SpaceMediator: Preventing Spatial and Privacy Attacks in Mobile Augmented Reality

by

Luis Manuel Claramunt

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved March 2022 by the
Graduate Supervisory Committee:

Gail-Joon Ahn, Chair
Carlos E. Rubio-Medrano
Jaejong Baek

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Mobile Augmented Reality (MAR) is a portable, powerful, and suitable technology that integrates 3D virtual content into the physical world in real-time. It has been implemented for multiple intents as it enhances people's interaction, e.g., shopping, entertainment, gaming, etc. Thus, MAR is expected to grow at a tremendous rate in the upcoming years, as its popularity via mobile devices has increased. But, unfortunately, the applications that implement MAR, hereby referred to as MAR-Apps, bear security issues. Such are imaged in worldwide recorded incidents caused by MAR-Apps, e.g., robberies, authorities requesting banning MAR at specific locations, etc. To further explore these concerns, a case study analyzed several MAR-Apps available in the market to identify the security problems in MAR.

As a result of this study, the threats found were classified into three categories. First, Space Invasion implies the intrusive modification through MAR of sensitive spaces, e.g., hospitals, memorials, etc. Then, Space Affectation means the degradation of users' experience via interaction with undesirable MAR or malicious entities. Finally, MAR-Apps mishandling sensitive data leads to Privacy Leaks.

SpaceMediator, a proof-of-concept MAR-App that imitates the well-known and successful MAR-App Pokémon GO, implements the solution approach of a Policy-Governed MAR-App, which assists in preventing the aforementioned mentioned security issues. Furthermore, its feasibility is evaluated through a user study with 40 participants. As a result, uncovering understandability over the security issues as participants recognized and prevented them with success rates as high as 92.50%. Furthermore, there is an enriched interest in Policy-Governed MAR-Apps as 87.50% of participants agreed with restricted MAR-Apps within sensitive spaces, and 82.50% would implement constraints in MAR-Apps. These promising results encourage the adoption of the Policy-Governed solution approach in future MAR-Apps.

DEDICATION

Aracely, mi dulce y generosa madre

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Augmented Reality (AR) alters the perception of the physical world by merging natural objects with additional virtual content, resulting in distinct users' sights of their surroundings. Noticeably, its popularity in the mobile sector has increased as Mobile Augmented Reality (MAR), via devices with lower accessibility costs, higher power, and the emergence of sophisticated communication infrastructure [2].

Currently, several types of applications implement MAR, hereby referred to as MAR-Apps. For example, there are MAR-Apps used for shopping (e.g., IKEA Place, Wayfair, eBay, etc.), entertainment (e.g., Snapchat, MARK, etc.), productivity (e.g., GeoGebra, Measure, etc.), and gaming (e.g., Jurassic World Live, etc.). Furthermore, the last category involves one of the most successful MAR-Apps: Pokémon GO. It had become a worldwide phenomenon since its release in 2016 when it experienced 21 million daily active users [3]. Through it, players capture AR as digital objects featuring Pokémon characters available globally. However, even as millions of people already use MAR, this technology is still in its early development stages [4].

MAR has succeeded in value, as users and implementation raised considerably [5]. Likewise, further development on libraries facilitates MAR execution, e.g., ARCore, ARKit, Vuforia, etc. Therefore, it is no surprise that Allied Market Research anticipates the MAR market to reach $184.61 billion by 2030, from $12.61 billion in 2020, with a compound annual growth rate of 31.40% from 2021 to 2030 [6].

Thus, with such tremendous potential and with no standard over how to regulate MAR-Apps, it is crucial to consider their safety as some, i.e., Pokémon GO, have been problematic, as we discuss next. Therefore, we analyzed and identified vulnerabilities

in several MAR-Apps available in the market and categorized them into three security issues; based on recorded incidents and possible outbreaks.

Space Owners are the entities who possess *sensitive space*, .e.g., memorials, hospitals, etc. Therefore, they must have the opportunity to regulate MAR-Apps operations within such locations, as some MAR content might be unwanted or lead to unwelcomed behavior; otherwise, they would suffer from *Space Invasion*. Such incident has already occurred throughout the world with Pokémon GO as Space Owners dealt with intrusive MAR, e.g., the 9/11 Memorial in New York City [7], Auschwitz WWII Holocaust [8], etc.

There is also a possibility for digital graffiti as MAR leaves physically unnoticeable traces, e.g., stickers, drawings, messages, 3D objects, etc. Currently, there are no restrictions on such content, allowing hostile entities to place malicious content easily. Furthermore, such entities have already exploited MAR-Apps compromising users' security to execute robberies, fights, assaults, etc. [9] Overall, users' experience depreciation via dangerous MAR content and risky multi-user interactions leads to *Space Affectation* issues.

MAR-Apps also deal with sensitive information, which leads to *Privacy Leak* if gathered without explicit consent or unwillingly shared with third parties. This issue is not limited to MAR-Apps [10]. Still, some of the analyzed MAR-Apps mishandled sensitive information, with an unspecified range for data, e.g., device facts, location, data generated by MAR-Apps, etc. [11]

We propose the prevention of the mentioned attacks by regulating the operations of MAR-Apps. For example, protected sensitive spaces via the implementation of the *sensitive space access control* (SSAC) mechanism. As a result, Space Owners can adequately restrain the MAR-App utilization within their domain and prevent Space Invasion. Similarly, user interchange is controlled, as they interact through

regulated *Rooms* with restrained admission through *attribute-based access control* (ABAC). Each Room receives unique MAR objects, and as policies created by users determine regulations for access and acceptable MAR content, it prevents Space Affectation. Alongside, users must know all the sensitive information collected from them through an *Attribute Wallet*, which also handles the data gathering. Therefore, resulting in possible prevention for Privacy Leak. Overall, the enforcement of such constraints leads to a Policy-Governed MAR-App. We demonstrate its implementation in `SpaceMediator`, a *proof-of-concept* Policy-Governed MAR-App that imitates Pokémon GO. It represents a multiplayer geolocation-based MAR-App, where multi-user interaction is possible through assigned locations to available MAR objects. Although, it respects protected sensitive spaces, restrains interaction among users, and allows them to manage gathered sensitive information.

Alongside, we sampled `SpaceMediator`'s usability through a user study with 40 participants. Without a requirement of prior computer science knowledge or exposure to MAR, they were introduced to the security issues found in MAR-Apps, prevented them in `SpaceMediator`, and provided feedback reflecting their experience. The results were satisfactory as, for example, participants comprehended the attacks with rankings as high as 4.65 on a scale from 1 to 5; also, 87.50% of them agreed on Policy-Governed MAR-Apps over sensitive spaces, and 82.50% would implement user regulations. Likewise, they wrote policies to regulate the operations of `SpaceMediator`, which assisted us in testing the feasibility of leaving the regulation responsibilities to ordinary users. We determined the efficiency of these policies, managing authorization as expected, through an automated evaluation mechanism which also led to satisfactory results.

The thesis follows the following organization: Chapter 2 provides background information on relevant topics, Chapter 3 compares it to related works and distinguish

its contributions, Chapter 4 provides a deeper explanation of the problems we address, Chapter 5 explains our solution model to handle such issues, Chapter 6 presents the implementation of the solution throughout `SpaceMediator`, Chapter 7 presents the research questions we addressed in a conducted user study and their answers, Chapter 8 reviews the results and identifies areas for future work, and Chapter 9 presents a conclusion.

Chapter 2

BACKGROUND

This chapter explains the technology `SpaceMediator` regulates, security issues that inspired the project, and necessary cybersecurity concepts. It covers Mobile Augmented Reality in Chapter 2.1, incidents caused by it in Chapter 2.2, the relevant access control mechanisms in Chapter 2.3, and an implementation model for such mechanisms in Chapter 2.4.

2.1 Mobile Augmented Reality

Mobile Augmented Reality (MAR) is a portable implementation of *Augmented Reality* (AR), that enables real-time interaction between 3D digital content and the actual physical world [5] [12]. It is commonly implemented in mobile applications accessible through smartphones, tablets, etc., hereby referred to as MAR-Apps. Its popularity has considerably grown as it tends to enrich users' experience and improve satisfaction [13].

MAR-Apps have diverse categories, e.g., games, shopping, entertainment, productivity, education, etc. Also, there are some geolocation-based MAR-Apps in which MAR objects are granted a specific location [14]. As a result, the virtual AR content displayed depends on the user's location. For example, through Live View Google Maps provides directions with AR arrows which are consistently updated to guide the user to navigate the surroundings [15]. Another example is the very successful MAR-App Pokémon GO. Users must reach the precise spot assigned to a Pokémon to interact and capture it dynamically by touching the screen to throw a Poké Ball

to the AR Object.

Furthermore, as the requirement for MAR is for AR technology to be portable, it is worth pointing out that MAR is not limited to MAR-Apps, as they are implemented in three different categories: smartphones, AR headsets, and AR glasses [16]. Generally, as AR headsets and AR glasses are wearable, they lead to more extended utilization with constant modification of surroundings through virtual content. However, the high-quality AR output they tend to offer brings affordability issues. Therefore, we focus on regulating the operations of MAR-Apps as they are the major trend in MAR utilization and are also more accessible since no extra gear is required.

2.2 Mobile Augmented Reality Incidents

Currently, there is an absence of regulation over how MAR-Apps operate. For example, there is no limitation over where MAR-Apps can be launched, an absence of restrictions over the MAR content available to users, and how MAR objects are distributed among users. As a result, as the popularity of MAR-Apps increases, more incidents caused by MAR have been recorded.

The absence of authority over where MAR-Apps can be started has led to users interacting with it in locations considered disrespectful. For example, people have been able to play Pokémon GO at the 9/11 Memorial in New York City, which was viewed as irreverent by many within the community [7]. Similar situations occurred in Poland's Auschwitz Memorial and Washington's D.C. Holocaust Museum, which requested MAR-Apps to be unplayable sites [8].

The regulation deficiency over how MAR objects are distributed has also compromised users. It is common for everyone in a MAR-App to have access to the same MAR objects, which has compromised users' security as malicious users waited at places where interactive MAR objects were available to assault or rob them [9]. It

has also caused crowds of hundreds of players, or more, leading to unpleasant noisy environments [17].

MAR-Apps users have also been involved in general incidents. For example, users broke into private properties as they did not respect the boundaries of deployed MAR objects, had car accidents as they used MAR-Apps while driving, or were injured because of distracted behavior while utilizing the MAR-Apps [18].

2.3 Access Control

An *Access Control* (AC) model regulates the operations a user may be able to perform on a resource, e.g., reading, writing, interacting with AR Content, etc., through *authorization decisions*, i.e., permit or deny, preventing improper entrance. Such protected resources are commonly referenced as *objects*, e.g., files, database records, videos, displays, etc. The entities who ask for access to the objects are known as *subjects*. They provide an *access request* describing who they are, their intentions, alongside any other relevant information. These are evaluated through a mechanism against *policies*, which are high-level structures that administer access [19]. There are several access control paradigms, and this Chapter introduces the ones relevant for `SpaceMediator`.

2.3.1 Attribute-Based Access Control

Attribute-Based Access Control (ABAC) is a logical AC model that controls access to objects by evaluating rules against the attributes of the entities, e.g., subject, object, environment, etc., relevant to a request. Such attributes may be characteristics of anything that may be defined and to which a value may be assigned [20]. These attributes are structured in a 4-tuple subset containing valuable information as {*ID, Data Type, Values, Category*}, e.g., {Username, String, Bob, Subject}, {OS, Integer,

28, Object}, {Current Time, Time, 12:00:00, Environment}.

ABAC is a very flexible model since collectible data can be used as attributes in well-structured policies accessible through multiple devices [21]. Due to such flexibility when evaluating an access request, information might be gathered from different sources. As a result, for a secure ABAC attributes must follow: *accuracy*, the attributes' values are correct; *integrity*, information is securely exchanged among parties; and *availability*, attributes are securely retrieved from the relying parties [22].

The dynamics of ABAC have made it a suitable AC model for large enterprises with complex control lists [19]. For example, an ABAC policy limits MAR interaction only to adults after 5:00 p.m. Then, the required information to evaluate the policy is efficiently gathered through a proper deployment. For example, getting the subject's age associated with an account from a database, scanning the object requested for access in a mobile device to identify it correctly, and updating the server's environmental condition of time as the request is evaluated.

2.3.2 Space-Sensitive Access Control

Space-Sensitive Access Control (SSAC) is a subset of ABAC as it has policies based on attributes that regulate functionality over predefined locations known as *sensitive spaces* [23]. Therefore, just like ABAC, SSAC is a flexible, efficient, and accessible model, as explained in Chapter 2.3.1. The main difference between these AC models is that SSAC always defines a geographical area, e.g., square, triangle, diamond, etc., over which authorization decisions will take place. For example, it is possible to have a policy that only grants access to MAR objects to adults after 5:00 p.m. while within a company's facility.

2.4 XACML

The eXtensible Access Control Markup Language (XACML) is an access control language for policies, requests, and responses approved by the Organization for the Advancement of Structured Information Standards (OASIS) [20]. Also, it is the *de facto* standard language for ABAC as it provides expressive and extensible capabilities in XML syntax, e.g., policy specifications, policy combining algorithms, etc. [24]

A model of basic XACML usage is shown in Figure 2.1. Everything starts when a subject submits an access request to the *Policy Enforcement Point* (PEP). This request is handled appropriately in the *Context Handler*, which manages proper communication among the different points. There, more relevant information could be added by the *Policy Information Point* (PIP). Once the request is ready, it is given to the *Policy Decision Point* (PDP). Next, *Policy Administration Point* (PAP) retrieves the applicable policy and assigns it to the PDP. Once the PDP has been delegated the necessary information, it will evaluate the request against the policy to make an authorization decision. This decision is sent back to the PEP, which will inform the subject concisely of the authorization, i.e., a JSON file with a Permit statement.

2.4.1 Policies

Authorization decisions are taken through the requirements established in a policy. Syntax elements in a policy determine attributes' {*ID, Data Type, Values, Category*}, their relation, and other relevant information for the decision process [25].

- **Policy:** This element must be present for proper *PDP* evaluation. It includes a series of Rules elements with specified *Rule Combining Algorithm*, which is the procedure taken for authorization decisions. As a result, subjects are granted access or not through a combination of satisfied statements within a policy

and the Rule Combining Algorithm. For example, *deny-unless-permit* only authorizes subject if Permit statements are met, *permit-unless-deny* exclusively rejects a subject if a Deny regulation is fulfilled, and *deny-overrides* prioritizes Deny decisions.

- **Rule:** They have an *Effect*, either Permit or Deny, which establishes an authorization decision taken if the rule is satisfied. A Rule's parameters are stated with attributes at the Rule's Target and/or Condition.

- **AttributeDesignator:** Retrieves values from attributes in the request with the same name. If value MustBePresent="true" then missing attributes will return a value of *Indeterminate*.

- **AnyOf:** Elements joined by this element end up with a disjunctive sequence, appended by an *AND* logical operator ($Attr_1 \wedge Attr_2$).

- **AllOf:** Elements joined by this element end up with a conjunctive sequence, appended by an *OR* logical operator ($Attr_1 \vee Attr_2$).

- **Condition:** A Boolean function, stated within the *Rule*, stating requirements before evaluating the attributes in the *Rule's* Target.

An example of a sample policy with reduced XACML syntax is shown in Listing 2.1. This policy works as a deny list that keeps track of unauthorized subjects. For example, with this policy, any of the stated subjects, $User_0 \vee User_1$, would be denied access when requesting it.

As shown in Figure 2.2, the deny list policy covered in Listing 2.1 is represented in a manageable graph that covers the most relevant information for an ABAC policy, including the applicable combining algorithm, rules stated within the policy, their ef-

```
1  <Policy RuleCombiningAlgId="permit-unless-deny">
2    <Description/> <Target/>
3    <Rule RuleId="Deny_List" Effect="Deny">
4      <Target>
5        <AnyOf>
6          <AllOf>
7            <Match MatchId="string-equal">
8              <AttributeValue>User0</AttributeValue>
9              <AttributeDesignator AttributeId="Username"/>
10           </Match>
11         </AllOf>
12         <AllOf>
13           <Match Id="string-equal">
14             <AttributeValue>User1</AttributeValue>
15             <AttributeDesignator AttributeId="Username"/>
16           </Match>
17         </AllOf>
18       </AnyOf>
19     </Target>
20   </Rule>
21 </Policy>
```

Listing 2.1: Deny List Policy.

fect, attributes in each rule, how these attributes are related, and pertinent attribute's data as id, value, and operation.

2.4.2 Requests

As previously explained in Chapter 2.3, access requests reflect the identity of a subject via attributes. Syntax in requests is not as complex as in policies as attributes are simply written in their respective category. As shown in Listing 2.2 security-relevant information is provided via an access request. For example, this request reflects that $User_0$ is the entity asking for access through a Google device at 8:35 p.m. If we provided this request to a PDP assigned the Deny List policy shown in Figure 2.2 $User_0$ shall be unauthorized.

```
{    "Request":{
     "AccessSubject":[{
       "Attribute":[{
         "AttributeId":"Username",
         "Value":"User0",
         "DataType":"string"
       }]}],
     "Resource":[{
       "Attribute":[{
         "AttributeId":"Device_Manufacturer",
         "Value":"Google",
         "DataType":"string"
       }]}],
     "Environment":[{
       "Attribute":[{
         "AttributeId":"current-time",
         "Value":"20:35:00",
         "DataType":"time"
       }]}]
}}
```

Listing 2.2: XACML Access Request.

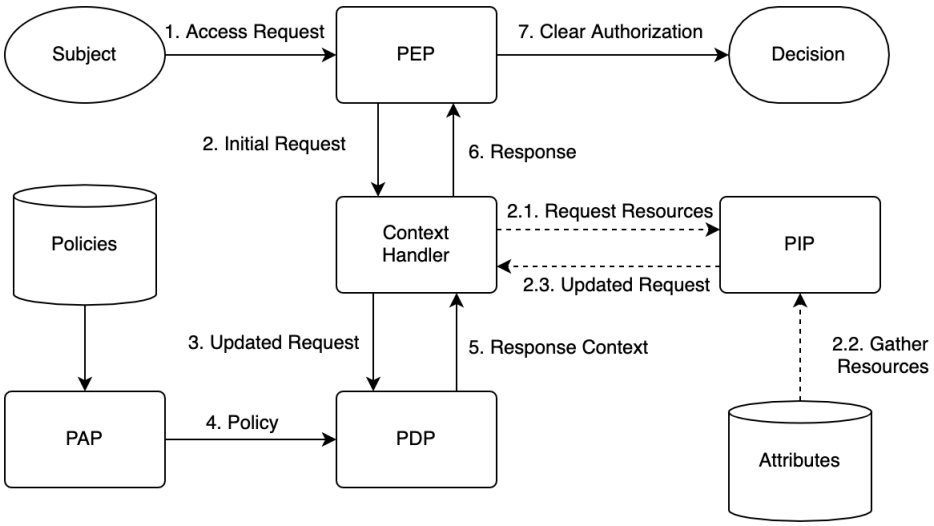


Figure 2.1: XACML Workflow Adapted from [1].

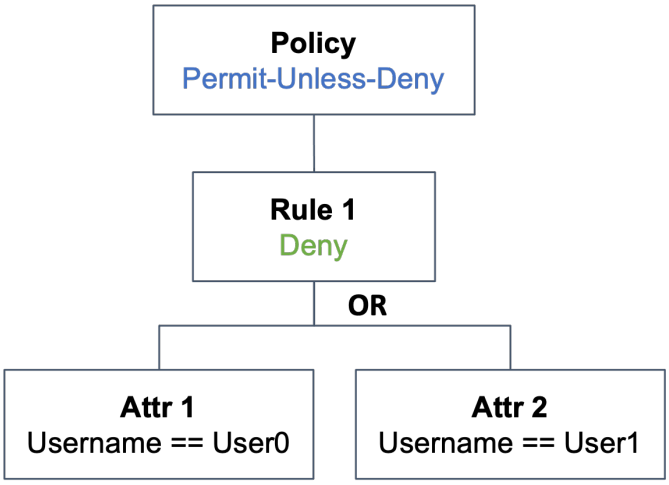


Figure 2.2: Deny List Policy Graph.

Chapter 3

RELATED WORK

Similarly, as the investment in AR increases, so has the concerns over its risks. It is a promising technology that remains in its first development stages, has caused issues with its success, and still misses well-defined regulations. Regardless, the computer science research community has kept its initiative and explored AR operations, identified vulnerabilities, and proposed remediations. Therefore, we will look through relatable research to our project.

Rubio-Medrano et al. [23] proposed the SSAC mechanism to regulate MAR-Apps over claimed physical spots, e.g., museums, hospitals, memorials, etc. Such SSAC represented an efficient authorization model suitable for MAR-App developers. Although, only Space Owners and developers were capable of regulating MAR-Apps' performance. Regular users, identified as the ultimate target of MAR-Apps, were left vulnerable, for example, to hostile MAR content as featured by the Space Affectation attacks discussed in Chapter 4.2.

Lebeck et al. [26] recognized security risks in virtual content. Although they utilized virtual reality, i.e., HoloLens, while we focus on accessible MAR-Apps, both technologies output virtual content that merges with the physical world and alters users' perspective. Furthermore, virtual content genuinely impacts the physical world, affecting users' actions and behaviors. Thus, maliciously placed AR content might cause dangerous or undesired actions among users. Alongside, a threat was detected in multi-user AR, especially among co-located users who modified each other via available AR, e.g., drawing, placing AR objects, etc. Similarly, users were concerned about inappropriate or hostile AR content. As a result, the necessity for further AR

regulation was acknowledged, along with challenges in its multi-user implementation. Through it, users shall manage their personal space, interact with AR objects personally, and control access to them, resulting in avoidance of unwanted interchange with others.

Lebeck et al. [27] identified security risks involved in MAR-Apps' abilities to modify users' surroundings, as malicious MAR-Apps were identified as capable of causing incidents by obscuring the real world. As a result, to prevent such happenings, MAR-Apps constraint their visual content through policies, which modify MAR content through specified attributes (e.g., size, rotation, etc.). However, while such policies restructured the output, they did not contain regulations that limited available MAR content (e.g., Spiders, Foxes, etc.). Also, the distribution of the utilized pre-defined policies in C# was outside the research's scope.

As previously mentioned, there is a wide range of AR content. Some are considered safety-critical as risks over incorrect AR output lead to dangerous side effects (e.g., driving, medicine, airplane maintenance, etc.). Therefore, research to analyze and prevent threats over such AR output has been conducted [28]. However, the safety-critical AR is less accessible than MAR-Apps because it requires more expensive tools. Nonetheless, we agree that AR output impacts users as they perceive the AR content, meaning varies, along with decisions taken afterward. Thus, mitigation of risky AR implies avoiding particular AR objects and reducing dangerous usage consequences, i.e., limiting usage time.

Finally, privacy issues over AR implementation is also primary concern as several researchers have assessed it. For example, Shang et al. [29] developed a successful tracking system to follow users' location in multi-user geolocation-based MAR-Apps. While we did not implement such a system, we recognize a threat to mishandling sensitive information exchange between mobile devices and a cloud service, alongside

limiting MAR-Apps permissions. Also, Zhang et al. [30] noticed missing mechanisms in Android to check if MAR collected unnecessary information, leading to a suggested framework to evaluate all information sent to a server. Alongside, we let users manage data gathered from them, as we explain later in Chapter 6.4.

Chapter 4

PROBLEM STATEMENT

This chapter presents security and safety issues on MAR-Apps. As MAR offers a wide variety of services, it is possible to find distinct types of vulnerabilities throughout its implementation in MAR-Apps. Therefore, it is essential to specify which ones are our priority and are addressed throughout the thesis.

We indicate in a threat model displayed in Figure 4.1 security implications involved in a MAR-App implemented with a cloud service. Some of the denoted threats are ways attackers could exploit mobile apps in general, i.e., stealing poorly stored login credentials (T1), modifying data provided by cellular devices to apps (T2), and intercepting insecurely exchanged information (T6) [31, 32, 33].

In this Thesis, we focus mostly on threats applicable to unregulated MAR-Apps with possible malicious MAR content (T3), leading to dangerous interaction (T4), and with forbidden access to sensitive data (T5). We analyzed several MAR-Apps, shown in Table 4.1, and found them vulnerable to at least one of such threats. Furthermore, each security issue is explained in detail, starting with *Space Invasion* in Chapter 4.1, *Space Affectation* in Chapter 4.2, *Privacy Leak* in Chapter 4.3, and an overview of the process used to examine the MAR-Apps in Chapter 4.4.

4.1 Space Invasion

This issue occurs in *sensitive spaces* where the *Space Owner*, the entity responsible for it, is unsatisfied with the executable MAR-Apps within the location. There are two possible ways MAR-Apps negatively affect a sensitive space. First, unwanted MAR

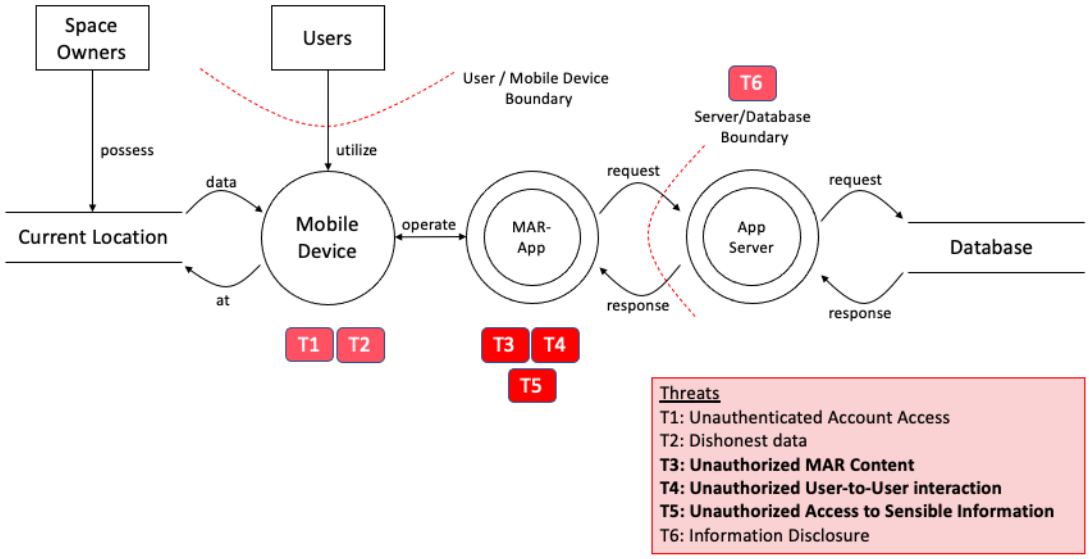


Figure 4.1: Threat Model.

content that merges with the physical world conducts negative interaction and virtual editing of its surroundings, as described in Chapter 2.1. Second, geolocation-based MAR-Apps could lead users to sensitive spaces and stimulate undesired behaviors, e.g., conglomerations, noisy environments, etc.

The security issues mentioned in Chapter 2.2 reflect real-world scenarios of space invasion. For example, the successful MAR-App Pokémon GO was unwelcomed at the 9/11 Memorial in New York City, Poland's Auschwitz Memorial, and Washington's D.C. Holocaust Museum [7] [8]. Also, players of the same MAR-App have broken into private properties [18].

4.2 Space Affectation

The Space Affectation issue is a result of meanly degraded MAR-Apps users' experience, triggered by intrusive MAR content, through which users must interact with MAR objects they despise, and negative user-to-user interaction.

As shown in Figure 4.2, geolocation-based MAR-Apps may lead towards user-to-

user interaction as two players meet at the exact spot assigned to a MAR object to play with it. Unfortunately, malicious users have taken advantage of such scenarios, and the multiplayer concept implemented throughout certain MAR-Apps has led to robberies, armed assaults, and other security compromising situations [18].

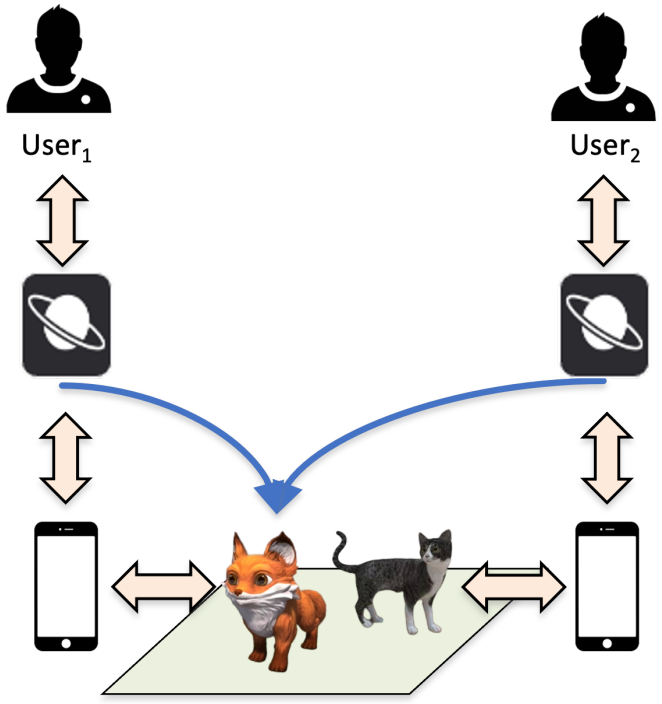


Figure 4.2: MAR-App Users Interaction.

4.3 Privacy Leak

There have been several mobile applications with recorded privacy incidents [10]. Therefore, privacy issues are not restricted to MAR-Apps. Although, it is noticeable that MAR-Apps share sensitive information between users and even without their explicit consent resulting in Privacy Leak. There is no specific range over the collected data as it could be distributed or generated, i.e., location [11].

4.4 Vulnerable MAR-Apps

We allocated relevant MAR-Apps on Google Play by running a search with relevant keywords, i.e., *augmented reality*, and exploring the available AR category. The MAR-Apps were installed on a Samsung S9 running Android 10 and a Motorola G6 running Android Pie. We utilized two devices to operate the MAR-Apps with different accounts and replicate multi-user interaction, one represented a benign entity while the other a malicious one. Through such process, we examined vulnerabilities and possible attacks.

We attempted to use each of the studied MAR-Apps within a series of physical spaces for the Space Invasion attack. If the operation was possible, exposing Space Owners to intrusive MAR, an attack was carried out successfully. For Space Affectation attacks, we evaluated the MAR content offered by the MAR-Apps and how it handled multi-user interaction. A successful attack was conducted by dangerous MAR content, and if the malicious user could compromise other's security via the MAR-App. Finally, we looked at the MAR-Apps sensitive information collection and handling for the Privacy Leak attack.

As shown in Table 4.1, all MAR-Apps were vulnerable to Space Invasion as they executed in the physical locations, and there was no provided way to limit their operations. In addition, several of the surveyed MAR-Apps were found vulnerable to Space Affectation. Some were geolocation-based MAR-Apps (e.g., Pokémon GO, Jurassic World, etc.) where the location assigned to MAR objects was publicly known. As described in Chapter 4.2, this has led to security incidents. Others were social MAR-Apps with no limitations over where MAR content could be shared or published, e.g., Snaappy, RealTag, WallaMe, MARK, etc. One user left traces with hostile MAR content as digital graffiti, and the attack was possible if the other user could interact

with such MAR content. Also, some of the MAR-Apps had violent MAR content, i.e., Weapon AR, leading to possible user experience degradation. Finally, some MAR-Apps demonstrated Privacy Leak attacks as they gathered sensitive information but did not handle it properly. For example, the user's current location was part of a public post without any previous warning.

MAR-App	**Space Invasion**	**Space Affectation**	**Privacy Leak**	**Downloads**	**Rating**
Pokémon GO	✓	✓	-	100M	4.1
Jurassic World Live	✓	✓	-	10M	4.4
The Walking Dead	✓	✓	-	5M	4.2
Color Quest AR	✓	-	-	1M	3.6
Snaappy	✓	✓	✓	1M	4.2
AR Real Driving	✓	-	-	500K	4.2
Just a Line	✓	-	-	500K	3.5
Weapon AR	✓	✓	-	100K	3.9
vTime XR	✓	✓	✓	100K	3.9
WallaMe	✓	✓	✓	100K	3.6
RealTag	✓	✓	-	100K	3.6
Real Note	✓	✓	✓	50K	3.6
My world	✓	✓	✓	10K	3.7
Tendar	✓	-	-	5K	3.9
MARK	✓	✓	-	1K	3.7

Table 4.1: MAR-Apps with Security/Safety Issues - March 2021.

Chapter 5

SOLUTION APPROACH

To prevent the security issues covered in Chapter 4, we propose the adoption of Policy-Governed MAR-Apps, which regulate MAR functionality at run-time. The *Control Model* (CM) used throughout Policy-Governed MAR-Apps is shown in Figure 5.1, composed of a set of relatable Entities associated with attributes that distinguish them.

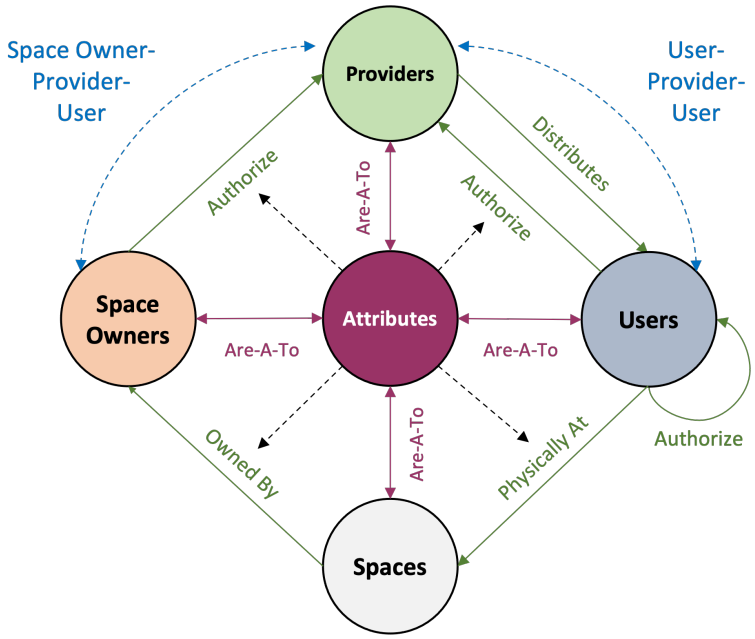


Figure 5.1: Policy-Governed MAR-Apps Control Model.

The *Provider* is the Entity primarily associated with developing and maintaining a MAR-App as long as the service is available. Generally, they can be related to attributes as IP addresses as mobile apps are associated with servers. In addition, they distribute MAR content for interaction.

The *Users* are the Entities who regularly operate a MAR-App. Therefore, they are the ones who interact with the supplied MAR content. As each User is unique, they are associated with attributes that distinguish them or personal information, e.g., ID, name, date of birth, etc. In a Policy-Governed MAR-Apps, Users shall have the opportunity to authorize who they interact with if multi-user interchange is possible and regulate the distributed MAR by the Provider.

The *Space* represents Users' location while interacting with MAR. As previously discussed in Chapter 2.1, MAR merges 3D digital content and the physical world. Therefore, leading to virtual manipulation or alteration of the surrounding. Some attributes identify Spaces, the areas impacted by MAR content, e.g., set of geographical coordinates, distance, etc.

Finally, the *Space Owners* are the Entities who *own* a Space. There is a wide range of possible Space Owners as MAR can influence various areas, e.g., parks, museums, businesses, residential areas, schools, etc. They are associated with attributes that assist in distinguishing them since each one is unique, e.g., ID, name, etc. Space Owners shall be able to avoid unwanted MAR content and antagonistic behavior on their property. Therefore, Providers must consider authorizations supplied by Space Owners. In this thesis, we assume legit ownership from the zone claimed by a Space Owner has been previously determined by external means. Such a process is interesting but demanding and is outside the scope of our research.

To recapitulate, we have described the Entities involved in the CM, their roles, how attributes are suitable to identify them, and their essential relationship with one another. Although, we must look deeper into how the proposed *Authorizations*, implementation of access control policies, prevent security issues. Therefore, it is worth examining the two *Modes of Interaction* that result from the presented regulations among Entities:

- **Space Owner-Provider-User**: Space Owners set regulations over MAR content and usage in a sensitive space, a claimed area, to Providers. Afterward, Providers will consider such regulations before delivering MAR to Users within the established sensitive space. As a result, Users are limited to authorized interaction within the specified boundaries. Therefore, the enforced restrictions potentially prevent the Space Invasion Attack.

- **User-Provider-User**: Users shall also establish regulations to Providers over the operation of the MAR-Apps that might impact them. Overall, they must regulate two parameters. First, the scope of MAR content they authorize for interaction. Then, if the MAR-App has multiplayer interaction, Users shall establish who they are willing to encounter. As a result, Providers will only distribute benign MAR content and avoid user-to-user interaction with unwanted parties. Therefore, limiting malicious third-parties exposure via MAR and potentially preventing the Space Affectation Attack. Likewise, Users must control the data MAR-Apps collect from them. They must be aware of any gathered sensitive information and manage to deliver it to Providers according to their will. Thus, stopping unawareness of personal data received by third parties and potentially preventing the Privacy Leak Attack.

Through the Modes of Interaction shown in the CM, there is potential prevention of Space Attacks. On the other hand, privacy issues depend on how developers handle sensitive information in Policy-Governed MAR-Apps. We show in Figure 5.2 the proper operation of a Policy-Governed MAR-App that respects Users' privacy:

(1) Write Policies: A Policy Maker, i.e., a User or Space Owner, establishes the applicable regulations over a MAR-App. For example, a Space Owner can determine that Foxes are unwanted within the specified sensitive space.

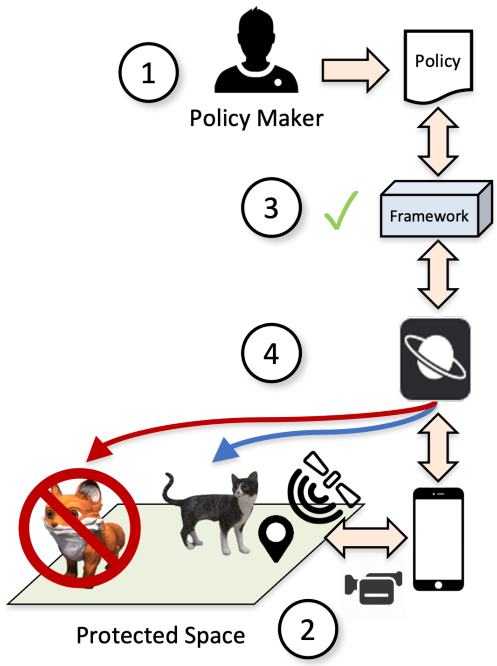


Figure 5.2: Policy-Governed MAR-App.

(2) Operation: When the MAR-App is up and running, it shall assemble relevant security information, e.g., location, usernames, device manufacturer, etc. To ensure the Users' privacy, they must be aware of all the data collected from them and have the authority to prevent the MAR-App from gathering some of it.

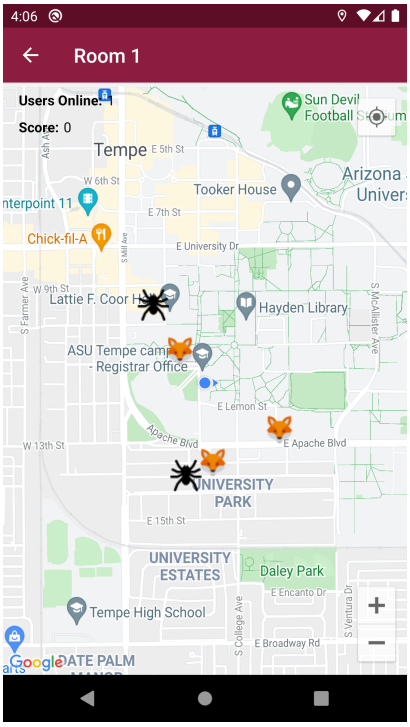
(3) Evaluation: The policies supplied to the Providers shall be evaluated whenever applicable. Such process involves the relevant security information recollected to reflect the Entity requesting access, and an authorization decision shall be taken.

(4) Enforce Regulations: Through the implementation of access control, a Policy-Governed MAR-App shall only allow proper functionalities. For example, only authorized MAR objects must be distributed in a sensitive space.

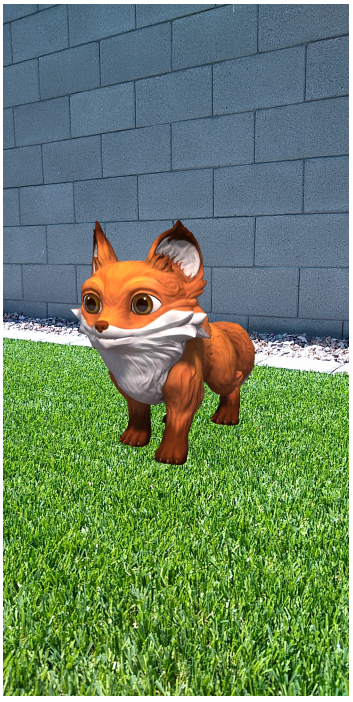
Chapter 6

IMPLEMENTATION

This Chapter covers the implementation of the Policy-Governed MAR-App concepts from Chapter 5 into `SpaceMediator`. This *proof-of-concept* application provides users with a safer environment by preventing the issues covered in Chapter 4. Specifically, it explains the architecture used for `SpaceMediator` in Chapter 6.1, regulation of sensitive spaces in Chapter 6.2, restrictions over user interaction in Chapter 6.3, how users' privacy is respected in Chapter 6.4, and the usability of `SpaceMediator` in Chapter 6.5.



(a) Distributed MAR Objects.



(b) Interaction with MAR Objects.

Figure 6.1: `SpaceMediator` Geolocation-Based MAR-App

6.1 Client-Server Architecture

`SpaceMediator` is a MAR-App that imitates Pokémon GO, the most popular MAR-App as denoted by the number of downloads up to March 2021 (Table 4.1). It is developed in Android and implements Augmented Reality through Google's library *ARCore* [34]. It replicates the execution of geolocation-based MAR-Apps by assigning specific coordinates to MAR objects. The content of such MAR objects is limited to Foxes and Spiders, shown in Figure 6.1(b), since it is a *proof-of-concept* MAR-App. However, it would be easy to extend the MAR content to cover other types of MAR content, e.g., robots, dinosaurs, etc. Similar to the interaction of Pokémon GO, users move around different locations to capture the available MAR objects and score some points.

It is a common practice to offload tasks into servers or clouds to improve the performance of mobile devices [35]. Since we observe such technique, as shown in Figure 6.2, we must look into the Client-Server Architecture to comprehend the Policy-Governed MAR-App conducted by `SpaceMediator`, two Servers, a Database, and an API.

1. **SpaceMediator:** It is a Policy Governed MAR-App in which Users can interact with MAR around in a regulated manner. Therefore, they can create policies to protect sensitive spaces and restrict User-to-User interaction and procedures to safeguard the sensible space.

2. **Game-Engine:** This server is responsible for handling the game functionalities that imitated Pokémon GO and implements the *Socket.IO* library for real-time communication among users. Within the CM explained in Chapter5, this server represents the Providers that distribute MAR objects.

3. **Query:** This server executed tasks that involved database information, e.g.,

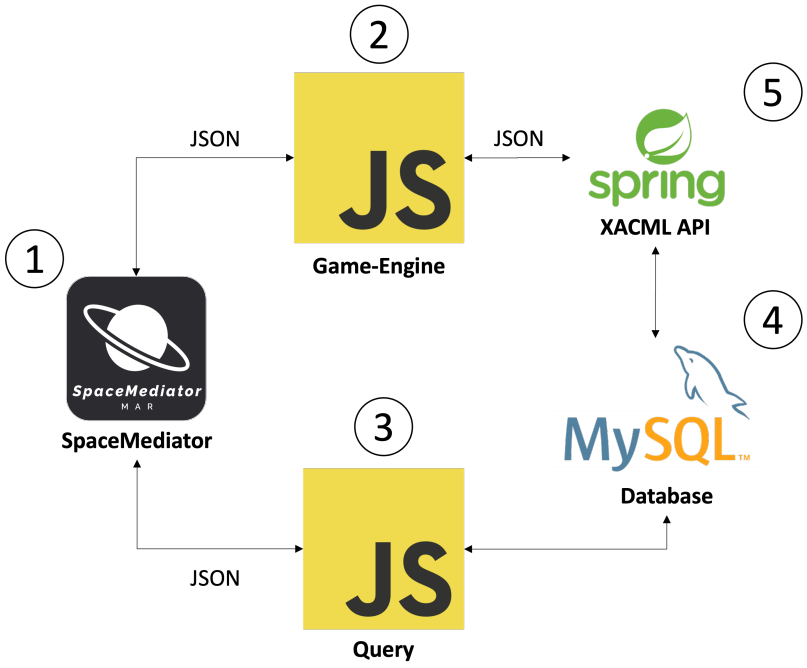


Figure 6.2: `SpaceMediator` Architecture.

request specific data, insert new data, delete existing material, and more.

4. **Database:** Collection of relevant information used throughout `SpaceMediator`, e.g., usernames, space policies, user interaction policies, available MAR objects, available rooms, etc.

5. **Authorization API:** It is responsible for making authorization decisions through the implementation of an XACML framework build by AT&T [36]. This is done by evaluating received requests against applicable policies. The API only provides two outputs in terms of authorization: *Permit* or *Deny*.

Along with letting users interact with MAR objects, `SpaceMediator` also has user-friendly interfaces to create policies, as shown in Figure 6.11. This would use, for example, the services offered by the Query Service and protect a sensitive space.

6.2 Regulate Sensitive Spaces

As previously explained in Chapter 4.1, the sensitive spaces are areas exposed to Space Invasion attacks by mishandled MAR. Therefore, it is necessary to offer Space Owner, the entities in charge of such sites, the possibility to regulate the operation of the MAR-Apps within such locations. As a result, we implement through `SpaceMediator` the *Space Owner-Provider-User* Mode of Interaction, described in Chapter 5. They are capable of enforcing restrictions over their sensitive spaces through a three-step process, shown in Figure 6.3:

(1) Policy Creation: Space Owners write a policy specifying how `SpaceMediator` shall operate within the claimed protected sensitive space. For example, the ban of MAR Foxes in a specified area.

(2) Sensitive Space Entry: As a geolocation-based MAR-App, `SpaceMediator` is aware of the location of its users and deployed MAR objects. It is necessary to detect if any of these are within a sensitive space, to enforce possible established regulations.

(3) Protect Sensitive Space: `SpaceMediator` evaluates applicable policies against the users or new MAR objects within the sensitive space. Therefore, only allowing authorized MAR content and interaction through SSAC. For example, a Provider trying to deploy a Fox MAR object must be denied if such content has been forbidden within a regulated sensitive space.

6.2.1 Policy Creation

Space Owners write policies to establish how they want to regulate MAR-App operations over their claimed sensitive space. This process starts with Space Owners specifying their claimed area via geographical points, where the policy will go into

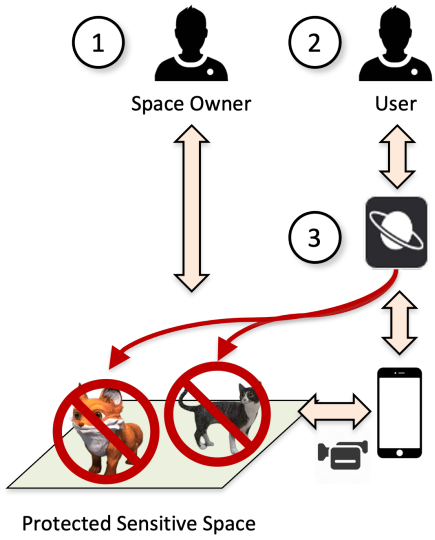
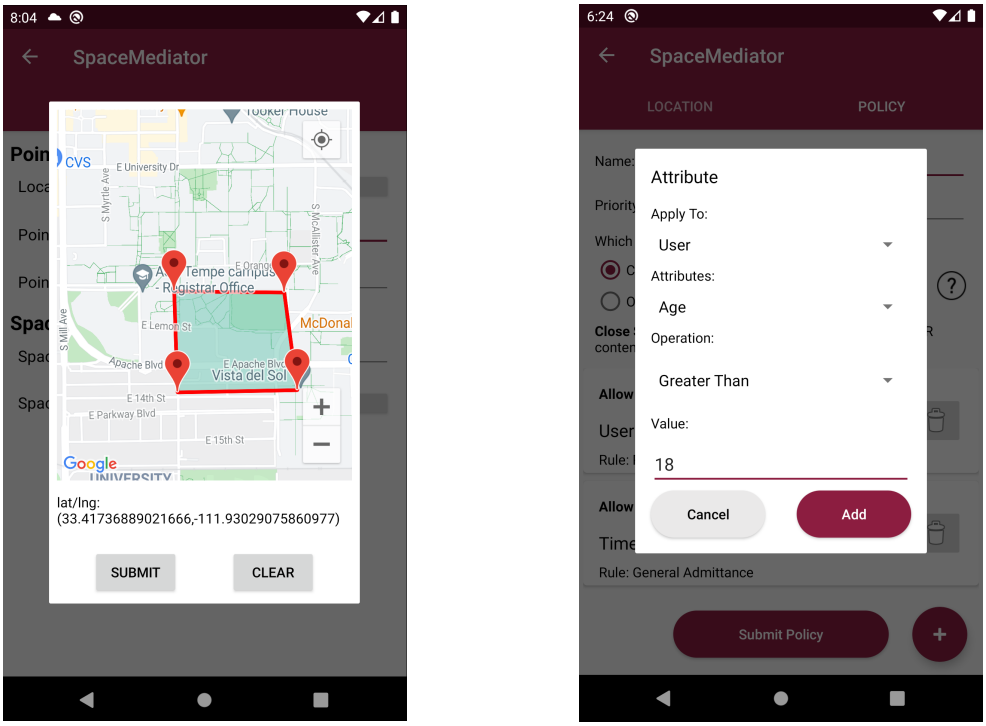


Figure 6.3: Protected Sensitive Space.

effect, as shown in Figure 6.4(a). Then, they select the regulation type they want to implement in the policy, which will decide the policy's combining algorithm and rules' effect (Chapter 2). There are two regulation types offered in `SpaceMediator` to offer a wider variety of possible policies. First, the *Open Space*, designed for Space Owners with low restrictive parameters, results in a *permit-unless-deny* policy with Deny rules. Second, the *Close Space* facilitates high restrictive constraints, resulting in a *deny-unless-permit* policy with Permit rules. Finally, as displayed in Figure 6.4(b), they specify the attributes they want as part of the policy, e.g., Age $>$ 18, Username $=$ $User_1$, Time $\leq$ 12:00:00, etc. It is important to point out that whether such attributes are permitted or denied depends on the selected regulation type. Therefore, let us look into each of the mentioned regulation types, i.e., policy structure and applicable rules, to apprehend how they would restrain a sensitive space.



(a) Specify Sensitive Space.

(b) Constraint for Sensitive Space.

Figure 6.4: Policy Creation for Sensitive Space.

Open Space

This regulation consists of two policies used for different purposes. Selecting an Open Space implies a predefined structure for these policies. As shown in Figure 6.5, both have a *permit-unless-deny* combining algorithm, rules with Deny permission, and attributes appended by *OR* logical operators. As a result, met statements in the policy result in a Deny authorization when evaluated, otherwise in a Permit. Therefore, Space Owners just define reject parameters through an Open Space.

- **MAR Distribution Policy:** In `SpaceMediator`, this policy describes how to handle MAR content within a sensitive space. It is used throughout MAR object distribution from the Provider, protecting sensitive spaces.

- **MAR Interaction Policy:** In `SpaceMediator`, this policy controls users' in-

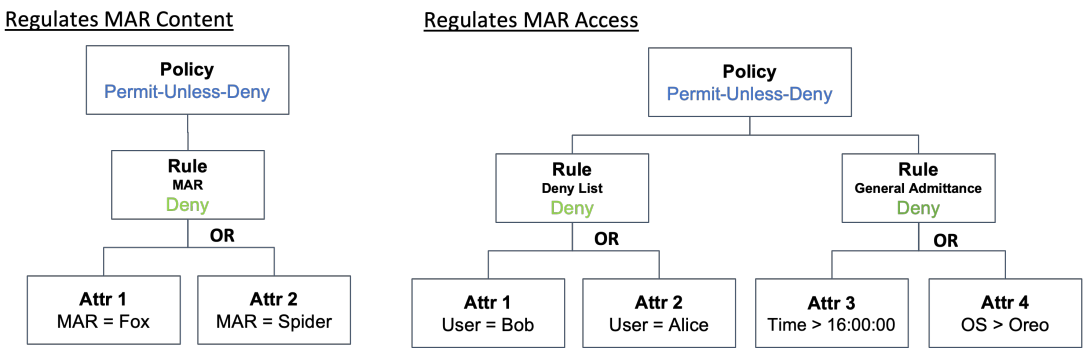


Figure 6.5: Open Policies.

teraction with available MAR objects within a sensitive space. There are two rules applicable to users.

- **Deny List Rule:** This rule consists of unauthorized usernames that shall not interact with the available MAR objects within the sensitive space.
- **General Admittance Rule:** This rule considers attributes that apply to all users, e.g., Time, OS Version, Device Manufacturer, etc. Those who meet any of the specified conditions in this rule shall be unauthorized.

Close Space

The overall structure of Close Spaces is very similar to the one implemented in Open Spaces, as shown in Figure 6.6. It also consists of two policies, but the resulting limitations are different as it utilizes a *deny-unless-permit* combining algorithm, rules have Permit permission, and the General Admittance Rule has its attributes appended by *AND* the logical operators. Therefore, these are more restrictive as policy statements are used are requirements for authorization, and failing to meet them results in denial. Although, the two policies within a Close Space hold the same purpose as in an Open Space.

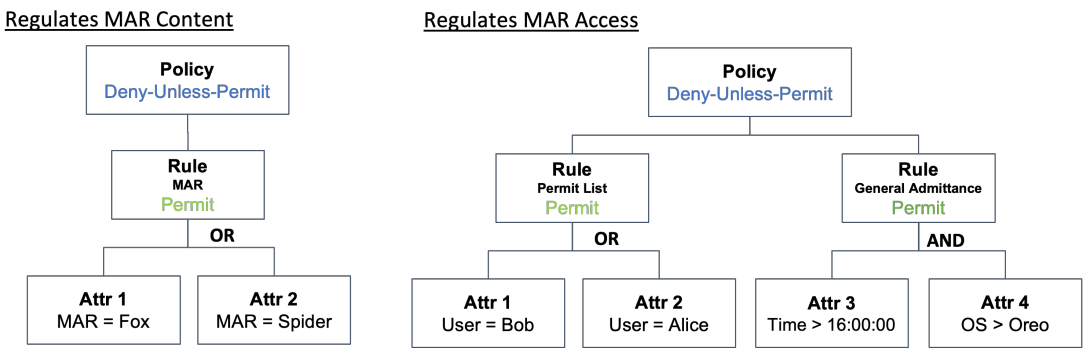


Figure 6.6: Close Policies.

6.3 Regulate Users Interaction

As previously mentioned in Chapter 4.2, malicious parties have compromised the security of users by exploiting the known locations of MAR objects in multi-user geolocation-based MAR-Apps. Thus, to possibly prevent such scenarios and Space Affectation overall, we implement in `SpaceMediator` the *User-Provider-User* Mode of Interaction, described in Chapter 5. As a result, MAR objects distribution among users is done through *Rooms*, isolated and regulated MAR environments for users to join.

6.3.1 Rooms

Regulating multi-user interaction brings alongside the challenge of assembling an easy-to-use process for users. Because of this, `SpaceMediator` implements such regulations through Rooms, an extra layer to protect users from Space Affectation. Through them, users are separated into different groups, they decide whom to interact with, access is restrained, and each Room is provided with unique MAR objects for interaction. In addition, as shown in Figure 6.7, `SpaceMediator` offers a Lobby that displays the available Rooms and applicable constraints. At the Lobby, users select a Room to join or create a new one, implementing their desired user interaction

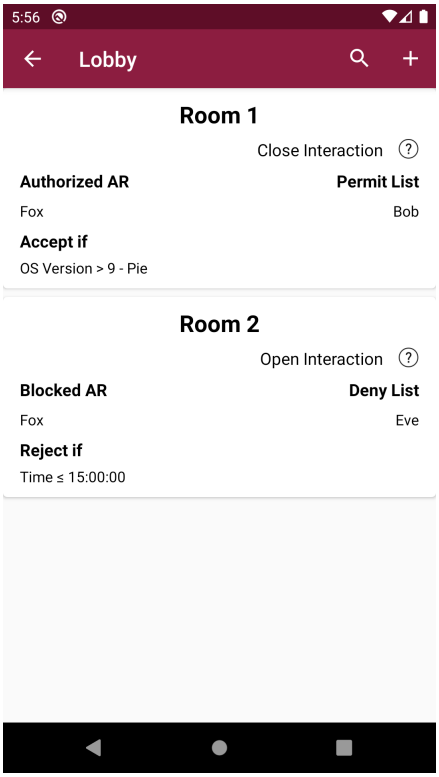
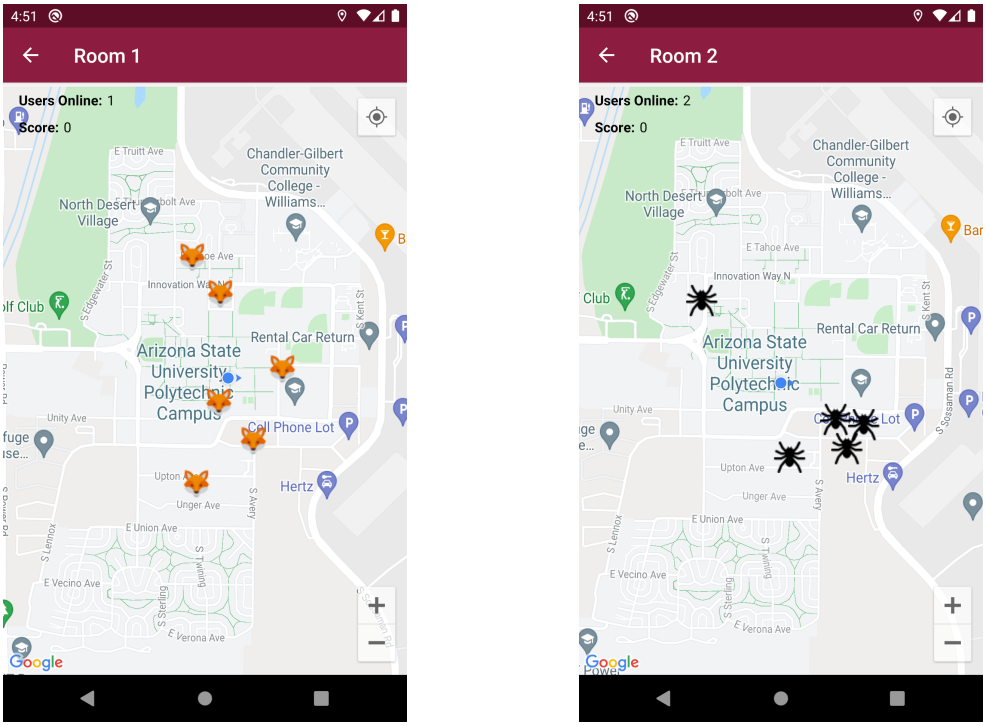


Figure 6.7: Lobby Displaying Available Rooms.

regulations. Afterward, users enter a Room and operate `SpaceMediator` as a regular geolocation-based MAR-App, go around capturing available MAR objects.

Rooms are isolated as a user can only be in one Room at a time, and the MAR objects provided to each Room are distinct. Also, they are regulated as they have admission requirements for users, and it filters the content of MAR objects to avoid undesirable ones. As shown in Figure 6.8, the two Rooms available in the Lobby are provided with their respective MAR objects for interaction, and the location and content of such MAR objects vary since Room 1 only allows foxes while Room 2 rejects them. There could be several users within a Room, but only one with the role of *HOST* establishes the applicable policies. Rooms implementation through `SpaceMediator` is reflected in Figure 6.9, and can be outlined in the following three steps:



(a) Room 1. (b) Room 2.

Figure 6.8: Distribute MAR Objects per Room

(1) Join Room: In `SpaceMediator`, users find available Rooms through a Lobby. There they try joining an existing Room, regulated by the HOST's policies. For example, a Room could only allow underage players. Users could also create new rooms if unable to join any. Overall, multi-user interaction is allowed only with authorized participants.

(2) Room Interaction: Users request new MAR objects for interaction within the Room. For example, such demands are generated as they move to distinct locations where no MAR objects are available. In general, `SpaceMediator` submits automated requests to the Provider for new MAR objects and keeps its users entertained.

(3) Regulated MAR: If the new content request is authorized, the Provider distributes new MAR objects to the users within the Room.

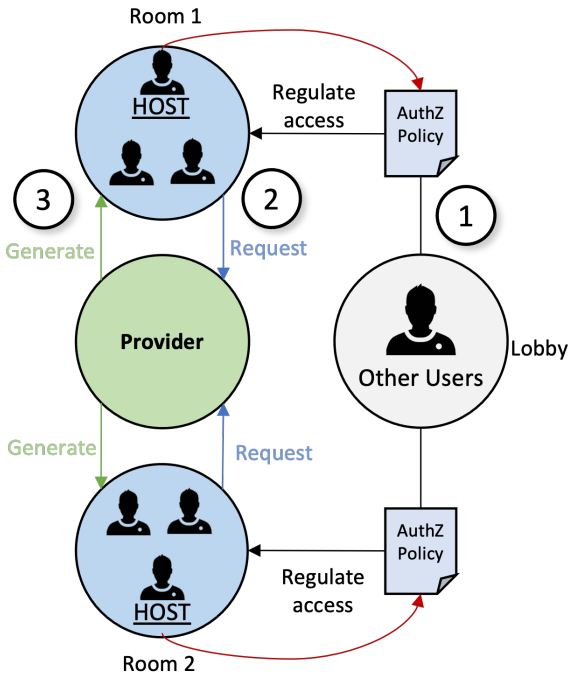


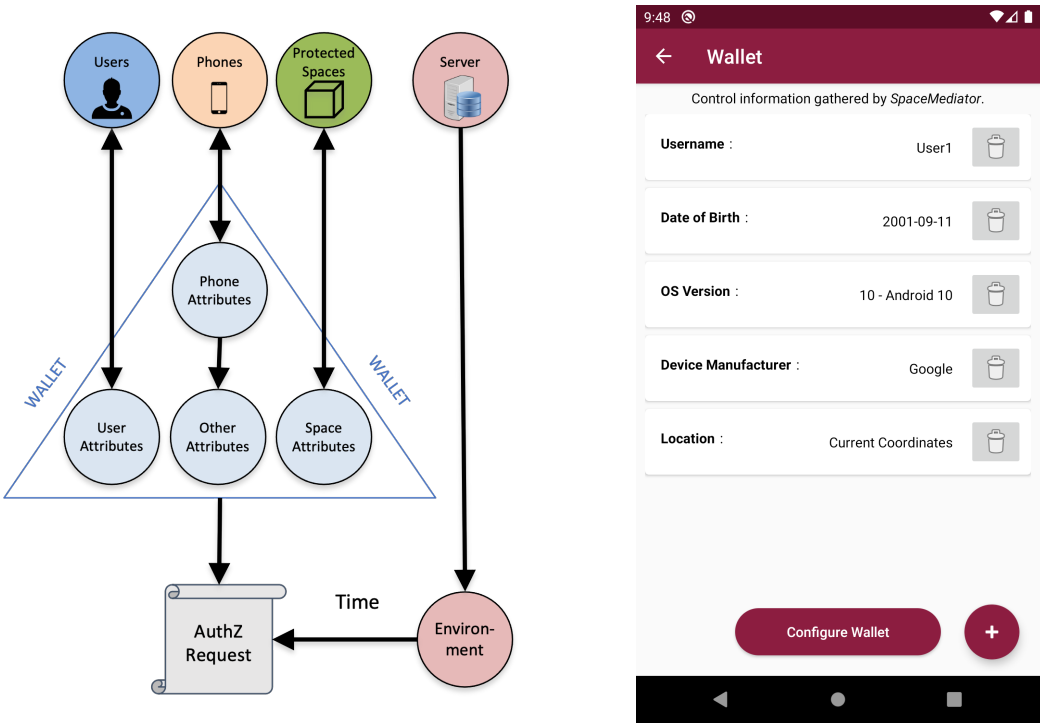
Figure 6.9: Regulated User Interaction.

6.3.2 Policy Creation

In `SpaceMediator`, the user assigned the *HOST* role is in charge of a Room's policy. This role is automatically appointed to whoever created a Room, and it is reassigned in a First-In-First-Out order if the HOST leaves. Thus, users efficiently implement their desired regulations by creating a Room in `SpaceMediator`'s Lobby. Besides, there is no limitation on having one HOST per Room as creating a new Room is a simple process.

`SpaceMediator` offers two regulation types for User Interaction: *Open Interaction* and *Close Interaction*, which define the structure of the policies. Overall, these structures' design is the same as those used for Open and Close Spaces. Indeed, the Open Interaction structure is shown in Figure 6.5, and the Close Interaction format is displayed Figure 6.6. They also apply the same combining algorithms, rules' permission effect and relations among attributes. Although, the policies had a different purpose.

- **MAR Distribution Policy:** In `SpaceMediator`, the Provider evaluates this policy when distributing MAR objects to a Room, omitting intrusive MAR content that degrades the HOTS experience.
- **MAR Interaction Policy:** `SpaceMediator` this policy evaluates users who want to join a Room. Thus, only authorized personnel by the HOST may enter and view available MAR objects.



(a) Access Request.

(b) Attribute Wallet.

Figure 6.10: Respecting Users Privacy.

6.4 Respecting Privacy

MAR-Apps are also vulnerable to Privacy Leak issues when gathering data from users, as explained in Chapter 4.3. Likewise, `SpaceMediator` collects data from its users, for example, as they move around to interact with MAR objects. Furthermore,

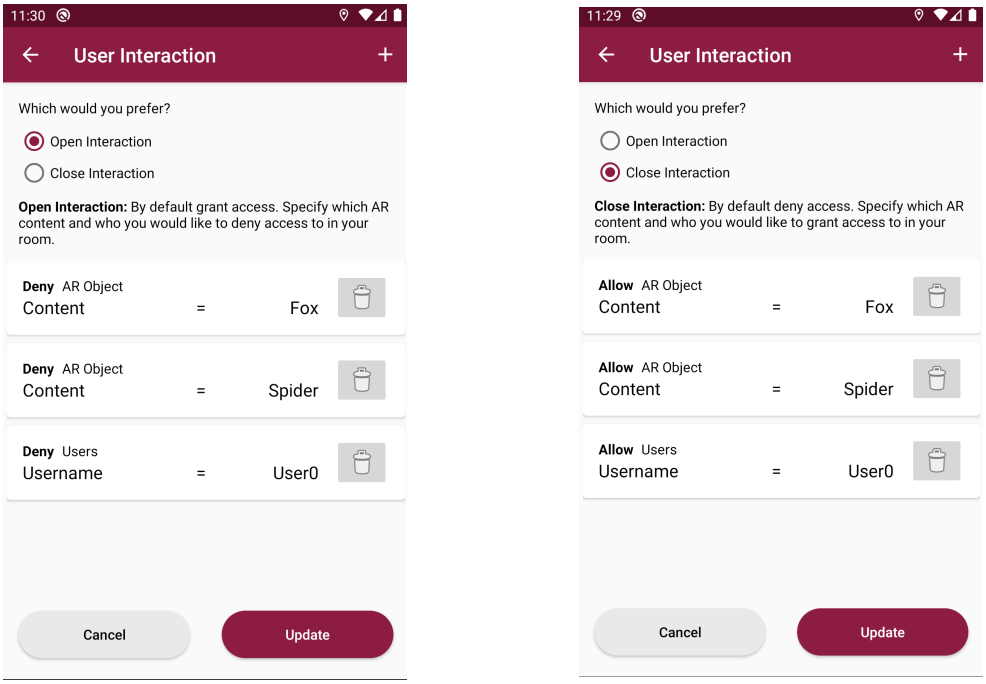
information is retrieved from users to create an *access request*, which contains valuable facts for authorization decisions as it is evaluated against a policy, as described in Chapter 2. Therefore, to respect users' privacy while enforcing regulations, we implement in `SpaceMediator` an *Attribute Wallet*.

Through it, users are aware of any information gathered from them. Indeed, as shown in Figure 6.10 almost all data used throughout `SpaceMediator` is within the Attribute Wallet's scope. Most of it is utilized for authorization purposes and represents attributes, e.g., birth date, device manufacturer, current geographical coordinates, etc. The Attribute Wallet also allows users to stop `SpaceMediator` from collecting sensitive information they do not want to provide. As shown in Figure 6.10, they are aware of gathered information and can manage it. Although, there is data outside the Attribute Wallet's range as it is appended at the servers, i.e., time. Users' privacy is respected in `SpaceMediator`, as there is clarity over the compiled information, and users can control it.

6.5 Policy Writing and Understanding

As previously explained in Chapters 6.2.1 and 6.3.2, `SpaceMediator` offers different regulations types, Close and Open to Space Owners and users. Alongside, they could choose among several available attributes, displayed in Table 6.1. Furthermore, *Other* attributes give them the possibility to specify *name* and *value* of the desired attribute. Prevailing, Space Owners and users need to understand the effect of the specified attributes to comprehend their crafted regulations.

Thus, as shown in Figure 6.11, `SpaceMediator` provides in its graphical user interface (GUI) for writing policies a description of the applicable regulation and its consequence on the attributes. Precisely, Figure 6.11(a) displays an Open policy where attributes are marked as Denied, and Figure 6.11(b) applies Close regulation



(a) Open Policy.

(b) Close Policy.

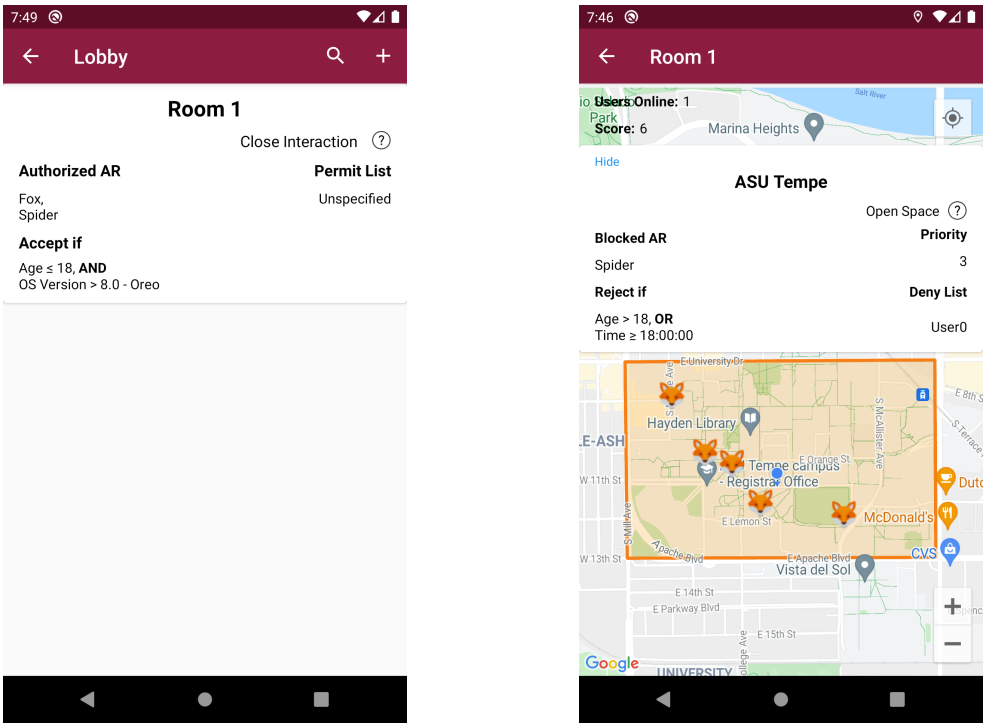
Figure 6.11: Policy Description Open and Close.

leading to Allowed attributes. Noticeably, the attributes in the dynamic GUI display other relevant information, e.g., ID, category, value, operator, etc.

Since `SpaceMediator` is a *proof-of-concept* MAR-App, ordinary players are presented with applicable regulations. As shown in Figure 6.12, a description of how a policy regulates a Room and a sensitive space is offered. Thus, users can understand why they are limited if rejected. Alongside, they can be aware of a requirement of providing sensitive information for access and avoid providing it through the Attribute Wallet if against their will.

Policies	Rules	Attributes	Operations
MAR Distribution	MAR	Content	$=$
MAR Interaction	Permit/Deny List	Username	$=$
	General Admittance	Age	$=, >, \geq, <, \leq$
		OS Version	$=, >, \geq, <, \leq$
		Device Manufacturer	$=$
		Time	$\geq, \leq$
		Other	$=$

Table 6.1: `SpaceMediator` Regulations.



(a) Regulated User Interaction.

(b) Protected Sensitive Space.

Figure 6.12: Policies in User Interface.

Chapter 7

EVALUATION AND RESULTS

This chapter evaluates a conducted user study formally approved by Arizona State University's Institutional Review Board (IRB) office, which oversees research integrity and ethics. Through this study, participants utilized our Policy-Governed MAR-App `SpaceMediator`. The chapter starts with a general overview of the user study in Chapter 7.1, covering our objectives, implementation methods, and evaluation techniques. It concludes by presenting the results in Chapter 7.2, along with possible explanations.

7.1 User Study

As previously discussed in Chapter4, several MAR-Apps are available across the different mobile operating systems with millions of downloads, growing popularity, and vulnerability to space and privacy attacks. In our approach to prevent these attacks, we allow users to regulate the functionality of a MAR-App. It intends to be helpful to all users, regardless of their prior knowledge, e.g., access control, computing, etc. To verify the feasibility of our approach, we conducted a user study involving seven *research questions* (RQ):

RQ1. Can participants understand the concepts of space invasion, space affectation, and privacy leak attacks?

RQ2. Can participants identify security issues, with respect to the three attacks just mentioned?

RQ3. Can participants write effective Space Protection Policies?

RQ4. Can participants write effective User Interaction Policies?

RQ5. Can participants understand the policies to counteract space attacks?

RQ6. Can participants utilize `SpaceMediator`'s attribute wallet properly?

RQ7. Do participants agree with the regulation of MAR-Apps?

For the user study, we recruited 40 participants through advertisements placed throughout the university campus. Furthermore, we focused on having participants with distributed background knowledge to identify if prior familiarity with computing was necessary to properly utilize a Policy-Governed MAR-App. As a result, half of the participants identified as having a background in Computer Science (CS). In contrast, the other half pursued degrees in different fields (Non-CS), e.g., engineering, arts, business, etc.

7.1.1 Procedure

The user study was conducted in timeframe group sessions with an average of 60 min. Through them, we gathered data from participants anonymously to evaluate the efficiency of the proposed methodology to regulate MAR-Apps to prevent space and privacy attacks. In addition, as participation was voluntary and we appreciated their collaboration, each participant received a $20 Amazon gift card by the end of each session.

The procedure implemented in each group session throughout the user study consisted of three phases: introduction, MAR-App interaction, and a questionnaire. Through them, we assured participants had a basic knowledge on relevant topics,

ID	Policy Description	Regulation
1	At ASU Tempe campus, block spiders MAR content, and deny Eve or anyone with an OS less than Android Pie.	Open Space
2	At ASU Brickyard, allow spiders MAR content and grant access to adults only after 6:00 p.m. or Bob.	Close Space
3	Within the room, allow MAR content of foxes and grant access to ASU students.	Close Interaction
4	Deny access to Eve or anyone else who has a Samsung device, is underage, or has an OS version less than Android 10.	Open Interaction

Table 7.1: User Study Policy Exercises.

used `SpaceMediator` when ready, and finalized by gathering feedback, all within a reasonable timeframe to maintain focus.

Introduction

In this first phase of the user study, within 15 minutes, we explained our project's scope to participants. This covered topics such as the current status of MAR-Apps, security issues triggered by MAR-Apps (Chapter 2.2), vulnerabilities on MAR-Apps (Chapter 4), our approach to preventing such vulnerabilities (Chapter 6), etc. We covered the topics gently for understandability regardless of familiarity with cybersecurity. By the end of the introduction, we wanted participants to understand MAR, its vulnerabilities, and the regulations implemented in `SpaceMediator`.

MAR-App Interaction

Once participants were familiar with the purpose of our project and essential topics covered within it, we allowed them to use `SpaceMediator`, our MAR-App with regulated functionality. Using `SpaceMediator`, they followed a set of predefined exercises to write four policies. Table 7.1 shows the English-written policy descriptions provided to the participants. Through such descriptions, we specified them the authorized or unauthorized entities.

As a result, each participant wrote two policies to prevent space invasion as Space Owners of a specified location and two to avoid space affectation by regulating user interaction in a room. The crafted policies were associated with an account given to each participant, stored in a database, and analyzed afterward. In addition, provided supplementary material for the first exercise of each category offered a quick review of the topics covered in the introduction phase, i.e., graphs of policy structure.

With `SpaceMediator` installed on four different devices, participants could complete these exercises in an average of 30 min. We utilized two Google Pixel 3XL with Android 11 and 4 GB of RAM, a Samsung S9 with Android 10 and 4 GB of RAM, and a Motorola G6 with Android Pie and 2 GB of RAM.

Questionnaire

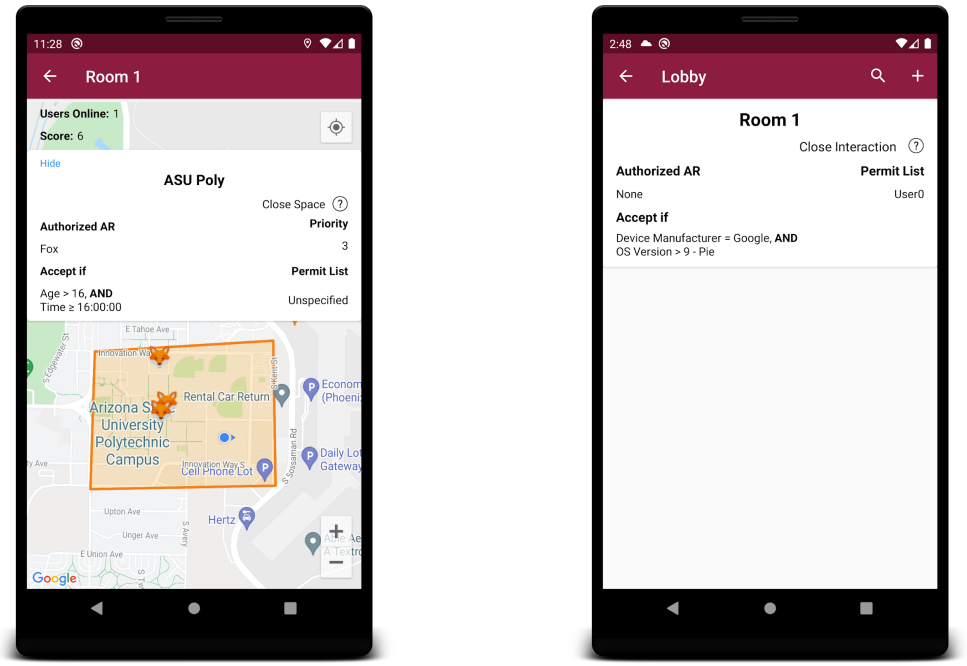
To conclude a user study session, participants answered a questionnaire with relevant inquiries to reflect their understanding of covered topics and provide feedback. We gathered this data through an adequately structured online questionnaire divided into four sections, completed in an average of 15 minutes. Content used throughout the questionnaire is available in Appendix A.

The first section, *scenario recognition*, consisted of five scenarios with different

Policy Description	Answer
At ASU Polytechnic campus allow foxes and deny interaction with users who are over 16 years of age after 4:00 p.m.	-
At ASU Polytechnic campus allow foxes and authorize interaction with users who are over 16 years of age after 4:00 p.m.	✓
At ASU Polytechnic campus allow foxes and deny interaction with users who are less than 16 years of age after 4:00 p.m.	-
At ASU Polytechnic campus allow foxes and authorize interaction with users who are over 16 years of age before 4:00 p.m.	-

Table 7.2: Questionnaire Policy Making - Descriptions for Sensitive Space.

security issues; and participants had to identify the undergoing attacks. Next, the policy-making section consisted of two types of questions involving `SpaceMediator`'s GUI. First, *policy-making description* through which participants associated a displayed policy, as Figure 7.1(a), with its proper description, as shown in Table 7.2. Second, the *policy-making attribute wallet* consisted of selecting the attributes required to gain access over a stated policy, as Figure 7.1(b), while protecting their privacy. Subsequently, participants provided a scale representation, ranged 1 to 5, to reflect comprehension of the security topics throughout the *policy understanding* section. Finally, they let us know their agreement on MAR-Apps regulations by the *exit* section.



(a) Policy for Sensitive Space.

(b) Policy for User Interaction.

Figure 7.1: Questionnaire Policies Displayed.

7.1.2 Policy Evaluation

By following the English-written policy descriptions present in Table 7.1 and using `SpaceMediator`, each participant created a total of four policies to regulate MAR and prevent space attacks. These policies had specific regulation goals stated in the descriptions, i.e., specifying the regulation type and applicable attributes. To evaluate if a policy was written correctly, we evaluated it against a request sequence that tested how authorization was handled over expected entities. For further clarity, let us consider the following example in which Exercise 1 states requirements to block three attributes (MAR spiders, user Eve, OS Pie) in the following way:

> At ASU Tempe campus, *block* spiders MAR content, and *deny* Eve or anyone with an OS less than Android Pie.

In `SpaceMediator`, users selected the regulation type for the policy, i.e., open or

Attribute	Request 1	Request 2	Request 3
MAR Content	Spider	-	-
Username	-	Eve	Alice
OS Version	-	Android 10	Oreo

Table 7.3: Access Requests to Evaluate `SpaceMediator` Testing Exercise 1.

close, and added relevant attributes. The side effects of the regulation type were adequately reflected on the GUI, as discussed in Chapter 6.5. Nonetheless, participants could miswrite the policy, e.g., incorrect regulation type, miss relevant attributes, etc. By evaluating each policy against a sequence of requests containing essential details, as Table 7.3 for Exercise 1, we assessed if a policy managed authorization properly. These policy-request evaluations were conducted through an automated process using the same API implemented in `SpaceMediator` and described in Chapter 6. Furthermore, policy syntax was also reviewed manually to verify each request's Permit/Deny results. Finally, we followed an evaluation scheme to categorize a policy as: *ideal*, carried out all expected regulations; *permissive*, vulnerable to security problem; *restrictive*, compromised functionality.

7.2 Results

As previously described in Chapter 7.1, participants were evenly distributed in terms of background-field, CS vs. Non-CS. However, we also worked with a population with distinct educational ranks since 22.50% recognized high school as their highest level, 42.50% had concluded an undergraduate major, and 35.00% had achieved a graduate degree. Also, they identified different experience levels of familiarity with MAR as 65.0% had no prior knowledge, 32.5% held medium experience, and only 2.5% rated it as well known. As a result, we worked with a diverse population, gathered

helpful information, and further analyzed it, ultimately focusing on answering the RQs.

RQ1. Can participants understand the concepts of space invasion, space affectation, and privacy leak attacks? To address RQ1, we performed the questionnaire's policy understanding described in Chapter 7.1. The results are shown in Figure 7.2, with an average on each security issue per background-field. Overall, participants successfully comprehended the issues described throughout the user study, as they provided good ratings reflecting it. However, space affectation had the lowest ranking with 3.55 within the CS, 4.30 among the Non-CS, and a prevailing norm of 3.93. On the other hand, space invasion had better ratings with 4.30 within the CS, 4.60 in the Non-CS, and an average of 4.45. Finally, privacy leak was the best-understood security issue with 4.70 for CS, 4.60 for Non-CS, and a standard of 4.65.

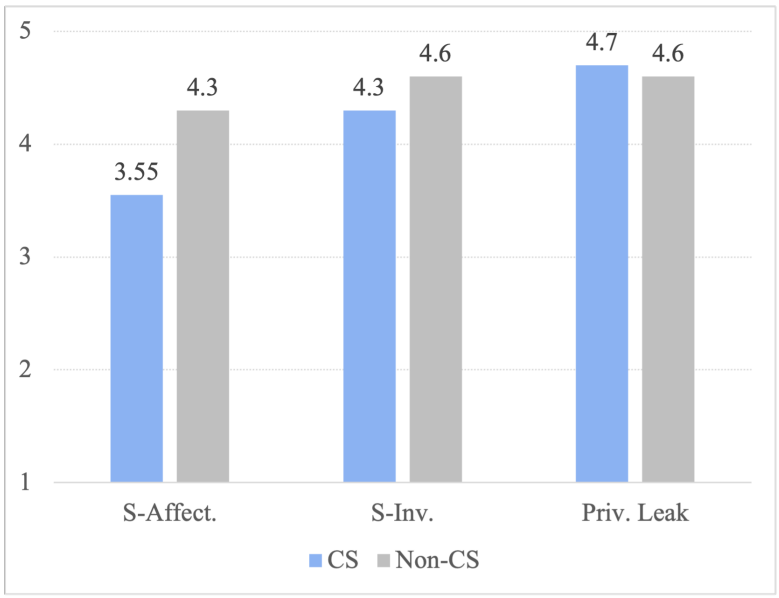


Figure 7.2: Comprehension of Security Issues.

RQ2. Can participants identify security issues, with respect to the three attacks just mentioned? To manage RQ2, we conducted the questionnaire's

scenario recognition. The outcomes are shown in Figure 7.3. Privacy Leak was the most recognizable security issue, with 92.50% of participants identifying such problem in the expected scenario. Afterward, space invasion had a distinction rate of 82.50%, followed by space affectation with 72.50%. Also, an uncompromised scenario with no undergoing attacks was identified by 65.00% of participants. Finally, with a 60.00% success rate, participants recognized simultaneous space invasion and space affectation attacks.

Noticeably, the understandability reflected in RQ1 goes along with the identifiability success rates in RQ2. For example, privacy leak was the most understandable security issue by participants in RQ1, and at the same time, it had the highest identifiability success in RQ2. Furthermore, the exact trails apply to space invasion and space affectation in second and third places. Therefore, we can notice consistency over the user study data reflecting comprehension over security issues.

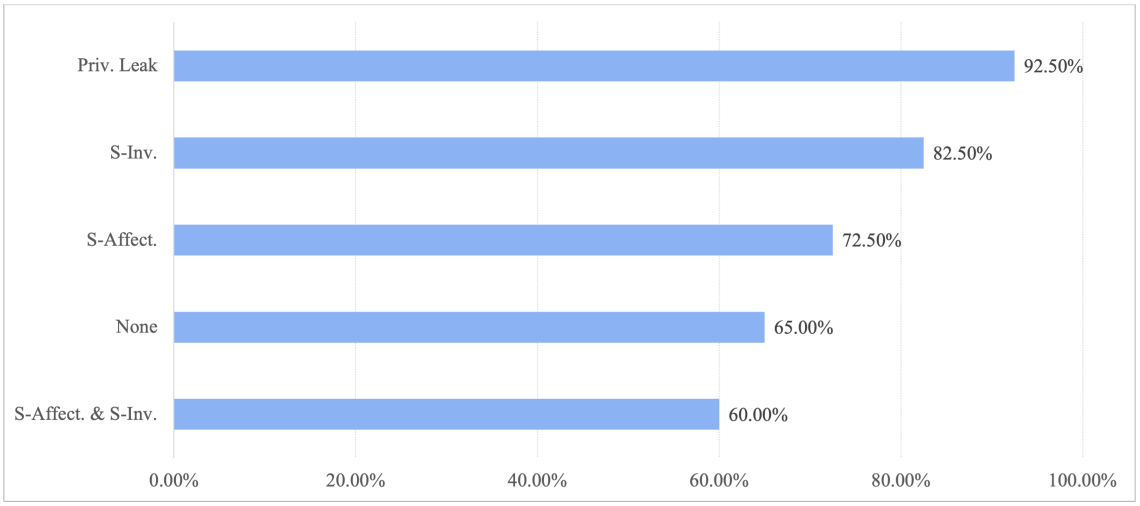


Figure 7.3: Detection of Security Issues.

RQ3. Can participants write effective Space Protection Policies? To answer RQ3, we evaluated the policies participants wrote as Space Owners throughout the MAR-App interaction via a procedure described in Chapter 7.1.2. The results

are displayed in Figure 7.4, with the results from Table's 7.1 Exercises 1-2. Overall, 55.00% of the policies were ideal as they effectively regulated a sensitive space, preventing a space invasion attack. The remaining set of improperly written policies contained different types of errors. For example, most of the incorrect policies for introductory Exercise 1 were restrictive with 30.00%, and the remaining 15.00% were permissive; on the other hand, the more challenging Exercise 2 had the opposite results with 30.00% permissive and 15.00% restrictive.

It is noticeable that throughout both Exercises 1 and 2, Non-CS participants had a higher success rate since at least 50.00% of them wrote ideal policies. As a result, the hypothesis arises that a coherent, friendly, and easy-to-use GUI is critical in writing effective policies. Although, further research is necessary to verify such a statement.

RQ4. Can participants write effective User Interaction Policies? We followed the same procedure for RQ4 as in RQ3. Therefore, results are also shown in Figure 7.4, but with results from Table's 7.1 Exercises 3-4. Interestingly, the success rate of ideal policies was higher for user interaction, with 70.00% in introductory Exercise 3 and 65.00% in the more demanding Exercise 4. Although, there were still unsuccessful policies in terms of regulations. In Exercise 3, the mistaken policies had 15.00% for both permissive and restrictive; meanwhile, Exercise 4 had results of 22.50% permissive and 12.50% restrictive.

The higher success rate on Exercises 3-4 may be related to increased familiarity with `SpaceMediator`. By the time participants reached these exercises, they had written the space protection policies from Exercises 1-2. Therefore, they likely had a better understanding of how to operate `SpaceMediator`. As a result, we consider the importance of a step-by-step guide to ensure the GUI offered to write policies to regulate MAR-Apps is well understood. Although, more research is necessary to confirm this idea.

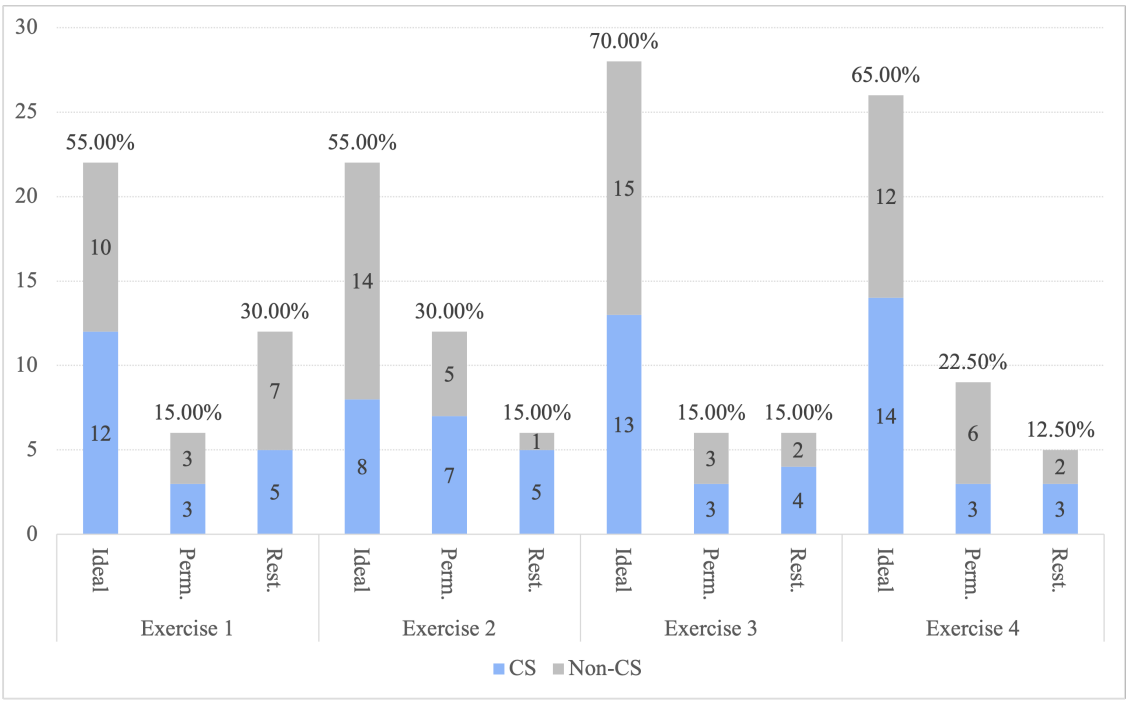


Figure 7.4: Performance in User Study Policy Writing.

RQ5. Can participants understand the policies to counteract space attacks? To handle RQ5, we performed the questionnaire's policy making, described in Chapter 7.1. In general, participants performed pretty well throughout these exercises. For example, the space regulation policy displayed in `SpaceMediator` GUI was associated with its appropriate description by 87.50% of the participants. In contrast, the user regulation policy had a lower success rate with 75.00%. Still, these are satisfactory results as they reflect comprehension by the majority of the population over the regulations implemented in a MAR-App.

It is possible the long and complex description used through the questionnaire's policy making confused participants. Therefore, breaking such depictions into multiple easy-to-read questions could improve the outcomes. But, of course, further research is necessary to understand the requirements for better results.

RQ6. Can participants utilize `SpaceMediator`'s attribute wallet prop-

erly? To address RQ6, we conducted the questionnaire's policy making - attribute wallet. The results are shown in Figure 7.5. In the first question, which consisted of two details, i.e., username and SSN, 92.50% of the participants successfully selected necessary features for proper policy evaluation as one rule could be satisfied. Concurrently, 37.50% of them provided additional unnecessary information for the policy, e.g., date of birth, device manufacturer, OS version, etc. The results were similar throughout the second question in terms of access with 95.00%. Although, there was better awareness for privacy as only 18.42% of such participants supplied unneeded traits. Overall, a significant portion of participants provided only the necessary attributes. It is an excellent first step towards evaluating how an attribute wallet would respect users' privacy without compromising the functionality of MAR-Apps.

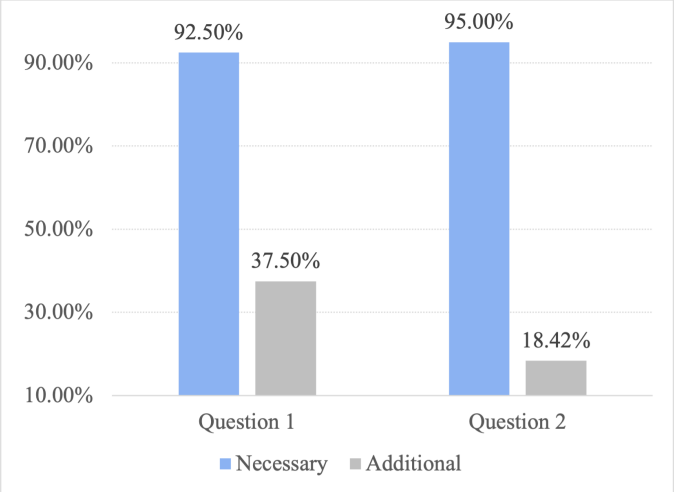


Figure 7.5: Performance in Privacy.

RQ7. Do participants agree with the regulation of MAR-Apps? To manage RQ7, we performed questionnaire's exit described in Chapter 7.1 and the outcomes are displayed in Figure 7.6. We found out that 87.50% of participants agreed that businesses and institutions should be able to regulate MAR-Apps, 7.50% were uncertain, and 5.00% were against it. Similarly, 82.50% of participants would

regulate MAR-Apps if possible, 15.00% would consider it, and only 2.50% discarded it. Overall, there is high interest in the MAR-Apps regulation, preventing space invasion attacks and space affection.

(a) Would you regulate MAR-Apps? (b) Should businesses regulate MAR-Apps?

Figure 7.6: Questionnaire Exit.

Chapter 8

DISCUSSION AND FUTURE WORK

In the user study, we addressed the participants' understandability of the space and privacy attacks covered in Chapters 4.1-4.3. As a result, we found out that a significant majority of the participants correctly comprehended the security issues. Furthermore, they successfully identified threats that compromised security on a given set of scenarios, properly discussed in Chapter 7.2. There was no noticeable difference in the performances between CS and Non-CS participants, which indicates users can handle these issues without any specific background, such as CS.

We also addressed the usability of our *proof-of-concept* Policy-Governed MAR-App `SpaceMediator`, with the Control Model covered in Chapter 5 and implementation in Chapter 6. Overall, participants' performance was decent as most of their policies enforced ideal regulations. Nonetheless, there are areas for improvement in this field. For example, considering that Non-CS participants had a slightly better performance than CS, along with the high prevailing understandability of the security issues, we suggest that better results on policy writing depend on further development in `SpaceMediator`'s front-end.

As covered in Chapter 6.5, `SpaceMediator` wrote policies through a GUI that reflected applicable attributes and their effect on them, i.e., permit or deny. Therefore, we did take care of having an understandable GUI. However, this was not our top priority, and several participants missed the data pointed out, leading to erroneous policies. As a result, we are now aware of the importance of MAR-Apps front-end when crafting regulations. Thus, `SpaceMediator`'s subsequent versions should bring an upgrade within this scope, and there is a wide possibility of advancements. For

example, a noticeable distinguishment between permit and deny, pointing out the relationship between attributes, building one rule at a time for better interpretation of policy structure, and vibration when updating policy's regulation type.

As a result, there might be a higher result on ideal policies. Furthermore, we should also take into account the policies evaluation types. As addressed in Chapter 7.1.2, policy evaluation resulted in three categories: ideal, permissive, and restrictive. Through these evaluations, we classified the possible side effects an erroneous approach could have while regulating a MAR-App. Although, the reality is that participants had different errors within the same type. For example, permissive policies had security problems, but some only allowed one unauthorized entity while others had no restrictions. Therefore, through our evaluations, we know whether erroneous policies tend towards security or usability issues, but further analysis is required to adequately assess the scalability of their consequences.

Since recorded real-world incidents inspire the security problems we want to prevent with our approach, as described in Chapter 2.2, we also addressed participants' agreement on the regulation of MAR-Apps. As a result, we found out that most of them are highly interested in the rule of MAR-Apps and would implement it if possible. Since `SpaceMediator` is a proof-of-concept approach, we conducted a regulated user study with only 40 participants, but our primary goal is to prevent those real-world incidents. Therefore, with knowledge of high interest from the community, it is possible to consider upgrading for future versions of `SpaceMediator` the databases implemented, and described in Chapter 6.1, to handle data at scale.

Finally, we are aware that participants were capable of specifying the sensitive spaces whenever writing a policy as a Space Owner, as explained in Chapter 6.2.1. Still, there is a concern for further action to verify ownership over the claimed areas. Since `SpaceMediator` was a *proof-of-concept* Policy-Governed MAR-App, we consid-

ered such verification process out of our scope. Although its implementation will be helpful through future versions of `SpaceMediator` to prevent malicious entities from meanly regulating a space they do not own.

Chapter 9

CONCLUSION

MAR-Apps have already been problematic due to a lack of regulations since they are still in early development. However, as the MAR market is expected to grow at substantial rates, it is crucial to evaluate recorded issues to prevent further ones. In this thesis, we introduce the concept of Policy-Governed MAR-Apps, which is implemented in the *proof-of-concept* `SpaceMediator`. It replicates the famous, successful, and troubling MAR-App Pokémon GO, but with regulated operations to prevent malicious activities. As a result, it protects sensitive spaces as only authorized MAR merges with the physical surroundings; it only allows benign multi-user interchange through controlled user interaction; its respects users' privacy by granting management over gathered sensitive information. Additionally, there is a high interest throughout the user study community for further implementation of Policy-Governed MAR-Apps. Also, high understandability over the risks MAR-Apps involve, and effective success rates in enforcing `SpaceMediator`'s regulations. Therefore, `SpaceMediator` might be the starting point towards well-regulated MAR-Apps, as Policy-Governed MAR-Apps is an implementable and discernible regulatory mechanism to protect Space Owners and ordinary users.

REFERENCES

[1] C. D. P. K. Ramli, H. R. Nielson, and F. Nielson, "The logic of XACML," *Science of Computer Programming*, vol. 83, pp. 80–105, 2014. Formal Aspects of Component Software (FACS 2011 selected and extended papers).

[2] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[3] A. K. Tang, "Key factors in the triumph of Pokémon GO," *Business Horizons*, vol. 60, no. 5, pp. 725–728, 2017.

[4] "CEOS of snap, XTMIF, OGGFF and MQ leading disruptive innovation and revenue growth in fintech, augmented reality and plant-based foods." `https://finance.yahoo.com/news/ceos-snap-xtmif-oggff-mq-130400034.html`, Feb 2022.

[5] M. Billinghurst, A. Clark, and G. Lee, "A Survey of Augmented Reality," *Foundations and Trends in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.

[6] A. M. Research, "Global Mobile Augmented Reality Market to garner $184.61 billion by 2030: Allied Market Research." `https://www.globenewswire.com/news-release/2021/09/15/2297215/0/en/Global-Mobile-Augmented-Reality-Market-to-Garner-184-61-Billion-by-2030-Allied-Market-Research.html`, Sep 2021.

[7] M. Chan, "Pokémon Go Players Anger 9/11 Memorial Visitors: 'It's a Hallowed Place'." `https://time.com/4403516/pokemon-go-911-memorial-holocaust-museum/`, 2016.

[8] "Holocaust Museum, Auschwitz want Pokémon Go hunts out." `https://www.usatoday.com/story/tech/news/2016/07/12/holocaust-museum-auschwitz-want-pokmon-go-hunts-stop-pokmon/86991810/`, 2016.

[9] T. Mullen, "Hundreds of Pokemon Go incidents logged by police." `https://www.bbc.com/news/uk-england-37183161`, 2016.

[10] A. Beach, M. Gartrell, and R. Han, "Solutions to Security and Privacy Issues in Mobile Social Networking," in *2009 International Conference on Computational Science and Engineering*, vol. 4, pp. 1036–1042, 2009.

[11] R. Minch, "Privacy issues in location-aware mobile devices," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pp. 10 pp.–, 2004.

[12] T. Höllerer and S. K. Feiner, "Chapter Nine Mobile Augmented Reality," 2004.

[13] J. M. Mota, I. Ruiz-Rube, J. M. Dodero, and I. Arnedillo-Sánchez, "Augmented reality mobile app development for all," *Computers & Electrical Engineering*, vol. 65, pp. 250–260, 2018.

[14] G. Meyer-Lee, J. Shang, and J. Wu, "Location-leaking through Network Traffic in Mobile Augmented Reality Applications," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, 2018.

[15] T. Haselton, "Google Maps has a wild new feature that will guide you through indoor spaces like airports." `https://www.cnbc.com/2021/03/30/google-maps-launches-augmented-reality-directions-for-indoor-spaces.html`, Mar 2021.

[16] Skarredghost, "My predictions for augmented reality in 2022." `https://skarredghost.com/2022/02/04/predictions-augmented-reality-2022/`, Feb 2022.

[17] "Pokemon go away: Troublesome sydney pokestop shut down." `https://www.bbc.com/news/technology-36948331`, 2016.

[18] L. Gornstein, "Terrible things happening to Pokemon Go players." `https://www.cbsnews.com/pictures/terrible-things-happening-to-pokemon-go-players/2/`, 2016.

[19] Chung, D. Ferraiolo, and D. Kuhn, "Assessment of Access Control Systems." `https://doi.org/10.6028/NIST.IR.7316`, 2006-09-29 2006.

[20] Chung, D. Ferraiolo, D. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (abac) definition and considerations." `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927500`, 2019-02-25 2019.

[21] Chung, D. Ferraiolo, R. Chandramouli, and D. Kuhn, *Attribute Based Access Control*. Artech House, Norwood, MA, 2017-11-30 2017.

[22] C. E. Rubio-Medrano, Z. Zhao, A. Doupe, and G.-J. Ahn, "Federated Access Management for Collaborative Network Environments: Framework and Case Study," in *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, SACMAT '15, (New York, NY, USA), p. 125–134, Association for Computing Machinery, 2015.

[23] C. E. Rubio-Medrano, S. Jogani, M. Leitner, Z. Zhao, and G.-J. Ahn, "Effectively Enforcing Authorization Constraints for Emerging Space-Sensitive Technologies," in *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, SACMAT '19, (New York, NY, USA), p. 195–206, Association for Computing Machinery, 2019.

[24] F. Turkmen, J. den Hartog, S. Ranise, and N. Zannone, "Formal analysis of XACML policies using SMT," *Computers & Security*, vol. 66, pp. 185–203, 2017.

[25] "eXtensible Access Control Markup Language (XACML) version 3.0," 2013.

[26] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner, "Towards security and privacy for multi-user augmented reality: Foundations with end users," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 392–408, 2018.

[27] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner, "Securing Augmented Reality Output," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 320–337, 2017.

[28] R. R. Lutz, "Safe-AR: Reducing Risk While Augmenting Reality," in *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 70–75, 2018.

[29] J. Shang, S. Chen, J. Wu, and S. Yin, "ARSpy: Breaking Location-Based Multi-Player Augmented Reality Application for User Location Tracking," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 433–447, 2022.

[30] X. Zhang, R. Slavin, X. Wang, and J. Niu, "Privacy Assurance for Android Augmented Reality Apps," in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 114–1141, 2019.

[31] P. Stirparo, I. N. Fovino, M. Taddeo, and I. Kounelis, "In-memory credentials robbery on android phones," in *World Congress on Internet Security (WorldCIS-2013)*, pp. 88–93, 2013.

[32] C. Lyu, A. Pande, X. Wang, J. Zhu, D. Gu, and P. Mohapatra, "CLIP: Continuous Location Integrity and Provenance for Mobile Phones," in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 172–180, 2015.

[33] L. Onwuzurike and E. De Cristofaro, "Danger is My Middle Name: Experimenting with SSL Vulnerabilities in Android Apps," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, (New York, NY, USA), Association for Computing Machinery, 2015.

[34] "ARCore." `https://developers.google.com/ar/develop/downloads`, 2021.

[35] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 1451–1455, 2016.

[36] "AT&T XACML." `https://github.com/att/xacml-3.0`, 2021.

APPENDIX A

USER STUDY

Scenario	Security Issues
A new MAR-App allows for AR-augmented text messages to be displayed over physical spaces all over the world. Once users approach a physical space, they can see the messages left by other users in the past. Messages are not monitored and can be placed without restrictions. In a popular shopping mall, messages insulting different social groups have been left by users, and can be seen by anybody, without permission of the mall owners.	Space Invasion, Space Affectation
A MAR-App allows for users to upload pictures from their device camera and augment them with MAR content, e.g., digital objects. The pictures are then not shared with anybody, are not stored in remote servers, and will disappear from the device memory after several minutes.	None
The Communications Director from the Holocaust Museum in Washington, D.C., wants the Museum excluded from Pokémon Go. The game is considered inappropriate for the memorial.	Space Invasion
Bob started using a social MAR-App. He posted a picture featuring a beautifully designed organic MAR object in his backyard, and his current location was posted alongside, which he did not intend to share.	Privacy Leak
Bob happily plays with Pokémon Go. He approaches the closest Pokéstop and sees two guys at that location playing the same game. However, when he gets there, the situation changes as they assault him.	Space Affectation

Table A.1: User Study Questionnaire Scenario Recognition.

User	Reg.	MAR	Username	Age	Time	Other
1	Close	= *Spider*	= *Bob*	–	$\geq 18:00:00$	–
2	Open	–	–	–	–	–
3	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
4	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
5	Close	= *Spider*	= *Bob*	$= 18$	$\geq 18:00:00$	–
6	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
7	Close	–	= *Bob*	≥ 18	$\geq 18:00:00$	–
8	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
9	Close	= *Fox*	–	–	–	–
10	Close	–	–	–	$\geq 18:00:00$	–
11	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
12	Close	= *Spider*	= *Bob*	–	$\geq 18:00:00$	–
13	Close	= *Spider*	= *Bob*	–	$\geq 18:00:00$	–
14	Close	= *Spider*	= *Bob*	$= 18$	$\geq 18:00:00$	–
15	Close	= *Spider*	= *Bob*	> 18	$\geq 18:00:00$	–
16	Close	= *Spider*	= *Bob*	–	$\geq 18:00:00$	–
17	Close	= *Spider*	= *Bob*	–	$\geq 18:00:00$	–
18	Close	= *Spider*	= *Bob*	> 18	$\geq 18:00:00$	–
19	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
20	Close	= *Spider*	= *Bob*	–	$\geq 18:00:00$	–
21	Close	–	= *Bob*	≥ 18	$\geq 18:00:00$	–
22	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
23	Open	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
24	Open	–	–	–	–	*Student* = *Bob*
25	Close	= *Spider*	= *Bob*	> 18	$\geq 18:00:00$	–
26	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
27	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
28	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
29	Close	= *Spider*	–	–	–	–
30	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
31	Close	= *Spider*	= *Bob*	> 18	$\geq 18:00:00$	–
32	Open	–	–	–	–	–
33	Close	–	–	–	$\geq 18:00:00$	–
34	Close	= *Spider*	= *Bob*	> 35	$\geq 18:00:00$	–
35	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
36	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
37	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
38	Close	= *Spider*	= *Bob*	≥ 18	–	–
39	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–
40	Close	= *Spider*	= *Bob*	≥ 18	$\geq 18:00:00$	–

Table A.2: User Study - Participants' Policies for Exercise 2