**
Document title:	Security analysis of cloud applications and specification of security objectives
Version:	1.2
Due date of document:	31th of March 2013
Actual submission date:	5th of May 2014
Lead beneficiary:	IMT
Participants:	Guillaume DUC (IMT), Renaud PACALET (IMT), Ignacio GARCÍA VEGA (WT)

Project co-funded by the European Commission within the 7th Framework Programme		
DISSEMINATION LEVEL		
PU	Public	**X**
PCA	Public with confidential annex	
CO	Confidential, only for members of the consortium (including Commission Services)	

EXECUTIVE SUMMARY

The security of cloud computing applications is a very topical subject. The aim of the TRESCCA project is to develop a secure cloud computing platform that is resistant against software and hardware attacks.

In order to design this platform, we at first need to have a common understanding of the different typology of cloud computing infrastructures and applications (from Infrastructure-as-a-Service to Software-as-a-Service). In the next step, by describing a typical application, we identify the entities involved, the assets manipulated, the security properties to guarantee on these assets and the attacks that threaten them.

Then, by analyzing the existing security solutions and their weaknesses, we will define the security objectives and the threat model that will be considered in the scope of the TRESCCA project. The security objectives are based on the concept of trusted compartment. They comprise five different objectives:

- Existence, confidentiality and integrity of a Trusted Compartment.
- Robustness of the TRESCCA platform against software denial of service.
- Attestation upon request of the integrity of the TRESCCA platform.
- Authentication of computing tasks delegated to the TRESCCA platform.
- Integrity of the TRESCCA platform in case of software upgrade and from boot to boot.
- Proper isolation between applications running on the TRESCCA platform.
- Robustness against software exploits and other logical attacks.

The threat model considers that the potential attackers of the light client of the TRESCCA platform are intermediate class attackers. The threats are thus limited to software exploits and on-board probing attacks against the TRESCCA platform. Side channel, fault and on-silicon attacks are not considered because many other research and development projects already deal with these classes of attacks and also because some of them are very sophisticated and out of scope of the intermediate class of attackers.

Finally, the hardware and software support for security is sketched. Two kinds of hardware extensions and their software cunterparts are envisaged:

- Support for virtualization.
- Protection of the external memory bus against on-board probing attacks.

CONTENTS

1 INTRODUCTION

1.1 Purpose of the Document

This document is deliverable D1.1 of the project. The main aim of this document is to establish the security objectives and the attacker model that will be considered during the course of the project.

In Chapter 2, the different cloud computing architectures are presented. In Chapter 3, a typical cloud application is detailed in order to present the entities involved, the assets manipulated, the security objectives to guarantee and the existing attacks. Finally, Chapter 4 introduces the security objectives of the TRESCCA cloud platform, the considered threat model and the envisaged hardware and software support for security.

1.2 Document Versions Sheet

Version	Date	Description, modifications, authors
1.0	2013-04-30	Version submitted to European Commission.
1.1	2013-06-21	Addition of the «Robustness against software exploits and other logical attacks» security objective.
1.2	2014-05-05	Fix spelling and language. Improvements of the content of Annex A following comments from the 1st Review.

2 TAXONOMY OF CLOUD COMPUTING ARCHITECTURES

TRESCCA project deals with some of the main security challenges identified in cloud computing. Several taxonomies of cloud computing can be found, most of them created from the perspective of vendors that are part of the landscape[5]. Within this section the common taxonomy of cloud computing is being established in order to have a common understanding and vocabulary.

2.1 Infrastructure as a Service (IaaS)

IaaS allows users to run any applications they prefer on cloud hardware of their own choice. Instead of buying and maintaining computer hardware, virtualized hardware (Storage, Computing Power and Networks) is rented on demand from the cloud. The user of IaaS installs and runs any applications needed on this virtualized hardware. Using computing infrastructure on demand from the cloud, instead of maintaining it locally yields multiple advantages: one-time applications and applications which are rarely used become affordable. The used hardware is scalable, i.e. a sudden increase or decrease of needed hardware is possible without problems. Unused capacities can be used by someone else. New software can easily be tested on a variety of (virtualized) platforms. Examples for IaaS are Amazon S3 for storage and Amazon EC2 for computing power as well as Microsoft SQL Azure (computing power, storage and network).

There are four different categories for IaaS (and also PaaS and SaaS) defined by the American National Institute for Standard and Technology (NIST) [4]. The private cloud is the most secure and costly option. In this category a specific number of physical servers are dedicated to a single customer. The cloud hardware is most possible separated from that of other users. In the community cloud category infrastructure is shared among several parties known to each other from a specific community. As infrastructure in the community cloud is less costly than a private cloud, it can also be dynamically scaled. This means that customers are able to increase or decrease the number of servers they are using and will only have to pay for it on a daily or even hourly basis. In the third category, the hybrid cloud, a mix of physical server and virtual server is rented on demand by the customer in the effort to reduce cost and to increase flexibility. Once again, the whole offer is dynamically scalable, with both dedicated and virtual servers able to be added or taken away as required. The last one is the public cloud, where a customer rents virtual server instances on demand. This means that customers share all of the servers used with other customers. Some companies still estimate this proceeding as too risky. However, cloud hosting is the lowest-cost form and by far the most technically and environmentally efficient form of IaaS. It results from the fact that cloud hosting allows an IaaS provider to run all of their physical servers in use to capacity and to close down those that are not required.

To emphasise such categorization schemes one can compare a similar differentiation of the Experton Group AG in 2012. They have observed the advances of IaaS market during the last three years. They claimed that today available IaaS offers are much more sophisticated and aligned to IT operations models of customers. Finally they proposed a classification among four different types.

- *Managed Cloud / Cloud Hosting* describes managed services on shared infrastructure offered by a

provider using enterprise grade Service Level Aggreements (SLAs).

- *Managed Private Cloud* describes managed services on dedicated infrastructure offered by a provider using enterprise grade SLAs.
- *Public Cloud* describes self serviced and unmanaged shared infrastructure offered by a provider using a Pay-as-you-Go or Pay-as-you-Use model with standardized SLAs.
- *Private Cloud* describes the establishment and operation of individual cloud infrastructures within an own data centre or a co-location.

The appropriate choice of an IT operations model is of tremendous importance for the successful transformation of a company. The reason lies in the determination of parameters and properties like scalability, SLAs and the opportunity of individual service adjustments.

2.2 Platform as a Service (PaaS)

Platform as a Service (PaaS) allows users to create their own cloud application using supplier specific tools and languages. The PaaS provider provides a programming environment and / or runtime environment which can easily be used via the internet for designing, developing and testing own software. The user does not have to invest into the necessary hardware or software nor their administration. Possible services within PaaS include tools for team collaboration and versioning as well as monitoring the security or middle ware services for data storage. Scalability and high availability of the service allow a flexible use for fast and efficient software development. In contrast to IaaS the user has no access to the operating system level. Instead of it, the service is accessible via APIs or web interfaces. The categories private cloud, community cloud, hybrid cloud and public cloud described above also apply for this service model.

There are different types for PaaS: Stand-alone development environments provide a general development environment for design, development and operation of web applications, which can then be used via an API or a graphical interface by others. Add on development facilities allow to adapt and customize existing applications. *Application delivery-only* environments support the operation of applications without development, debugging and testing tools. Open PaaS provides open source software which allow users to run applications choosing their programming language, OS, libraries and (virtual) servers. Windows Azure and Google app engine are two examples.

2.3 Software as a Service (SaaS)

In a *Software as a Service* (SaaS) scenario everything is decentralized in the private, community, hybrid or public cloud (servers, storage, software) and clients access the software provided by the cloud by using a web browser only. The service provider maintains hardware and selected software including updates and patches as well as data storage. The user does not pay for software licenses, instead a *pay per use* (e.g on a monthly basis) model is employed. Since everything is done inside the cloud, the user is able to access the service from anywhere. There are two types of SaaS: Either a single application (one version) is provided for all users. For scalability reasons the application is installed on multiple machines. Or the user is able to configure the software as needed and uses an individualized version. Google Docs and Sales-force CRM are examples of SaaS.

3 STUDY OF A TYPICAL CLOUD APPLICATION

One of the attractions of cloud computing is the cost efficiency afforded by economies of scale, reuse, and standardization. To get the full benefit of these efficiencies, cloud providers have to provide services that are flexible enough to serve the largest customer base possible, maximizing their addressable market. Unfortunately, it is often perceived that integrating security into these solutions is making them more rigid.

This rigidity often manifests in the inability to gain parity in security control deployment in cloud environments compared to traditional IT. This stems mostly from the abstraction of infrastructure, and the lack of visibility and capability to integrate many familiar security controls, especially at the network layer.

In this chapter, TRESCCA is willing to do a security analysis of a "typical" cloud application. In order to achieve this we firstly develop an analysis and identification of the different actors in a cloud environment and valuable assets to protect. After this step, we will analyze the security properties to guarantee on these assets.

3.1 Involved Elements

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable resources (e.g., networks, servers, storage, applications and services). The cloud model envisages a world where components can be rapidly orchestrated, provisioned, implemented and decommissioned, and scaled up or down to provide an on-demand utility-like model of allocation and consumption.

From the architectural perspective, there is much confusion surrounding as there are existing many definitions today. In TRESCCA we will analyse a high-level cloud architecture consisting on building blocks that will abstract the details about the equipment being used. The cloud architecture can be de-constructed in the following set of building blocks:

Clients These might be thin clients, smart phones, tablet computers, virtual desktops, or traditional PCs and laptops.

Client networks Regardless of the type of client, clients will need to connect to the data center via a network.

Data center abstraction In order to support a multi-data center environment for Disaster Recovery/Business Continuance (DR/BC) purposes or using a blend of internal/external cloud, a layer of abstraction needs to be implemented between clients and data centers.

Client/server data center networks Each data center requires a high-performance and a high-reliability LAN. To support application mobility between data centres or between internal and external clouds, high-speed connectivity between the data centres along with layer 2 extension between data centres and "virtualization-aware" intelligence is needed all the way to the network edge.

Server abstraction Most of the management of the physical resources and cloud servers takes place here: whatever will be delivered "as a service" needs a management layer that is located on top of the physical platforms.

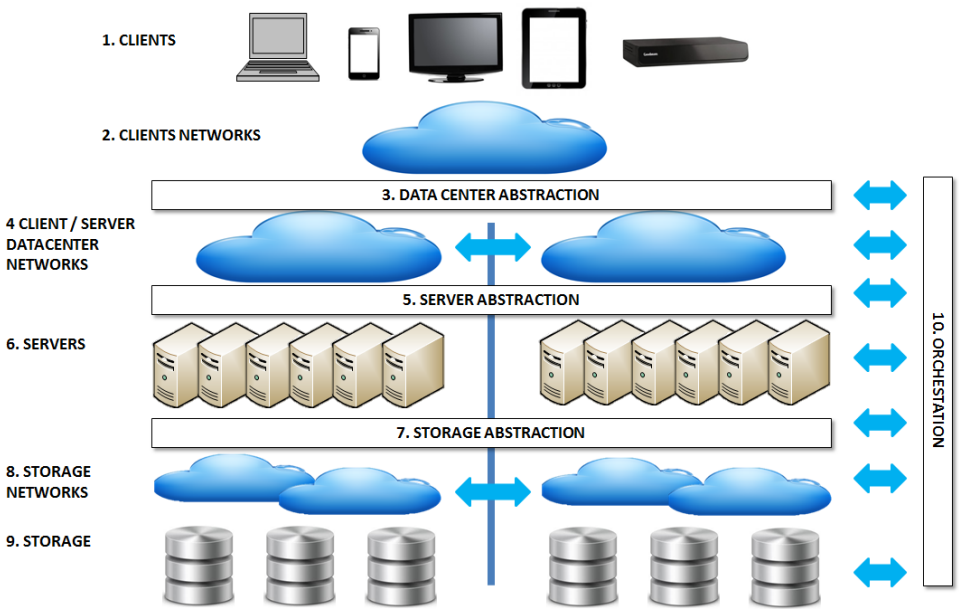

Figure 3.1: Elements involved in a typical cloud computing application

Servers At some point, virtual servers must reside on physical platforms. These will need to be very robust, as the value proposition of server consolidation does not work well if only a few Virtual Machines (VMs) can be deployed per platform.

Storage abstraction In order to support non-stop operations, VMs cannot be taken down whenever there is a need to upgrade a RAID array.

Storage networks All servers must have access to shared storage to support VM mobility. Any type of High Availability requires at least two physically separated Storage Area Networks (SANs) per data centre. This is a long-standing best practice, and there are very good reasons why SANs do not follow the LAN model of a single, fully connected network.

Storage Historically, SAN administrators used rule-of-thumb ratios of initiators-to-targets and similar strategies that relied on knowledge about the network endpoints' behaviour. Because endpoints can move dynamically, these strategies might break down, thus storage systems and SAN paths must be fast and reliable enough to support any application.

Orchestration Orchestration requires coordinating LAN and SAN behaviours - whether or not LAN and SAN traffic is converged.

3.2 Actors and roles

A new set of roles can be enumerated in cloud computing due to the increased service orientation and the opportunities of offering value-added and complex services that integrate different component services offered by different providers.

In this way, NIST whose scope includes Cloud Computing has published a number of documents that are considered as the baseline reference models for Government Cloud Computing. The document "NIST Cloud

Computing Reference Architecture (RA) and Taxonomy (Tax)"[3] accurately communicates the components and offerings of cloud computing.

The Overview of the Reference Architecture describes five major actors and their roles and responsibilities using the newly developed Cloud Computing Taxonomy. The five major participating actors are the Cloud Consumer, Cloud Provider, Cloud Broker, Cloud Auditor and Cloud Carrier. These core individuals have key roles in the realm of cloud computing.

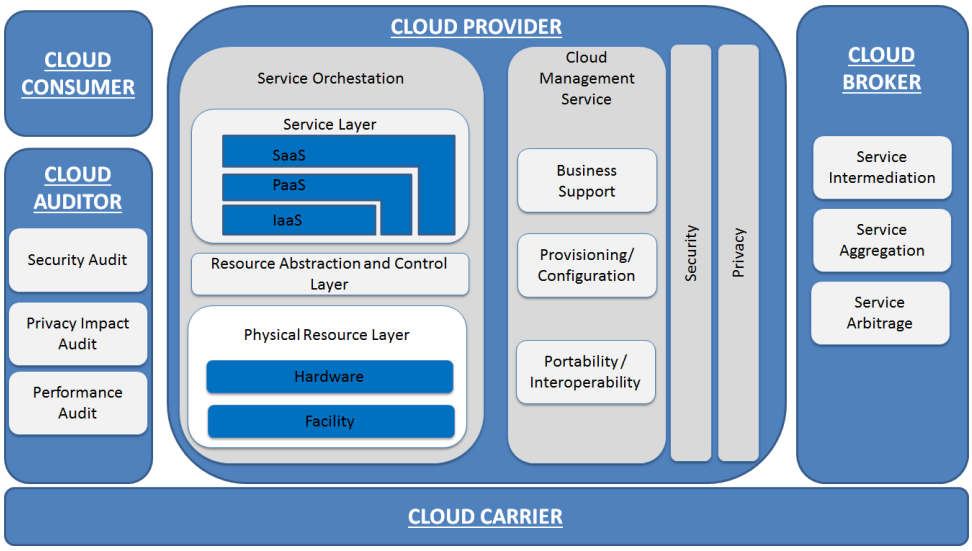

Figure 3.2: Actors involved in a cloud computing environment

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, *Cloud Providers*.
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between *Cloud Providers* and *Cloud Consumers*.
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from *Cloud Providers* to *Cloud Consumers*.

3.3 Key Security Properties

Cloud Security is generally not much different from security in any IT environment. However, because of the cloud service models employed, the operational models, and the technologies used to enable cloud services, cloud computing may present different risks to an organization than traditional IT solutions[1].

Within the study of a typical cloud application it is important to provide a description of some of the key security properties that the TRESCCA project should look after carefully in the design of the platform. It is important to highlight that due to the multiple entities and actors involved in a cloud environment the security analysis is even more complex.

A stronger emphasis is thus put on security measures and controls objective in place to achieve security

goals. Networks and data are vulnerable without a security plan in place, therefore at first it is important to understand the different key properties of security in cloud computing:

Confidentiality Confidentiality means, that no one is able to access data without authorization. This typically requires that users possess the right credentials, such as encryption keys. This requires that the appropriate tools for managing these credentials (distributing, verifying and revoking) are included in the management infrastructure. Similar to integrity, it means confidentiality of both, the data stored inside the cloud networking infrastructure, as well as the data that is contained in the communication.

Integrity Integrity means, that a subject is not able to alter secured data without authorization; one instantiation of this is that all modifications can be detected after the fact, for instance using electronic signature schemes. For cloud networking, the integrity of the data that is stored on the cloud networking infrastructure is important. Also the integrity of communication with and inside the infrastructure elements has to be realized, so that no man-in-the-middle attacker is able to alter data that is send to, from, or inside the cloud networking infrastructure.

Availability Availability means, that a subject is not able to impact the ability of a system to deliver services to its users in an unauthorized way. Availability is generally a quantitative metric against measurable parameters (e.g. number of users serviced in parallel, network bandwidth, response time). It has strong ties to (perceived) quality of service. In the case of cloud networking this means that no user without administrative privileges on the cloud networking infrastructure is able to impact the service of the other users.

Authenticity Authenticity means, that one can prove that someone or something is the one/thing that it claims to be. Authenticity is needed for the cloud networking infrastructure so that a user can verify that he is communicating with the correct infrastructure. Also users can authenticate their identity in order to access the infrastructure.

Non-Repudiation Non-Repudiation means, that a subject that performed an action is not able to disclaim it afterwards. For cloud networking this is strongly related to traceability, i.e., to verify where the virtual infrastructure is located and if it is conform to the agreed policies. Non-Repudiation is also important for accounting.

Privacy Privacy means that a subject is able to decide which personal information it wants to reveal. Anonymity, i.e. to hide the subject's identity in a set of other identities (anonymity set), and pseudonymity, i.e. the use of pseudonyms instead of real names, are ways to enforce privacy. Privacy is always a trade-of between information that is necessarily needed to provide a service and the user, who wants to provide as little as possible personal information. Within cloud networking, at first the needed information has to be identified and additionally it has to be ensured that current legal directives from not only the country where the physical infrastructure is located but also from virtual infrastructure's location are followed as they can pass legal borders.

Traceability Data traceability means that all the information flowing through the cloud is logged about accesses to an asset enabling to verify and track these flows and their content.

3.4 Conclusion

A list of all the identified high level security threats that target a cloud application is available in Appendix A. Although TRESCCA will not design cloud applications, it will enable features that will allow to build a more

secure cloud environment. TRESCCA features altogether with the proper security design of the cloud applications will ensure strong logical and physical security in the whole cloud architecture. The next section will list the security objectives on which the TRESCCA platform will focus and will introduce the considered threat model as well as the envisaged corresponding hardware and software supports.

4 TRESCCA CLOUD PLATFORM

Considering all the above mentioned assets and security properties, the TRESCCA project aims at proposing new solutions to overcome the lack of trust between cloud operators and cloud end users. The ultimate and long term goal is to allow trustable cooperation between both ends even when sensitive computations are delegated by one to the other. TRESCCA deals mainly with the *light* clients at the cloud's edges, so the security objectives and threat models are asymmetrical. By adding hardware and software support for security to the light clients, the TRESCCA platform will offer enhanced security about their behaviour. A whole family of hardware and software attacks against light clients will be rendered much more difficult (expensive), if not infeasible, even when the attacker is the end user itself. But the TRESCCA platform also aims at improving the security for the end users. Computations on the light client side, on behalf of the end user, shall thus also be protected against malevolent or compromised remote cloud infrastructures. In the following we will analyse the TRESCCA security objectives, the corresponding threat models and the hardware and software support for security that shall be added to the light clients in order to reach the security objectives.

4.1 Security objectives

> The penalties and costs for lost or compromised customer, employee, or financial data make it imperative that IT managers not lose control of their systems. This means they must implement the best tools available for protecting their infrastructure and validating the integrity of the computing environment on an ongoing basis. Establishing a root of trust is essential. Each server must have a component that will always behave in the expected manner and contain a minimum set of functions enabling a description of the platform characteristics and its trustworthiness.
>
> James Green, Intel Corporation[2]

4.1.1 Trusted compartment

The most important security objective of the TRESCCA platform is to guarantee that a hardware and software *trusted compartment* exists inside the light client and that it cannot be compromised, neither by software exploits or hardware attacks. The considered attackers include the end users of light clients themselves, because they are frequently in the best possible position to mount attacks, but also the cloud infrastructure, because if malevolent or compromised it could benefit from its higher administration privileges to remotely tamper with the trusted compartment. The goals of malevolent end users can be anything from bypassing access control features to retrieve protected contents, high value secrets or to avoid tolling, to attacks targeting the heart of the cloud to get advantages against other cloud end users or cloud operators. The targets of malevolent or compromised cloud infrastructures could be end users' privacy or any end users' assets which access control is delegated to the light client. For the considered threat models, it shall thus be impossible[1] to:

- retrieve confidential information from the trusted compartment,

[1] *Impossible*, here, must be understood as more difficult or expensive for the considered class of attackers than the expected benefits of a successful attack.

- alter the nominal behaviour of the trusted compartment without the alteration being detected by the cloud infrastructure and the end user,
- tamper with the communication channels between the cloud infrastructure or the end user and the trusted compartment without this being detected by the cloud infrastructure and the end user.

This primary objective shall hold even on light clients allowing end users or third party services providers to install and run their own applications. The light client shall exhibit a clear partitioning between the trusted compartment and the remaining, potentially untrusted, areas. The trusted compartment shall allow the cloud infrastructure and end users to delegate sensitive computing to the light client. It is important to note that full availability of the trusted compartment is not a requirement because there is no way to protect a light client against denial of service attacks while it is in the hands of potential attackers. Wired and wireless network disruption or power supply removals are always possible and can reasonably not be prevented.

4.1.2 Robustness against software denial of service

Instead of full availability of the trusted compartment, the second security objective is the availability of the trusted compartment when only software attacks are considered. The hardware and software extensions of the light client shall guarantee that, whatever the currently running applications in the untrusted area and the load they put on shared resources, the trusted compartment is still reactive and is granted a sufficient access to the shared resources to render its nominal services.

4.1.3 Attestation

In order to set up a root of trust between the cloud infrastructure, the end user and the light client, the trusted compartment shall be equipped with an attestation mechanism. The attestation mechanism shall answer attestation requests from the cloud infrastructure or the end user and the answers shall constitute an unforgeable certificate that the trusted compartment has not been tampered with.

4.1.4 Authentication of delegated computing tasks

The trusted compartment shall provide means to authenticate computing tasks that the cloud infrastructure or the end user delegates to the light client. It shall provide unforgeable guarantees that the delegated computing tasks were executed as expected, that confidential information have not been disclosed during the processing and that the returned results or resulting state changes of the light client have not been tampered with. If useful, authentication can rely on the attestation mechanism.

4.1.5 Software upgrade and boot-to-boot integrity

The software stack of the light client, including the trusted compartment, shall be upgradable. This is required to fix discovered bugs or security flaws or to add new features. Updating the software stack shall not compromise the trusted compartment. Upgrades of the trusted compartment shall be authentic and there must be a way for the cloud infrastructure and / or the end user to control the upgrade. In applications where both are involved, they shall agree before the upgrade can take place. It shall be impossible for an adversary to tamper with the upgrade process and prevent it to put the light client in the expected final state. After the successful completion of an upgrade it shall be impossible to downgrade to an old version of the software stack. From one boot of the light client to the next, it shall be impossible to tamper with the software stack without this being detected at boot time. The boot sequence shall check all software components in such a way that even downgrades are trapped. In case the boot sequence detects that the software stack has been tampered with, the sequence shall end in a state where attestation and authentication of delegated computing tasks are not possible any more. It shall then become obvious to the cloud infrastructure and the end user that the trusted compartment is not running and that the light client cannot be trusted any more.

4.1.6 Isolation between applications

The trusted compartment shall have the full control of the loading and unloading of other software services. It shall guarantee the perfect memory isolation between services, unless cooperating services require memory sharing and apply the pre-defined sharing protocol. This is an essential feature to offer the cloud infrastructure guarantees against the end users and conversely. The trusted compartment shall act as a trusted third party between the cloud infrastructure and the end users.

4.1.7 Robustness against software exploits and other logical attacks

Completely avoiding software exploits and other logical attacks is probably infeasible on large, complex software stacks that frequently consist in multi-millions lines of code. But as the TRESCCA platform will rely on several critical software components, the most critical of all being the Trusted Compartment, these software components shall be kept as simple and minimal as possible. Whenever applicable, they shall be amenable to the most aggressive design and verification techniques, like, for instance, formal specification, design and verification. Moreover, even when applications are large, complex, and thus potentially bogus and exploitable, TRESCCA shall provide a certain level of protection against software exploits and other logical attacks by properly isolating applications. The consequences of a software exploit shall thus be limited to the attacked application (containment of software exploits).

4.2 Threat model

When designing security solutions, it is important to define against which class of adversary the solutions shall be effective. The mechanisms implemented to guarantee the security objectives can be very different if the considered attackers are undergraduate students with limited equipment, money and skills or if they are governmental or criminal large organisations.

In the scope of the TRESCCA project, we consider that the adversary has a complete physical access to the light client hardware, except the internals of the main SoC. Indeed, on-silicon attacks against integrated circuits manufactured in nano-scale technologies are extremely complex and require very sophisticated equipments (focused ion beams, e-beam testers, micro-probes and micro-positioners, etc.). These attacks are very difficult to defeat. On the other hand, they are so sophisticated that it is unlikely that they are used against low to medium value secrets like a game console, a set top box or an Internet provider's box unless the gained advantages are massively reusable. TRESCCA will focus on reasonably well designed security infrastructures where retrieving one secret from one embedded system does not compromise hundreds or thousands of equipments. As a consequence, the gap between the potential value of gained advantages and the cost of on-silicon attacks is large and these attacks will not be considered.

TRESCCA will not consider direct hardware attacks against the main SoC, like, for instance, fault attacks or side channel attacks. These attacks are perfectly practical, with far less resources than on-silicon attacks, and by intermediate class attackers. A lot of research works deal with them and many countermeasure proposals have been proposed and implemented. TRESCCA will consider that such countermeasures are used every time it is required to protect the SoC. As usual in the security field, TRESCCA follows a modular approach, where each problem is dealt with as independently as possible from the others.

On the software side we consider light clients on which the cloud infrastructure and the end user can load and launch arbitrary applications, under control of the trusted compartment (which role is to guarantee the isolation between them).

With a complete physical access to the light client hardware and software components the adversary is able to spy at the memory bus, inject her own data on the memory bus, intercept communication between the light client and the cloud servers, replace the content of the flash memories and other mass storages of the light client, execute her own applications on the system, etc. Her attacks can thus be either purely software exploits

or hardware or a combination of the two. A typical example of combined attack can be the attacker populating as much external memory as allowed with custom code and probe the memory bus to flip an address bit while the light client runs in the highest privileged mode, forcing the light client to run the custom code in privileged mode and leading to a privilege escalation.

In the context of a cloud platform, Section A.1 lists the attacks that threaten an application running inside the cloud. Section A.2 presents the results of an analysis of these threats and lists those that are concerned, and thus taken into account, by the threat model and the security objectives of the TRESCCA project.

4.3 Hardware and software support for security

In order to guarantee the security objectives under the threat model described in the previous section, the TRESCCA project focuses on the light client which is in the hand of the end users. Two kinds of hardware and software extensions will equip the light client, one for the virtualization and the other targeting the protection of the external memory bus.

The protection of the external memory bus will rely on cryptographic primitives to guarantee the confidentiality and integrity security properties. As the cryptographic primitives used for encryption, decryption and integrity checking are computationally intensive they will be accelerated by dedicated hardware engines. The hardware accelerators will be embedded on the SoC (as required by the threat model) and driven by software drivers, parts of the trusted compartment. Altogether, the hardware accelerators and their software counterparts shall protect the memory bus according dynamically configurable security policies, with the smallest possible impact on the performance of the SoC.

Several solutions exist to guarantee some of the security objectives we considered for the TRESCCA platform against our threat model (or a subset of it). These solutions and their weaknesses are described in the Deliverable D2.1 of the TRESCCA project.

A SECURITY THREATS

This appendix provides a list of the security threats identified against a cloud application. As described in Chapter 4, the TRESCCA platform only focuses on a limited number of these threats. However, it is important to keep in mind that a large number of threats are not tackled by the TRESCCA platform and must be mitigated by other means.

A.1 Threats

A.1.1 Introduction

The security is characterized by the maturity, effectiveness, and completeness of the risk-adjusted security controls implemented on a cloud environment. These controls are implemented in one or more layers ranging from the facilities (physical security), to the network infrastructure (network security), to the IT systems (system security), all the way to the information and applications (application security). Additionally, controls are implemented at the people and process levels, such as separation of duties and change management, respectively.

There is currently no way for a naive consumer of cloud services to simply understand what exactly he/she is responsible for, but there are efforts underway by specific entities like the CSA and other bodies to define standards around cloud audit. The CSA defined the following security model based on different layer which will help TRESCCA to define and classify the different identified risks.

A.1.2 Vertical Threats involving all layers

Insider IT sabotage or internal theft of intellectual property. Internal management and sabotage is one of the most critical security points in a cloud environment. The own IT infrastructure managers generally have access to most of the platform, with the possibility of carrying out tasks outside legal or not legal. To avoid this, among other tasks it would be necessary to provide a nominal access to the infrastructure, as well as a complete management of access permissions by roles and external audit systems to IT administrators. This risk applies to the entire stack as any level of the cloud platform is vulnerable against this type of attack.

Breach of actual legislation by inability to certify environments. Failure to follow the legislation that apply to existing information systems involve complete inability to certify the safety of the environment.

Detect and tackle malicious insiders, whom could compromise cloud provider or other client services. In this case, it is necessary to detect and intercept potential internal attacks to the cloud platform or current cloud consumers, either from a third user who might be trying to make malicious use of the application to extract information or carrying out uncontrolled attacks.

A.1.3 Physical Threats

Unauthorized physical access to infrastructure. It is necessary to conduct a detailed access record to the physical infrastructure, as well as an exhaustive control of the tasks carried out on them. In this way unauthorized accesses would be detected in order to prevent security leaks. Besides it will provide detailed information in case it is needed.

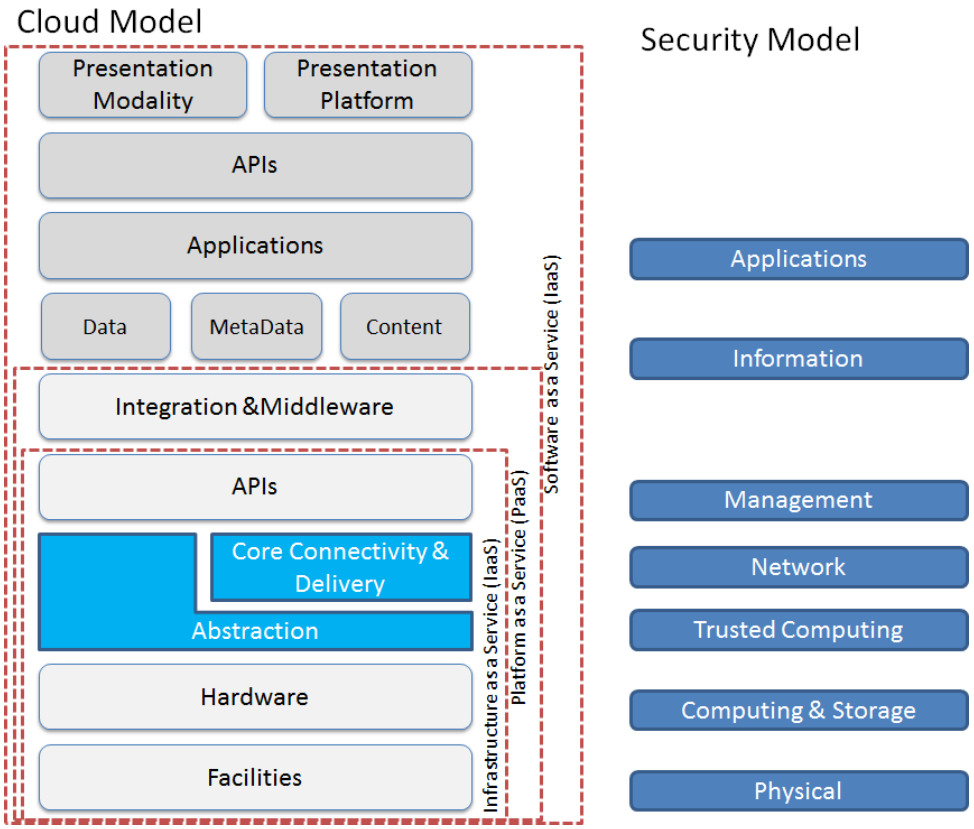

Figure A.1: Security Model

Availability failure. A power outage would complete shutdown the infrastructure, as well as service interruption. This requires having a physical redundancy in the electrical sockets, having all the computer hardware configured with redundant power supplies and various electrical connections, thus safeguarding the interruption of one of the electrical circuits. Additionally it is required to have generators powered by an additional system, e.g. diesel engines, which will safeguard the infrastructure from an electrical disconnection.

Other safety measures. It is necessary to safeguard the infrastructure with security measures to certify the continued protection of the information, between the measures we can find fire safety measures, or air and particle filtering.

A.1.4 Computing and Storage

Illegal storage access. Encrypted cloud storage keeps the information data private at all times. However, managing the decryption keys in the cloud can be challenging unless a new approach to cloud key management is adopted.

Accept inauthentic data (Data masking). Data masking is a method of creating structurally similar but inauthentic versions of an organization's data

Within almost any organization there is a risk associated with data breaches: unauthorized individuals viewing and potentially leaking personally identifiable and other sensitive pieces of information that reside in your databases and applications. While these types of threats are most often associated with malicious outside hackers, according to a 2012 study conducted by the Ponemon Institute, 88 percent of breaches

come from inside an organization while only 1 percent is committed by hackers. These insider threats could be regular full-time employees, part-time employees and interns, temporary workers, former employees brought back for special projects, or outsourced employees and vendors.

A.1.5 Trusted Computing

Unauthorized information access. Virtualization and software platform must offer strong isolation to prevent underlying and unauthorized information access. The world of virtualization is based on the implementation of various machines, in this case virtual ones, onto one unique physical machine in order to perform an optimization and consolidation of hardware resources. Sharing the same environment carries the risk of information transfer or accessibility from one virtualized machine to the hypervisor or other virtual machines. This makes necessary to implement security measures and techniques that enclose these problems.

A.1.6 Network

Cloud-side sniffing. In a cloud environment there is a sharing of hardware resources, so it is essential to ensure the correct logical division of the environment. Otherwise the existing traffic in either LAN or SAN could be accessed from undesired. LAN & SAN architecture must warrant total independence of network traffic to avoid sniffing and different network traffic task mix or unwanted elements visibility.

Identity Spoofing (IP Address Spoofing). Most networks and operating systems use the IP address of a computer to identify a valid entity. In certain cases, it is possible for an IP address to be falsely assumed identity spoofing. An attacker might also use special programs to construct IP packets that appear to originate from valid addresses inside the corporate intranet. After gaining access to the network with a valid IP address, the attacker can modify, reroute, or delete your data. The attacker can also conduct other types of attacks.

Denial of Service. Unlike a password-based attack, the denial-of-service attack prevents normal use of your computer or network by valid users. After gaining access to your network, the attacker can do any of the following:

- Randomize the attention of your internal Information Systems staff so that they do not see the intrusion immediately, which allows the attacker to make more attacks during the diversion.
- Send invalid data to applications or network services, which causes abnormal termination or behavior of the applications or services.
- Flood a computer or the entire network with traffic until a shutdown occurs because of the overload.
- Block traffic, which results in a loss of access to network resources by authorized users

Man-in-the-Middle Attack. As the name indicates, a man-in-the-middle attack occurs when someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. For example, the attacker can re-route a data exchange. When computers are communicating at low levels of the network layer, the computers might not be able to determine with whom they are exchanging data. Man-in-the-middle attacks are like someone assuming your identity in order to read your message. The person on the other end might believe it is you because the attacker might be actively replying as you to keep the exchange going and gain more information. This attack is capable of the same damage as an application-layer attack, described later in this section.

Client side sniffing. A cloud side sniffer is an application or device that can read, monitor, and capture network data exchanges and read network packets. If the packets are not encrypted, a sniffer provides a full view of the data inside the packet. Even encapsulated (tunneled) packets can be broken open and read unless

they are encrypted and the attacker does not have access to the key. Using a sniffer, an attacker can do any of the following:

- Analyze your network and gain information to eventually cause your network to crash or to become corrupted.
- Read your communications.

A.1.7 Management

Unauthorized access. An essential policy for a good management of a cloud environment or any IT environment, independently of the layer, is to establish a restricted access policy to the explicitly necessary sources from the network as well as the obligation of using strong passwords with a minimum security requirements for thus avoid brute force Access to the infrastructure.

Undetected exploits or bugs. As a security measure it is needed to take control and the preventive maintenance of the application in order to avoid undetected exploits. In order to achieve this, a policy of updating packages, dependencies, applications, etc. should be undertaken.

Data loss and unwanted behaviour. To prevent data loss or unwanted behaviors through the complete cloud platform is interesting to have the necessary platforms and procedures to combat viruses or malware hosting as for example updated antivirus software.

Intrusion. In order to provide a quicker response time to intrusion, it is interesting to host predefined operating procedure, defense protocols and accessible by prior IT administrators.

Insecure software API, does it provide more functionality than intended. Access to application management by command lines should be minimized to optimal tasks. The overcapacity could lead to malicious use of the application.

A.1.8 Information

Key or password leakage. Key distribution, key location and key storage must be equal or more important than the strength of the key. Having an automatically and random generated password is not enough if the custody of the key is not adequate.

Information leakage. It is essential to ensure the client or in-house information continuity, without partial or complete loss of it. It is an essential need to keep persistence in customer or internal information, due to law or legal contracts.

Information loss. In order to isolate customer information from unauthorized access, the data containers that host customer data in devices must be configured to be capable of using encryption and being logically accessible exclusively by the client. Int his way customers' information will be kept encrypted and logically inaccessible, even for Cloud provider administrators.

Data Modification. If an attacker accesses the customer information, he can modify the data in the packet without the knowledge of the sender or receiver. Even if all the communication do not require confidentiality, it is not desirable to have the transmission messages to be modified in transit. For example, if you are exchanging purchase requisitions, you do not want the items, amounts, or billing information to be modified.

A.1.9 Applications

Password and key cracking. One of the main techniques for application-level intrusion is the password cracking aiming access to it using various techniques such as brute force, dictionary attack, heuristics, etc. Therefore it is necessary to have policies and procedures that allow a higher rate of strength in the password and key definition.

Misuse platform by malicious data or botnet hosting or creation of rainbow tables, captcha solving.

- Hosting malicious data. The need for measures to counter the vulnerability of hosting malicious information has been previously discussed at various levels, this information could directly or indirectly cause havoc on the infrastructure of the customer or the cloud provider itself.
- Botnet command and control. One of the main threats on the Internet for cloud providers is the hosting of applications or environments that are potentially targeted for massive denial of service attacks. One way how attackers achieve this is the conversion of online machine to zombies' machines that will perform recursive attacks that once they have a considerable set of elements disposed can be controlled and concreted directed to the real target.
- Building rainbow tables. This is one of the techniques well-known nowadays to crack passwords using reverse coding.
- Captcha solving. This measure is taken into consideration to avoid the use of automatic delivery of emails, registries, etc. that could affect to the integrity or image of a system. In order to avoid this, the use of measures such as "Captcha Solving" is recommended to certify that the interface is being managed by an human entity

Account or service Hijacking. Impersonation threats due to legitimate users stealing user's credential and accessing illegitimate information.

Eavesdropping. In general, the majority of network communications occur in an unsecured or "clear text" format, which allows an attacker who has gained access to data paths in the network to "listen in" or interpret (read) the traffic. When an attacker is eavesdropping on your communications, it is referred to as sniffing or snooping. The ability of an eavesdropper to monitor the network is generally the biggest security problem that administrators face in an enterprise. Without strong encryption services that are based on cryptography, the information data can be read by others as it traverses the network.

Application-Layer Attack. An application-layer attack targets application servers by deliberately causing a fault in a server's operating system or applications. This results in the attacker gaining the ability to bypass normal access controls. The attacker takes advantage of this situation, gaining control of the application, system, or network.

A.2 Threat analysis

The following table indicates which of the previously described threats are taken into account in the scope of the project considering the threat model and the security objectives described in Chapter 4.

Table A.1: TRESCCA Threat Analysis

Category	Threat	Probability	Impact	Addressed by TRESCCA	How
Vertical risks	Internal Theft	2	5	-	
	Breach legislation by inability to certify environment	2	5	-	
	Malicious insiders	2	5	-	
Physical risks	Unauthorized physical access to infrastructure	3	5	-	
	Availability failure	2	5	-	

Category	Threat	Probability	Impact	Addressed by TRESCCA	How
Computing and Storage risks	Illegal storage access	3	4	+	Targeted on client side by HSM-Mem, HSM-Noc and VM isolation
	Accept inauthentic data	2	4	+	Addressed by HSM-Mem integrity checking
Trusted Computing risks	Unauthorized information access	2	4	+	Addressed by VM isolation
Network risks	Cloud-side sniffing	3	4	-	
	Identity spoofing	2	4	+	Addressed by HSM-Mem, SSM, VM encryption + secure channel
	Denial of Service	1	5	+	Addressed by HSM-Noc
	Man-in-the-Middle Attack	2	4	+	Addressed by HSM-Mem, SSM, VM encryption + secure channel
	Client-side sniffing	2	4	+	Addressed by HSM-Nem, HSM-Noc, VM isolation
Management risks	Unauthorized access	2	5	-	
	Undetected exploits	2	4	-	
	Data loss and unwanted behaviour	1	5	-	
	Intrusion	1	5	-	
	Insecure software API	1	3	+	Provides protection to sensitive API by combining a secure boot procedure and the Trusted compartment
Information risks	Key or password leakage	3	4	+	Keys and passwords can be protected locally by HSM-Mem
	Information leakage	3	3	+	Addressed by HSM-Mem, HSM-Noc, VM isolation
	Information loss	2	3	+	Addressed in the client side by using HSM-mem and VM isolation
	Data modification	2	3	+	Addressed by HSM-Mem, HSM-Noc, VM isolation
Applications risks	Password and key cracking	3	4	-	
	Misuse platform by malicious data or Botnet Hosting or creation of Rainbow tables, captcha solving	2	4	-	
	Account or service Hijacking	2	3	-	
	Eavesdropping	2	3	+	Addressed by HSM-Mem, HSM-Noc and VM-Isolation
	Software Flaws	2	4	-	

5	Very high
4	High
3	Medium
2	Low
1	Very low

ACRONYMS

API	Application Programming Interface
BIOS	Basic Input / Output System
CA	Certification Authority
CPU	Central Processing Unit
CSA	Cloud Security Alliance
DR/BC	Disaster Recovery / Business Continuance
EEPROM	Electrically Erasable Programmable Read-Only Memory
GHz	Giga Hertz
GB	Giga Byte
IaaS	Infrastructure as a Service
ID	IDentifier
IO	Input / Output
IP	Internet Protocol
IP	Industrial Property
IP	Intellectual Property
IT	Information Technology
LAN	Local Area Network
LCP	Low-Count Pin
MAC	Message Authentication Code
MMU	Memory Management Unit
NIST	National Institute of Standards and Technology
OS	Operating System
PaaS	Platform as a Service
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCR	Platform Configuration Registers
R&D	Research and Development
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RAM	Read Only Memory
SaaS	Software as a Service
SAN	Storage Area Network
SIM	Subscriber Identity Module
SLA	Service Level Agreement
SoC	System on Chip
SSL	Secure Socket Layer

Continued on next page –

– continued from previous page

TCG	Trusted Computing Group
TLS	Transport Layer Security
TPM	Trusted Platform Module
TXT	Trusted eXecution Technology
US, USA	United States of America
VM	Virtual Machine

BIBLIOGRAPHY

[1] Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing v3.0. Technical report, Cloud Security Alliance, 2011. `https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf`.

[2] James Greene. Intel Trusted Execution Technology. Technical report, Intel Corporation, 2012.

[3] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Dawn Leaf. NIST cloud computing reference architecture. Technical report, National Institute of Standards and Technology (NIST), 2011.

[4] Peter Mell and Timothy Grance. The NIST definition of cloud computing. *NIST Special Publication*, 800-145, 2011.

[5] B.P. Rimal, Eunmi Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, pages 44–51, 2009.