
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JOSÉ BENARDI DE SOUZA NUNES

# AN EXTENSIBLE TAXONOMY ON PRIVACY AND CONFIDENTIALITY

CAMPINA GRANDE - PB
2022

**JOSÉ BENARDI DE SOUZA NUNES**

# AN EXTENSIBLE TAXONOMY ON PRIVACY AND CONFIDENTIALITY

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande, pertencente à linha de pequisa Privacidade e Segurança da Informação e área de concentração Ciência da Computação, como requisito para a obtenção do Título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Andrey Elísio Monteiro Brito

**CAMPINA GRANDE - PB**
**2022**

N972e Nunes, José Benardi de Souza.
An extensible taxonomy on privacy and confidentiality / José Benardi de Souza Nunes. – Campina Grande, 2022.
57 f.: il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2022.
"Orientação: Prof. Dr. Andrey Elísio Monteiro Brito"
Referências.

1. Classificação. 2. Privacidade. 3. Confidencialidade. 4. Taxonomia. I. Brito, Andrey Elísio Monteiro. II. Título.

CDU 025.4(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA MARIA ANTONIA DE SOUSA CRB 15/398

MINISTÉRIO DA EDUCAÇÃO
**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**
POS-GRADUACAO CIENCIAS DA COMPUTACAO
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

## FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

**JOSÉ BENARDI DE SOUZA NUNES**

AN EXTENSIBLE TAXONOMY ON PRIVACY AND CONFIDENTIALITY

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 14/03/2022

Prof. Dr. ANDREY ELÍSIO MONTEIRO BRITO, Orientador, UFCG

Prof. Dr. REINALDO CÉZAR DE MORAIS GOMES, Examinador Interno, UFCG

Profa. Dra. KEIKO VERÔNICA ONO FONSECA, Examinadora Externa, UTFPR

Prof. Dr. EDUARDO DE LUCENA FALCÃO, Examinador Externo, UFRN

Documento assinado eletronicamente por **REINALDO CEZAR DE MORAIS GOMES**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 15/03/2022, às 07:19, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Eduardo de Lucena Falcão**, **Usuário Externo**, em 15/03/2022, às 08:35, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Keiko Veronica Ono Fonseca**, **Usuário Externo**, em 15/03/2022, às 14:42, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **ANDREY ELISIO MONTEIRO BRITO**, **PROFESSOR 3 GRAU**, em 16/03/2022, às 14:33, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **2173662** e o código CRC **BE49FBF2**.

# Resumo

O desafio de garantir privacidade e confidencialidade é um que frequentemnte temos de adaptar às aplicações em particular que observamos. Atualmente existe a falta de um enquadramento em comum para comparar e avaliar as diferentes soluções em termos de privacidade e confidencialidade. Se torna mais difícil navegar em uma área em busca da solução apropriada se não podemos facilmente posicionar uma dada aplicação e suas alternativas no contexto daquela área. Tudo isso torna mais difícil para um interessado estabelecer se é possível re-aplicar estratégias a novas aplicações e problemas. Consequentemente, isso nos impede de convergir a soluções melhores. Este trabalho propõe uma taxonomia centrada nas vulnerabilidades de privacidade e confidencialidade de uma aplicação como forma de prover o enquadramento em comum que discutimos. Para prover esta taxonomia relacionada à privacidade e confidencialidade neste trabalho são analisadas e classificadas vinte e uma aplicações, as quais são agrupadas em dezenove tipos distintos de serviços. A taxonomia deste trabalho foi validada com sucesso através de uma demonstração de ortogonalidade (demonstrar que dimensões são disjuntas) e uma demonstração de utilidade, tendo sido aplicada com sucesso em um sistema de análise inteligente de infecção enquanto sistema caso de uso. Também são explicadas as facetas e níveis da taxonomia e provido o método empregado para construí-la.

**Palavras-Chave:** Privacidade, Confidencialidade, Taxonomia

# Abstract

The challenge of ensuring privacy and confidentiality is often tailored to the specific application under observation. Currently, there is a lack of a common framework to compare and assess different solutions in terms of privacy and confidentiality. This hinders navigating a field to search for the appropriate solution if we cannot easily place an application and alternative solutions in the context of its area. All of this makes it harder for an interested party to establish whether it is possible to reapply strategies to new applications and problems. Consequently, it hinders us from converging to better solutions. This work proposes a taxonomy centered on applications' privacy and confidentiality vulnerabilities to provide the common framework I have discussed. To provide this taxonomy on privacy and confidentiality, I analyzed and classified twenty-one applications, grouped into nineteen distinct types of services. I have successfully validated this work's taxonomy through an orthogonality demonstration (show that dimensions are disjoint) and a utility demonstration. Furthermore, I have successfully applied this work's taxonomy in an intelligent infection analysis system in a use case analysis. I have also explained the taxonomy's facets and levels and provided the method used to build it.

**Keywords:** Privacy, Confidentiality, Taxonomy

# Acknowledgments

I thank my mother and brother for always being at my side and believing in me.

I thank my friends for listening to my problems and pushing me forward.

I thank professor Andrey Brito for the guidance and feedback.

I thank professor Francisco Neto for the advice and insights.

I thank my work colleagues, with which I have learned so much along the way.

# Acronyms

**API** Application Programming Interface. 11, 34, 39

**AWS** Amazon Web Service. 46

**DCAP** Data Center Attestation Primitives. 13

**DFD** Data Flow Diagram. 6, 7

**ECDSA** Elliptic Curve Digital Signature Algorithm. 13

**GDPR** General Data Protection Regulation. 1, 53

**hTMM** Hybrid Threat Modeling Method. 6

**IAS** Intel SGX Attestation Service. 13

**LAS** Local Attestation Service. 15

**LGPD** Lei Geral de Proteção de Dados. 1, 53

**LINDDUN** linkability, identifiability, nonrepudiation, detectability, disclosure of information, unawareness, noncompliance. 6, 7, 53

**mTLS** Mutual Transport Layer Security. 47

**NIST** National Institute of Standards and Technology. 11

**OCTAVE** Operationally Critical Threat, Asset, and Vulnerability Evaluation. 6, 53

**PnG** Personae non Grata. 6

**SCONE** Secure CONtainer Environment. 14

**SDK** Software Development Kit. 14

**SGX** Intel Software Guard Extensions. vi, viii, 12–14, 47–50

**SPIFFE** Secure Production Identity Framework for Everyone. 11, 12, 45, 46

**SPIRE** SPIFFE Runtime Environment. vi, viii, 11, 12, 45–47

**SQUARE** Security Quality Requirements Engineering Method. 6

**STRIDE** Spoofing identity, Tampering with data, Repudiation, Information Disclosure, Denial of service, Elevation of privilege. 7, 53

**SVID** SPIFFE Verifiable Identity Document. 12, 46, 47

**TCB** Trusted computing base. 10, 11, 14

**TEE** Trusted Execution Environment. 10–13, 45, 47

**TLS** Transport Layer Security. 13

**TPM** Trusted Platform Module. 12

**VM** Virtual Machine. 11, 45

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Ensuring privacy and confidentiality often becomes a challenge that we must tailor to a specific application. We currently lack a common framework to compare and assess different solutions in terms of privacy and confidentiality. All of this makes it harder to establish whether we can apply known strategies to new applications and problems.

## 1.1 Motivation and Context

The lack of a common framework makes it harder to provide visibility to existing solutions, and by extension, employ the progress made in the area. From the point of view of an application's stakeholder, it is harder to navigate and search for the correct alternative if we cannot easily contextualize the application in terms of privacy and confidentiality vulnerabilities. There is also a vested public interest in ensuring compliance with legislations that demand a commitment with privacy, such as Lei Geral de Proteção de Dados (LGPD) in Brazil and General Data Protection Regulation (GDPR) in the European Union (TEMER et al., 2018) (BREITBARTH, 2019).

A taxonomy centered on applications' privacy and confidentiality vulnerabilities would provide the common framework we have discussed. Unfortunately, in the field of technology, most efforts for the creation of taxonomies have been made without a proper methodology, falling back to an *ad hoc* approach. This reliance on *ad hoc* approaches murks the logic applied in the creation of the taxonomy, which makes it hard to evaluate its consistency and correctness. To avoid these problems, this work employs a distinct methodology envisioned for the area of Information Systems to the context of privacy and confidentiality (NICKERSON; VARSHNEY; MUNTERMANN, 2013) (USMAN et al., 2017).

The importance of correctly approaching matters of privacy and confidentiality has been recently highlighted in view of the pandemic waves and the need for tracking contagion and notify new cases (GHAYVAT et al., 2021) (MUNZERT et al., 2021) (SUDRE et al., 2021). Applications in the realm of intelligent infection analysis, not unlike my use case system, handle extremely privacy-sensitive data in all stages, from an input in the form of a user's medical information to equally sensitive information in diagnosis. This particular format of applications recently has become even more relevant through applications to track Covid-19

(Ahmed et al., 2020). It becomes necessary to track and regulate the access of possibly sick individuals to spaces to contain the spread of disease. It is in everyone's best interest that systems capable of detecting symptoms of a possible infection do inform the possibly sick individuals, so they can take the necessary precautions or seek medical help. On the other hand, such systems deal directly with incredibly privacy-sensitive data, going from pictures and videos of people to preliminary diagnoses (GHAYVAT et al., 2021) (MUNZERT et al., 2021) (SUDRE et al., 2021). This management of privacy-sensitive data raises the need to ensure that these systems deal responsibly and sensibly with such information.

Furthermore, the different applications within the context of smart cities described in Margarita *et al.* (ANGELIDOU et al., 2017) fall into an application profile pertinent to this work. Smart city applications include examples that manage extremely privacy-sensitive information such as users' trajectories, energy profiles, and pictures of the users themselves. Among these applications, we have examples of inputs and outputs with different levels of privacy sensibility. For instance, smart grids have very sensitive input in a user's energy profile, but its output can be less sensitive such as a residence's total energy consumption.

In this work, I contribute to the state of the art by creating a taxonomy to help position a given application in the context of privacy and confidentiality vulnerabilities. The value of this taxonomy becomes even more palpable given the lack of taxonomies focused on privacy (USMAN et al., 2017).

I have also elected to employ this work's taxonomy on a use case analysis, the use case being an intelligent infection analysis system endowed with body temperature monitoring and facial recognition. By evaluating how my taxonomy classifies each system's services, I can discern possible vulnerabilities and address them accordingly. Finally, I have used the solutions I have formulated to create a new version of each service where said vulnerabilities have been dealt with or mitigated.

## 1.2 Objectives

It is the overall objective of this work to construct a taxonomy to provision a common ground for the classification of applications in terms of privacy and confidentiality.

The specific objectives of this work are the following:

- Conciseness, the taxonomy is meaningful without being unwieldy or overwhelming.
- Extensibility, the taxonomy can easily comport a new dimension or characteristic of an existing dimension.
- Explainability, the taxonomy dimensions and characteristics explains well an object.
- Reproducibility, the taxonomy can be reproduced insofar as the arbitrary decisions are outright described.

## 1.3 Method

The steps that combined form the method employed in this work are the following:

- Elect data-driven services for the taxonomy construction.
- Assess the characteristics of data-driven services regarding privacy and confidentiality vulnerabilities.
- Pool the insights taken from the elected data-driven services into an artifact in the form of a taxonomy.
- Elect a relevant data-driven and privacy-sensitive system for a use case analysis.
- Discern mitigations for vulnerabilities recognized in the use case system through the taxonomy's employment.

## 1.4 Document Organization

I have organized the rest of this document in the following manner. In Chapter 2, I discuss related works. In Chapter 3, I discuss the theoretical background that I depend upon, such as key concepts in the realm of privacy and adjacent but vital attributes such as confidentiality and integrity. In Chapter 4, I present the taxonomy I have created in this work and reference the method I have employed to do so. In Chapter 5, I showcase my use case study system and its services. In Chapter 6, I apply my taxonomy to the use case system's services and discuss the implication of the resulting classifications. In Chapter 7, I discuss the improvements I have implemented while keeping in mind what the taxonomic classification disclosed about

the use case system's services. Finally, in Chapter 8, I conclude by briefly discussing the contributions of my work.

# 2 Related Work

In this chapter, I review relevant works related to this one. More specifically, I discuss references associated with the construction of taxonomies, privacy-enhancing technologies, privacy-sensitive applications, and threat modeling.

## 2.1 Taxonomy construction

Nickerson *et al.* (NICKERSON; VARSHNEY; MUNTERMANN, 2013) point out the absence of taxonomies created through a well-defined method in the fields of Information Systems, Computer Science, and Non-Information System Business. The authors present an iterative method to construct taxonomies that inherently assesses existing elements and applies the taxonomy on elements as we build the taxonomy. The method's authors showcase it by creating a taxonomy on mobile applications. Demonstration through illustration is employed to provide validity to taxonomies. The method includes establishing ending conditions to the taxonomy, and only deems the taxonomy finished once the built taxonomy has met the requirements specified in the ending conditions.

Usman *et al.* (USMAN et al., 2017) make a systematic mapping study on taxonomies in Software Engineering and point out the lack of taxonomies created through a well-defined method. The authors also notice a lack of privacy-focused taxonomies. The authors distinguish between three different forms of validation for taxonomies: orthogonality demonstration, benchmarking, and utility demonstration. Orthogonality demonstration refers to the orthogonality of the taxonomy dimensions, whether the taxonomy dimensions are mutually exclusive. "Benchmarking" consists of comparing the taxonomy with similar classification schemes. Utility demonstration is done by actually classifying subject matter examples. The authors point to facet-based taxonomies as an alternative to new and emerging fields. Other works such as Erlenhov *et al.* (ERLENHOV et al., 2019) have proposed facet-based taxonomies for the field of Software Engineering.

## 2.2 Threat Modeling

In the study done by Schevchenko *et al.* (SHEVCHENKO et al., 2018) twelve different threat modeling techniques are discussed and compared. Some methods deal with perspectives fun-

damentally different from the one guiding my work. For instance, the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) method focuses on assessing organizational risks and does not address technical risks such as a weak encryption. One could complement techniques that work with distinct perspectives with the taxonomy presented in this work. It is not unheard of to use different threat modeling techniques to complete each other. For instance, the authors discuss the Hybrid Threat Modeling Method (hTMM) that itself is the combination of the methods Security Quality Requirements Engineering Method (SQUARE), Security Cards, and Personae non Grata (PnG).

The methods encompassed by hTMM are substantially different. Let us compare Security Cards to PnG. Security Cards are not a formal method and are typically used to brainstorm. Analysts use a deck of cards to help them answer questions about an attack, such as who the attacker might be, the assets of interest, how the attackers can carry out the attacks, and why they might target the system. On the other hand, PnG is a people-centric framework. Analysts construct people's profiles with an adversarial mindset seeing people as potential attackers. The profiles describe a person's motivations, skills, and goals. The idea is to help an analyst understand the system's vulnerabilities through the lens of a human attacker. The act of joining these profoundly different methods into one more comprehensive strategy sets the precedent of distinct approaches complementing each other.

The linkability, identifiability, nonrepudiation, detectability, disclosure of information, unawareness, noncompliance (LINDDUN) method serves as a systematic approach to privacy assessment. One of the core elements of the LINDDUN framework is the Data Flow Diagram (DFD). The DFD is a graphical representation of the "flow" of data through an information system. We can break down LINDDUN into six different steps. First, we formulate the DFD for the target system. Next, we map privacy threats to DFD elements. Once we have mapped the privacy threats, we identify threat scenarios. The fourth step consists in prioritizing the identified threat scenarios. Next, we elect mitigation strategies. Lastly, we select appropriate privacy-enhancing technologies.

As LINDDUN is a privacy-centered approach to threat modeling, this work's taxonomy parallels LINDDUN. A privacy-centered taxonomy that describes how data enters and leaves a system contributes to the same task as a DFD. In other words, a privacy-centered taxonomy is an artifact analogous to a DFD. With this in mind, we might be able to employ this work's

taxonomy in an existing privacy-centered approach to threat modeling such as LINDDUN. Although, this idea elicits further research.

Another technique that relies on DFD is the Spoofing identity, Tampering with data, Repudiation, Information Disclosure, Denial of service, Elevation of privilege (STRIDE) method. Users typically use STRIDE to identify system events, entities, and the system's boundaries. As a threat modeling technique that demands understanding how data flows from and to the system, we might augment it by using a privacy-centered taxonomy such as the one constructed in this work.

# 3 Theoretical Background

This chapter defines key terms and presents some of the technologies I have employed throughout my work. I define privacy attributes my taxonomy works on, as these terms can have a whole range of meanings. Then, I discuss concepts and solutions needed to safeguard attributes adjacent to privacy, such as confidentiality and integrity. Readers versed in these topics may want to skip this chapter.

## 3.1 Privacy attributes

We can break down the creation of a secure and privacy-preserving application into five challenges: confidentiality, integrity, availability, accountability, and privacy. Confidentiality implies that an entity's data and computations are to be kept confidential from unauthorized parties. Integrity refers to whether data has been tampered with or are as expected. If we ensure integrity, we guarantee that data and the computations on it have not been tampered with by some party. Integrity also ensures that any violation is to be detected. Intuitively, availability refers to the quality of a given service being able to be used or obtained. It is not uncommon to expect a service to meet a Service Level Agreement. Accountability refers to the capability of identifying a party with undeniable evidence as responsible for something (Xiao; Xiao, 2013).

In contrast, privacy is a more subtle concept. Confidentiality, integrity, and accountability influence the attribute privacy. A camera placed to record and transmit images of passersby is by design a privacy problem. On the other hand, failing to safeguard the camera's communication to its back-end server is a confidentiality problem that also leads to a privacy problem. (Eckhoff; Wagner, 2018)

To have a clear concept of privacy, I follow Nissenbaum's definition of privacy as contextual integrity, where the societal norms of a given context constrain the expectations for data collection and dissemination. For instance, in the context of a therapist's visit, we expect that only relevant data is collected, and the therapist should not disseminate these data outside of their practice. Any collection or use of data beyond these expectations represents a privacy violation. This privacy definition agrees with the work of Ponciano *et al.* (PONCIANO et al., 2017) who discuss how privacy-related notions and expectations also depend

on social factors. Throughout this work, I focus on privacy, integrity, and confidentiality (NISSENBAUM, 2004).

In the review done by Eckhoff *et al.* (Eckhoff; Wagner, 2018), the authors make a distinction between privacy types, which they divide into five classes: Location, State of Body & Mind, Behavior & Action, Social Life and Media. By choosing to group both privacy of thoughts and body-related privacy into the category "State of Body & Mind" they diverge from Finn *et al.* (FINN; WRIGHT; FRIEDEWALD, 2013), who keep body and mind in two distinct categories.

Discerning whether data is privacy sensitive is challenging. If it becomes necessary to understand what data cutout is privacy sensitive, such as when using strategies as data splitting, we find ourselves with a possibly more significant challenge. (DOMINGO-FERRER et al., 2019) It is necessary to keep in mind that seemingly innocuous data may expose users' privacy. Logging history, which represents users via deterministic pseudo-anonymized ids, may seem harmless at first. However, this logging history could reveal behavioral patterns of the users.

## 3.2 Privacy-preserving technologies

Domingo-Ferrer *et al.* (DOMINGO-FERRER et al., 2019) go over a whole slew of privacy techniques, both cryptographic and non-cryptographic. Among non-cryptographic methods, the authors discuss data anonymization and data splitting, which both rely on deciding which attributes are direct identifiers and quasi-identifiers to work correctly (SAMARATI; SWEENEY, 1998). Furthermore, data splitting depends on the cloud service providers that host the data chunks not colluding (YANG; LI; NIU, 2015). The discussed cryptographic techniques do not entail such problems and produce inherently safer results by allowing operations on encrypted data but have their challenges. Secure Multi-Party Computation inexorably distributed and iterative nature limits its practicality to specific applications (e.g., voting) (GOLDREICH; MICALI; WIGDERSON, 1991). Homomorphic Encryption is not iterative but provides minimal operations or otherwise prohibitive costs (SHAN et al., 2018).

Domingo-Ferrer et al. (DOMINGO-FERRER et al., 2019) extend their discussion to the privacy model of differential privacy. It is worthy of notice that it provides quantifiable privacy guarantees through its statistical model. However, its most practical form is iterative

and assumes that we keep the data away from malicious parties, and access to sanitized queries is limited to ensure privacy guarantees (DWORK; ROTH et al., 2014).

In the review done by Eckhoff *et al.* (Eckhoff; Wagner, 2018), the authors lay out applications in smart cities as a significant threat to users' privacy. The pervasiveness of its applications and their inter-connectivity raise a sizable challenge to the matter of user sovereignty given the volume of data collected, and the various ways said data are employed. The authors report that privacy protection or information on privacy policies was still scarce with few exceptions. The authors notice that the privacy solutions for different solutions are often similar, which would indicate the possibility of generic privacy patterns.

## 3.3 Confidentiality-preserving technologies

As I have discussed in Section 3.1 ensuring confidentiality is a requirement to ensure privacy. Intuitively, adjusting what datum we should deliver is entirely pointless if we can't withhold data from those who should not access it. With this in mind, I shall explore some concepts and solutions related to the challenge of protecting confidentiality.

From the perspective of ensuring privacy, we can understand confidentiality as protecting data by controlling who access it. Protecting data is a known problem that requires protecting the applications that manage it. One approach is to verify the integrity of the platform and the code running on it. This process is known as attestation. If we verify the platform and code before providing any sensitive data, there will be evidence for the robustness of what will handle these data. For example, only code approved by a committee or properly reviewed should access sensitive data. In this Section, I explore the concept of Trusted Execution Environment (TEE) as it is recurrent in solutions that aim at preserving confidentiality. I also discuss some of the confidentiality-preserving technologies I have employed in this work.

### 3.3.1 The concept of Trusted Execution Environment

A TEE is an isolated processing environment that provides a specific set of guarantees. It offers isolated execution of applications (commonly called trusted applications), the integrity of the application code executing on the environment, the confidentiality of the data processed, and integrity of the Trusted computing base (TCB) (SABT; ACHEMLAL; BOUAB-

DALLAH, 2015). According to the National Institute of Standards and Technology (NIST) definition, a TCB is the totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination responsible for enforcing a security policy. Moreover, a TEE may support remote attestation. The TEE can prove its trustworthiness and security properties to other entities through the remote attestation process.

Demigha and Larguet (DEMIGHA; LARGUET, 2021) present a set of properties to consider for TEEs. They give several criteria in the cloud computing perspective, from security to functionality and deployability. Security properties such as isolation, confidentiality, and integrity protection are must-have features for confidential applications. On top of that, in the context of cloud computing in untrusted environments, we should consider functional criteria such as TCB size and remote attestation support. The deployability criteria resume essential features which make it easier to deploy confidential applications into a cloud-native environment. The use of legacy application code, the performance impact in TEE usage, Virtual Machine (VM) migration, and ecosystem size are essential criteria to inspect.

### 3.3.2 Provisioning identities with SPIRE

One example of a framework that helps validate platforms and applications is the Secure Production Identity Framework for Everyone (SPIFFE). There are multiple implementations of SPIFFE. SPIRE is one of its implementations, and its authors envisioned it as a reference implementation. A SPIRE deployment comprises a SPIRE Server and at least one SPIRE Agent. The server is the signing authority for identities issued to a group of workloads mediated by agents. In this context, a workload is equivalent to an application that requires an identity. The agent provides the Application Programming Interface (API) known as SPIFFE Workload API locally to workloads which is why it must be at the node on which workload is running. The SPIRE server maintains a registry of workload identities and the requirements that must be verified for those identities to be issued. (FELDMAN et al., 2020)

SPIRE utilizes various attestor plugins to carry out attestation for either nodes and workloads. We can use different attestors depending on the deployment type, such as a Kubernetes or an OpenStack attestor. The SPIRE server must perform attestation on the agent nodes before carrying out their respective roles. We perform node attestation by collecting the results from plugins related to node attestation available on both the server and the client. Work-

load attestation works similarly with a set of available pertinent plugins. Once attestation is successful, the SPIRE components issue to workloads a cryptographic identity in the form of a SPIFFE Verifiable Identity Document (SVID). An SVID is a document through which a workload proves its identity to a resource or caller. We consider an SVID valid if a relevant authority within the SPIRE deployment has signed it. An SVID contains a single SPIFFE ID, a string that identifies workloads that fit the criteria via attestation and represents the service's identity. The SVID encodes the SPIFFE ID in a cryptographically verifiable document, usually an X.509 certificate. (FELDMAN et al., 2020)

Once the workload has been verified and granted an SVID it can use the credential to join the application. For example, the workload can use the SVID to get access to configurations, databases, and message channels.

The security guarantees of the workload will depend on the attestation mechanisms. SPIRE supports attestor plugins based on software-measured properties (such as the container image hosting the workload) and hardware-measured properties (such as attestation via Trusted Platform Module (TPM) (TASSYANY, 2021), or SGX (SILVA, 2021). We can, in turn, use these security guarantees to help uphold the confidentiality of a workload, which, as I have discussed in Section 3.1 is a requirement to ensure privacy. In the following Subsection, I explore the SGX solution and its properties, as this is one of the primarily employed solutions in this work.

### 3.3.3 INTEL SGX

Although attestors based on hardware support in SPIRE are still limited, they provide a much more robust verification. Therefore, in this work, I consider hardware-measured properties for attestation. More specifically, I consider SGX which is a set of instructions and changes to the memory access mechanisms added to the x86 architecture. SGX is a hardware-assisted TEE that allows the creation of encrypted isolated memory regions in the memory address space of an application. These protected memory regions are named enclaves and have access control enforced by hardware. An SGX-enabled processor checks the operating system memory mapping, ensuring that only the proper enclave instructions can access protected memory. The TEE's trusted computation base is made up of hardware mechanisms that control access to these protected memory regions. A processor with SGX verifies the memory

mapping decisions from the operating system to ensure that only enclave code has access to protected memory pages. (ANATI et al., 2013a)

An SGX application has one untrusted portion but can include several trusted independent parts (COSTAN; DEVADAS, 2016). As an enclave can not perform system calls due to the TEE isolation properties, the untrusted application part is always responsible for input and output operations and instantiating enclaves. In any case, enclaves must be designed never to leak sensitive data in plain text.

To ensure integrity, SGX provides local and remote attestation functionalities (ANATI et al., 2013a) in the forms of memory protection and remote attestation. Once the system has loaded the application code into an SGX enclave, the code cannot be modified or inspected. Even code with higher privileges, such as the operating system, cannot do it. On top of that, one cannot steal the data if this code receives said data through protected channels, such as Transport Layer Security (TLS) connections. This implementation employs new instructions and units added directly to the processor, enabling higher security levels than what software alone can provide (COSTAN; DEVADAS, 2016).

The remote attestation feature completes the enclave's confidentiality and integrity protection. Third parties can use remote attestation to ensure that the execution of an expected piece of code happens inside an enclave in an SGX endowed server. The attestation process is a mechanism by which a challenger can gain confidence that an enclave is running in an SGX-enabled platform (SCARLATA et al., 2018). Also, through this process, a challenger can verify the identity associated with an enclave. The enclave identity, also known as MRENCLAVE, is an SHA-256 digest of an internal record of all the activities done during the enclave construction (ANATI et al., 2013b).

Thus, the digest includes all pages of code, data, stack, the heap, relative position of the pages, and all security flags associated. Intel currently provides two approaches to perform remote attestation. The first one, using the Intel SGX Attestation Service (IAS), is a client platform-focused approach that uses a privacy-preserving group signature scheme that is only verifiable by IAS. The second one, the Intel's Data Center Attestation Primitives (DCAP), is based on Elliptic Curve Digital Signature Algorithm (ECDSA) signatures (SCARLATA et al., 2018). These primitives allow the construction of on-premise attestation services, enabling third parties to build their attestation infrastructure, especially useful for enterprises

that do not want to outsource trust decisions to Intel and for environments that need to avoid internet access latencies.

### 3.3.4 SCONE

Using SGX is by no means trivial. Some limitations, such as not invoking system calls directly, make for a steep learning curve. Some tools set themselves to address SGX 's adoption barriers by enabling the execution of unmodified applications inside enclaves. The Secure CONtainer Environment (SCONE) toolkit is one of those (ARNAUTOV et al., 2016), also providing tools for Docker and Kubernetes integration. We can look at SCONE as a *runtime* built on top of SGX that offers the inclusion of applications without the need for modifications. SCONE's features are made possible by re-implementing base libraries that are intermediaries to the operating system.

This approach makes SCONE a better option than the SGX Software Development Kit (SDK) to port existing workloads to SGX enclaves. To run workloads inside enclaves using the SGX SDK, a developer must rewrite the code using specific libraries with limited features. Also, the software development kit limits the developer to C and C++ programming languages.

To use SCONE in a remote environment, a client enables the creation of configuration files that create containers and sets up the environment for secure communication. SCONE securely delivers sensitive configuration such as credentials during the container start-up (injected into the application's enclave as environment variables, files, or command line parameters).

The Configuration and Attestation Service (PALÆMON) makes possible the transfer of confidential information. PALÆMON is an SGX -based component that persists confidential information and provides this information only to attested parties. Being itself protected by SGX , PALÆMON could run either inside or outside the user's premises (GREGOR et al., 2020). Using PALÆMON, an operator can describe the workload, properties, security constraints, and the configuration for an application. Then, the operator deploys the application remotely and, to receive the configuration, the application goes through an attestation process that ensures the properties and security constraints. In this way, PALÆMON verifies the enclave identity and the properties of the SGX TCB. Another service that runs at each

server, Local Attestation Service (LAS), makes the connection between PALÆMON and the executing application.

# 4 Taxonomy

To assess the situation of a given service in terms of privacy, confidentiality, and security threats, one needs to understand how the service interacts with the information it receives and produces. I have envisioned a taxonomy concerning a service's interaction with its input and output vectors as a solution. In this section, I discuss the method used to create this taxonomy, its definition, its ending conditions, showcase its organization, and describe each iteration in its creation.

## 4.1 Construction method

To make it easier for others to evaluate, reproduce, and extend my taxonomy, I make use of a well-defined method instead of following an *ad hoc* approach. I have employed the iterative process proposed by Nickerson et al. (NICKERSON; VARSHNEY; MUNTERMANN, 2013), who originally envisioned it for the area of Information Systems. The steps one has to follow in this method can be seen in Figure 1.

We can roughly divide the method into three parts. Determining the meta-characteristic, determining the ending conditions, and the iterative loop that only stops when the taxonomy meets the ending conditions. The meta-characteristic is the most comprehensive characteristic that acts as the basis to devise the taxonomy's features. We must align the meta-characteristic with the taxonomy's purpose, and each of the taxonomy's characteristics must be a logical consequence of the meta-characteristic.

The ending conditions stipulate the requirements that the taxonomy must meet to determine when to terminate its construction. We can divide these conditions into objective and subjective. We can readily understand objective conditions without relying on a subjective perception in opposition to subjective conditions. For instance, given an arbitrary definition of a taxonomy, an objective ending condition is that the created taxonomy must meet said definition. Meanwhile, a subjective ending condition would be that the taxonomy is "concise". The meaning of "concise" is not readily understood and can vary from person to person.

All steps going from three to seven in Figure 1 represent the iterative loop. The loop has two branches, an empirical-to-conceptual approach, and a conceptual-to-empirical approach.

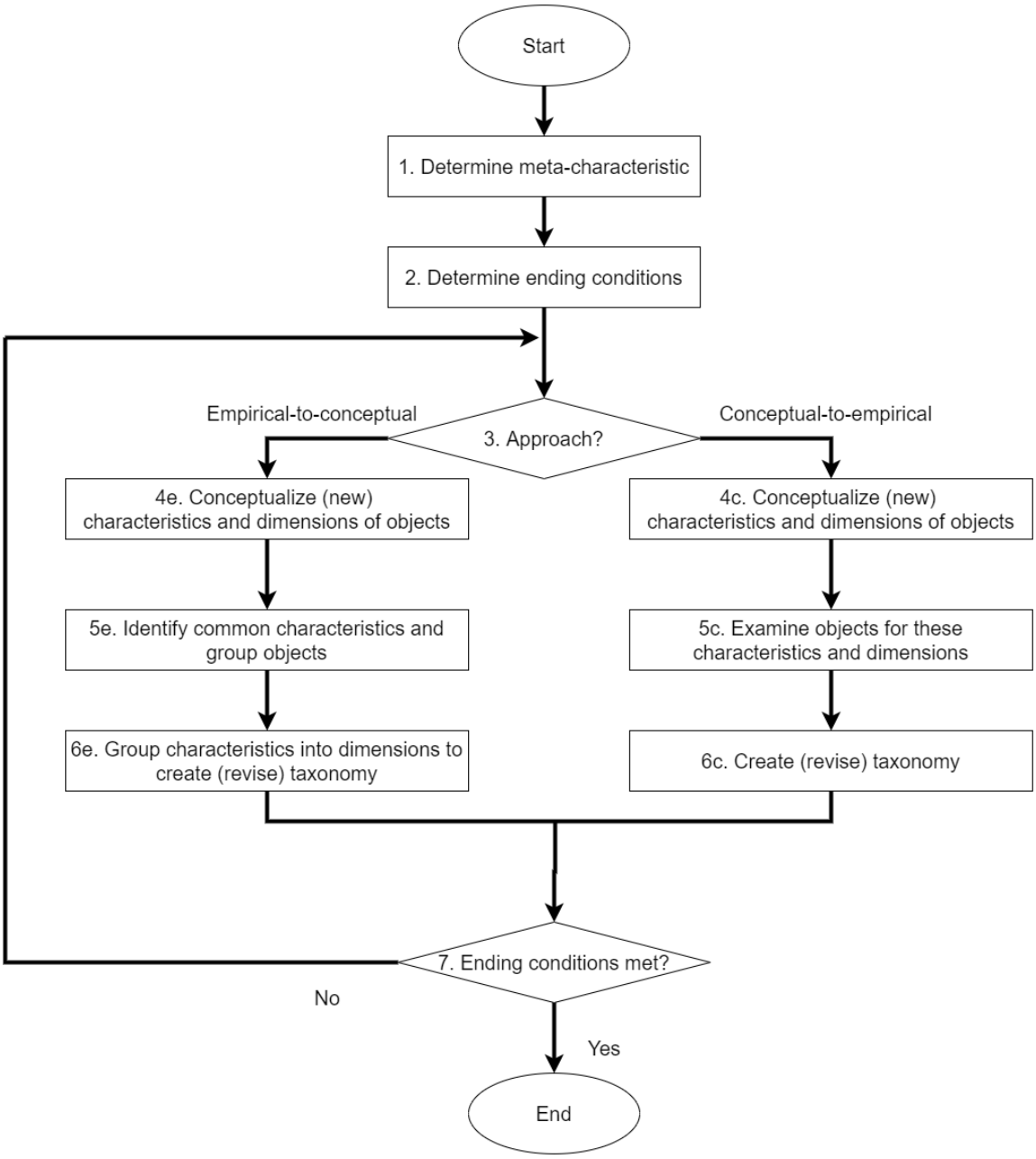


Figure 1: Taxonomy Construction, based on (NICKERSON; VARSHNEY; MUNTERMANN, 2013)

In the empirical-to-conceptual approach, the taxonomy's creator elects objects groups and classifies them, and from these classifications, creates or revises the taxonomy. In the conceptual-to-empirical approach, the creator conceptualizes new characteristics and applies them to the objects already examined, then creates or revises the taxonomy.

Adding a new object to the taxonomy helps improve its generality. If the taxonomy

successfully classifies a new entity, the success attests to the taxonomy's generality. On the other hand, if the taxonomy fails to classify a new entity, this forces a revision, improving the taxonomy generality. In other words, the more iterations we apply to the taxonomy following the method I have elected, the more we strengthen its generality by definition.

## 4.2 Construction and validation

Firstly, let's state my definition of taxonomy. For the sake of clarity, I subscribe to the concept of taxonomy provided by Nickerson et al. (NICKERSON; VARSHNEY; MUNTERMANN, 2013). I define a taxonomy $T$ as a set of $n$ dimensions $D_i(i = 1, ..., n)$ each consisting of $k_i$ $(k_i \geq 2)$ mutually exclusive and collectively exhaustive characteristics $C_{ij}(j = 1, ..., k_i)$ such that each object under consideration has one and only $C_{ij}$ for each $D_i$. Let's put it into an example to make it easier to understand my definition, looking at Figure 2. In the taxonomy shown in Figure 2 we have seven dimensions ($n = 7$): "Access Input", "Access Output", "Cardinality", "Sensitivity", "Privacy of Location", "Privacy of State of Body & Mind" and "Privacy of Behavior & Social Life". Now let's take a look at the dimension "Cardinality", which I name as $C_1$ to make it fit the definition. We have two mutually exclusive characteristics in "Cardinality": "Single Input" and "Multiple Input". We have two characteristics, so $k_1 = 2$, and we can think of "Single Input" and "Multiple Input" as $C_{11}$ and $C_{12}$.

I have opted to construct a facet-based taxonomy suitable for new and evolving fields such as mine. A facet-based taxonomy refers to classification with multiple dimensions. Each facet or branch represents a distinct dimension and contains a value. It is important to notice that the facet scheme fits my definition of taxonomy in the following way: each facet corresponds to a dimension $D$, and the facet's levels correspond to the characteristics $C_{ij}(j = 1, ..., k_i)$ of that dimension. (USMAN et al., 2017)

I display the taxonomy I have built on Figure 2. The meta-characteristic I expect my taxonomy to represent is the interaction between an application and its input and output vectors, that is, its input and output. I have established the following ending conditions to signal that the taxonomy has been built and ensure the orthogonality of the taxonomy dimensions:

1. We have examined all objects or a representative sample of objects.
2. No object was merged with a similar object or split into multiple objects in the last iteration.
3. We have classified at least one object under every characteristic of every dimension.
4. We have not added new dimensions or characteristics in the last iteration.
5. No dimensions or characteristics were merged or split in the last iteration.
6. Every dimension is unique (i.e., there is no dimension duplication).
7. Each characteristic does not overlap semantically with other characteristics in its dimension (i.e., no characteristic duplication within a dimension).
8. Each combination of characteristics is unique and not repeated (i.e., there is no duplication).

For one of the conceptual-to-empirical iterations in the construction of my taxonomy, I used the types of privacy summarized by (Eckhoff; Wagner, 2018). I have built the taxonomy by analyzing tens of applications. I have categorized these applications through the lenses of the service they provide. The applications included in this process either dealt with potentially privacy-sensitive data or performed computations that demanded confidentiality.

I attest to the validity of my taxonomy through an orthogonality demonstration and a utility demonstration. I prove the orthogonality of my taxonomy's dimensions by the taxonomy's ending conditions that ensure the dimensions are mutually exclusive, in particular the sixth ending condition, which prohibits dimension duplication. Secondly, I showcase utility by classifying actual applications. We naturally classify application via the empirical iterations of the taxonomy construction. Furthermore, for a more rigorous validation approach we perform a case study we discuss in Chapter 5.

The applications analyzed during the construction of my taxonomy can be seen in Table 1. I present these applications as a starting point for my taxonomy. One can readily expand it thanks to its facet-based format (USMAN et al., 2017). I map each application to the service it provides, as can be seen in Table 1. To ensure the ending condition eight, I

Table 1: Analyzed applications and their descriptions

| Application | Service | Url | Description |
|---|---|---|---|
| Smart Gridsoft | Smart grid | www.smartgridsoft.in | Energy benchmarking by residence using smart meters. |
| AirBeam | Hyperlocal air pollution monitoring | www.habitatmap.org/airbeam | Air quality instrument to measures hyperlocal concentrations of harmful particles in the air. |
| EverImpact | Holistic air pollution monitoring | www.everimpact.com | Platform using data from satellites, sensors, traffic and buildings on emissions. |
| BigBelly | Smart waste management | bigbelly.com/products | Waste management system with automatic tracking of fill level of waste units. |
| Philly Building Benchmarking | Building energy benchmarking | www.phillybuildingbenchmarking.com | Energy benchmarking by building in the city of Philadelphia. |
| Enevo | Smart waste management | www.enevo.com/waste-solutions-services | Analytics platform using sensors to track fill level of waste units. |
| Reroute.it | Personal trajectory tracking | icos.urenio.org/applications/reroute-it | App for calculation of costs and environmental impacts of one's transportation choices. |
| LeakView | Real-time detection of pipe breakages | www.visenti.com | Real-time leak detection and non-revenue water management. |
| GoEzy | Ai driven commuting | www.metropia.com/goezy-app | AiI driven management for driver navigation, dynamic carpool pairing, transit, ride hailing, micro-transit, biking, and walking. |
| MyBuildingDoesntRecycle | Crowdsourced recycling monitoring | mybuildingdoesntrecycle.com | Crowd sourced tracking of recycling services by buildings in Chicago. |
| MyCity360 | Smart parking | mycity360.co.il | Parking solution with tracking of available spots using sensors. |
| OpenTreeMap | Tree inventory and tracking | www.opentreemap.org | Tree inventory and tracking via maps. |
| Water Storage | Public water storage management | www.bom.gov.au/water/dashboards | Open dashboard with information on availble water storages over periods of time. |
| WasteOS (Dumpster monitoring) | Ai driven dumpster visual monitoring | compology.com | Visual tracking of fill level of waste units with prescription of adjustments to container size, service frequency and service days |
| WasteOS (Trailer monitoring) | Ai driven visual trailer monitoring | compology.com | Visual trailer monitoring with AI models calculating space utilization over time. |
| Octopus Card App | Universal Smart Card | www.octopus.com.hk | Universal smart card with digital wallet integrated to transport systems. |
| MoodFit | Mental health coach | www.getmoodfit.com/ | Tracker for mental health and personalized daily goals. |
| Erouska | Intelligent infection analysis | erouska.cz/en | Intelligent healthcare integrating social networking and health data for infection analysis. |
| Ubibot | Ambient temperature monitoring | www.ubibot.com | Iot platform fed by sensors monitoring environmental conditions. |
| LiteMe | Smart grid | www.liteme.com.br | Energy monitoring and analysis (including segregation of the usage of individual appliances). |
| Google Fit | Health tracker | www.google.com/fit | Platform that records physical fitness activities and compares them to user's fitness goals. |

classify the application through the lenses of the service it provides. For instance, relatively similar waste management solutions such as "BigBelly" and "Enevo" fall under the service "Smart Waste Management" and receive the same classification.

The iterations that comprise the construction of the taxonomy can be seen in greater detail in Section 4.5, which details each decision in adding or removing a dimension, the iteration at which an object was examined, and why at a given iteration the process was stopped or continued.

## 4.3 Facets and Levels

In this section, I discuss the facets and levels of my taxonomy, represented in Figure 2. I represent facets as rounded rectangles, and the facet's levels by text field pointed to by arrows. It may come to notice that one can group multiple facets under an outer facet. For

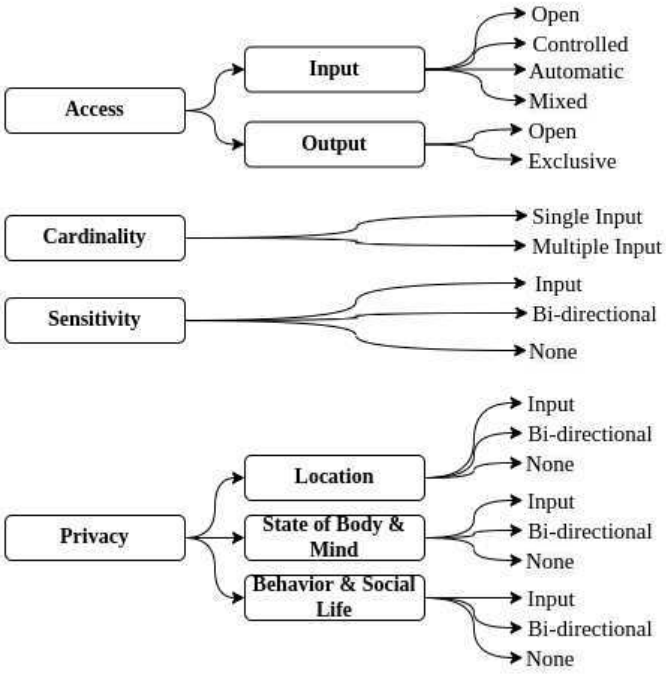


Figure 2: This work's taxonomy on privacy and confidentiality.

instance I grouped the facets **Input** and **Output** under the facet *Access*. Only the innermost facets represent a dimension $D$ as described in my definition of taxonomy. Outer facets such as *Access* only serve to group semantically related facets and do not represent a dimension, unlike the other facets shown in Figure 2.

My taxonomy's classification of an application designates the relationship between the application and its input and output vectors. The categorization applied to each service incorporating the facets and levels I discuss in this Chapter is available in Table 2. Although in Table 2 I have attached a link for each service, during the services' evaluation, I have gathered different sources of information going from documentation to demonstrations.

1) *Access:* This facet describes the mechanisms through which data vectors are made available to or by the application. Such a mechanism can be internal or external to the application. For instance, although there may not be an access control mechanism in the service, a group or organization may have restricted access to the service to its members. I divide this facet into two sub-facets: Input and Output.

The **Input** facet refers to how the data is made available to the application. The *Open* level denotes a case where there is no access control over the data provided as input to the

application. The *Controlled* level represents cases where one must go through some control access mechanism. The *Automatic* level refers to applications where automated means such as sensors provide input. The *Mixed* level refers to applications that employ a mix of *Open, Controlled*, and *Automatic* means.

The **Output** facet refers to how data is made available by the application. The *Open* level denotes the case where there is no access control mechanism, that is anyone can freely access the output. The *Exclusive* level represents the case where either one must qualify via an appointed access control mechanism to get access or the data is automatically delivered to another service or storage bypassing users.

2) *Cardinality:* This facet refers to the number of entities whose data must be consumed for the application to provide its service. A personal trajectory monitoring application only needs a single user's data to display their trajectory. The *Single Input* describes the case where the application only needs data from the entity itself to provide the service to the same entity. This case allows us to disclose the user's data via the service's functionalities to no one but them. The *Multiple Input* denotes when the application must rely upon the data of multiple entities to provide its service to a single entity. This case implies that when we furnish the service operations to someone, the service inevitably leaks users' data to a degree despite not necessarily outright disclosing it. This "privacy leakage" inexorably happens because information on people other than whoever receives the data escapes via the output. A commuting application would need input from various people to allow a user to use its service.

3) *Sensitivity:* This facet describes the application in terms of the flow of sensitive data. The *Input* level describes the case when sensitive data is received but not outputted. The *Bi-directional* level represents the case where both the input and the output are sensitive. The *None* level represents the case where no sensitive data is either received as input or produced as output. Initially, I proposed a level *Output* that would signify a service whose input is not sensitive but whose output is. I did so for completeness, that is, to have one level for each combination of input and output sensitivity. However, during the taxonomy construction, I did not encounter an example of this level or characteristic. Following the third ending condition, I removed it.

4) *Privacy:* This facet describes the privacy type within the flow of sensitive data. I

Table 2: Services' classification

| Services | Access | | Cardinality | Sensitivity | Privacy | | |
|---|---|---|---|---|---|---|---|
| | Input Provisioning | Output Provisioning | | | Location | State of body & Mind | Behavior & Social life |
| Smart grid | Automatic | Open | Multiple | Input | Input | None | Bi-directional |
| Hyperlocal air pollution monitoring | Open | Open | Single | Input | Input | None | None |
| Holistic air pollution monitoring | Automatic | Controlled | Multiple | None | None | None | None |
| Smart waste management | Automatic | Exclusive | Multiple | None | Input | None | None |
| Building energy benchmarking | Automatic | Open | Multiple | Input | None | None | Input |
| Personal trajectory tracking | Controlled | Exclusive | Single | Bi-directional | Input | Input | Bi-directional |
| Real-time detection of pipe breakages | Automatic | Exclusive | Multiple | Bi-directional | None | None | None |
| Ai driven commuting | Automatic | Exclusive | Multiple | Bi-directional | Bi-directional | None | Bi-directional |
| Crowdsourced recycling monitoring | Open | Open | Multiple | None | None | None | Input |
| Smart parking | Automatic | Open | Multiple | Input | Input | None | Bi-directional |
| Tree inventory and tracking | Controlled | Open | Multiple | None | None | None | None |
| Public water storage monitoring | Automatic | Open | Multiple | None | None | None | None |
| Ai driven dumpster visual monitoring | Automatic | Exclusive | Multiple | Input | Input | None | Input |
| Ai driven visual trailer monitoring | Automatic | Exclusive | Multiple | Bi-directional | Bi-directional | None | Bi-directional |
| Universal Smart Card | Automatic | Exclusive | Single | Bi-directional | Bi-directional | Input | Bi-directional |
| Mental health coach | Controlled | Exclusive | Single | Bi-directional | None | Bi-directional | Input |
| Intelligent infection analysis | Controlled | Exclusive | Multiple | Bi-directional | Input | Bi-directional | Input |
| Ambient temperature monitoring | Automatic | Open | Multiple | None | Input | None | Input |
| Health Tracker | Automatic | Exclusive | Single | Bi-directional | Bi-directional | Bi-directional | Bi-directional |

divide this facet into three sub-facets: Location, State of Behavior & Mind, and Behavior & Social Life.

The **Location** facet describes data that may compromise one's privacy by exposing details on location at a given time or over some time. The semantic of the levels *Input, Bi-directional and None* are the same as the levels in the sensitivity category.

The **State of Body & Mind** facet describes data that may compromise one's privacy by exposing details on physical health, genetics, and mental health. The semantic of the levels *Input, Bi-directional and None* are the same as in the sensitivity category.

The **Behavior & Social Life** facet describes data that may compromise one's privacy by exposing details on habits, opinions, and affiliations. An example of this would be one's purchase history or a post with political views on social media. The semantic of the levels *Input, Bi-directional and None* are the same as the levels in the sensitivity category.

## 4.4 Classification granularity

With enough information on an application's components, it becomes possible to apply the taxonomy directly to the application's components. Such as with a conventional threat analysis, a classification of the application's parts would provide a more fine-grained representation of how the application handles data and what particular segments of the application must be protected or revised.

I argue that it is worthwhile to pursue the most granular classifications available within the limit of feasibility. The analysis needs to be compatible with the attack model of the rele-

vant system. For instance, what are the system's borders? What are the possible adversaries? Only the users or also the administrators? To illustrate how a more granular categorization might be helpful, let us look at an example. On a high level, we can categorize a given application as one of exclusive output that can be accessed only by entities represented in it. However, that same application could have a database component containing sensitive data without proper access control hosted in a cloud provider, which flew under our radar on a higher level categorization. To provide an example using my taxonomy, in Chapter 6 I classify the services from the use case system presented in Chapter 5.

## 4.5 Taxonomy's iterations

This section describes my taxonomy's iterations to reach the stage discussed in this work. The steps follow the same nomenclature as the original method showcased in Figure 1 for clarity's sake.

### 4.5.1 Step 1: Meta-Characteristic

This taxonomy's meta-characteristic refers to the interaction between application and input vectors and output vectors (e.g. data and users). The meta-characteristic serves as a statement on what the taxonomy fundamentally aims to represent.

### 4.5.2 Step 2: Ending Conditions

In this subsection, I describe the subjective and objective ending conditions as explained in Section 4.1. The following are the objective conditions I establish this taxonomy must fulfill to ensure dimension orthogonality: All objects or a representative sample of objects have been examined. No object was merged with a similar object or split into multiple objects in the last iteration. At least one object is classified under every characteristic of every dimension. No new dimensions or characteristics were added in the last iteration. No dimensions or characteristics were merged or split in the last iteration. Every dimension is unique and not repeated (i.e., there is no dimension duplication). Every characteristic is unique within its dimension (i.e., there is no characteristic duplication within a dimension). Each combination of characteristics is unique and is not repeated (i.e., there is no duplication).

The following are the subjective conditions this taxonomy must fulfill: Conciseness, be meaningful without being unwieldy or overwhelming. Robustness, provides for differentiation among objects sufficient to be of interest. Comprehensiveness, can classify all objects or a random sample of objects within the domain of interest. Extensibility, can easily comport a new dimension or characteristic of an existing dimension. Explanatory, the taxonomy's dimensions and characteristics explains well an object.

### 4.5.3 Iteration 1

We begin with **Step 3** which refers to choosing an approach to apply on the taxonomy. I have arbitrarily chosen to begin with the Conceptual-to-Empirical approach. Now let's proceed with **Step 4c** which consist in conceptualizing new characteristics. There could be a different number of people with access to the output of the application, access could be restricted to the individual represented in the data or be provided to the general public. To represent this I have created the **Output access** dimension: *exclusive* (only who's in the data can access), *open* (people other than who's represented in the data can access the output).

I conceive that the relation of those represented in the consumed data with the application could vary. Only the application owner could be in the data, or people external to the application could also be represented in the data. To represent this I have created the **Input composition** dimension: *Self* (only app owner in input), *general* (people other than the owner are in the input). I conceive that the number of people represented in the data consumed by the application could vary. A single person could be represented or a multitude of people could also be represented in the data. To represent this I have created the **Input cardinality** dimension: *Single* (one person in input), *Multiple* (more than one person in input). I conceive that the number of people represented in the data outputted by the application could vary. A single person could be represented or a multitude of people could also be represented in the data. To represent this I have created the **Output cardinality** dimension: *Single* (one person in output), *Multiple* (more than one person in output).

Let us now follow **Step 5c** which consists in examining objects. First, let's list the examined objects. Smart Grid, object taken from (ZHANG et al., 2017). Trajectory Monitoring, object taken from (ZHANG et al., 2017). Intelligent infection analysis, object taken from (ZHANG et al., 2017). This object becomes even more relevant in the context of Covid-19

detection. Room Temperature Monitoring, object taken from (ZHANG et al., 2017). Room Temperature Monitoring focus on improving air-conditioning, and reducing energy bills. For now I must keep only **Output cardinality** out of **Input composition**, **Input cardinality** and **Output cardinality** as they are redundant. They are redundant because all objects have the same characteristics for the aforementioned dimensions. To fulfill **Step 6c** I must create the taxonomy. Below we have our first iteration of the taxonomy:

$$
\begin{aligned}
&T1 = \{ \\
&OutputAccess(Exclusive, Open), \\
&OutputCardinality(Single, Multiple) \\
&\}
\end{aligned}
$$

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have created the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.4 Iteration 2

To start the second iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Hyperlocal air pollution monitoring (AIRBEAM, 2022). Smart waste management (BIGBELLY, 2022). Building energy benchmarking on the city of Philadelphia (BUILDING ENERGY BENCHMARKING, 2022). Holistic air pollution monitoring (EVERIMPACT, 2022). Smart waste management (ENEVO, 2022). Personal trajectory tracking (REROUTE IT, 2022). I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. I perceived that the objects I examined display the folllowing characteristics: An application generates output with either low or high level of privacy sensibility. Application receives input with either low or high level of privacy sensibility. Following **Step 6e** I must revise the taxonomy. Below we have the current state of the taxonomy:

$$
\begin{aligned}
&T2 = \{ \\
&OutputAccess(Exclusive, Open), \\
&OutputCardinality(Single, Multiple), \\
\end{aligned}
$$

$InputSensibility(Low, High)$,
$OutputSensibility(Low, High)$
}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have revised the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.5 Iteration 3

To start this iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Real-time detection of pipe breakages (LEAKVIEW, 2022). AI driven commuting (GOEZY, 2022). Crowdsourced recycling monitoring (MY BUILDING DOESNT RECYCLE, 2022). Smart parking (MYCITY360, 2022). Tree inventory and tracking (OPENTREEMAP, 2022). I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. I perceived that the objects I examined display the folllowing characteristics: Application either allows input from anyone whatsoever, input is granted via control access, or the input is automatically generated and delivered. Following **Step 6e** I must revise the taxonomy. Below we have the current state of the taxonomy:

$T3 = \{$
$InputProvisioning(Open, Controlled, Automatic)$,
$OutputAccess(Exclusive, Open)$,
$OutputCardinality(Single, Multiple)$,
$InputSensibility(Low, High)$,
$OutputSensibility(Low, High)$
}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have revised the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.6 Iteration 4

To start this iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Public water storage management in Australia (AU WATER STORAGE, 2022). AI driven dumpster visual monitoring (WASTEOS DUMPSTER MONITORING, 2022). AI driven visual trailer monitoring (WASTEOS TRUCKING MONITORING, 2022). I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. I perceived no new characteristics in the objects I examined. Following **Step 6e** I must analyze whether I must revise the taxonomy, which in this iteration is unnecessary as I have not seen new characteristics. Below we have the current state of the taxonomy:

$$
\begin{aligned}
&T4 = \{ \\
&InputProvisioning(Open, Controlled, Automatic), \\
&OutputAccess(Exclusive, Open), \\
&OutputCardinality(Single, Multiple), \\
&InputSensibility(Low, High), \\
&OutputSensibility(Low, High) \\
&\}
\end{aligned}
$$

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I do not deem subjective ending conditions such as Comprehensiveness met, and for this reason, I shall continue the iterative process.

### 4.5.7 Iteration 5

To start this iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Trainer service describe in Chapter 5. Classifier service describe in Chapter 5. Alert service describe in Chapter 5. I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. I perceived no new characteristics in the objects I examined. Following **Step 6e** I

must analyze whether I must revise the taxonomy, which in this iteration is unnecessary as I have not seen new characteristics. Below we have the current state of the taxonomy:

$$
\begin{aligned}
&T5 = \{ \\
&InputProvisioning(Open, Controlled, Automatic), \\
&OutputAccess(Exclusive, Open), \\
&OutputCardinality(Single, Multiple), \\
&InputSensibility(Low, High), \\
&OutputSensibility(Low, High) \\
&\}
\end{aligned}
$$

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I do not deem subjective ending conditions such as Comprehensiveness met, and for this reason, I shall continue the iterative process.

### 4.5.8 Iteration 6

We begin with **Step 3** which refers to choosing an approach to apply on the taxonomy. I have chosen the Conceptual-to-Empirical approach having in mind the work of Eckhoff *et al.*. Now let's proceed with **Step 4c** which consist in conceptualizing new characteristics. Based on the work of Eckhoff *et al.* (Eckhoff; Wagner, 2018) I have conceptualized the following new characteristics: Privacy of Location. Privacy of State of Body & Mind. Privacy of Behavior & Action. Privacy of Social Life. Privacy of Media. Following **Step 5c** I must examine the objects already seen during this process employing the lens of the concepts I have conceptualized in this iteration. After examining seen objects I have noticed that all of them have been given a value *No* for the category **Privacy of State of Mind** making it redundant. At this point the category **Privacy of State of Mind** will not be acknowledged and instead I'll add **Privacy of State of Body**. Following **Step 6e** I must revise the taxonomy. In this iteration I have added the following categories: Privacy of Location, Privacy of State of Body, Privacy of State of Body, Privacy of Behavior & Action, Privacy of Social Life, and Privacy of Media. Below we have the current state of the taxonomy:

$$T6 = \{$$

$InputProvisioning(Open, Controlled, Automatic)$,
$OutputAccess(Exclusive, Open)$,
$OutputCardinality(Single, Multiple)$,
$InputSensibility(Low, High)$,
$OutputSensibility(Low, High)$,
$PrivacyofLocation(Yes, No)$,
$PrivacyofStateofBody(Yes, No)$,
$PrivacyofBehavior\&Action(Yes, No)$,
$PrivacyofSocialLife(Yes, No)$,
$PrivacyofMedia(Yes, No)$
}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have revised the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.9 Iteration 7

To start this iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Universal Smart Card (OCTOPUS CARD/APP, 2022). Mental health coach (MOODFIT, 2022). I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. The new object (MOODFIT, 2022) has a clear mental component in terms of privacy. **Privacy of State of Mind** is no longer irrelevant. At this point the category **Privacy of State of Body** will be replaced with **Privacy of State of Body & Mind**. Following **Step 6e** I must analyze whether I must revise the taxonomy, I have replaced **Privacy of State of Body** with **Privacy of State of Body & Mind**. Below we have the current state of the taxonomy:

$T7 = \{$
$InputProvisioning(Open, Controlled, Automatic)$,
$OutputAccess(Exclusive, Open)$,
$OutputCardinality(Single, Multiple)$,
$InputSensibility(Low, High)$,

$OutputSensibility(Low, High)$,

$PrivacyofLocation(Yes, No)$,

$PrivacyofStateofBody\&Mind(Yes, No)$,

$PrivacyofBehavior\&Action(Yes, No)$,

$PrivacyofSocialLife(Yes, No)$,

$PrivacyofMedia(Yes, No)$

}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have revised the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.10 Iteration 8

We begin with **Step 3** which refers to choosing an approach to apply on the taxonomy. I have arbitrarily chosen the Conceptual-to-Empirical approach. Now let's proceed with **Step 4c** which consist in conceptualizing new characteristics. I have conceptualized the following new characteristics: Replace **Output Cardinality** with **Input Cardinality** due to the possibility of multiple slices of data being presented for different perspectives (admin, user, ...) making it ambiguous to determine cardinality for the output. Replace **Privacy of Behavior & Action** and **Privacy of Social Life** with **Privacy of Behavior & Social Life**. The two seem too close to each other to be taken as different characteristics. Remove **Privacy of Media** as it seems redundant, whether video or textual information whatever data is deemed privacy-sensitive must be protected. Replace **Input Sensibility (Yes, No)** and **Output Sensibility (Yes, No)** with **Sensibility (Input, Output, Bi-directional, None)**. Following **Step 5c** I must examine the objects already seen during this process employing the lens of the concepts I have conceptualized in this iteration. The option *Output* in **Sensibility**, **Privacy of Behavior  Social Life**, **Privacy of State of Body & Mind** and **Privacy of Location** had no matches and for this reason must be removed. Following **Step 6e** I must revise the taxonomy. The option *Output* was removed from **Sensibility**, **Privacy of Behavior  Social Life**, **Privacy of State of Body & Mind** and **Privacy of Location**. Below we have the current state of the taxonomy:

$T8 = \{$

$InputProvisioning(Open, Controlled, Automatic)$,
$OutputAccess(Exclusive, Open)$,
$InputCardinality(Single, Multiple)$,
$Sensibility(Input, Bi\text{-}Directional, None)$,
$Privacy\ of\ Location(Input, Output, Bi\text{-}directional, None)$,
$Privacy\ of\ State\ of\ Body\&Mind(Input, Output, Bi\text{-}directional, None)$,
$Privacy\ of\ Behavior\&SocialLife(Input, Output, Bi\text{-}directional, None)$
}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have revised the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.11 Iteration 9

To start this iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Smart grid (LITEME, 2022). Energy consumption monitoring (LITECAMPUS, 2022). I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. The LiteCampus application receives input from multiple sources: "Controlled" from authenticated users and automatically from sensors. The LiteCampus application shows a new characteristic for the dimension **Input Provisioning**. LiteCampus application encompasses two distinct services: Open dashboard with energy consumption data for buildings on campus. Energy consumption dashboard with a control access mechanism. Logged in users can look at the energy consumption of different devices (very intrusive). Following **Step 6e** I must revise the taxonomy. I have replaced sensibility with sensitivity. I have add the characteristic "Mixed" to the dimension **Input Provisioning**. Below we have the current state of the taxonomy:

$T9 = \{$
$InputProvisioning(Open, Controlled, Automatic, Mixed)$,
$OutputProvisioning(Exclusive, Open)$,
$Cardinality(Single, Multiple)$,
$Sensitivity(Input, Bi\text{-}directional, None)$,

$Privacy\ of\ Location(Input, Output, Bi\text{-}directional, None)$,
$Privacy\ of\ State\ of\ Body\&Mind(Input, Output, Bi\text{-}directional, None)$,
$Privacy\ of\ Behavior\&SocialLife(Input, Output, Bi\text{-}directional, None)$
}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. I have revised the taxonomy at this iteration, and for this reason, the iterative process must continue.

### 4.5.12 Iteration 10

To start this iteration I must choose an approach. I choose the approach Empirical-to-Conceptual which constitutes **Step 3e**. Now, I must identify new objects in accordance with **Step 4e**. Let us list the objects I have gathered: Health tracker (GOOGLEFIT, 2022). I must now follow **Step 5e** which consists in identifying common characteristics in the objects I have examined. I perceived no new characteristics in the objects I examined. Following **Step 6e** I must analyze whether I need to revise the taxonomy, and I deem no revision needed. Below we have the current state of the taxonomy:

$T10 = \{$
$InputProvisioning(Open, Controlled, Automatic, Mixed)$,
$OutputProvisioning(Exclusive, Open)$,
$Cardinality(Single, Multiple)$,
$Sensitivity(Input, Bi\text{-}directional, None)$,
$Privacy\ of\ Location(Input, Output, Bi\text{-}directional, None)$,
$Privacy\ of\ State\ of\ Body\&Mind(Input, Output, Bi\text{-}directional, None)$,
$Privacy\ of\ Behavior\&SocialLife(Input, Output, Bi\text{-}directional, None)$
}

Finally, in **Step 7** I analyze whether the taxonomy meets the ending conditions. The ending conditions described in 4.5.2 have been met. The objective conditions have been maintained, and I deem the subjective conditions as explained in 4.5.2 fulfilled. For this reason I stop the construction of the taxonomy at this iteration.

# 5 Case study

To further validate my taxonomy, I have employed it in a case study. I have centered my case study around an intelligent infection analysis system responsible for temperature monitoring and face recognition. The use case study covers all the services within the use case system. This system is part of a smart campus initiative in the Federal University of Campina Grande. In such a context, this system manages data of multiple people, which inevitably raises concerns about confidentiality and privacy. As a real-life application that I or others will eventually deploy, the system became a meaningful candidate for my case study.

## 5.1 Overview

The use case system performs three distinct tasks, that we can see in Figure 3. All tasks are preceded by the same configuration process. The service needs to communicate with a secret manager who provides the service configuration, such as its encryption key.

The system's first task is to enroll users. The system performs this task through steps 1 and 2 shown in Figure 3. The consenting users provide the system with images showing their faces. The system feeds these images to a machine learning model with facial recognition capabilities that encodes each image. The model can use such encoding to discern whether a given photo shows the user. The system maps each encoding to the name of the user it represents.

The second task consists in identifying enrolled users in the sensor feed the system receives. The system performs this task through steps 3, 4, and 5 shown in Figure 3. The system is given pictures with temperature readings (e.g., from a camera that includes thermal readings). Once a sensor reading is received, it is matched against the encodings for the enrolled users through a neural network. If the system recognizes one of the enrolled users in the picture, it tags it with the user name; otherwise, it defaults to an unknown person. The system then stores the tagged images.

The system's third and final task is to allow users to check for anomalous readings on them. In this context, anomalous readings refer to body temperatures too far below or above the median human body temperature. This task is performed by steps 6, 7, and 8 as illustrated in Figure 3. An enrolled user consults an API provided by the system. The API will inform

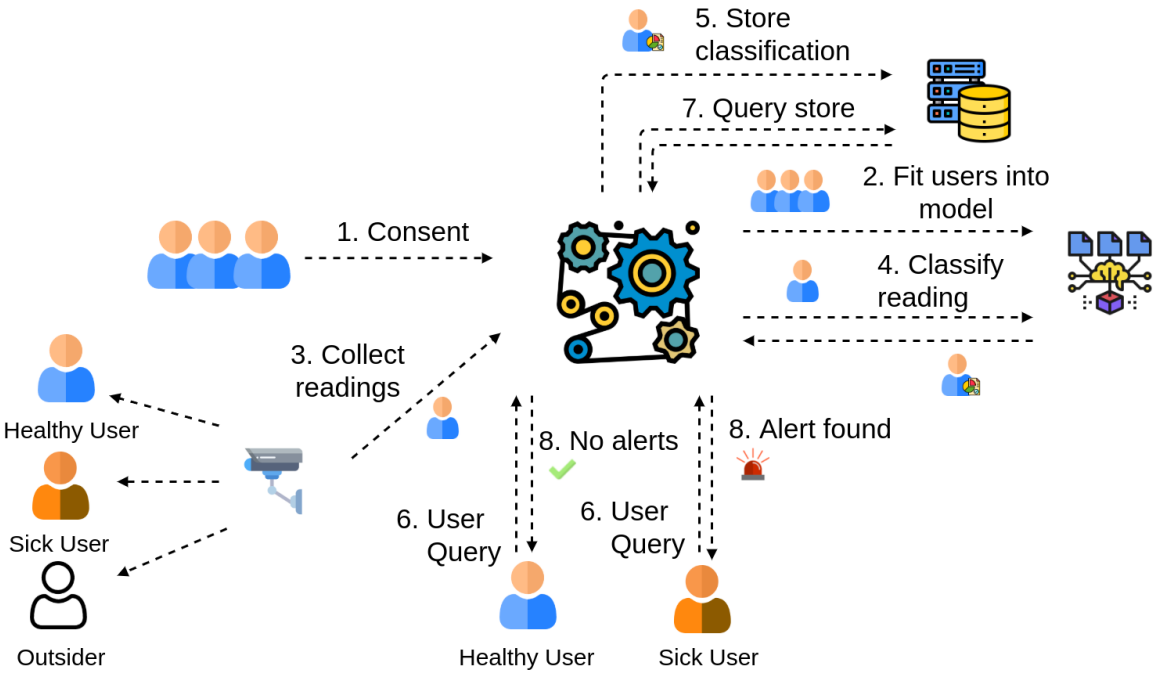


Figure 3: Case study system.

the user of anomalous readings matched with the said user in a given period. With this information in hand, the user can now be aware of the possibility of a health condition. A user can employ this same service to query whether other people in the vicinity encompassed by the system's sensors had anomalous readings, so the querying user can be aware of health risks by going to that vicinity.

Figure 4 represents the use case system's overall behavior in terms of a sequence diagram. A user submits to the Trainer service a request to be enrolled in the use case system, represented in Figure 4 by the call `enrollUser` that includes as arguments a name (`name: String`) and face picture (`facePicture: bytes`) from the user. The Trainer service processes this call converts the user's face picture into a format an artificial neural network can understand represented by an encoding (`encoding: Tensor`). The type `Tensor` refers to an algebraic object that describes a multi-linear relationship between sets of algebraic objects related to a vector space. The Trainer service delegates the persistence of the user-related data received and produced to a Storage service via the call `persistUser` that includes as arguments the user's name (`name: String`) and encoding (`encoding: Tensor`). This sequence encapsulates the responsibilities of the Trainer

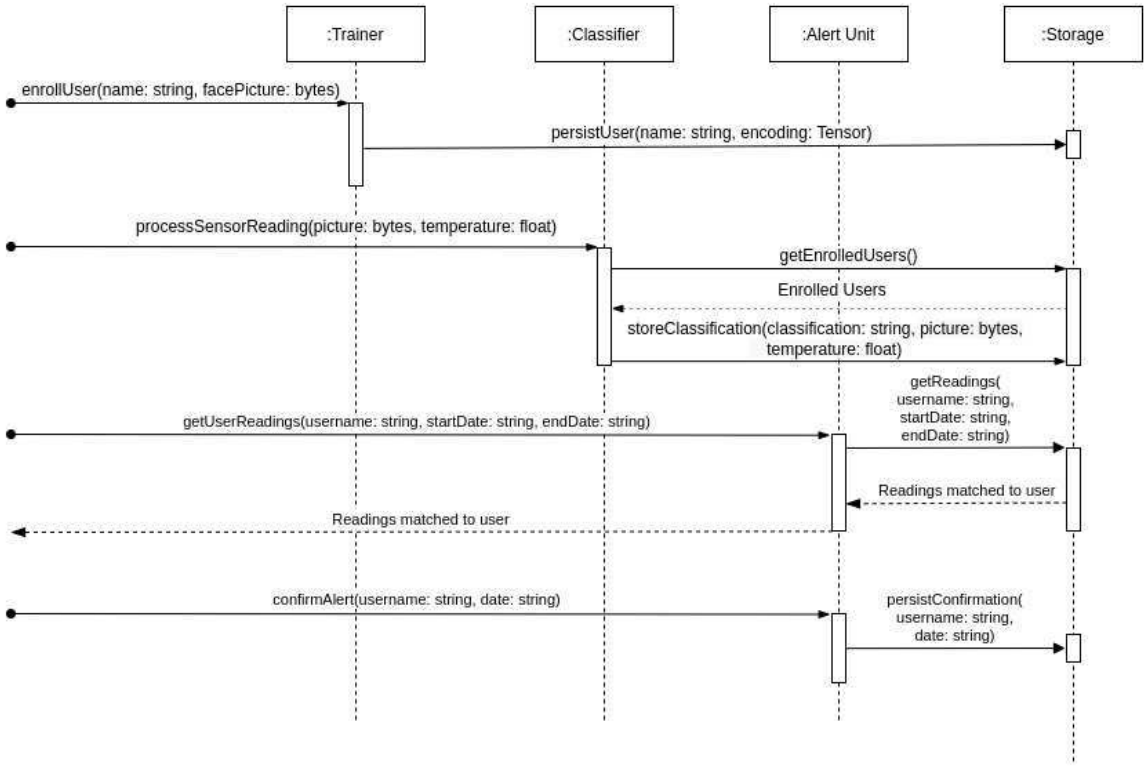


Figure 4: Use case system's sequence.

service.

Visual sensors with temperature monitoring capabilities send their feed to the Classifier service via the call `processSensor` which includes as arguments the picture (`picture: bytes`) and the temperature associated with the person captured in the shot (`temperature: float`). The Classifier service retrieves from the Storage service the users already enrolled via the call `getEnrolledUsers`. The Classifier service employs the information from the enrolled users to match the person shown in the sensor feed to one of the enrolled users. The result of this matching is sent into storage by the Classifier service via the call `storeClassification` which adds the classification (`classification: string`) to the data the Classifier has received from the sensor.

We can see the sequences related to the Alert service in Figure 4. The call `getUserReadings` represents a user demand for sensor reading matched to that user (`username: string`) in a given period (`startDate: string, endDate: string`). The Alert Service processes this call and consults the relevant storage service to retrieve sensor readings matched to the user.

The Alert service then forwards the matched readings to the user. The call `confirmAlert` represents a user confirming to the system that she was sick on a given date. The Alert service consumes this call and persists the confirmation with the help of a storage service.

## 5.2 Trainer Service

The Trainer service is responsible for enrolling a user in the system by consuming an image of the user's facial features. We can see the components of the Trainer service in Figure 5. The Trainer service consumes encrypted messages from a message bus, dedicated to "Pictures of Enrolled Users". The Trainer service decrypts the messages received from the message bus using an encryption key that the services received as part of their configuration. Each message constitutes a user to be subscribed and contains the user's facial image plus their name as a label.

The service contains a machine learning model tasked with facial recognition, we have employed a deep neural network trained on the open dataset "Labeled Faces in the Wild" (HUANG et al., 2007). No further training is done on the neural network. The model encodes the images extracted from each message, represented in Figure 5 as "Package". The neural network consumes the image and produces an encoding. The encoding is represented in Figure 5 as a purple circle leaving the "Package". Such encoding represents the user's facial features, and the neural network can use it to say whether the user's face is in a given image. The model groups the encodings it produces in a collection and creates a map to link the labels to their encodings.

Once the service has consumed the available messages, it uses an encryption key retrieved from its configuration to encrypt the encodings and the mapping between labels and encodings. The service exports the encrypted files to a shared volume, represented in Figure 5 as "Model Objects".

## 5.3 Classifier Service

The Classifier service is responsible for processing sensor images and matching them with the facial features of the enrolled users. We can see a representation of the Classifier service with its components in Figure 6. The Classifier service employs the same neural network

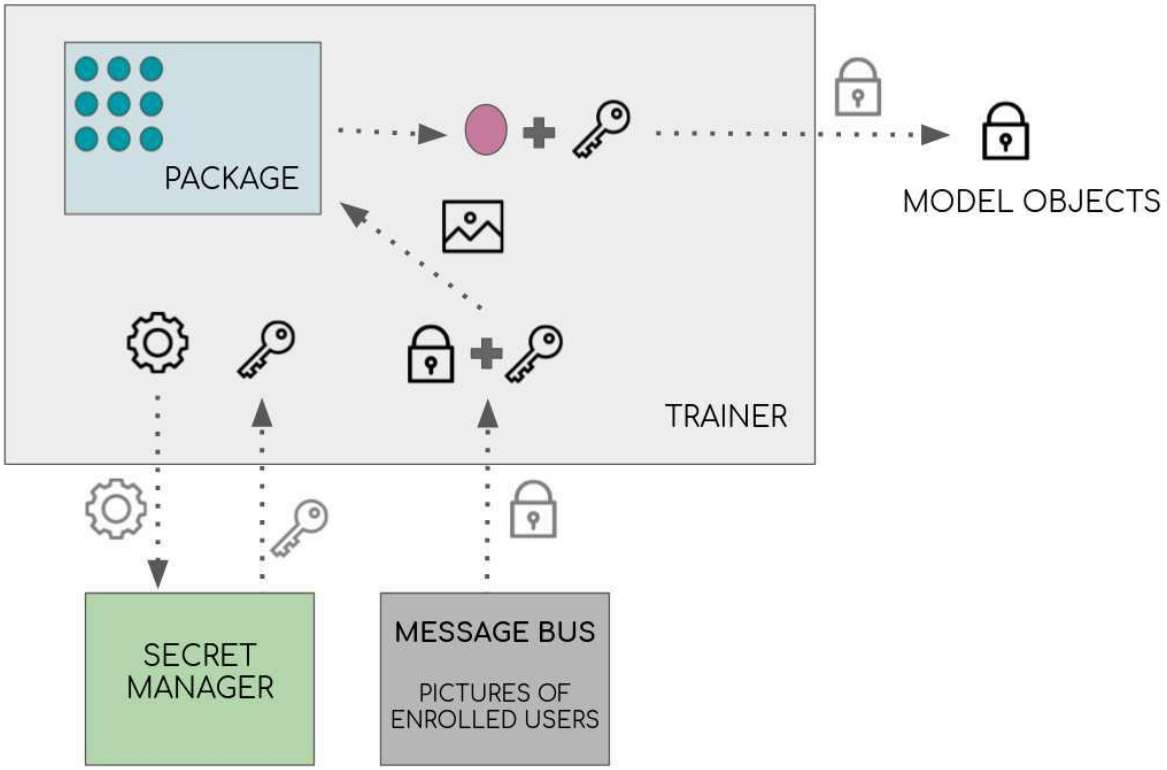


Figure 5: Trainer Component.

as the Trainer Component also represented as “Package”. The service receives as input encrypted encodings and the mapping between labels and encodings, designated as “Model Objects”. The encodings are represented in Figure 6 as a purple circle. The Classifier consumes encrypted messages from a message bus set apart with readings of unknown people from an area with temperature monitoring, represented in Figure 6 as “Pictures of Unknown People”. Each message contains an image of a passerby attached to a temperature reading.

The Classifier retrieves a message, decrypts it, extracts the temperature reading and the passerby’s picture from the message, and employs its neural network to classify the image. The image is labeled as one of the subscribed users or marked as “Unknown”. Suppose the temperature reading is unusually high and by that I mean above the median human body temperature, the Classifier tags that message with an “alert” status. Lastly, the message is encrypted and stored in a different message bus, represented in Figure 6 as “Classified Pictures”.

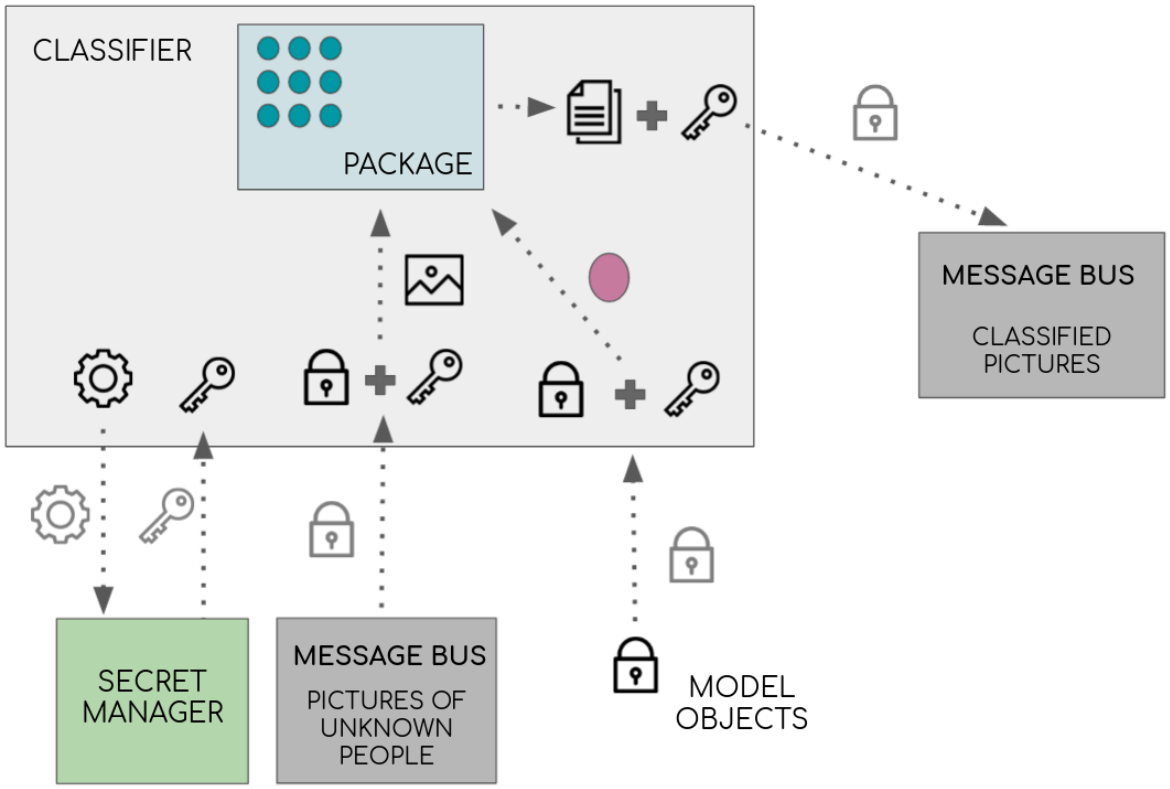


Figure 6: Classifier Component.

## 5.4 Alert Service

The Alert service is responsible for providing an API to enrolled users. The API allows users to query whether the system has linked an anomalous reading to them. We can see a representation of the Alert service and its components in Figure 7. The Alert service consumes encrypted messages from the message bus, which stores temperature readings that have been matched, successfully or not, to the facial features of enrolled users. I represent the message bus as "Classified Pictures". The service receives an encrypted file with the expected user-base as input. Requests to the Alert service are only allowed to users present in this user base. The aforementioned file is represented in Figure 7 as "User Base". The service also receive encodings that serve as a rendition of the users, these encodings are represented in Figure 7 as a purple circle originating from "Model Objects". The Alert Component exposes an API where enrolled users can request whether there have been abnormal readings on them or to confirm that they have been sick on a given date. The Alert Component only provides the API to enrolled users. Furthermore, when a sick individual is recognized an alert

is sent identifying the sick individual to the enrolled users so they can protect themselves.

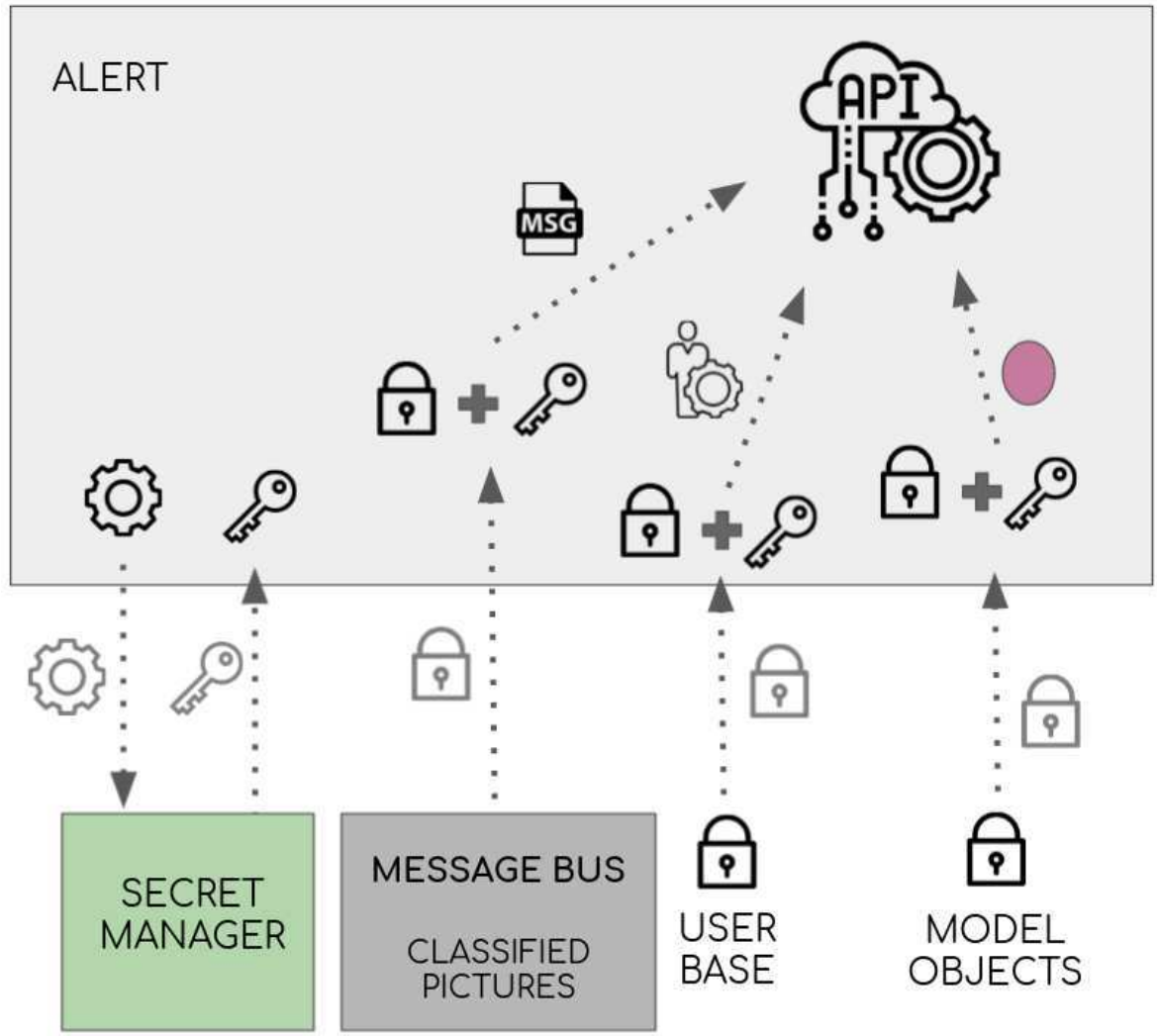


Figure 7: Alert Component.

# 6 Taxonomic Classification

This chapter discusses the results gathered from applying the taxonomy to all three services within my use case system. This discussion extends to all the features the services provide a user. The classification is shown in Table 3.

1) *Access:* In terms of **Input** only the Trainer service is under the category "Controlled" as it accepts input from users to be enrolled. The Classifier service automatically receives input data via sensors. The Alert service, in turn, is fed the output data generated by the Classifier. In terms of **Output**, all three services fall under the "Controlled" category. The Trainer and Classifier encrypt their output using encryption keys received as part of their configuration. The Alert service gate-keeps access to its results through an authentication method (e.g., client certificates).

2) *Cardinality:* The Trainer and Classifier services only need to employ data from a given user to link sensors reading to that user. Thanks to the system's neural network that has already been trained on the open dataset "Labeled Faces in the Wild" (HUANG et al., 2007), these services only require a user's data to provide their functionalities to that user. The people in the dataset "Labeled Faces in the Wild" (HUANG et al., 2007) are disregarded in terms of cardinality because the dataset is open, and therefore we do not consider it privacy-sensitive. However, a user can employ the Alert service to query whether the system has spotted people with anomalous readings in the area it monitors. For this reason, the Alert service uses data from multiple people to provide its functionalities to a single user. 3) *Sensitivity:* The Trainer service receives sensitive data in the form of pictures with the facial features of users. However, its output is much less sensitive because it consists of a not so easily understood mathematical representation of the users' facial features in the form of tensors. However incomprehensible, a party with enough knowledge about the neural network I employed can use this tensor to identify the user's facial features represented in the tensor. The Classifier only adds information to its privacy-sensitive input by tagging it with its classification value, resulting in an equally sensitive output. The Alert input matches the Classifier input. The Alert's results expose whether a given user shows symptoms and may harbor disease. Therefore its output is also privacy sensitive.

4) *Privacy of Location:* The Classifier and Alert services link a possibly named individ-

Table 3: Use case system's classification

| APPLICATIONS/CHARACTERISTICS | | | TRAINER | CLASSIFIER | ALERT |
|---|---|---|---|---|---|
| ACCESS | INPUT | OPEN | | | |
| | | CONTROLLED | X | | X |
| | | AUTOMATIC | | X | |
| | OUTPUT | OPEN | | | |
| | | CONTROLLED | X | X | X |
| | | AUTOMATIC | | | |
| CARDINALITY | | SINGLE INPUT | X | X | |
| | | MULTIPLE INPUT | | | X |
| SENSITIVITY | | INPUT | | | |
| | | BI-DIRECTONAL | X | X | X |
| | | NONE | | | |
| PRIVACY | LOCATION | INPUT | X | | |
| | | BI-DIRECTONAL | | X | X |
| | | NONE | | | |
| | STATE OF BODY & MIND | INPUT | | | |
| | | BI-DIRECTONAL | X | X | X |
| | | NONE | | | |
| | BEHAVIOR & SOCIAL LIFE | INPUT | | | X |
| | | BI-DIRECTONAL | | X | |
| | | NONE | X | | |

ual, if it is an enrolled user, to the vicinity where the sensors operate. In contrast, the Trainer consumes a picture for each user not readily tied to any location or period.

5) *Privacy of State of Body & Mind:* All three services handle data that pertain to either a user's body or mind. The Trainer operates on pictures with the user's facial features. The Trainer and Classifier produce and consume, respectively, a representation of the user's facial features. The Classifier and Alert handle temperature readings of passersby and possibly of enrolled users that a malicious party could use to infer whether they have some affliction.

6) *Privacy of Behavior & Social Life:* The sensor readings employed as input and output by the Classifier tie a user's comings and goings to the vicinity where the sensors operate. With enough of these, one can have a glimpse of a user's schedule. The Alert service consumes these same readings as input.

## 6.1 Discerned vulnerabilities

As represented in the taxonomic classification, the three services have sensitivity in their data flow. The Classifier and Alert have data with sensitivity in their entire flow. The Trainer receives a sensitive input that eventually becomes less sensitive due to its computations. Unless I shield these computations, somehow, the sensitive data employed in these operations are exposed.

As shown in Table 3, the Alert service diverges from the other services in terms of the dimension "Cardinality" as it disposes the characteristic "Multiple Input". This characteristic signifies that the Alert service includes data from multiple people to provide its functionalities. Beyond that, this classification denotes that the Alert service's functionalities inherently expose the privacy of people represented in the employed data as a user has access to information that describes other people. The Alert service identifies the sick individual to the enrolled users so they can protect themselves. I had to mitigate this privacy threat or handle it appropriately. Otherwise, I would risk allowing users to misuse the Alert service. Ill-intentioned individuals with access to the service could expose the privacy of others who transit in the area monitored by the use case system by leaking the data to which they had access.

As their taxonomic classification shows, the Classifier and Alert services include "privacy of location" in their data flow. The inputs that pertain to a user link the user in a time frame to the area where the system's sensors operate. However, as the classification shows, the cardinality of the Classifier and Trainer services is under the category "Single Input". That means that these services only need to employ data from the user to provide them their functionalities. Therefore, a user's privacy need not be exposed through the functioning of the services as only the user receives the data related to them.

For all three services, the configuration stage represents a reason for concern. A malicious party can either compromise the secret manager or pretend to be the secret manager to replace the configuration used by the service. Unless I tie the secret manager to the guarantees of a verified application, it represents a risk. In other words, I see the need for a trusted secret management service.

# 7 Use case improvements

This chapter showcases improvements I have discerned for the services in the use case system. More importantly, this chapter delineates an example of the intelligence provided by the taxonomy being employed on a use case. Therefore, this chapter helps augment this work's use case analysis, which provides a more rigorous utility demonstration for the taxonomy.

Before discussing the nature of this work's solution, I should clarify what parts of the use case system the solution encompasses and how so. I have factored all of the three services that make up the use case system in this work's solution. For each service all of their features are taken into account. The broader discussion applies to all of the services. I give special attention to the Alert service as it raises a more complex privacy problem with its features that consume data from several people to provide said feature to a single person.

I look at the services through the lens of their input and output vectors. This perspective allows the acknowledgment of the services as a whole in terms of privacy. This perspective also allows a user to apply the taxonomy to any service that includes input and output vectors. Therefore, it helps ensure that we can employ the taxonomy as a generic categorization tool instead of being tied to the use case system.

How does this discussion extend to confidentiality? I have used trusted execution environments to tackle the matter of confidentiality. Once I have discerned that a given service deals with sensitive data, I acknowledge whether the service warrants values such as integrity and confidentiality. From here, I can encompass the service with the appropriate trusted execution environment to meet its needs. Generality is maintained because we can pick the proper trusted environment for a particular situation. I showcase this by using two distinct trusted execution environments with disparate requirements and benefits.

## 7.1 Broader solution

To avoid tampering during the configuration stage, I make all the components follow the following configuration process. I employ an identity manager to gatekeep access to the secret manager. This identity manager demands a specific identity from the service interested in retrieving some configuration from the secret manager. Some form of attestation must provide the identity. Given the role of the identity manager, it must undergo some form

of attestation. The services and their environments also warrant attestation because of the computations on privacy-sensitive data they perform. I have also edited the services to ensure they encrypt data that they must persist. For instance, the services encrypt all the messages stored in a message bus with encryption keys they received via their initial configuration.

As far as attestation is concerned, one alternative would consist of employing an implementation of SPIFFE, such as SPIRE (see Section 3.3.2). Together, the SPIRE server and agent play the role of identity manager. The attestation process for the SPIRE agent (node attestation), as we perform it on the node the agent runs on, is presented in steps 1, 2, 3, and 4 in Figure 8. The agent and server can utilize various attestor plugins managed by their "Node Attestor" component in this process. In the figure, a node attestor that relies on evidences from the cloud provider (such as cloud tenant and VM image) are used. The SPIRE server functions as a root of trust in the deployment (ZIMMER; KRAU, 2016) (FELDMAN et al., 2020). For this reason, it should run on a trusted platorm (e.g., inside TEEs or on user's premises).

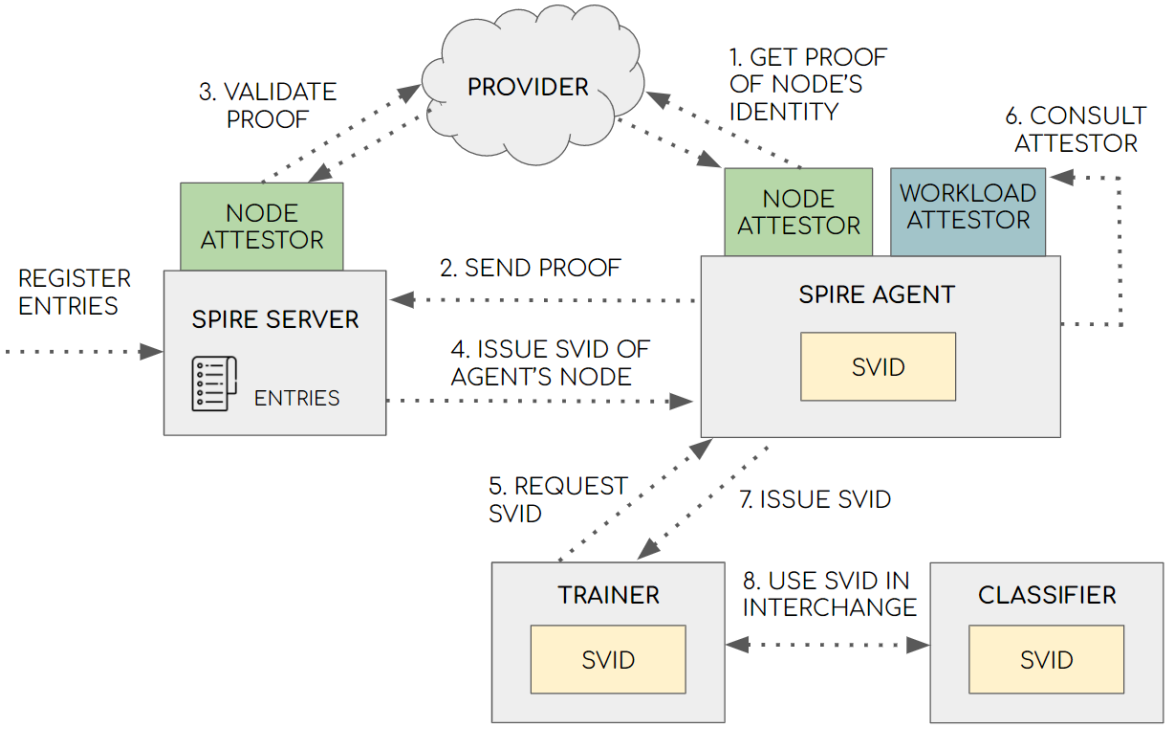


Figure 8: Overview of Trainer service interacting with SPIRE deployment.

We can follow Figure 8 to get an overview of how a workload gets an identity. Let us assume the SPIRE Server, SPIRE Agent, and the workload Trainer have been raised but

have not communicated with each other. In this scenario, the workload Trainer is the one that needs to receive its identity/SVID to maintain communication with different workloads, such as the workload Classifier also shown in Figure 8.

To request an identity, we must create an appropriate entry in the SPIRE server. An entry maps a SPIFFE ID to a set of verifiable properties of a node or workload known as selectors. By mapping a set of selectors to an identity, we ensure that only an entity that matches all the expected properties receives the appropriate identity in the form of an SVID . This process starts by registering an entry for the Agent in the SPIRE Server shown as "Register entries" in Figure 8.

Firstly, the SPIRE Agent and Server need to validate the machine that hosts the Agent through the process known as Node attestation. The Agent retrieves information from the provider hosting it using plugins managed by its "Node attestor" component. For instance, if hosted at Amazon Web Service (AWS), the Agent can employ the AWS Instance Identity Document to prove its identity (FELDMAN et al., 2020). I have summed up this via step 1 in Figure 8. The Agent then sends the information collected to the SPIRE Server illustrated in step 2. In step 3, the SPIRE Server consults the provider to confirm the details sent by the Agent by using plugins managed by its own "Node attestor" component. If the collected details show that the Agent's node matches the selectors described in the related entry, the SPIRE Server issues the identity for the Agent as shown in step 4. The node hosting the Agent now has its identity, concluding the node attestation process.

Once we have attested the SPIRE Agent's node, we then apply an attestation on the workloads with pertinent plugins, known as workload attestation. This process starts with creating an entry through the SPIRE server, displayed as "Register Entries" in Figure 8. The workload requests an SVID from its Agent, seen in step 5. The Agent consults its subcomponent, "Workload Attestor", which manages the employed plugins, analogous to the "Node Attestor" as portrayed in step 6. If the information collected by the "Workload Attestor" matches the selectors in the workload's entry, the Agent issues an identity to the workload via a cryptographic identity in the form of an X.509 certificate as presented in step 7. (FELDMAN et al., 2020).

By using attestation and issuing ids as X.509 certificates, SPIRE provides a solid addition to the interaction between the services. Once a SPIRE agent has performed an attestation on

a given service, it is granted an appropriate identity via an X.509 SVID , and workloads can use it in Mutual Transport Layer Security (mTLS) connections. We can then ensure that the attested services only communicate with other attested. This is represented in step 8 in Figure 8.

Executing an application entirely in a TEE is also an alternative. This option is more costly as we are not just using hardware-based attestation to perform the function of a plugin. On the other hand, this alternative ties the whole execution of the application to the TEE guarantees. With this in mind, we are running the entire application within a TEE with hardware-based attestation. The execution within TEEs, for example, using the SCONE framework detailed in Section 3, could be done by itself or in conjunction with SPIRE. Given the sensitive nature of the data manipulated by my use case system and the computations on privacy-sensitive data the services perform, I have opted to execute the system's services and the secret manager using a TEE. I also make a point to ensure the services encrypt data that must be persisted outside the TEE, such as the messages in the message bus.

I have chosen SCONE's trust management service PALÆMON to occupy the role of identity manager. Only after the proper attestation has been performed, PALÆMON provides some initial configuration so each service can reach the secret manager. I have chosen a MariaDB instance executed inside a TEE to serve as my secret manager (WIDENIUS, 2022). I have chosen MariaDB to store configurations as it is a widely used and consolidated storage solution (GREGOR et al., 2020). In Figure 9, Figure 10, and Figure 11 we can see a representation of a cog leaving PALÆMON for the service which represents the configuration being given to the service by PALÆMON. The configuration is used by the service to communicate with the MariaDB instance which is shown as the same representation of a cog being sent from the service to the MariaDB instance. In response, the MariaDB sends the relevant secrets to the service shown as a key being sent from MariaDB to the service.

I have chosen SGX as the TEE because of its strong guarantees ensured via hardware-based attestation (ANATI et al., 2013a). As depicted in Figure 9, the Trainer component was fully executed within SGX. We can see that the same was done with the Classifier via Figure 10, and with the Alert service via Figure 11. The same treatment was extended to the the MariaDB. As I have made sure the message bus handles only encrypted messages, it can be left unchanged. I chose Apache Kafka as the message bus (KREPS, 2011). In Figure 9,

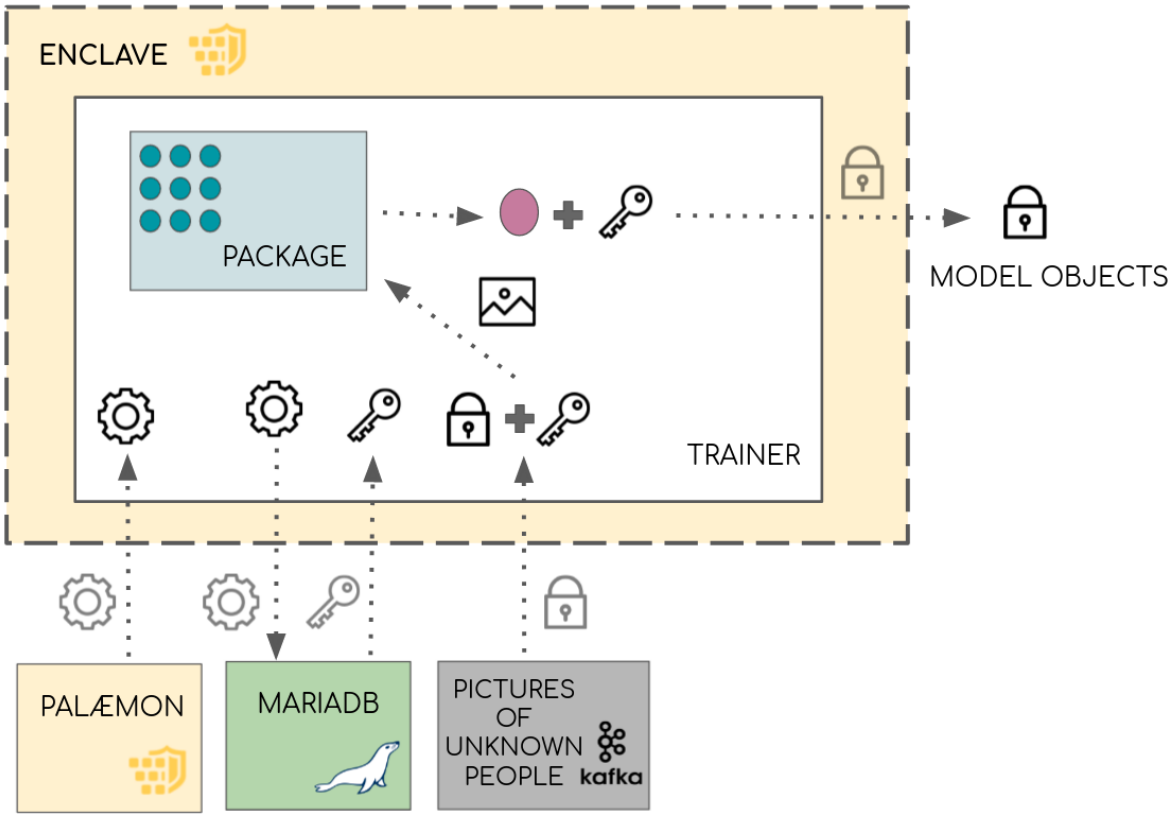


Figure 9: Trainer component after SGX integration

Figure 10, and Figure 11 the encodings serving as a rendition of the users' faces that a neural network can understand are represented as purple circles.

## 7.2 The Alert service problem

As showcased in the taxonomic classification, there is a problem with the Alert service's functionalities. As the Alert employs multiple people's data to provide its service to someone, it inherently exposes people's privacy in the input data. In other words, there is a privacy loss associated with the features of the Alert service. We have employed the following strategies to mitigate this issue within the context of this work. I have added an access control mechanism to ensure access to the service is limited to enrolled users. The functionality that encompasses data from multiple people primarily checks whether the system has detected someone with unusual temperature readings, in other words, someone possibly sick in a particular window of time. The system gives no information on the person it has spotted, and the user making the query is not given access to the sensor readings involved. However,

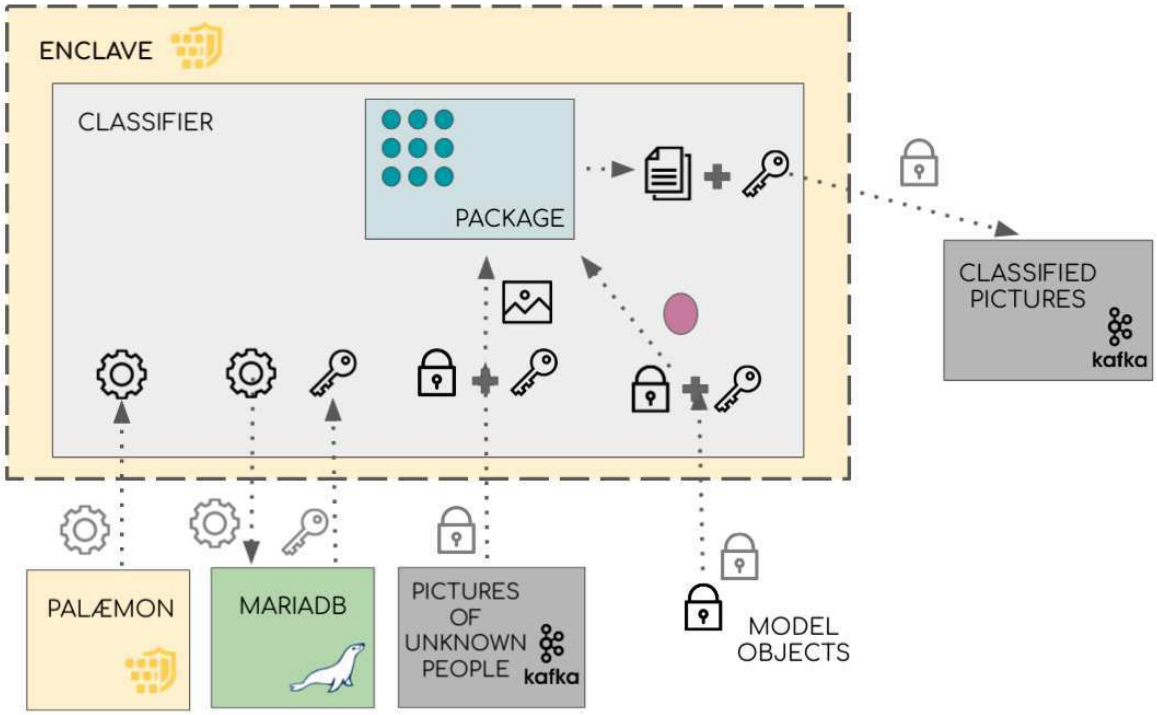


Figure 10: Classifier component after SGX integration

we must acknowledge that given few enough people in an area, access to queries on sick individuals would enable a malicious individual with knowledge of the people in the area at the given time to surmise the ill individual through this feature. For instance, with only two users, it becomes trivial for one of the users to deduce whether the other one is ill or not. The enrolled users are still allowed to query about themselves and check whether the system has detected them at a given time with either regular or unusual temperature readings. This way, I provide a user the full functionality based on the data that refers to them, and I restrict the functionality when it possibly includes data from several people. Before, an alert would identify the sick individual to the enrolled users. Now, the Alert notifies the enrolled users of the presence of a sick individual without identifying them. It is vital to notice that, although I have limited a user's access to data concerning other individuals for privacy reasons, the functionality still alerts people about possibly sick individuals in their vicinity. Lastly, enrolled users can still check on readings on themselves, and at any rate, the system can not reach people who are not registered.

I deem the restrictions I have imposed on the Alert service's features that extend over multiple people adequate for the scope of this work. However, I must clarify that I envision

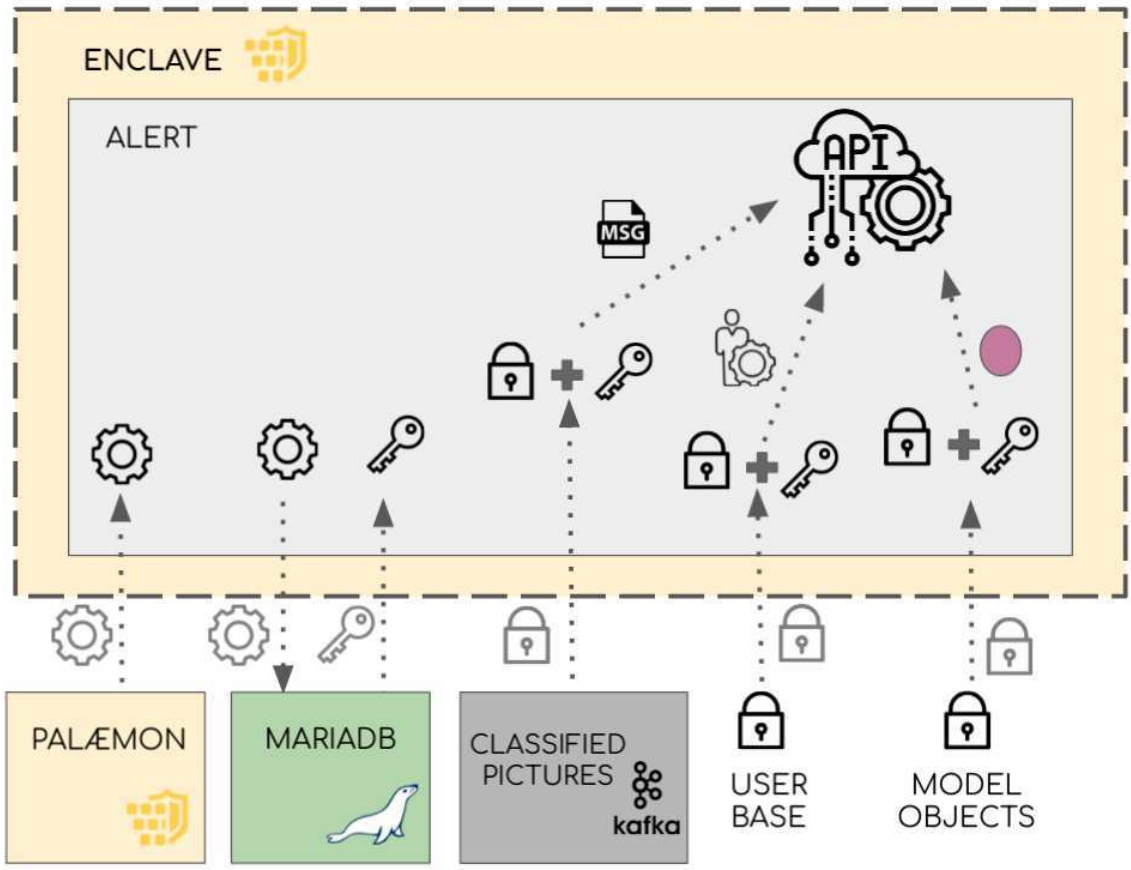


Figure 11: Alert component after SGX integration

these features as the appropriate entry point to employ well-known privacy solutions such as k-anonymity and differential privacy (SAMARATI; SWEENEY, 1998) (DWORK; ROTH et al., 2014). In particular, I believe that differential privacy is a good fit.

Firstly, I have classified the Alert service given these features as Bi-Directional in terms of the facet "Privacy of Location" which justifies a privacy solution. Secondly, the Alert service itself intermediates user access to query results. Lastly, although the restrictions on the Alert service's features I have discussed in this section limit the "privacy leakage" that intelligent methods, such as machine learning models, can exploit, there is no quantification of this escape. Differential privacy would act precisely to quantify this escape of data and help further minimize the threat of intelligent methods.

There is still, however, the challenge of misclassifications. As the system matches passersby to enrolled users, there is the possibility that the system misclassifies a passerby as an user. This problem is inherent to the employment of a machine learning model. There are

two ways I can address this situation. First, someone under an administrator role could go over the classifications, searching for the wrong ones. I can then use these misclassifications to fine-tune the model and make other mismatches more unlikely. Simply granting access to an administrator to all the classifications would unnecessarily expose people's privacy. For this reason, I have instead chosen to allow an enrolled user to assess matches made on them and report misclassified ones.

# 8 Conclusions

In this work, I have analyzed and classified applications employing and learning on potentially privacy-sensitive real-world data. I showcase a taxonomy on privacy and confidentiality to classify twenty-one different applications, grouped into nineteen distinct services. It is important to note that my taxonomy's capacity to categorize these tens of vastly different applications attests to its generality, and therefore it is not tied or limited to the use case system I have chosen. I have detailed the taxonomy's facets and levels and the method used to build it. I have also gone into greater detail into each particular decision I have taken to create the taxonomy in Chapter 4, and in particular in Section 4.5. By doing so, I evaluate the taxonomy's validity and correctness tangibly and meaningfully and preserve the reproducibility of my work in contrast with taxonomies built *ad hoc*. Creating a taxonomy following strict requirements serves as a positive contribution to the lack of taxonomies following a distinct methodology in areas related to or within Computer Science (NICKERSON; VARSHNEY; MUNTERMANN, 2013)(USMAN et al., 2017).

While analyzing third-party applications helps us develop and validate the taxonomy, applying it to individual components helps better illustrate its value. Therefore, I apply this taxonomy to a use case system capable of temperature monitoring and facial recognition. I used the insights I have extracted from the classification to devise improvements to the services within my use case system. Through these improvements, I aimed at solving or mitigating the privacy and confidentiality problems discerned in the classification. I present this taxonomy as a starting point that can be expanded or used with other taxonomies. Finally, I have validated my taxonomy through an orthogonality demonstration and a utility demonstration. Thanks to the very nature of the method I have employed to construct the taxonomy, evaluating the taxonomy with new applications always remains a venue to improve the taxonomy further.

The Alert service's categorization displayed particular characteristics when I applied my taxonomy to this work's use case system. I have argued that by necessarily employing multiple people's data to provide its functionalities to a single person, the Alert service inherently exposes the privacy of the people its data represents. I have provided a solution to curtail this problem by restricting the functionality that raises concern in terms of privacy. A venue

for future work lies in exploring the use of the privacy-preserving techniques discussed by Domingo-Ferrer et al. (DOMINGO-FERRER et al., 2019), as well as others, in terms of viability and trade-off as ways to solve the problem presented by the Alert service. For instance, one could explore employing differential privacy to manage the privacy loss of the Alert Service with quantifiable privacy guarantees.

Analyzing how this work's taxonomy synergizes with known threat modeling approaches remains a meaningful research direction. On the one hand, one could explore how we might combine fundamentally different techniques such as OCTAVE with this work's taxonomy to create a more comprehensive method. On the other hand, one could study how we might augment threat modeling techniques already focused on privacy, such as LINDDUN or STRIDE, with a privacy-oriented artifact such as this work's taxonomy.

Lastly, this work's artifact could be instrumental in the context of LGPD and GDPR. The matter of data privacy is remarkably nuanced, and a taxonomy centered on this subject could help guide the protocols devised by private or governmental entities to protect their data or even help them decide which components and systems are critical in terms of privacy.

# References

Ahmed, N. et al. A survey of covid-19 contact tracing apps. *IEEE Access*, v. 8, p. 134577–134601, 2020.

AIRBEAM. 2022. <https://www.habitatmap.org/airbeam>. Accessed on January 1st, 2022.

ANATI, I. et al. Innovative technology for cpu based attestation and sealing. In: ACM NEW YORK, NY, USA. *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy*. [S.l.], 2013. v. 13, p. 7.

ANATI, I. et al. Innovative technology for cpu based attestation and sealing. In: . [S.l.: s.n.], 2013.

ANGELIDOU, M. et al. Enhancing sustainable urban development through smart city applications. *Journal of Science and Technology Policy Management*, v. 9, 12 2017.

ARNAUTOV, S. et al. SCONE: Secure linux containers with intel SGX. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016. p. 689–703. ISBN 978-1-931971-33-1. Disponível em: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/arnautov>.

AU WATER STORAGE. 2022. <http://www.bom.gov.au/water/dashboards/#/water-storages>. Accessed on January 1st, 2022.

BIGBELLY. 2022. <https://bigbelly.com/products/>. Accessed on January 1st, 2022.

BREITBARTH, P. The impact of GDPR one year on. *Network Security*, v. 2019, n. 7, p. 11–13, 2019. ISSN 1353-4858. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1353485819300844>.

BUILDING ENERGY BENCHMARKING. 2022. <http://visualization.phillybuildingbenchmarking.com/>. Accessed on January 1st, 2022.

COSTAN, V.; DEVADAS, S. *Intel SGX Explained*. 2016. Cryptology ePrint Archive, Report 2016/086. <https://ia.cr/2016/086>.

DEMIGHA, O.; LARGUET, R. Hardware-based solutions for trusted cloud computing. *Computers  Security*, v. 103, p. 102117, 2021. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404820303904>.

DOMINGO-FERRER, J. et al. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications*, v. 140-141, p. 38–60, 2019. ISSN 0140-3664. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0140366418310740>.

DWORK, C.; ROTH, A. et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, v. 9, n. 3-4, p. 211–407, 2014.

Eckhoff, D.; Wagner, I. Privacy in the smart city—applications, technologies, challenges, and solutions. *IEEE Communications Surveys Tutorials*, v. 20, n. 1, p. 489–516, 2018.

ENEVO. 2022. <https://www.enevo.com/waste-solutions-services>. Accessed on January 1st, 2022.

ERLENHOV, L. et al. Current and future bots in software development. In: *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*. [S.l.: s.n.], 2019. p. 7–11.

EVERIMPACT. 2022. <https://www.everimpact.com/#home>. Accessed on January 1st, 2022.

FELDMAN, D. et al. *Solving the Bottom Turtle — a SPIFFE Way to Establish Trust in Your Infrastructure via Universal Identity*. 1. ed. [S.l.: s.n.], 2020. This book presents the SPIFFE standard for service identity, and SPIRE, the reference implementation for SPIFFE. https://spiffe.io/book/.

FINN, R. L.; WRIGHT, D.; FRIEDEWALD, M. Seven types of privacy. In: *European Data Protection*. [S.l.: s.n.], 2013.

GHAYVAT, H. et al. Recognizing suspect and predicting the spread of contagion based on mobile phone location data (counteract): A system of identifying covid-19 infectious and hazardous sites, detecting disease outbreaks based on the internet of things, edge computing, and artificial intelligence. *Sustainable Cities and Society*, v. 69, p. 102798, 2021. ISSN 2210-6707. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2210670721000901>.

GOEZY. 2022. <https://www.metropia.com/goezy-app>. Accessed on January 1st, 2022.

GOLDREICH, O.; MICALI, S.; WIGDERSON, A. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, Association for Computing Machinery, New York, NY, USA, v. 38, n. 3, p. 690–728, jul 1991. ISSN 0004-5411. Disponível em: <https://doi.org/10.1145/116825.116852>.

GOOGLEFIT. 2022. <https://www.google.com/fit>. Accessed on January 1st, 2022.

GREGOR, F. et al. Trust management as a service: Enabling trusted execution in the face of byzantine stakeholders. In: *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. [S.l.: s.n.], 2020. p. 502–514.

HUANG, G. B. et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. [S.l.], 2007.

KREPS, J. Kafka : a distributed messaging system for log processing. In: . [S.l.: s.n.], 2011.

LEAKVIEW. 2022. <http://www.visenti.com/?page_id=206>. Accessed on January 1st, 2022.

LITECAMPUS. 2022. <https://litecampus.lsd.ufcg.edu.br/>. Accessed on January 1st, 2022.

LITEME. 2022. <https://www.liteme.com.br/>. Accessed on January 1st, 2022.

MOODFIT. 2022. <https://play.google.com/store/apps/details?id=com.robleridge.Moodfit&hl=en&gl=US>. Accessed on January 1st, 2022.

MUNZERT, S. et al. Tracking and promoting the usage of a covid-19 contact tracing app. *Nature Human Behaviour*, v. 5, n. 2, p. 247–255, Feb 2021. ISSN 2397-3374. Disponível em: <https://doi.org/10.1038/s41562-020-01044-x>.

MY BUILDING DOESNT RECYCLE. 2022. <http://mybuildingdoesntrecycle.com/>. Accessed on January 1st, 2022.

MYCITY360. 2022. <https://mycity360.co.il/>. Accessed on January 1st, 2022.

NICKERSON, R. C.; VARSHNEY, U.; MUNTERMANN, J. A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, Taylor Francis, v. 22, n. 3, p. 336–359, 2013. Disponível em: <https://doi.org/10.1057/ejis.2012.26>.

NISSENBAUM, H. Privacy as contextual integrity. *Washington Law Review*, University of Washington School of Law, v. 79, n. 1, p. 119–157, fev. 2004. ISSN 0043-0617.

OCTOPUS CARD/APP. 2022. <https://www.octopus.com.hk/tc/consumer/mobile-payment/octopus-app/about/index.html>. Accessed on January 1st, 2022.

OPENTREEMAP. 2022. <https://www.opentreemap.org/>. Accessed on January 1st, 2022.

PONCIANO, L. et al. Designing for pragmatists and fundamentalists: Privacy concerns and attitudes on the internet of things. In: *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2017. (IHC 2017). ISBN 9781450363778. Disponível em: <https://doi.org/10.1145/3160504.3160545>.

REROUTE IT. 2022. <https://icos.urenio.org/applications/reroute-it/>. Accessed on January 1st, 2022.

SABT, M.; ACHEMLAL, M.; BOUABDALLAH, A. Trusted execution environment: What it is, and what it is not. In: *2015 IEEE Trustcom/BigDataSE/ISPA*. [S.l.: s.n.], 2015. v. 1, p. 57–64.

SAMARATI, P.; SWEENEY, L. *Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression*. Computer Science Laboratory, SRI International, 1998. Disponível em: <http://www.csl.sri.com/papers/sritr-98-04/>.

SCARLATA, V. et al. Supporting third party attestation for intel® sgx with intel® data center attestation primitives. In: . [S.l.: s.n.], 2018.

SHAN, Z. et al. Practical secure computation outsourcing: A survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 51, n. 2, feb 2018. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3158363>.

SHEVCHENKO, N. et al. *Threat Modeling: A Summary of Available Methods*. [S.l.], 2018.

SILVA, M. S. L. d. *Integrating SPIFFE and SCONE to enable universal identity support for confidential workloads.* Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2021. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/21598>.

SUDRE, C. H. et al. Symptom clusters in covid-19: A potential clinical prediction tool from the covid symptom study app. *Science Advances*, v. 7, n. 12, p. eabd4177, 2021. Disponível em: <https://www.science.org/doi/abs/10.1126/sciadv.abd4177>.

TASSYANY, M. *Um Mecanismo de aprovisionamento de Identidades para Microsserviços Baseado na Integridade do Ambiente de Execução*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2021. Disponível em: <https://doi.org/10.5753/sbrc.2021.16758>.

TEMER, M. et al. *LGPD ruling*. 2018. <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm>. Accessed: 12-11-2021.

USMAN, M. et al. Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method. *Information and Software Technology*, v. 85, p. 43–59, 2017. ISSN 0950-5849. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584917300472>.

WASTEOS DUMPSTER MONITORING. 2022. <https://compology.com/industries/waste-and-recycling/>. Accessed on January 1st, 2022.

WASTEOS TRUCKING MONITORING. 2022. <https://compology.com/industries/trucking>. Accessed on January 1st, 2022.

WIDENIUS, U. M. *MariaDB*. 2022. <https://mariadb.org/>. Accessed on January 1st, 2022.

Xiao, Z.; Xiao, Y. Security and privacy in cloud computing. *IEEE Communications Surveys Tutorials*, v. 15, n. 2, p. 843–859, 2013.

YANG, J.; LI, J.; NIU, Y. A hybrid solution for privacy preserving medical data sharing in the cloud environment. *Future Gener. Comput. Syst.*, v. 43-44, p. 74–86, 2015.

ZHANG, K. et al. Security and privacy in smart city applications: Challenges and solutions. *Comm. Mag.*, IEEE Press, v. 55, n. 1, p. 122–129, jan 2017. ISSN 0163-6804. Disponível em: <https://doi.org/10.1109/MCOM.2017.1600267CM>.

ZIMMER, V.; KRAU, M. Establishing the root of trust. *Accessed: Jun*, v. 15, p. 2021, 2016.