Project Title	Trust-aware, Reliable and Distributed Information Security in the Cloud
Project Acronym	TREDISEC
Project No	644412
Instrument	Research and Innovation Action
Thematic Priority	Cybersecurity, Trustworthy ICT
Start Date of Project	01.04.2015
Duration of Project	36 Months
Project Website	www.tredisec.eu

D2.2 REQUIREMENTS ANALYSIS AND CONSOLIDATION

Work Package	WP 2, Requirements and architecture for a secure, trusted and efficient cloud
Lead Author (Org)	Kaoutar Elkhiyaoui, Melek Önen, Dimitrios Vasilopoulos (EURC)
Contributing Author(s) (Org)	David Vallejo García (ARSYS), Beatriz Gallego Nicasio Crespo, Rosa Maria Vieira Alvarez (ATOS), Hubert Ritzdorf (ETH), Kaoutar Elkhiyaoui, Melek Önen, Dimitrios Vasilopoulos (EURC), Panos Louridas (GRNET), Angelo de Caro, Anil Kurmus, Alessandro Sorniotti (IBM), Roch Lescuyer (MPH), Ghassan Karame, Wenting Li (NEC), Andreas Fischer, Benny Fuhry, Mathias Kohler (SAP)
Reviewers	Julien Bringer (MPH), Rodrigo Díaz (ATOS)
Due Date	31.12.2015
Date	22.12.2015
Version	Final

Dissemination Level

X	PU: Public
	CO: Confidential, only for members of the consortium (including the Commission)

The work described in this document has been conducted within the project TREDISEC. This project has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 644412. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content.

Versioning and contribution history

Version	Date	Author	Notes
1.0	16.11.2015	EURC	Initial Outline
1.1	22.11.2015	ETH, EURC, IBM, MPH, NEC,	Initial contributions to section 3
1.2	23.11.2015	ARSYS, ETH, GRNET, IBM	Updated contributions to Section 3 and initial contributions to section 2
2.0	24.11.2015	EURC	Merged partners' contributions and released v2
2.1	25.11.2015	ATOS	Initial contribution to section 4
2.2	26.11.2015	ETH, EURC, IBM, MPH, NEC, SAP	Updated contributions to Section 3
2.3	26.11.2015	ARSYS	Updated contributions to section 2
2.4	26.11.2015	ATOS	Updated contributions to section 4
3.0	27.11.2015	EURC	Review of all sections, initial version of section 1
4.0	30.11.2015	ATOS, ETH, EURC, IBM, MPH, NEC, SAP	Updates of sections 1,2,3,4 and addition of executive summary, conclusion
5.0	01.12.2015	ATOS, EURC, IBM, MPH, SAP	Finalized version ready for review
5.1	14.12.2015	MPH	Review from Julien Bringer (MPH)
6.0	15.12.2015	ATOS, EURC,	Revised following MPH's feedback.
7.0	16.12.2015	EURC	Added a new figure on consolidated requirements. Document ready for approval phase.
7.1	21.12.2015	ATOS	Approval from Rodrigo Diaz (additional comments)
8.0	22.12.2015	EURC	Added table 12 regrouping the pair of

			security and functional requirements and architectural requirements. Document ready for quality check.
Final	28.12.2015	B. Gallego-Nicasio Crespo (ATOS)	Quality check

Disclaimer

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

List of Tables

Executive Summary

The objective of the TREDISEC project is to develop tools that enhance the confidentiality and integrity of the data and computations outsourced to the cloud. While a number of solutions already address some cloud security problems, the new TREDISEC framework will be designed to integrate various security primitives into a unified framework without sacrificing the scalable advantages of cloud computing.

The purpose of this deliverable is to explore the various functional and non-functional requirements (including security and privacy requirements) of cloud storage and computation systems and identify not only the most relevant ones but also those which may not be met simultaneously. With this aim, the following methodology has been applied:

- The six representative TREDISEC use cases have been analysed and a complete set of functional requirements is derived: these requirements must basically be fulfilled for the correct operation of the cloud system. On the other hand, the major security and privacy requirements of these use cases are also highlighted targeting the protection (privacy and integrity) of storage and computation operations.
- Since the description of the use cases and the derived security requirements are high-level, the deliverable further focuses on the different primitives the project aims at designing (in WP3, WP4 and WP5): Once the dedicated security and privacy requirements are defined the document explains how these requirements affect the functional requirements and specify the ultimate (and sometimes conflicting) TREDISEC requirements which basically combine one security requirement with one or several functional requirements.
- As the final target of the project is the development of a unified framework (WP6) integrating the different security primitives, this document also outlines the requirements with respect to the architecture of the framework that will help TREDISEC developer and administrators to choose the most convenient architectural approach and specify technical details. These requirements are differentiated with respect to their technical, business, and quality nature.

Thanks to the specification of the requirements combining security and operational aspects, the TREDISEC project is now moving into the design of the various security primitives (WP3, WP4 and WP5) and further into the orchestration of these individual modules.

Glossary of Terms

AC	Access Control
BIT	Built-in-test
CC	Cloud Customer
CRM	Customer Relationship Management
CS	Cloud Server
DO	Data Owner
DoW	Description of Work
ERP	Enterprise Resource Planning
EU	European Union
IaaS	Infrastructure as a Service
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ISP	Internet Service Provider
M	Mandatory
O	Optional
PaaS	Platform as a Service
PDF	Portable Document Format
PDP	Proof of Data Possession
PET	Privacy Enhancing Technology
PoR	Proof of Retrievability
PoW	Proof of Ownership
RAM	Random-Access Memory
SaaS	Software as a Service
SLA	Service Level Agreement
SQL	Structured Query language
TREDISEC	Trust-aware, REliable and Distributed Information SEcurity in the Cloud
TRL	Technology Readiness Level
UC	Use Case
WP	Work Package

1 Introduction

1.1 Purpose and Scope

The TREDISEC project deals with security and privacy challenges for the storage and processing of the data on the cloud while taking into account various functional requirements such as storage efficiency and scalable computation. The project aims at designing and implementing a set of security primitives that will ensure the confidentiality and integrity of the outsourced data and computations to the cloud; these newly conceived TREDISEC primitives will be compatible with different cloud storage and computation techniques that address the problem of storage and computation efficiency and multi-tenancy. The ultimate goal of the project is to integrate these modular TREDISEC primitives into a unified framework.

The purpose of this document is to gather a broad spectrum of requirements related to the cloud environment ranging from different operational pre-requisites to specific security and privacy challenges. The document first starts by analysing the TREDISEC use cases described in deliverable D2.1 [1] which also introduces an initial set of functional and non-functional requirements. Based on the analysis of the use cases which are regrouped into two main categories, the deliverable derives a generic list of functional requirements and the global security needs for cloud storage and computation solutions.

Many of the identified functional requirements tend to conflict with security requirements. Therefore, the document further focuses on the specification of the main security and privacy requirements for each technical work package delivering TREDISEC security primitives, namely WP3 (verifiability), WP4 (confidentiality and access control) and WP5 (privacy preserving data processing) and analyses how these requirements can influence the functional requirements and vice versa. The TREDISEC specific requirements are further defined by combining conflicting requirements (functional and security requirements). While some requirements are considered as optional the majority of them are mandatory and will drive the work in WP3, WP4 and WP5.

Furthermore, the deliverable gives an overview over architectural, business and quality requirements which will help the design of the TREDISEC framework's architecture that will be introduced in deliverable D2.3.

1.2 Structure of the document

The document is structured as follows:

- Section 2 summarizes the TREDISEC use cases and further extracts the main functional and the generic security requirements.
- Section 3 focuses on TREDISEC security measures and on the specific requirements, and further identifies the conflicting requirements between these specific security requirements and functional requirements. The security requirements are classified with respect to the cloud service (storage or computation) and the main security property (integrity or privacy).
- Section 4 describes the main requirements from the architectural point of view and classifies them into three categories: architecture, quality and business requirements.
- The deliverable concludes with Section 5.

2 Summary on Use Cases

2.1 Overview

Within WP2 activities, six use-cases were defined from which a number of functional and security requirements that cloud service providers should attend to were elicited (cf. D2.1 [1]). These use-cases also illustrate that TREDISEC activities when completed will answer a real world need for security guarantees that do not come at the expense of efficiency.

The six use-cases defined in D2.1 [1] deal with various scenarios; however, they can easily be categorised into two main categories:

- **File sharing services:** This category subsumes use-cases 1, 2 and 3 (see Section 2.2), which deal with data outsourcing to the cloud in a *multi-tenant* environment. Namely these use-cases consider scenarios wherein multiple end-users claim ownership of the same portion of data. This entails that these end-users will access and edit this shared data concurrently. Therefore, the cloud service provider should make sure that only authorized end-users can read and update the shared data, and that the end-users always get the latest version of their data.
- **Big Data services:** This class of use-cases includes use-cases 4, 5 and 6 (cf. Section 2.3), which mainly focus on the case where a single cloud customer outsources large amount of data (that most of the time are sensitive and thus encrypted) to the cloud service provider for processing purposes. The security requirements derived from these use-cases describe primarily the need for enabling the cloud service provider to process (encrypted) data efficiently without decryption, and the need for allowing the cloud customer to verify the outcome of data processing efficiently.

2.2 File sharing services

UC1: Storage efficiency with security

This first use-case describes the upload, storage and deletion of user data using the *~okeanos* cloud storage service[1] in such a way that (a) data confidentiality is guaranteed throughout the data life-cycle and (b) storage and computational efficiency are preserved.

UC2: Multi tenancy and access control

The second use case, envisaged by GRNET, concerns the management of multiple users that share resources (that could be infrastructure or data) on the Cloud. In such an environment, the cloud is required to support multi-tenancy and to deploy mechanisms for fine grained access control and resource isolation. Such a service should not of course undermine the confidentiality and integrity of the outsourced data.

UC3: Optimised WebDav Service for confidential storage

The third use case provided by Arsys is focused on a shared storage service that wants to provide multi-tenancy access control through WebDav[2] access. The use case aims to build a WebDav service upon a confidential storage whereby even ISP administrators cannot violate the confidentiality of users' data. The service will allow tenants (customers) to manage more than one user with different data access and permissions while data isolation among tenants is guaranteed. Finally, original services or tasks such as user data sharing and storage optimization should not be sacrificed for the security; neither should be the performance of the provided service.

[1] https://okeanos.grnet.gr/home/
[2] http://www.webdavsystem.com

2.2.1 Stakeholders

From UC1, UC2 and UC3, we identify the following Stakeholders:

- **Cloud customers:** These are tenants with an active account on the cloud storage product. The cloud customer is an organisation outsourcing the storage of its data (which may be sensitive) to the cloud in the aim of reducing its financial expenditures while enjoying reliable and secure storage services. As an organisation, the cloud customer is a legal entity that cannot directly access its data; rather (read/write) access rights are delegated to authorized members in the organization. These authorized members in turn will constitute what we call hereafter cloud end-users (or end-users interchangeably).
- **Cloud end-users:** These are the entities that upload the data, update it and delete it when required. Depending on their role within the cloud customer's organisation (and sometimes the data in question), these end-users will be given different access rights. While legally, the cloud customer is the owner of the outsourced data, it is generally an end-user (e.g. the organisation administrator) that modifies the outsourced data and accordingly we assume that anyone with such rights is a data owner.
- **Cloud service providers:** These are the entities that offer the storage service. An end-user authenticates to these cloud service providers through *frontend servers*. These servers are a balanced set of servers that are directly accessible by end-users over the Internet. Besides frontend servers, each cloud service provider controls a set of backend servers that actually host the data storage service and a set of Database servers that store users' registration information. Furthermore, a cloud service provider could be an ISP offering network services to its customers.

2.2.2 Goals & needs

Cloud storage providers anticipate that both the number of their users and their needs in terms of storage will keep on increasing. To be able to accommodate these needs in a cost-efficient manner, advanced deduplication techniques must be employed, covering all types of data, ranging from files to volumes.

At the same time, it is desirable that these cloud service providers augment their services with security primitives like encryption and secure deletion. The use of these primitives will foster users' trust in the provided cloud services and will encourage these users to outsource their sensitive and private data.

Furthermore, to broaden their catalogue of services, it is necessary for cloud storage providers to enable resource sharing among their users. Ideally, cloud storage providers should support multi-tenancy over any kind of resources. To that end, within TREDISEC, we aim to devise advanced access control mechanisms that will allow tenants to define fine-grained policies on a per user basis, and which can be easily integrated in current cloud environments.

Another goal of TREDISEC is to design solutions for end-to-end encryption and data at rest (when possible) while ensuring that such solutions do not obstruct cloud operations. The main advantage of such mechanisms is that even when part of the cloud environment is undermined, the confidentiality of the outsourced data can still be guaranteed.

The aforementioned goals call for a unified solution that supports deduplication, encryption and multi-tenancy access control. This combination constitutes an important research challenge, which has not yet been addressed in its entirety by any major cloud provider.

2.3 Big Data services

UC4: Enforcement of Biometric-based Access Control

This first use-case, supplied by Morpho, considers biometric-based online authentication. It assumes that a user has to perform some authentication before accessing to a service. The biometric-based authentication is delegated to a third party (called Cloud Authentication Server) who performs the authentication and, in addition, provides a proof that the authentication has been correctly performed.

UC5: Secure Upgrade of Biometric Systems

The second use-case supplied by Morpho considers major upgrades of biometric systems. Due to the evolution of algorithms and/or biometric data format, updates of existing biometric data should be performed, and sometimes a large amount of biometric data should be processed. In this use-case, these updates are outsourced to a cloud service. For privacy reasons, biometric data must be encrypted. As a result, the cloud provider should be able to carry out computations over a large amount of encrypted biometric data.

UC6: Database Migration into a Secure Cloud

This use case describes the migration of a company's legacy data into a secure cloud environment, requiring the data to be encrypted, yet stored in such a way that SQL queries can be executed over it. Encrypting large sets of legacy data (potentially multiple gigabytes of data) could take several months, has potentially impact on the running business, and the result requires a larger storage footprint, such that a specific encryption process for the legacy data should be used.

2.3.1 Stakeholders

From UC4, UC5 and UC6, we identify the following stakeholders:

- **Cloud customers:** These are the tenants that outsource the big data on which the processing will be performed. They are the data owners and as such require that the cloud service provider ensures the integrity of the outsourced data, guarantees its correct processing. When the outsourced data is sensitive the data owner may either encrypt the data before outsourcing and in that case, it should make sure that the cloud still could process it without decryption, or ask the cloud service provider to assure that the data is always encrypted at rest.
- **Cloud service providers:** These are the entities that provide the processing service to the data owners. It follows that they are interested in offering a processing service that provides guarantees on the correct processing of the outsourced data or enable the processing of encrypted data.

2.3.2 Goals & Needs

The goal of big data services is to provide a service capable of processing large-scale data on behalf of the data owner. Usually, big data service providers often have a lot more capacities for fast and reliable online data processing than companies outsourcing data and data processing. Hence, for cost saving and performance optimization reasons companies want to use these cloud offerings.

The use-cases above introduce two types of computations over biometric data. The first type of computations is the comparisons over biometric templates. It raises the need of verifiable techniques in order to ensure that the comparisons are correctly performed. The second type of computations is the extraction of biometric templates from raw biometric images. The process of converting and exporting the biometric images to the cloud should be efficient. The cloud provider should be able to process computations over encrypted images. A particular focus will be made in TREDISEC on the transposition over the encrypted domain of the signal-processing-based computations over the biometric images. Additionally, verifiable techniques should be compatible to the above goals, in order to ensure that the computations over encrypted data were correctly performed.

Companies wanting to outsource the data they own, do usually not start from scratch with their business process executions and data in their enterprise systems (ERP, CRM, etc.), but rather have possible large-scale legacy data stored on their on-premises solutions. This data must be migrated

towards the cloud solution such that it is available for further usage. The process for outsourcing data and processes must, of course, be as reliable and smooth as possible, with having a minimum impact on the day-to-day business. A (nearly) zero downtime migration process is often desired or requested. Moreover, the time frame for the actual migration should be as short as possible. Possible data preparations and modifications such as the encryption of the data (such that it can be securely stored with the cloud provider) must be optimized and executed in a highly efficient way. Finally, the data stored at the service provider must be stored in such a way, that no other clients of the service provider have access to it, and it must be stored in a form that still allows processing over it, even if it is encrypted. Typical SQL queries, for instance, should be possible to be executed over the data.

2.4 Summary on Use Case requirements

Based on the needs derived in D2.1 [1], we define two categories of requirements:

- **Functional requirements**: This class of requirements encompasses the basic functionalities that cloud service providers aim at achieving, to either (a) make their services scalable through for instance, storage efficiency mechanisms and parallelization techniques, or (b) encourage new customers to purchase the service by putting in place mechanisms that ensure that data is available 99.999% or that access is restricted to authorized users only.
- **Security Requirements:** These requirements deal with the set of security functionalities that a cloud service provider should implement to assure cloud customers of the correct execution of the outsourced operations (whether storage or processing) and the privacy of their data.

2.4.1 Functional requirements

- **Multi-tenancy:** The cloud services provided by TREDISEC should accommodate a multi-tenant environment. That is, an environment in which multiple users share the ownership of outsourced data, or are permitted to operate on the data without being actually owners. This requirement is more relevant to the use-cases pertaining to file sharing services (namely, UC1, UC2, and UC3).
- **Storage Efficiency:** Cloud service providers take advantage of deduplication and compression mechanisms to minimise their storage needs and therefore, their expenditures. Thus, it is very important for TREDISEC solutions not to hinder the deployment of such mechanisms and to work seamlessly on top of them. While storage efficiency is a very important requirement for cloud services, it is more crucial to enable it for the file sharing use-cases.
- **Computation Efficiency:** By possessing very powerful machines and using parallelization techniques cloud service providers are able to operate on huge amount of data very fast. It follows that TREDISEC security services should maintain this low cloud provider's latency by making sure that the implemented security services do not add too much complexity to the cloud environment. This requirement is derived from the use-cases dealing with the big data services (notably UC4, UC5, and UC6).
- **Data Access:** The owner of outsourced data should be given the possibility to control who accesses her data and how. More specifically, the data owner should be allowed to share the ownership of her data, give read/write rights to users of her choice and finally revoke such rights at any point in time. Therefore, we envisage in TREDISEC to develop mechanisms for access policy enforcement. Given the multi-tenant nature of file sharing use-cases, they require solution that control and regulate data access.
- **Data Processing:** Since the cloud provider possesses plenty of computational resources that the lay customer does not, it can perform complex operations on big data very fast. This encourages customers not only to outsource storage but also to outsource data processing. To facilitate the adoption of TREDISEC security services, we should focus on how to

reconcile existing data processing functionalities and the pressing requirements of data confidentiality and computation integrity.

We note here that this requirement is more related to the use cases dealing with big data services, since, for the case of file sharing services the cloud provider is only supposed to store the data.

- **Dynamicity:** Most of the data outsourced to the cloud is prone to changes. Such changes include appending new data, modifying chunks of existing data, or deleting parts of the outsourced data. Besides the classical challenge of synchronization that cloud service providers should solve when multiple users update outsource concurrently, we should also ensure in TREDISEC that our security mechanisms work seamlessly in the presence of dynamic data. Namely, a cloud customer should not be impelled to download her (entire) data to perform a small change. Ideally this requirement should be met in all the TREDISEC use-cases. However, in TREDISEC we prioritise the file sharing use-cases.
- **Availability:** An important requirement that cloud service providers must meet is the requirement of data availability. Availability assures the cloud customer that she can download her (entire) data at her convenience. Although in the use-cases for big data processing, the cloud customer is not supposed to ever download her data, we believe that in TREDISEC this requirement should be met for all the use-cases.

Table 1 goes over these functional requirements and specifies whether they are mandatory (M) or optional (O) for the six TREDISEC use cases.

		Functional Requirements Mandatory(M)/Optional(O)						
	Use-Cases	Multi-tenancy	Storage efficiency	Computation efficiency	Data Access	Data Processing	Dynamicity	Availability
File Sharing Services	UC1	O	M		M		M	M
	UC2	M	M		M		M	M
	UC3	M	M		M		M	M
Big Data Services	UC4			M		M		
	UC5		O	M		M	O	
	UC6	O	M	M		M	O	O

Table 1: Functional Requirements

2.4.2 Security requirements

- **Storage integrity:** Cloud customers should be able to verify that their data is stored correctly. Namely, cloud customers should be provided with guarantees that assure that their data was not tampered by inadvertent or deliberate deletion. Solutions to meet this requirement include classical integrity mechanisms such as Merkle trees or more sophisticated and efficient solutions such as Proof of Retrievability (PoR). However, these solutions do not integrate well with existing cloud infrastructures, as their security guarantees come at the expense of storage efficiency (see Section 2.2 for more details).
- **Computation integrity:** When outsourcing computation, cloud customers want to be assured that the result returned by the server is correctly computed. For that, cloud customers may require to either receive proofs of correct computations or to be assured that the computation was performed in trusted environment. The advent of cloud computing has given rise to a plethora of techniques to answer this requirement from verifiable computation primitives to

trusted hardware, however, these solutions are either impractical or support limited functionalities (cf. Section 3.3).

- **Storage privacy:** One of the obstacles to a wider adoption of cloud services is privacy, especially among industry. This type of cloud customers is reluctant to disclosing their data that by nature is sensitive to third parties. Therefore, such customers either tend to encrypt their data before outsourcing or require the cloud service provider to put in place mechanisms for data encryption. Techniques to implement this requirement are diverse, we can name for instance, verifiable ownership (cf. Section 3.2.3) data confidentiality mechanisms, access control and resource isolation (see Section 3.4). In TREDISEC, we strive at designing solutions that do not cancel out the advantages of the cloud.
- **Computation privacy:** When the outsourced data is encrypted, a natural challenge that arises is how to process such data. It is desirable that the adopted encryption mechanisms enable the cloud server to operate on the data without the need for decryption.

  Note that computation privacy is partially achieved for some operations such as keyword search. However, it remains an open question how to empower the cloud to perform more complex and generic operations while counteracting any threats to the cloud customers' privacy (cf. Section 3.5. for more details).

Similarly to Table 1, Table 2 reviews the security requirements according to their priority level (mandatory/optional) with respect to the use cases.

		Security Requirements Mandatory(M)/Optional(O)			
	Use cases	Storage integrity	Computation integrity	Storage privacy	Computation privacy
File Sharing Services	UC1	M		M	
	UC2	M		M	
	UC3	M		M	
Big Data Services	UC4		M	O	O
	UC5	O	O	M	M
	UC6	O		M	M

Table 2: Security Requirements

3 TREDISEC Security measures

3.1 Overview

Given the classification of functional and security requirements in the previous section, we describe here how existing work tackled the issues of coupling security and utility and we identify the standing problems that we intend to address in TREDISEC. Notably, starting from the defined use-cases and building upon the detailed requirements provided in D2.1 [1], we give a comprehensive description of the security needs that TREDISEC aims at fulfilling. Then we show briefly how most of the times, the fact that when one aims at answering these security requirements, the attractive features of cloud services are hindered.

All requirements are named with the following convention: Work Package #, Task #, Requirement #.

3.2 Storage integrity requirements

As the first and foremost cloud service is data storage, customers outsourcing their data demand some guarantees on this operation. While classical data integrity solutions can directly be used for file sharing use cases (UC1, UC2, UC3) they fall short in the case of Big Data services (UC4, UC5, UC6) and the new problem is defined as verifiable storage.

3.2.1 Verifiable storage

Verifiable storage allows a cloud customer (CC) to check whether her (Big) data (D) is stored correctly at the cloud server (CS). As previously mentioned, classical data integrity techniques are not suitable anymore since they require the customer to download the entire data together with the integrity proof computed by the cloud. TREDISEC tackles this specific problem and currently investigates existing solutions which can be classified into two categories: Proof of Data Possession (PDP) and Proof of Retrievability (PoR).

The main security requirements verifiable storage aims to meet are:

WP31-R1: Efficient storage verification

Once her data D is outsourced, CC should be able to verify its integrity whenever needed. However, since this data is very large, this verification should not require CC to download the whole data.

WP31-R2: Data possession verifiability

To meet the efficiency requirement, verifiable storage solutions cannot fully guarantee the integrity of the entire data. TREDISEC solutions should therefore provide probabilistic guarantees on the availability of data.

WP31-R3: Data extractability

This additional requirement derives from the probabilistic nature of the data possession verification. Data extractability assures that whenever the data possession verification succeeds, CC can retrieve her data in its entirety with an overwhelming probability.

WP31-R4: Delegated verifiability

Delegated verifiability allows CC (who owns the data) to authorize third parties to similarly check the correct storage of her data.

WP31-R5: Public verifiability

In contrast with delegated verifiability, public verifiability assures that any third party, even if not explicitly authorized, can perform this storage verification

3.2.2 Verifiable ownership

To avoid client-side deduplication attacks, the new primitive called Proof of Ownership (PoW) was introduced with the aim of preventing leakage amplification in client-side deduplication. More specifically, the idea is that if an outside adversary somehow obtains a bounded amount of information about a given target user file F via out-of-band leakage, then the adversary cannot leverage this short information to obtain the whole file F by participating in client-side deduplication with the cloud storage server.

One of the main objectives of the project with respect to verifiability is the study of PoW protocols. There are indeed several open questions when it comes to this family of protocols, mostly revolving around performance and security. In addition, we plan to investigate PoW schemes that can be applied to encrypted data and/or data uploaded by participants that do not share mutual trust.

The main requirements follow.

WP33-R1: Efficient ownership verification

The main efficiency parameters of Proof of Ownership schemes are (a) the size of the summary information kept by the server for every file, (b) the communication complexity of the protocol, and (c) the computation complexity of executing the protocol (all with respect to the file size |F| and the security parameter).

WP33-R2: Verifiable Ownership with Data Confidentiality

Traditionally, Proof of Ownership protocols focus on a scenario where multiple users outsource unencrypted content. In TREDISEC we will investigate data outsourcing protected by PoW protocols that tolerate encrypted data as input, with the aims of matching the existing deduplication ratios and optimising client and server side efficiency.

3.2.3 Trade-off analysis

This section identifies requirements that combine storage integrity requirements with functional (and sometimes other security requirements).

WP31-R6: Verifiable storage with efficiency at the cloud

The TREDISEC verifiable storage primitive should not cancel out the (computational) performance advantages of cloud services and therefore should be optimized in terms of storage and computation overhead.

WP31-R7: Verifiable storage with dynamic data

The TREDISEC verifiable storage primitive should handle data updates (deletion, insertion, modification) inherently.

WP31-R8: Verifiable storage with storage efficiency

The TREDISEC verifiable storage primitive should adapt to existing storage efficiency techniques such as data deduplication or data compression. Even when data is deduplicated, CC should be able to verify its availability.

WP31-R9: Verifiable storage with multi-tenancy

The TREDISEC verifiable storage primitive should allow users sharing the outsourced data (i.e. when data is deduplicated) to perform data updates efficiently and without the need for synchronization.

WP33-R3: Verifiable ownership with multi-tenancy

The system should be able to deduplicate confidential data coming from users who do not share mutual trust (and as an example, are not prepared to share cryptographic material).

WP33-R4: Verifiable ownership with data reduction

To achieve storage efficiency, the cloud storage system should be able to perform the deduplication function over outsourced data. The adoption of Proof of Ownership protocols should not affect the deduplication function not should it ideally reduce its efficiency.

WP33-R5: Verifiable ownership with dynamicity

Proof of Ownership protocols should be able to support data modification in a way that is incremental to the size and scope of the change.

3.2.4 Summary

Table 3 summarizes the storage integrity requirements and captures their relationship with the TREDISEC use cases.

Requirement	Mandatory (M)/ Optional(O)	Related Use Cases
WP31-R1: Efficient storage verification	O	UC5
WP31-R2: Probabilistic data possession verification	O	UC5
WP31-R3: Data extractability	O	UC5
WP31-R4: Delegated verifiability	O	UC5
WP31-R5: Public verifiability	O	UC5
WP31-R6: Verifiable storage with efficiency at the cloud	O	UC5
WP31-R7: Verifiable storage with dynamic data	O	UC5
WP31-R8: Verifiable storage with storage efficiency	O	UC5
WP31-R9: Verifiable storage with multi-tenancy	O	UC5
WP33-R1: Efficient ownership verification	M	UC1, UC2, UC3
WP33-R2 : Verifiable ownership with data confidentiality	M	UC1, UC2, UC3
WP33-R3: Verifiable ownership with multi-tenancy	M	UC1, UC2, UC3
WP33-R4: Verifiable ownership with data reduction	M	UC1, UC2, UC3
WP33-R5: Verifiable ownership with dynamicity	O	UC1, UC2, UC3

Table 3: Storage Integrity Requirements Summary

3.3 Computation integrity requirements

3.3.1 Verifiable computation

While storage integrity requirements address the integrity of outsourced data, computation integrity requirements address the correctness of outsourced computation. The aim of TREDISEC is to deal with this problem. The main requirements are summarized below.

WP32-R1: Computation integrity

Computation integrity denotes that computations are correctly carried out. In particular, when computations are outsourced by some party (a client) to some server, the client is ensured that the delegated computations are correctly performed by the server. This property brings confidence in the result of the computation, since errors may happen. Errors may be induced by malicious actors, but also by purely technical failures.

WP32-R2: Public verifiability

For this requirement, verifiability is another name for computation integrity. The verifiability is said public when anyone can verify the correctness of the computation, not only the client who asked for a delegated computation.

WP32-R3: Public delegatability

While public verifiability asks that anyone can verify the correctness of a computation, public delegatability requires that anyone can query a computation, i.e. supply inputs to a delegated function.

WP32-R4: Managing big databases

Solutions for some particular cloud functionalities may exist. They might be efficient for small databases, but fail to work for bigger ones. As an example, efficient solutions exist for securely querying outsourced encrypted databases, but these solutions are only usable for databases with restricted size. The challenge brought by this requirement is to come up with efficient solutions for databases composed of a huge amount of records. Of course, the underlying notion of “database size” and “efficiency” depends on the application.

WP32-R5: Data confidentiality

This property denotes that only authorized entities have access to a particular data and from the opposite point of view, that the data remains hidden from unauthorized entities

3.3.2 Trade-off analysis

This section identifies requirements that combine computation integrity requirements with functional (and sometimes other security requirements).

WP32-R6: Verifiable computation with efficiency at the cloud

For a solution using cloud functionality, efficiency at the cloud means that the complexity of the solution is acceptable by the service owners and the clients who use it. What is acceptable in practice is of course application-dependent. As for the client, this efficiency can be measured by different metrics, as time, storage or interaction complexities, or also by deployment costs. Complexity of some solutions might also be compensated with other aspects, such as large scale deployment. The TREDISEC verifiable computation primitive should supply acceptable performance from the cloud’s point of view.

WP32-R7: Verifiable computation with efficiency at the client

For a client being the beneficiary of cloud functionality, efficiency at the cloud means that the constraints induced by the usage of the cloud on the client’s side are acceptable from the client’s point of view. This efficiency might be measured by the time or complexity needed for the interactions with the cloud; or by the size of the local storage induced by the cloud usage.The TREDISEC verifiable computation primitive should supply acceptable performance from the client’s point of view.

WP32-R8: Verifiable computation over big databases

The TREDISEC verifiable computation primitive should supply verifiable techniques that can process a large amount of data.

WP32-R9: Verifiable computation with data confidentiality

The TREDISEC verifiable computation primitive should provide confidence in the correctness of computations while maintaining the confidentiality of the data.

3.3.3 Summary

Table 4 summarizes the computation integrity requirements and captures their relationship with mainly UC4 and UC5.

Requirement	Mandatory (M) / Optional (O)	Related Use Cases
WP32-R1: Computation integrity	M O	UC4 UC5
WP32-R2: Public verifiability	M O	UC4 UC5
WP32-R3: Public delegatability	O O	UC4 UC5
WP32-R4: Managing big databases	O M	UC4 UC5
WP32-R5: Data confidentiality	O M	UC4 UC5
WP32-R6: Verifiable computation with efficiency at the cloud	M O	UC4 UC5
WP32-R7: Verifiable computation with efficiency at the client	M O	UC4 UC5
WP32-R8: Verifiable computation over big databases	M	UC5
WP32-R9: Verifiable computation with data confidentiality	O M	UC4 UC5

Table 4: Computation Integrity Requirements Summary

3.4 Storage privacy requirements

3.4.1 Access control and Policy enforcement

Access control is essential in protecting storage privacy. Customers must be able to trust the cloud service that only authorized parties can access their data. More complicated access control mechanisms provide extra or improved use cases for cloud storage. Additional policy enforcement solutions such as secure deletion allow customers tighter control over their data, enhance their storage privacy and can be essential in order to comply with business regulations.

Access control requirements that storage privacy should provide are:

WP41-R1: Semantic and contextually constrained policy enforcement

Semantic constraints (e.g. separation of duties) and contextual constraints (such as temporal, environmental, usage or isolation constraints included in an access request, e.g. physical geo-location of requests and/or resources, etc.) must be evaluated when determining access to services and resources. Usage control extends access control by adding restrictions to future data usage once data access has been granted.

WP41-R2: Privacy-respectful policy enforcement

The set of user attributes obtained should be limited to those strictly necessary to evaluate an access request against the corresponding AC policy. Support for including more advanced privacy policies regulating access to user attributes based on existing PETs (Privacy Enhancing Technologies), provided by a trusted third party for instance, and is also desirable. This way, policy enforcement would be respectful with the applicable data protection regulation (EU Directive 94/46/EC[3]) on data processing establishes that personal data should not be processed at all, except when certain conditions are met. These conditions fall into three categories: transparency, legitimate purpose, and proportionality.

WP44-R1: Secure deletion

Once a customer chooses to securely delete data, this data should be irrecoverable even if all cloud storage and all user secrets are compromised subsequently. Therefore, even the user can't recreate the data after a secure deletion.

WP44-R2: Shared ownership

In case data has been generated cooperatively by multiple cloud customers, these customers should have the option to jointly make access control decisions on this data. This way unilateral exclusion of a customer by the account manager should be prevented. Such a solution should, however, not require the deployment of Policy Decision Points for each customer. Instead it should be achieved without changes to the current cloud infrastructure.

WP44-R3: Assisted deletion

Customers usually try to delete a certain piece of information. In order to securely delete information, however, customers have to securely delete all files containing this information. This can become an obstacle as the number of files grows and as files might be embedded in other files, e.g. in PDF files, or might be contained in other files, e.g. in compressed archives. Assisted deletion should simplify this process by identifying other files containing the same information and thereby protecting storage privacy.

3.4.2 Resource isolation

To enforce resource isolation, systems may make use of access control and security policies. The entities enforcing these policies, such as hypervisors, operating system kernels, middleware or applications, are themselves vulnerable to attacks. Therefore, improving the security of such policy-enforcing-entities (monitors) improves the security guarantees provided by the policies.

The main objective of the project is to design mechanisms that improve the security of monitors either through: (i) removing vulnerabilities present in the code base, (ii) preventing such vulnerabilities from being reachable by attackers, or (iii) in the presence of attacker-reachable vulnerabilities, preventing their exploitation.

WP42-R1: Improved resource isolation

[3] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

The security of monitor-like components in the cloud infrastructure should be improved with either program analysis, attack surface reduction, or hardening techniques.

WP42-R2: Secure storage per tenant

When possible, the data of users should be stored encrypted and its integrity should be protected.

3.4.3 Data confidentiality

Cloud services introduce new security threats with respect to the confidentiality of the outsourced data. While the cloud providers are motivated to provide data confidentiality for their data storage services given the increasing security assurance demands from the cloud customers, they will also lose the advantage of optimizing their storage costs by de-duplicating the data once traditional encryption is applied to the data.

TREDISEC aims to provide strong data confidentiality guarantees while benefiting from the various advantages of data deduplication in the cloud. On the one hand, we aim to devise novel schemes which ensure data confidentiality despite a powerful adversary that has access to the user's secret material: such schemes are defined as key-exposure resistant schemes. We also plan to propose techniques which support deduplication of data encrypted by different mistrusting principals (tenants, users).

In what follows, we summarize the main data confidentiality requirements with respect to today's market:

WP43-R1: End-to-end security

During the entire lifecycle of data outsourcing to the cloud, no party, except the cloud user, should be able to break the confidentiality of the outsourced data. This is a mandatory requirement that all of our primitives should satisfy.

WP43-R2: Resistance to key leakage

Frequent data breach incidents often show that adversaries can acquire keys often due to system vulnerabilities, misconfiguration errors or careless user/administrator. We plan to devise optional primitives that ensure data confidentiality even if the encryption key is leaked to the adversary

3.4.4 Trade-off analysis

This section identifies requirements that combine storage privacy requirements with functional requirements.

WP43-R3: Data confidentiality with file-based deduplication

The cloud storage service is able to optimize the storage usage by de-duplicating the encrypted files that has the same content. File-based deduplication typically results in considerable storage savings (>50%). Our goal is to devise secure and efficient primitives which support file-based deduplication.

WP43-R4: Data confidentiality with block-based deduplication

Block-based deduplication results in 25% additional savings when compared to simple file-based deduplication techniques. Our primitives should be extended to support block-based deduplication.

WP43-R5: Data confidentiality with compression

In addition to data deduplication, the cloud storage service typically compresses data to optimize its storage usage. Our primitives should optionally support data compression.

WP44-R4: Secure deletion with efficiency at the cloud

The TREDISEC storage privacy primitive should not increase the storage and computation costs significantly. The additional information stored and the additional computation required should be as small as possible.

WP44-R5: Shared ownership with efficiency at the cloud

The TREDISEC storage privacy primitive should only incur additional costs if shared ownership is desired by the customers. Additionally its computational costs for reading and writing should be small.

WP44-R6: Assisted deletion with efficiency at the cloud

The TREDISEC storage privacy primitive should not significantly increase latency. Assisted deletion requires additional operations during read, write and delete requests. However, these operations should be performed out-of-band as much as possible. This leads to possible temporal inconsistencies. Therefore these operations should also be as optimized as possible.

3.4.5 Summary

Table 5 summarizes the storage privacy requirements and captures their relationship with TREDISEC use cases.

Requirement	Mandatory (M)/ Optional(O)	Related Use Cases
WP41-R1 Semantic and contextually constrained policy enforcement	M	UC2, UC3
WP41-R2 Privacy-respectful policy enforcement	O	UC2, UC3
WP42-R1 Improved resource isolation	M	UC2, UC3
WP42-R2 Secure storage per tenant	M	UC1, UC2, UC3
WP43-R1 End-to-end security	M	UC1, UC2, UC3, UC4, UC5
WP43-R2 Resistance to key leakage	O	All UCs
WP43-R3 Data confidentiality with file-based deduplication	M	UC1,UC3
WP43-R4 Data confidentiality with block-based deduplication	M	UC1,UC3
WP43-R5 Data confidentiality with compression	O	UC1,UC3
WP44-R1: Secure deletion	O	UC1, UC2, UC3
WP44-R2: Shared ownership	M	UC2, UC3
WP44-R3: Assisted deletion	O	UC1, UC2, UC3
WP44-R4: Secure deletion with efficiency at the cloud	M	UC1,UC3
WP44-R5: Shared ownership with efficiency at the cloud	O	UC1,UC3
WP44-R6: Assisted deletion with efficiency at the cloud	O	UC1,UC3

Table 5: Storage Privacy Requirements Summary

3.5 Computation privacy

3.5.1 Big data confidentiality

WP5-R1 Data Confidentiality

The original data of the data owner should be protected against unintended and unauthorized access. Within TREDISEC, data confidentiality should be enforced by means of encryption.

WP51-R1 Efficient initial encryption

The encryption of large data sets with one or multiple encryption schemes should be executed in a performance optimised manner.

The encryption schemes used to encrypt the original data set during the migration process should be selected in a way such that possible compression capabilities of the target system at the cloud provider (e.g., dictionary compression within databases) can be used.

This requirement refers to a “Hot State” described in D2.1 [1]. During the migration process and initial encryption of legacy data, the selection of the top-most layers and respective encryption schemes should be based on an analysis of SQL statements expected to be run on the database.

WP52-R1 Privacy preserving migration with minimum downtime

End-user application downtime needs to be minimised during the migration process in order to allow daily business operations to continue. Ideally, a zero-downtime migration is performed where clients seamlessly switch from the old infrastructure to the new one while their applications keep running.

3.5.2 Privacy preserving processing

Privacy preserving processing deals with the design of mechanisms that enable the cloud to process encrypted data. Ideally, cloud providers should be able to conduct any complex operations on the outsourced data. While advances in fully homomorphic encryption are promising, they are still too computationally intensive to represent a viable solution for privacy preserving processing. This is why, in TREDISEC, we focus on a different line of research, that aims at designing dedicated privacy preserving mechanisms for specific applications. More specifically, we address the problem of privacy preserving data processing for biometric data and privacy preserving word search: One of the most demanding operation for cloud application is word search. A data owner (DO) or another authorized third party should be able to search for some words over the data that has already been outsourced in an encrypted manner. The idea is to exploit the properties of the outsourced data and the functions we are interested in, to come up with efficient security solutions that do not negatively impact the performances of cloud computing.

WP51-R2 Query analysis for optimised SQL statement execution over remotely stored encrypted data

Migrating towards a hosted, encrypted database requires the execution of SQL statements on encrypted data. However, there (currently) are SQL operations that cannot be executed on encrypted data but instead require to be performed on the client. Performance largely depends on the selection of the parts to be executed on the clients. This requirement refers to the optimisation of an SQL query (or operator tree) for an optimised (with regards to performance) query execution.

WP53-R1: Privacy preserving data processing

A processing of a set of data is privacy-preserving if it enables both to obtain the desired result of the process and to maintain the confidentiality of the data.

WP53-R2: Search pattern privacy for word search

During the search phase, it is desired that the adversary does not derive any information from the search queries. More specifically, the adversary may not discover the content of the query (the

queried word); additionally, the adversary should not discover whether two queries were targeting the same word.

WP53-R3: Access pattern privacy for word search

In addition to search pattern privacy, the proposed privacy preserving word search primitive may also ensure access pattern privacy which mainly ensure the privacy of query responses. In addition to query pattern privacy, the adversary may not discover whether the queried (non-disclosed) word exists or not.

WP53-R4: Performance / Efficiency at the client

For a client being the beneficiary of cloud functionality, efficiency at the client means that the constraints induced by the usage of the cloud on the client's side are acceptable from the client's point of view. This efficiency might be measured by the time or complexity need by the interactions with the cloud; or by the size of the local storage induced by the cloud usage.

WP53-R5: Query expressiveness for word search

In most of privacy preserving word search solutions a user can only query one word at a time. An ideal solution would be to allow the user any types of queries (such as conjunction of several words).

3.5.3 Trade-off analysis

This section identifies requirements that combine computation privacy requirements with functional (and sometimes other security requirements).

WP51-R3 Computation friendly confidentiality

The data owner has the requirement that the respective data is stored in the most secure way possible. At the same time, the data owner requires that processing over the data is possible and happens with nearly the same performance as it was done on premise. This requires, however, that encryption schemes are used such that they allow processing over the data (e.g., searchable, additive homomorphic, deterministic, or order-preserving encryption schemes).

The goal is either to improve the performance of a given encryption scheme or enabling the data owner to decide between performance, functionality and security.

WP51-R4: Storage friendly confidentiality

The data owner has the requirement that the respective data is stored in the most secure way possible. The cloud provider as well as the data owner require that the data is stored in the most storage efficient way, minimizing the data storage footprint and with it operation costs.

Traditional encryption schemes do not allow data compression as, for instance, dictionary compressions used in databases. Data compressions can be more effectively applied with deterministically stored data.

Hence, the goal is either to improve the storage of the data by using deterministic encryption schemes or using randomized encryption schemes while having a larger data footprint on the cloud provider's server.

WP53-R6: Privacy preserving data processing with Big Data

Solutions for some particular cloud functionalities may exist. They might be efficient for small databases, but fail to work for bigger ones. As an example, efficient solutions exist for securely querying outsourced encrypted databases, but these solutions are only usable for databases with restricted size. The challenge brought by this requirement is to come up with efficient solutions for databases composed of a huge amount of records. Of course, the underlying notion of "database size" and "efficiency" depends on the application.

WP53-R7: Privacy preserving data processing with efficiency at the cloud

"Efficiency at the cloud" for a solution implementing cloud functionality means that the complexity of the solution is acceptable by the service owners and the clients who use it. What is acceptable in practice is of course application-dependent. As for the client, this efficiency can be measured by different metrics, as time, storage or interaction complexities, or also by deployment costs. Complexity of some solutions might also be compensated with other aspects, such as large scale deployment.

WP53-R8: Privacy preserving data processing with multi-tenancy

Several users may outsource different databases and further several users including data owners should be able to query different databases given that they have the authorisation. Additionally, a data owner should be able to revoke an authorised party at any time.

WP53-R9: Privacy preserving data processing with dynamic data

The new privacy preserving data processing solution should support data dynamicity. The addition, deletion or modification of data should not affect the performance of the underlying solution, seriously.

WP53-R10: Privacy preserving data processing with verifiability

The new privacy preserving data processing solution may additionally assure the correctness of the response to the query while being confidential.

3.5.4 Summary

Table 6 summarizes the computation privacy requirements and captures their relationship with mainly Big Data use cases.

Requirement	Mandatory (M)/ Optional(O)	Related Use Cases and if suitable Requirements
WP5-R1 Data Confidentiality	M	UC4, UC5, UC6
WP51-R1 Efficient initial encryption	M	UC6
WP51-R2 Query analysis for optimised SQL statement execution over remotely stored encrypted data	O	UC6
WP52-R1 Privacy preserving migration with minimum downtime	O	UC6
WP53-R1: Privacy preserving data processing	M O	UC5, UC6 UC4
WP53-R2: Search pattern privacy for word search	O	UC5, UC6
WP53-R3: Access pattern privacy for word search	O	UC5, UC6
WP53-R4: Performance / Efficiency at the client	M	UC4, UC5, UC6
WP53-R5: Query expressiveness for word search	M O	UC6 UC5
WP51-R3 Computation friendly confidentiality	O	UC5, UC6
WP51-R4: Storage friendly confidentiality	O	UC6
WP53-R6: Privacy preserving data processing with Big Data	M O	UC5, UC6 UC4
WP53-R7: Privacy preserving data processing with efficiency at the cloud	O	UC4, UC5, UC6
WP53-R8: Privacy preserving data processing with multi-tenancy	O	UC6
WP53-R9: Privacy preserving data processing with dynamic data	O	UC5, UC6
WP53-R10: Privacy preserving data processing with verifiability	O	UC4, UC5, UC6

Table 6: Computation Privacy Requirements Summary

3.6 Consolidated Requirements

Figure 1 provides a consolidated view on previously enumerated requirements classified with respect to the different use cases, the security requirements and their priorities. While the most challenging requirements for file sharing services are storage privacy with storage efficiency and verifiable ownership which are mainly tackled in WP4 and partly in WP3 (verifiable ownership), topics in WP5 and requirements with respect to verifiable storage and computation are more in line with Big Data services.

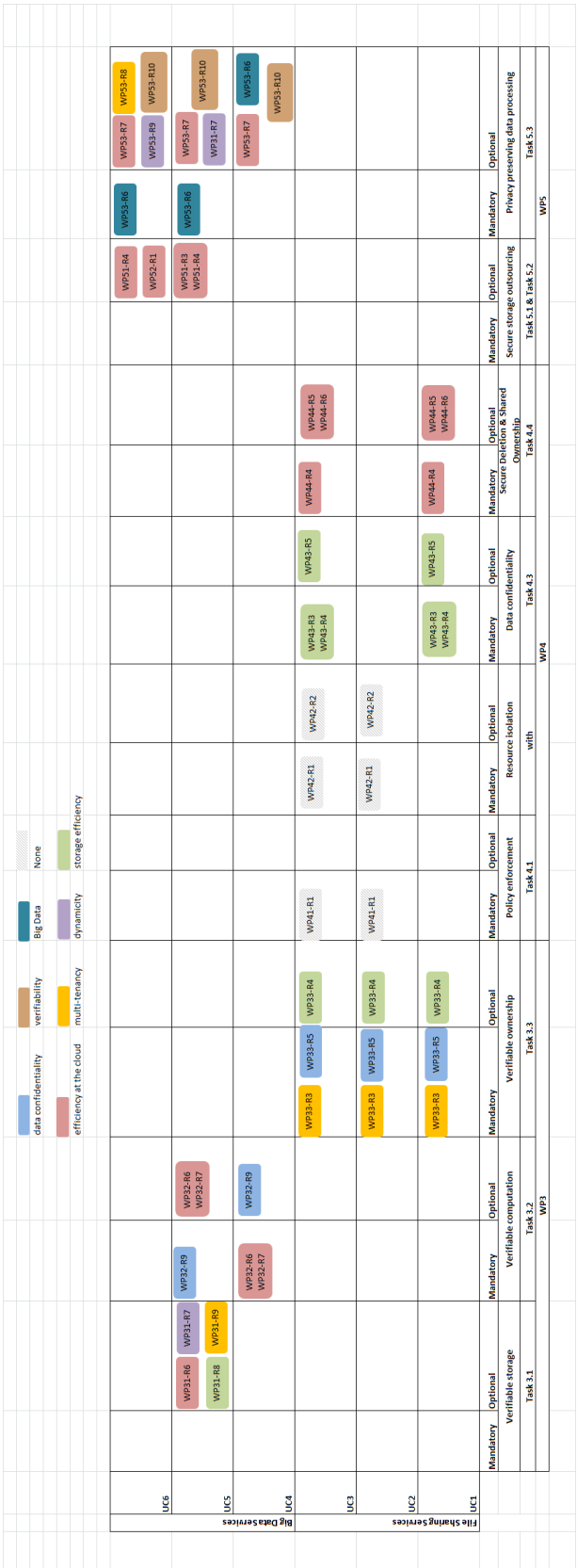

Figure 1: TREDISEC Consolidated Requirements

4 TREDISEC Architecture Requirements

4.1 TREDISEC Challenges

Following the analysis of requirements specific to each TREDISEC security primitive, this section now focuses on the identification of requirements related to the framework itself. Indeed, in addition to the design and implementation of TREDISEC security primitives, the aim of the project is to propose a unified TREDISEC framework that will allow users with different roles such as framework administrators, developers, and security experts, to configure and deploy these security services ensuring storage integrity, storage privacy, computation integrity and computation privacy; moreover additional utilities are also needed for general purposes, such as, visualization (user interfaces), monitoring, logging, accounting and billing, into the targeted cloud system. The idea behind this is to enable a secure and trustworthy yet operational cloud framework.

Since the TREDISEC framework will be deployed into a target cloud system, all requirements extracted from cloud computing architectures [2] are applicable to our case.

In the following sections we describe the requirements that the TREDISEC framework should additionally fulfil from three angles:

- **Architecture requirements**: these requirements complementing the functional requirements defined in section 2.4.1, refer to a specific function of the framework's architecture and are considered as high priority.
- **Quality requirements**: these requirements do not refer to functional aspects but rather to the framework usability and quality. Some of these requirements are achieved if some others from the "architecture" category are fulfilled.
- **Business requirements**: these are requirements from the business point of view. Most of these requirements are achieved if some others from the "quality" and "architecture" categories are fulfilled.

In section 4.5, two tables with the mapping of the requirements are provided.

4.2 Architecture Requirements

WP2A1: Measurable framework

Broadly speaking, the cloud architectures automatically control and optimize resource usage by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, active user accounts). Resource usage can be monitored, controlled, audited, and reported, providing transparency for both the provider and consumer of the utilized service. In this sense we devise a framework for TREDISEC supporting a metering/monitoring capability for those security primitives which are deployed into the target cloud architecture.

WP2A2: Configurable framework

The TREDISEC framework should allow users to configure the instance that wants to deploy in the target cloud system (i.e. which security primitives to deploy and which not, and with some custom features and parameters). This requirement does not aim at a full customization, since default tested configurations of the TREDISEC primitives and combinations of two or more of them should be the desired way of functioning of the framework.

WP2A3: Flexible deployment model

The TREDISEC framework architecture should be designed to be flexible enough to integrate with different cloud service models (SaaS, PaaS and IaaS), deployment options (hybrid, community, private, public), but also with different architectures offered by cloud service providers. This requirement contributes to achieving business requirement B4, since designing the framework to

support multiple deployment options is the basis to support different packaging options and business offerings.

WP2A4: Semi-automated recovery from failure

This requirement refers to computing cloud architecture that provides continuous business operation (i.e. highly resistant to disruption and able to operate in a degraded mode if damaged), rapid recovery if failure does occur, and the ability to scale to meet rapid or unpredictable demands.

WP2A5: Semi-automated build, configuration and deployment processes

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. With regard to this, we envisage a TREDISEC framework that enables an automated (or semi-automated) configuration, building and deployment of the selected security primitives over the target cloud system.

WP2A6: Visibility and reporting

The reporting will be useful for monitoring customer service level agreements (SLAs), system performance, compliance, and also for billing tasks. Visibility is related to transparency and contributes to achieving various business (such as B3, B5, and B6) and quality (e.g. Q6, Q7) requirements. This requirement is clearly related to A6 and has a dependency on it.

WP2A7: Provide User Interfaces

Delivering a user interface for the TREDISEC framework administrator, as well as for developers and security experts, it would be desirable. This requirement contributes to achieving business requirement B5 and quality requirement Q7 described in the next sections.

WP2A8: Scalability

Scalability deals with the ability of the software system to manage increasing complexity when given additional resources. Scalability with large data set operations is a requirement for Cloud Computing. Horizontal scalability is what clouds provide through load balancing and application delivery solutions.

Vertical scalability is related to resources used, much like the old mainframe model. If an application doesn't scale well vertically, it's going to increase the costs to run in the cloud

Hosting applications within an Infrastructure as a Service provider grants us the operational agility to scale our infrastructure very quickly, but the application needs to scale with our infrastructure. One of the benefits of cloud computing is that we can rapidly grow our infrastructure from 1 to 100 servers within a moment's notice, but applications must have some way of sharing work across these newly provisioned servers in order scale in any meaningful way.

Taking in account the aforementioned the TREDISEC security primitives that will be deployed into a cloud system should consider the scalability requirement during their design phase (e.g. designing stateless applications).

WP2A9: Interoperability

Broadly speaking, interoperability can be defined as a measure of the degree to which diverse systems or components can work together successfully. More formally, IEEE (https://www.ieee.org/education_careers/education/standards/standards_glossary.html) and ISO define interoperability as the ability for two or more systems or applications to exchange information and mutually use the information that has been exchanged.

By applying this concept to TREDISEC, the framework should enable the interoperability among the different security primitives and utilities as well as the interoperability with the cloud system where the framework will be deployed.

On the other hand the interoperability among different cloud providers should be also taken into account.

WP2A10: Modular design

The TREDISEC security primitives should follow the modular design approach. It is a design approach that subdivides a system into smaller parts called modules that can be independently created and then used in different systems. A modular system can be characterized by functional partitioning into discrete scalable, reusable modules, rigorous use of well-defined modular interfaces, and making use of industry standards for interfaces.

Table 7 identifies the mandatory and optional requirements among the previously described architecture requirements.

Requirement	Mandatory (M)/ Optional(O)
WP2A1 Measurable framework	O
WP2A2 Configurable framework	M
WP2A3 Flexible deployment model	M
WP2A4 Semi-automated recovery from failure	O
WP2A5 Semi-automated build, configuration and deployment processes	M
WP2A6 Visibility and reporting	O
WP2A7 Provide User Interfaces	M
WP2A8 Scalability	M
WP2A9 Interoperability	M
WP2A10 Modular design	M

Table 7: TREDISEC Architecture Requirements

4.3 Quality Requirements

WP2Q1: System Availability

One big advantage of all types of cloud computing architectures is that, by its nature, cloud computing removes single points of failure. The failure of one node of the system has no impact on Information Availability and does not result in perceivable downtime. Cloud computing provides a highly resilient computing environment.

As an example, the TREDISEC security primitives deployed within the cloud system could increase their availability running more than one instance of each of them. Moreover, ensuring that the security mechanisms are properly tested both at development time and at deployment/execution time will prevent unnecessary service discontinuities and downtimes. This can be enforced by incorporating to the security primitives' development lifecycle well documented testing procedures for developers and operation teams.

WP2Q2: Elasticity

It is the ability of a system to increase the workload on its current and additional (dynamically added on demand) hardware resources (scale out);

Cloud computing allows businesses to expand or contract computing power as required and allows 'bursts' of computing power to be utilised on an on-demand basis. It also allows very effective load balancing.

The TREDISEC security primitives that will be deployed into a cloud system should consider during their design phase to scale easily to handle cloud-sized workloads.

WP2Q3: Security

As the first aim of the project is providing an end-to-end secure TREDISEC framework, this mandatory requirement regroups all security requirements defined in various previous sections.

WP2Q4: Adaptability

The TREDISEC framework should be flexible enough to adapt to the user needs. This means that the user is able not only to reconfigure (and re-deploy) the currently deployed instance with some other set of TREDISEC security primitives, but also to easily adapt to changes in the target cloud infrastructure.

WP2Q5: Performance

Another key requirement that is already classified as functional requirement during the analysis of use cases is the performance of the TREDISEC security primitives deployed in the cloud architecture. For example, disk I/O or access to the RAM may cause intermittent spikes in performance. However, as with traditional software architectures, overall traffic patterns and peaks in system use, account for the majority of performance issues in cloud computing.

WP2Q6: Usability

To provide a friendly user interface which allows end-users to configure, deploy (and re-deploy) and monitoring the TREDISEC security primitives would be helpful.

WP2Q7: Maintainability

Generally speaking, this requirement measures how easily and rapidly a system or equipment can be restored to operational status following a failure, considering concepts like preventive maintenance and Built-In-Test (BIT), required maintainer skill level, and support equipment. Being more specific for the TREDISEC framework, this requirement expects security primitives to be designed and developed to facilitate an easy and fast recovery (automatic or not) from failure and thus, it is closely related to requirement A12.

Table 8 captures the mandatory and optional quality requirements for TREDISEC.

Requirement	Mandatory (M)/ Optional(O)
WP2Q1 System Availability	M
WP2Q2 Elasticity	O
WP2Q3 Security	M
WP2Q4 Adaptability	M
WP2Q5 Performance	M
WP2Q6 Usability	O
WP2Q7 Maintainability	O

Table 8: TREDISEC Quality Requirements

4.4 Business requirements

WP2B1: Quality for business

Quality requirements are understood as areas of demands, requirements or concerns that have an impact on the commercial relation between consumers of cloud services and providers of cloud services. They may also have an impact on the decision for or against the application of cloud computing and more specifically on the TREDISEC framework (deployed within a cloud system), in a specific context.

We can consider under this category the quality requirements described in section 4.3: availability, elasticity, security, adaptability, performance, usability and maintainability.

WP2B2: Market share

According to the DoW, the outcomes of the project will be around Technology Readiness Level, TRL 5/6 ("technology demonstrated in relevant environment, industrially relevant environment in the case of key enabling technologies") and the pilots will be extended to form products. It is also pointed out that individual project components (this is especially true for all the security primitives developed within the project) are planned to be integrated into existing products, such as the SAP's HANA Enterprise Cloud product[4], IBM's SmartCloud[5] and Softlayer[6] offerings, MORPHO's biometric system solutions[7] and other industrial partner's products that were already put into the market.

Therefore to release a unified TREDISEC framework with TRL 5/6 will be a mandatory requirement.

WP2B3: Flexibility

The TREDISEC framework should be flexible enough to support different services and deployment models (SaaS, PaaS and IaaS / Public, Private, Hybrid and Community), but also to adapt to different business exploitation strategies and chargeback models. In this regard, it is desirable that the TREDISEC framework architecture is designed to support modifications easily to handle pricing variations, for example, for promotions and specials that might vary over time or by region.

WP2B4: Stakeholder satisfaction

This requirement refers to identifying the key actors involved in the business ecosystem of cloud system in which the TREDISEC framework is deployed and how all these stakeholders get benefit.

WP2B5: Compliance

Compliance includes all aspects of confirming to regulations. This duty applies to the cloud user and owner of the data and applications hosted in the cloud. Cloud computing creates particular new questions in this context such as compliance in the context of cross-border cloud computing.

As shown in Table 9, apart from requirement **WP2B2**, all business requirements are considered as being mandatory for the TREDISEC framework.

Requirement	Mandatory (M)/ Optional(O)
WP2B1 Quality for business	M
WP2B2 Market share	O
WP2B3 Flexibility	M

[4] http://hana.sap.com/abouthana.html
[5] http://www.ibm.com/cloud-computing/
[6] www.softlayer.com/
[7] http://www.morpho.com/en/civil-identity/mastering-electronic-id-chain-secure-and-efficient-e-services

WP2B4 Stakeholder satisfaction	M
WP2B5 Compliance	M

Table 9: TREDISEC Business Requirements

4.5 Summary

In previous sections 4.2, 4.3, and 4.4, we have described the requirements that the TREDISEC unified framework should fulfil from three different points of view: architecture, quality and finally business. This does not mean that the envisaged framework should consider all of them individually, since many of the "quality" and "business" requirements are covered by some architectural requirements already.

Table 10 shows the mapping among quality requirements against architecture requirements; Table 11 displays the mapping between business and architecture requirements.

Requirement	**WP2A1: Measurable (O)**	**WP2A2: Configurable (M)**	**WP2A3: Flexible (M)**	**WP2A4:Semi-automated recovery from failure (O)**	**WP2A5: Semi-automated build, configuration and deployment (M)**	**WP2A6: Visibility and reporting (O)**	**WP2A7: Provide User Interfaces (M)**	**WP2A8: Scalability (M)**	**WP2A9 : Interoperability (M)**	**WP2A10: Modular Design (M)**
WP2Q1 Availability (M)				X	X			X		
WP2Q2 Elasticity (O)			X					X		X
WP2Q3 Security (M)										
WP2Q4 Adaptability(M)		X	X		X				X	X
WP2Q5 Performance (M)										
WP2Q6 Usability (O)		X				X	X			
WP2Q7 Maintainability (O)		X		X	X					X

Table 10: Mapping between "quality" requirements and "architecture" requirements.

Requirement	WP2A1: Measurable (O)	WP2A2: Configurable (M)	WP2A3: Flexible (M)	WP2A4:Semi-automated recovery from failure (O)	WP2A5: Semi-automated build, configuration and deployment (M)	WP2A6: Visibility and reporting (O)	WP2A7: Provide User Interfaces (M)	WP2A8: Scalability (M)	WP2A9 : Interoperability (M)	WP2A10: Modular Design (M)
WP2B1: Quality (M)	X	X	X	X	X	X	X	X	X	X
WP2B2 : Market share (O)	X	X	X	X	X	X	X	X	X	X
WP2B3: Flexibility (M)		X	X		X			X	X	
WP2B4:Stakeholder satisfaction (M)	X	X	X	X	X	X	X	X	X	X
WP2B5: Compliance (M)						X				

Table 11: Mapping between “business” requirements and “architecture” requirements.

Security & Functional Requirements for File Sharing services			(M/O)
Privacy	WP41-R1	Semantic and contextually constrained policy enforcement	M
	WP42-R1	Improved resource isolation	
	WP43-R3	Data confidentiality with file-based deduplication	
	WP43-R4	Data confidentiality with block-based deduplication	
	WP41-R2	Privacy-respectful policy enforcement	O
	WP43-R5	Data confidentiality with data compression	
	WP44-R5	Shared ownership with efficiency at the cloud	
	WP43-R5	Assisted deletion with efficiency at the cloud	
Integrity	WP33-R1	Efficient ownership verification	M
	WP33-R2	Verifiable ownership with data confidentiality	
	WP33-R3	Verifiable ownership with data multi-tenancy	
	WP33-R4	Verifiable ownership with data reduction	
	WP33-R5	Verifiable ownership with dynamicity	O
Security & Functional Requirements for Big Data services			
Privacy	WP51-R1	Efficient initial encryption	M
	WP53-R1	Privacy preserving data processing with Big Data	
	WP53-R5	Query expressiveness with word search	
	WP51-R3	Computation friendly confidentiality	O
	WP51-R4	Storage friendly confidentiality	
	WP53-R6	Privacy preserving data processing with Big Data	
	WP53-R7	Privacy preserving data processing with efficiency at the cloud	
	WP53-R8	Privacy preserving data processing with multi-tenancy	
	WP53-R9	Privacy preserving data processing with dynamic data	
	WP53-R10	Privacy preserving data processing with verifiability	
Integrity	WP32-R6	Verifiable computation with efficiency at the cloud	M
	WP32-R7	Verifiable computation with efficiency at the client	
	WP-R8	Verifiable computation over big databases	
	WP32-R9	Verifiable computation with data confidentiality	O
	WP31-R6	Verifiable storage with efficiency at the cloud	
	WP31-R7	Verifiable storage with dynamic data	
	WP31-R8	Verifiable storage with storage efficiency	
	WP31-R9	Verifiable storage with multi-tenancy	
Architecture, Quality & Business Requirements			
Architectural Requirements	WP2A2	Configurable framework	M
	WP2A3	Flexible deployment model	
	WP2A5	Semi-automated build, configuration and deployment processes	
	WP2A7	Providing user interfaces	
	WP2A8	Scalability	
	WP2A9	Interoperability	
	WP2A10	Modular design	
	WP2A1	Measurable framework	O
	WP2A4	Semi-automated build, configuration and deployment	
	WP2A6	Visibility and reporting	
Quality Requirements	WP2Q1	System availability	M
	WP2Q3	Security	
	WP2Q4	Adaptability	
	WP2Q5	Performance	
	WP2Q2	Elasticity	O
	WP2Q6	Usability	
	WP2Q7	Maintainability	
Business Requirements	WP2B1	Quality for business	M
	WP2B3	Flexibility	
	WP2B4	Stakeholder satisfaction	
	WP2B5	Compliance	
	WP2B2	Market share	O

Table 12: Summary of TREDISEC Requirements

5 Conclusions

This report first identified the generic functional and security requirements that TREDISEC solutions intend to meet (cf. Section 2). The starting point is the use-cases defined in D2.1 [1] that showcase that there is an actual need for security solutions that offer to cloud customers security and privacy guarantees, and at the same time have a little impact on the performance and the mode of operations of cloud systems. Next, in Section 3, a more comprehensive description of security requirements is provided; the document further shows how achieving security when in conflict with cloud services' functionalities. While these requirements are mostly related to cloud customers and servers, Section 4 describes the needs with respect to developers, administrators and cloud service providers and enumerates the architecture, business and quality requirements in order to reach a unified TREDISEC framework. As a summary, Table 12 (see previous page) regroups all these requirements and shows their priority (Mandatory (M)/ Optional (O)).

6 References

[1] C. Soriente, K. Doka, M. Kohler, J. F. Ruiz, R. M. Vieira, B. Gallego-Nicasio, J. Bringer, R. Lescuyer and D. Vallejo Garcia, "D2.1 Description of the context scenarios and use case definition," TREDISEC project, 2015.

[2] R. Bhaskar Prasad, A. Jukan, D. Katsaros and Y. Goeleven, "Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach," *Journal of Grid Computing,* pp. 3-26, 2011.