A Multi-Store Privacy Measurement of Virtual Reality App Ecosystem

Chuan Yan, Zeng Li, Kunlin Cai, Liuhuo Wan, Ruomai Ren,
Yiran Shen, Guangdong Bai

***Abstract*—Virtual Reality (VR) has gained increasing traction among various domains in recent years, with major companies such as Meta, Pico and Microsoft launching their application stores to support third-party developers in releasing their applications (or simply *apps*). These apps offer rich functionality but inherently collect privacy-sensitive data, such as user biometrics, behaviors, and the surrounding environment. Nevertheless, there is still a lack of domain-specific regulations to govern the data handling of VR apps, resulting in significant variations in their privacy practices among app stores.**

In this work, we present the *first* comprehensive multi-store study of privacy practices in the current VR app ecosystem, covering a large-scale dataset involving 6,565 apps collected from five major app stores. We assess both *declarative* and *behavioral* privacy practices of VR apps, using a multi-faceted approach based on natural language processing, reverse engineering, and static analysis. Our assessment reveals significant privacy compliance issues across all stores, underscoring the premature status of privacy protection in this rapidly growing ecosystem. For instance, one third of apps fail to declare their use of sensitive data, and 21.5% of apps neglect to provide valid privacy policies. Our work sheds light on the *status quo* of privacy protection within the VR app ecosystem for the first time. Our findings should raise an alert to VR app developers and users, and would encourage store operators to implement stringent regulations on privacy compliance among VR apps.

I. Introduction

Virtual Reality (VR) has exhibited remarkable growth in recent years, driven by the convergence of advancements in software and hardware technologies. It creates computer-generated digital environments, providing users with immersive experiences through specialized devices such as head-mounted displays. VR has gained widespread popularity across a broad spectrum of domains, such as gaming and entertainment, education and distance learning, remote collaboration, healthcare and rehabilitation. The global VR market is projected to achieve a compound annual growth rate of 14% from 2023 to 2032, as highlighted by a recent report [1].

The growth of VR can be partly attributed to the continuously enriching content offered by flourishing VR apps. Inspired by the great success of the mobile app ecosystem, VR service providers, such as Meta Oculus and Pico [2], have established app stores to facilitate third-party developers in releasing their apps to users. These stores have become a hub for users to access a wide variety of VR experiences. The growing popularity of VR apps also introduces significant privacy risks. Their immersive nature entails access to extensive privacy-sensitive data through multi-modal sensors deployed in VR devices, including user biometrics, movements, behaviors, and even the surrounding environment. This unprecedented level of access to sensitive data makes VR apps particularly vulnerable to privacy breaches.

A crucial line of research has focused on understanding privacy leakage related to VR-specific data. It reveals that the leakage of VR-specific data, such as virtual identities, social interactions, and biometrics [3], can lead to severe privacy issues such as precise identification of users or detailed profiling of their habits [4], [5], [6], [7]. Some recent studies [8], [9] have also begun exploring the privacy practices of VR apps' behavior within individual stores. However, the broader landscape of *how effectively VR app developers and app store operators can safeguard user privacy and rigorously enhance their measures* remains largely unexplored.

Evaluating privacy risks in the VR domain is a challenging task. At the app level, existing compliance checkers [10], [11] are not directly applicable due to their limitation in integrating VR domain-specific knowledge. Although regulations like the European Union (EU) General Data Protection Regulation (GDPR) [12] and California Consumer Privacy Act (CCPA) [13] have been proposed to impose obligations on data controllers and processors, they only provide high-level guidelines. Effective compliance checking in the VR domain requires a combination of these regulations with domain-specific insights to understand the real practices [14]. Moreover, VR apps encompass more complex use cases and designs, equipped with additional sensors to collect VR-specific sensitive data [4], such as eye tracking and hand movement data, which are beyond the scope of existing analyzers for traditional mobile device apps [15], [16]. For instance, the implications of motion data leaks in VR can vary significantly [17], [18], and the collection of additional child data in these environments may lead to severe consequences [19], [20], [21]. The complexity of the VR ecosystem also poses significant challenges for large-scale analysis. VR apps can be built using various APIs in different game engines, making them difficult to analyze. These apps are released through different app stores, each targeting different users, offering different functionality types, and adhering to their own store-specific policies that explain VR-specific requirements, such as Meta Quest's Virtual Reality Check.

Our Work. In this work, we conduct the first comprehensive

• *C. Yan, L. Wan, R. Ren and G. Bai are with The University of Queensland, Australia.*
• *Z. Li and Y. Shen are with Shandong University.*
• *K. Cai is with The University of California, Los Angeles (UCLA)*

study on the privacy practices of VR apps in the current app ecosystem. By combining domain-specific knowledge collected from VR research and various standards in VR privacy, we summarize two categories of VR-specific privacy-sensitive data and five VR-domain privacy concerns. Using this domain-specific knowledge, we create a compliance-checking pipeline and conduct a multi-store analysis to understand the privacy practices of existing VR apps in the wild. This approach addresses existing challenges and provides an effective and comprehensive understanding of VR privacy compliance issues.

Our multi-store analysis targets five major VR stores, including Oculus, Viveport, Pico, Microsoft, and PlayStation, which represent over 80% of the global market share of VR devices as of 2025 [2]. This allows us to identify the different privacy problems caused by various store designs and highlight the shared problems that the community should pay more attention to. Moreover, the results of the analysis will enable us to perform adaptive compliance checking, which can be used in the majority of future VR apps.

Our analysis includes investigations of the apps' privacy policy documents and privacy meta-information disclosed through different app stores. The former serves as the formal disclosure and contractual agreement between the user and the developer [22], [6], [23], [24], and the latter serves as the portal for the user to glean insights into the app's data collection practices. We aim to explore two research questions (RQs) regarding these declarative documents. ***RQ1)*** *What is the status of VR apps in providing users with adequate documents to declare their data handling practices?* This RQ investigates whether comprehensive documents are available and easy-to-interpret by average users, with a horizontal comparison of the types of meta-information mandated by different VR app stores (**Section V**). This comparison addresses the gap in current research that analysis of a single VR app store [9] is not sufficient and allows for a comprehensive understanding of the real VR store ecosystem, which contains multiple stores. ***RQ2)*** *Do existing VR apps provide complete and accurate content in their declarative documents?* This RQ assesses whether the contents provided cover the key components of concern to users and legislators, such as permission requests, data collection, and data retention. This assessment combines high-level regulations with VR domain-specific knowledge to provide an in-depth analysis of the thoroughness of privacy content in the VR domain, which is crucial for ensuring user trust and compliance with legal standards. We also specifically check child user privacy protections, considering the large child user base of VR apps (**Section VI**).

For privacy-relevant behaviors, we aim to explore ***RQ3)*** *what is the status of existing Android VR apps' actual behaviors in complying with their privacy policies?* To this end, we construct the *behavioral profiles* that represent apps' access to VR-specific data. We develop an analysis framework specifically designed for Android VR apps, given their wide use and the availability of their executable files (i.e., *.apk* files), which represents the mainstream of the VR market. We customize reverse engineering and static analysis techniques to overcome challenges arising from apps' dependency on game engines (**Section VII**). Specifically, unlike prior studies that focused only on one platform, we propose a reverse engineering technique that recovers semantic information from *.apk* files compiled using different game engines (e.g., Unity and Unreal) and compilation methods (e.g., IL2CPP, Mono). Recovering this information is crucial for identifying actual API usage and behaviors, enabling a more comprehensive analysis of unique compliance issues in VR applications.

Key Findings. Our study reveals the *status quo* of the privacy practices in the VR app ecosystem for the first time. Some main findings are summarized below.

- **Lack of a stringent privacy vetting process in different VR app stores**. VR app developers are not provided with a standardized process for disclosing privacy-related information and privacy policies. This has resulted in significant discrepancies in mandated disclosure and quality control of privacy-related declarations across VR app stores.
- **Low quality of privacy policy documents**. Privacy policies of existing VR apps exhibit several critical deficiencies, including insufficient detail (30%), inaccessible references (20%), limited non-English language support, lack of VR-specific content (34%), poor readability, and incompleteness. Such inadequate privacy policies may discourage users from adopting VR and hinder the healthy development of the VR ecosystem.
- **Inconsistency between behavioral profiles and declarative profiles**. In the Android-based VR app stores (Oculus and Pico), almost one-fifth of VR apps access VR-specific sensitive data without transparently and correctly declaring this behavior in their privacy policies.

Contributions. The main contributions of this work are summarized as follows.

- **Revealing the *status quo* of privacy practices in the VR app ecosystem**. We conduct the first comprehensive analysis of VR apps multi-stores, revealing findings that underscore the ongoing concerns regarding the current status of the VR app ecosystem. Our work serves as a call to privacy attention for this emerging and expanding domain. It encourages app developers and store operators to take measures to foster the robust advancement of the ecosystem.
- **Constructing the first multi-store dataset of VR apps for privacy analysis**. We construct a large-scale VR app dataset including 6,565 apps for five major VR stores, and their artifacts of 802 *.apk* files, 6,565 app profile pages, and 2,788 privacy policies.
- **An adaptive VR app analysis framework suitable for large-scale analysis**. To address the challenge posed by different store policies, diverse compilation methods, and multiple game engines in the VR domain, we extract VR-specific data APIs from relevant documentation. We then combine these with the VR corpus we collect to develop an adaptive framework that allows analyzing VR applications in different app stores.

II. Background

In this section, we first introduce the privacy features of VR apps (Section II-A). To facilitate the understanding of our app behavior analysis, we also introduce the game engines for VR app development and execution (Section II-B).

A. Privacy Features of VR Apps

Apps developed for various VR stores offer users an immersive and experiential means to engage with virtual reality content through specialized equipment like headsets, headphones, or masks. They often involve sensitive data to create a profound, immersive VR experience. For instance, some apps [25], [26] use iris data to enable their interaction based on eye tracking, and several game or exercise apps gather and model the surrounding physical space to customize their gesture interaction.

VR manufacturers opt for Android as their primary development system, to enable developers to create apps efficiently, such as the Oculus Quest and Pico Neo series. Therefore, Android-based apps are the main target in our analysis. Their devices naturally inherit Android's permission system as their resource access control system for sensitive resources like cameras, sensors, location services, and microphones. It mandates apps to explicitly declare their permission requests in the `AndroidManifest.xml` file and obtain consent from users.

B. Game Engines

The game engines are specialized software frameworks designed to create virtual reality experiences. These engines offer a suite of tools and features that enable developers to construct immersive 3D environments and interactive experiences in their apps. They also manage device sources (e.g., sensors and cameras), and provide program interfaces for apps to access these resources. Therefore, when analyzing the privacy behaviors of VR apps, we have to locate such API invocations.

So far, two game engines, Unity and Unreal, have been dominantly popular, which together account for 70% of the VR app engine market [27]. They are both renowned for their multi-store support, enabling developers to create apps compatible with various devices. Both engines manage the interaction with the underlying Android OS, eliminating the necessity to develop apps in Java. Instead, the app crafted using them is compiled into native code and packaged into an *.apk* file. Below, we present a brief introduction to each of the engines.

Unity. Unity is the most popular game engine, according to its recent market share [28]. It uses C# (built on the .NET framework) for scripting, providing compatibility with a variety of stores. In addition to the robust graphics rendering and physics simulation, Unity offers a vast array of APIs for data handling, such as accessing the device's acceleration (via the `Input.acceleration` API) and rotation (via the `Input.gyro`) in 3D space.

To analyze Unity-based apps, a challenge to address is Unity's *Intermediate Language to C++* (IL2CPP) mechanism [29]. For devices without the Mono runtime [30], IL2CPP generates C++ code during the app building process, and compiles it into native code. This process introduces a challenge for app analysis, resulting in the loss of high-level information (see Section VII-A1).

Unreal. The Unreal engine is well known for its high-fidelity graphics and real-time rendering capabilities. It uses C++ as the programming language for development, offering extensive control over game mechanics and visuals. Besides its advanced rendering engine that supports detailed graphic processing, Unreal supports personalized physical simulations based on user data such as eye and body movements. When Unreal packs the app into *.apk*, its source code is compiled together with the library files into `libUE4.so`, and its main configurations are wrapped into a *.pak* file. Therefore, we mainly focus on these files for the app's behaviors in our analysis (see Section VII-A2).

III. Methodology Overview and Domain Knowledge Aware Analysis

To address the challenge of performing domain knowledge-aware analysis on real-world applications, we begin by understanding the unique VR privacy threats and concerns. Then we design a methodology to measure the privacy of the VR app ecosystems. Figure 1 shows the workflow of our methodology. More specifically, We first collect app data from five major VR stores. For the textual artifacts of VR apps, we analyze the disclosure of privacy-relevant information and privacy policies in different stores to answer ***RQ1***. The details are shown in Section V. Next, we conduct an in-depth analysis of declarative documents, focusing on the provisions for protecting children's data, and cross-store permission misuse. This analysis aims to explain ***RQ2***. For ***RQ3***, We design a static analysis framework that extracts VR-specific privacy-sensitive data APIs from apps written with Unity and Unreal game engines, compiled using different approaches, and then constructs code representations as control flow graphs. We then combine the resulting control flow graph with the data types extracted from privacy policies to perform privacy compliance checking. It should be noted that our static analysis of game engines is limited to free-download VR apps (802 in total) available on the Android operating system's VR app stores, specifically Oculus and Pico. We provide a detailed explanation in Section VII.

A. VR-domain Privacy-Sensitive Data and Privacy Concerns

Since there is a lack of a comprehensive list of VR domain-specific privacy-sensitive data and privacy concerns of VR apps for our assessment use, we construct a privacy taxonomy for VR. This taxonomy is essential for guiding the design of our assessment methodology, ensuring that all relevant privacy aspects are thoroughly considered and addressed. We resort to the literature and data regulations like GDPR for this purpose. We start with Fernandez et al. [5] and Wu et al. [31], which are the most recent publications reflecting VR privacy, and track the studies they cite, continuing the process. The referenced literature covers a wide range of topics, including VR privacy policy [8], [32], VR interaction [6], data gathering [7], [33], [5], biometric authentication [34], [35]

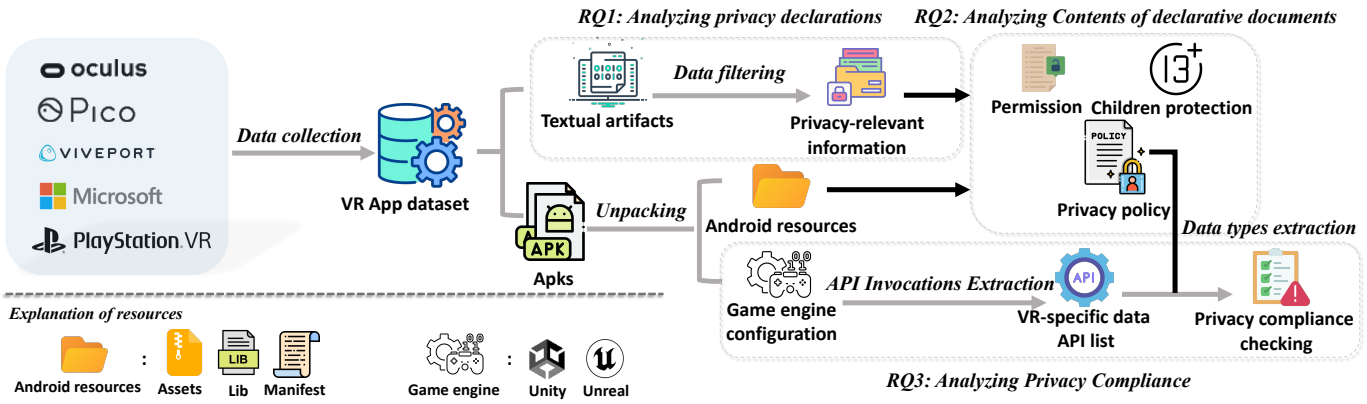

Fig. 1: The workflow of privacy measurement of virtual reality app ecosystem

TABLE I: VR privacy concerns consolidated from literature

Concern Type	No.	Privacy Concern	Referred by Literature
Declarative Concerns	C#1	Privacy Policy	[22], [36], [6], [7] [23], [37], [38], [24]
	C#2	Protection for Children	[19], [20], [21], [33]
	C#3	Permission Request	[39], [31]
Behavioral Concerns	C#4	Data Access	[5], [4], [40], [3]
	C#5	Behavior-declaration	[41], [15], [42]

and context-aware computing [6], [5], [8]. From them, we identify *privacy-sensitive Data of VR Domain* (Section III-A1), and consolidate privacy concerns on *declarative profiles* and *behavioral profiles* of VR apps (Section III-A2). The overall results are summarized in Table I, and next we detail our consolidation.

1) Privacy-sensitive Data of VR Domain: We first summarize the data types that are treated as privacy-sensitive data in studies relevant to privacy data analysis [34], [35], [6], [5], [8]. Two categories of data have been the main concerns of these studies.

Biometric data. VR apps have the capability to gather a wide spectrum of biometric data from users, including *eye movements, hand gestures, body motion*, and *facial expressions*. On the one hand, biometric data is essential for VR's user experience enhancement. For example, by tracking eye movements, VR apps can understand the user's focus and adjust the visual experience accordingly. Hand gesture tracking is crucial for enabling intuitive and natural interactions, enhancing the sense of immersion and engagement in the virtual world.

On the other hand, biometric data is typical personally identifiable information (PII), as it is highly personalized and can be used to identify an individual uniquely. A recent study [34] demonstrates that VR users can be uniquely and reliably identified across multiple sessions using just their head and hand motions. With a larger scope of data, an accuracy of 95.3% can be achieved on user identification among more than 500 individuals [35].

Ambient data. The data collected from a user's surrounding environment serves essential purposes within the realm of VR technology. By accurately mapping and modeling the physical space around the user, VR technology can establish virtual boundaries that correspond to real-world limitations, ensuring that users can move and interact without any physical constraints. Nonetheless, ambient data reveals key aspects of a person's life, such as frequently visited locations, daily routines, or social interaction, as demonstrated in previous studies [6], [5], [8].

2) Privacy Concerns of VR Apps: To recognize the privacy concerns of VR apps, we examine the topics of all collected papers and summarize the subjects their analyses are conducted on. We also read the main articles of GDPR, CCPA and Children's Online Privacy Protection Act (COPPA) to identify the privacy-related items that they specifically highlight. Three co-authors independently conduct the collection process, and then accumulate them via a discussion. By doing so, we identify three *declarative concerns* (C#1, C#2, C#3) to clarify data handling practices and restrictions on their user base, as well as two *behavioral concerns* (C#4, C#5) to focus on the actual behaviors of VR apps when they handle VR-specific privacy-sensitive data.

Concern #1: Privacy Policy. While data protection regulations vary slightly across different regions, they all require developers (i.e., *data processors*) to provide transparent and explicit privacy policies for their VR applications, ensuring users understand how their biometric data and immersive experiences are being collected, processed, and protected, e.g., Art.13 in GDPR [12], and Art.1798.130 in CCPA [13]. The common concerns regarding the privacy policies are on their availability [43], quality to be interpretable (or readability) [44], and alignment with regulatory requirements (or completeness) [45].

Concern #2: Protection for Children. Data regulations have been put in place specifically to protect children online, e.g., Children's Online Privacy Protection Act (COPPA) in the US [46] and GDPR Art.8. They all include stringent policies for handling children's data. For example, COPPA requires data handling for children under the age of 13 to be under parental control, and GDPR requires that information provided to children about data handling should be easy to understand. A recent study also indicates that children are more susceptible to addiction to VR technology compared to adults [47]. Additionally, children often exhibit lower privacy awareness and may have insufficient understanding level [48].

Therefore, it is essential for VR apps to enforce appropriate age restrictions and explicitly disclose their handling of child users' data, to mitigate potential privacy risks to children users [49].

Concern #3: Permission Requests. Recall that in Android-based systems, permissions are taken as credentials to access data and resources (Section II-A). As mentioned in Section III-A1, compared to traditional Android apps, VR apps often utilize users' biometric data, requiring special permissions beyond the default Android system permissions, such as facial tracking permission in Pico (`com.picovr.permission.FACE_TRACKING`) and gesture recognition permission in Oculus (`com.oculus.permission.HAND_TRACKING`) in VR devices. These special permissions are typically platform-specific. Mixing permissions in different platforms can lead to compatibility and security risks.

Concern #4: Data Access. The primary focus in the studies of app behaviors lies on data access and data collection [50], [8]. They examine the code [50] or the network traffic [8] to check whether the app collects and transmits sensitive data to remote servers. Following them, we put the focus of this work on the access of VR domain-specific data.

Concern #5: Behavior-declaration Compliance. Accessing sensitive data is not necessarily a violation, as some apps may need the data to fulfill relevant functionalities. This perspective has garnered a consensus in other domains such as mobile and virtual personal assistants [41], [42]. For example, a VR Yoga app that gathers body movement data should be considered benign, as long as it transparently discloses its data handling practices and adheres strictly to its privacy policy. Therefore, we highlight the importance of ensuring that the *behaviors* of apps should comply with the *declared* practices in their policies.

IV. Data Collection and Preprocessing

In this section, we introduce the process of collecting data relevant to building VR apps' declarative and behavioral privacy profiles, including textual artifacts and app executables (*.apk* files) (Section IV-A). We also outline the preprocessing of the collected *.apk* files (Section IV-B).

A. VR App Data Collection

Due to the absence of a dataset encompassing VR multi-store data, we take the initiative to construct one for our analysis. We collect all relevant data that can reflect the declarative and behavioral aspects of the privacy practices within the app.

Textual artifacts. App stores typically require developers to provide meta-information for their apps, as exemplified in Figure 2. This encapsulates the declarative aspect of the app's privacy practices, encompassing details such as the link to the privacy policy, requested permissions, age rating, and interaction styles. Hence, we create a crawler based on Selenium [51] for the data collection and parsing. The process starts at the portal page of the app catalog, systematically navigating through all available pages and extracting the textual content from each app's profile page. This method allowed us to compile a comprehensive dataset, resulting in the collection of textual artifacts from a total of 6,565 VR apps. The details and analysis of this dataset are summarized in Table II.

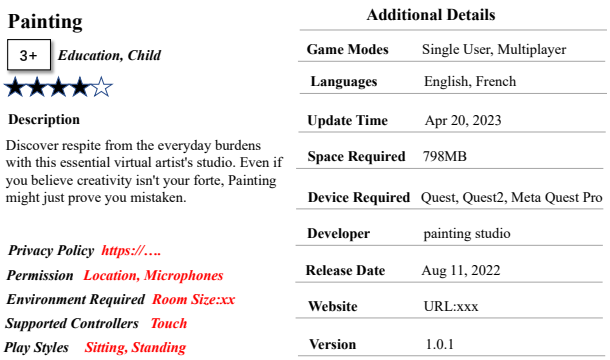

Fig. 2: An example of textual artifacts of a VR app (with privacy-related texts highlighted[1])

App executables. Given that Android-based apps are more accessible for analysis than apps for other stores, we mainly collect apps' *.apk* files for subsequent study. We first create multiple accounts on app stores and deploy them on six VR devices, including three Meta Quest 2, two Pico Neo 3, and one Pico 4. From the app stores, we install all free-to-use apps onto the devices. Then, we sideload the APK Extractor [52] on the devices, which is a tool designed for extracting *.apk* files installed on Android devices. Finally, we use the Android Debug Bridge (ADB) `pull` command to transfer these extracted *.apk* files to a PC. Among the five targeted VR stores, three of them, i.e., Oculus, Viveport, and Pico, primarily use Android as their operating system. In total, we have collected 802 *.apk* files, including 365 from Oculus and 437 from Pico. As Viveport has a limited number of free-to-use apps provided, and most of them are for PC (*.exe* files), apps from its app store are not included in our dataset.

B. APK Unpacking

After obtaining the *.apk* files, we unpack them to extract resources that our assessment relies on. We use the ApkTool [53] which has been widely used for reverse engineering and analyzing Android-based applications. The *.apk* files are built by the development engines, such that their internal structures are organized differently. We extract the following three key resources for subsequent analysis.

Assets. In traditional Android apps, the assets folder is used for storing resource files like text, audio, video, or configuration files that do not require compilation. In Unity-based apps, Unity uses it to store game models, bytecode and native code. Particularly, it stores bytecode (compiled from C# scripts) for apps targeting Mono runtime, and native code for apps running without interpreters. The extracted code is used for the app behavior analysis (Section VII-A1).

lib. The lib directory in *.apk* is designated for storing an app's native library files. In traditional Android apps, these libraries

[1]We consider *Environment Required* and *Play Styles* privacy-relevant, as when an app enforces requirements on ambiance and the interaction model, it usually checks using a camera upon being started

TABLE II: Textual data by apps in five popular VR app stores

VR App store	Operating System	Manufacture	Size (Apps)	Developer	Copyright	Age Rating	Category	Permission	Language	Review	Privacy Policy	Game Model[1]	Update Information	App Size	System Requirement	Environmental Requirement[2]	Device Requirement	Play Styles[3]	Supported Controllers[4]
Oculus	Android	Meta	2,233	✓		✓	✓		✓	✓	✓	✓	✓	✓			✓	✓	✓
Viveport	Android/Windows	HTC	3,103	✓		✓	✓		✓	✓	P†(274)	✓	✓	✓	✓	✓	✓		✓
Pico	Android	Pico Interactive	455	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
Microsoft	Windows	Microsoft	473	✓	P†(157)	✓	✓		✓	P†(83)	P†(206)	✓		✓	P†(451)		✓		
PlayStation	Psos	Sony	301	✓	✓	✓	✓		✓		P†(33)	✓			✓		✓	✓	

[1] Game Model: The number of users supported by the VR apps and whether it can be networked to interact with other users.
[2] Environment Requirement: Venue requirements for VR apps, such as room size and ground conditions.
[3] Play Styles: VR apps support the way users use them. such as standing, sitting, or lying.
[4] Supported Controllers: VR apps support controllers such as joysticks, steering wheels and more.
† P: Partial, indicating that part of apps make the disclosure. The boxed features stand for those relevant to apps' privacy profiles.

are typically written in low-level programming languages like C or C++, and are compiled into native code. It is utilized by Unreal to store the entire app developed based on it, the configurations of the app, and the Unreal runtime. All code and runtime are compiled into a library named `libUE4.so`, and all configurations are wrapped into a *.pak* file. Both are located in the lib directory. We mainly focus on these files for the app's behaviors in our analysis (Section VII-A2).

AndroidManifest. The `AndroidManifest.xml` file is a crucial configuration file that provides information about the app to Android OS, such as package information, application components, and intent filters. It is used by Android OS to understand the structure and behavior of the app at the installation time. In our analysis, we use the permission requests that are contained in this file (Section VI-B).

V. Analyzing Privacy Declarations (RQ1)

To address the question of how well VR apps provide users with adequate documentation on their data handling practices, we analyze the status of apps' declarative documents including privacy-relevant information and the privacy policy (**Concern #1**) among five app stores.

A. Overview of Privacy-relevant Information Disclosure

As shown in Table II, there is a significant variation in the data types of apps disclosed among app stores. Surprisingly, except for Pico, the other two Android-based app stores do not mandate the necessary app permissions in their textual data. On the other hand, the situation with privacy policies is also concerning. Apart from Oculus and Pico, apps on the other three stores often lack privacy policies, with specific instances being 56.45% on Microsoft, 89.04% on PlayStation, and 91.17% on Viveport.

We also treat the external controllers as privacy-relevant information, as they may involve gesture movement of the user. For this type of information, Microsoft and PlayStation do not mandate the disclosure. Interaction models and environmental requirements should be disclosed, too, as apps require them to conduct checking through cameras when apps are started. Pico and PlayStation are the only two stores that mandate the disclosure of the interaction model, and Viveport mandates the environmental requirements.

TABLE III: The distribution of (failure) status codes from links in privacy policies among app stores

	Oculus	Viveport	Pico	Microsoft	PlayStation
404 (%)	1.7	8.8	21.8	10.2	5.9
400 (%)	0.0	0.4	0.0	0.5	0.0
500 (%)	0.0	1.1	0.7	1.0	0.0
503 (%)	0.1	0.0	0.3	0.0	0.0
410 (%)	0.2	0.0	0.0	0.0	0.0
Timeout (%)	3.4	11.0	9.1	11.7	2.9
Total (%)	5.4	21.3	31.9	23.4	8.8

> **Finding 1: There is a lack of standard policy on the disclosure of privacy-relevant features by VR apps** The mandated information for disclosure exhibits significant discrepancies among VR app stores. In comparison, Oculus and Pico provide more comprehensive privacy-relevant information. This is not only because they focus exclusively on VR app stores, but also due to the benefits of the complete Android ecosystem. The current lack of comprehensive information disclosure in multiple VR stores has raised concerns about the transparency and accuracy of user privacy data.

B. Analysis of VR Privacy Policies

Previous research [54] highlights that compared to mobile apps, VR apps are more likely to collect sensitive user data. If these practices are not clearly disclosed in the privacy policy, it can lead to significant privacy risks, including unauthorized data use, potential breaches, and a lack of informed consent from users, ultimately compromising user trust and regulatory compliance.

Validity of privacy policies. Some apps provide non-informative privacy policies. For instance, a few privacy policies consist of only a single sentence, such as "*We will collect your data*". Such simplistic sentences fail to inform users about how their data is collected and processed by the data handlers, especially regarding VR-specific data like eye-tracking and body movement, which are closely tied to user

identity. As discussed in [55], an overly short privacy policy fails to provide sufficient information. We treat policies shorter than 100 words as insufficient [41].

On the other hand, we find that some apps (3.87%) provide the link to the app store's privacy policy as the policy for their apps. These policies broadly summarize the data handling practices of official applications provided by device manufacturers but often express data collection practices in obscure language qualifiers, such as "*We may collect your biometric data*". However, this ambiguity fails to clearly inform users whether and what specific VR-related biometric data is being collected. Since VR-collected data can introduce unique threats, failure to inform these threats can raise concerns about fairness to users and is explicitly prohibited by Article 29 Working Party [56].

Finding 2: 29.8% apps fail to provide informative privacy policies. In each store, there are apps that fail to provide informative privacy policies. Apps in the Oculus store perform slightly better due to Meta's strict policies for developers. However, it is still notable that over 20% of the apps have privacy policies that are less than 100 words long. Some apps simply provide the privacy policies of their app store as their own policies. All stores except PlayStation suffer from this issue, including Microsoft (12.68%), Viveport (3.28%), Oculus (2.46%), and Pico (7.46%).

Accessibility of Links. VR apps' privacy policies often reference other web pages for explanatory purposes, such as defining VR-specific terms or explaining data processing methods unique to virtual environments. To ensure privacy policies are clear and transparent, these links must stay functional and accessible, enabling users to understand data collection related to VR-specific features. We check the links and record the returned status codes. The distribution of inaccessible links across five VR app stores is detailed in Table III.

Finding 3: Privacy policies of existing VR apps largely suffer from the inaccessibility of referred resources. More than 20% VR apps fail to maintain the accessibility of the resources their privacy policies refer to. The most severe issues stem from a status code of 404 (5.69%) and access timeouts (5.34%). Among app stores, Pico exhibits the worst inaccessibility, where over 30% of privacy policy links cannot be accessed properly, largely due to Pico's lack of supervision over apps.

Language Support. In traditional app stores, most apps primarily support English, largely because the companies and developers are predominantly based in English-speaking countries. In contrast, with the advent of open-source game engines like Unreal, the development of VR app stores has become more diverse. For example, China's Pico and Japan's PlayStation VR store cater to a global audience. Developers from around the world can now upload their VR apps to different app stores, making them accessible to users worldwide. Hence, for apps that support multiple languages, we also consider the coverage of the multilingual versions as an important metric in the availability of privacy policies, as Article 29 Working Party [56] stipulates that when the target data subjects of a data collector use multiple languages, corresponding translated versions should be provided. We thus check if the language versions of the app's privacy policy match the list of language versions it claims to support.

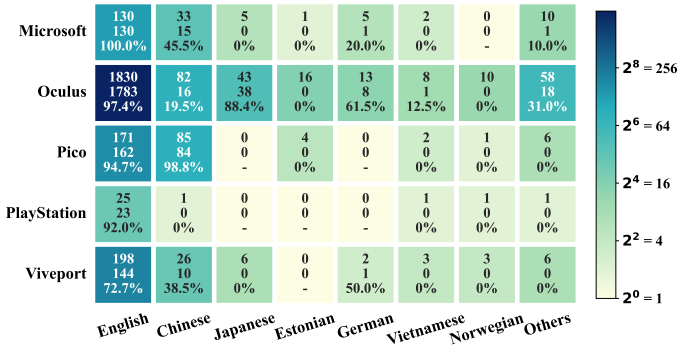

Fig. 3: The coverage of multilingual versions of privacy policies across VR stores[2]

We build a language detector based on Selenium [51] for this purpose. It detects whether a language selection option is available on the webpage of the privacy policy by searching for buttons and links using keywords of "*language*" and language names like "*English*" and "*French*". If such a link exists, it simulates a click action and checks its availability. It also mutates the language code in the privacy policy link, such as "*fr*" for French and "*de*" for German, and checks whether the mutants lead to accessible privacy policies. After obtaining these pages, it filters out invalid policies and uses *langdetect* [57] to check the languages of the pages.

Our language detector helps us identify the presence of many privacy policies in languages other than English, including Chinese (309, 11.08%), Japanese (86, 3.08%), German (25, 0.90%), Vietnamese (20, 0.72%), French (15, 0.54%) and other languages (97, 3.48%). Figure 3 summarizes apps' coverage of multilingual versions of their privacy policies.

Finding 4: Existing VR apps perform well in covering English privacy policy, but fall short in covering non-English languages as they claim in their "*Supported Language*" sections.

VR Specificity. Some developers simply reuse the privacy policy for their other apps as the policy for their VR apps. Such policies only provide general information to users, without important details of how data is processed specifically in VR apps. This behavior also increases the risk of non-compliance with privacy laws and regulations, potentially exposing both developers and VR app stores to legal and security issues. To check the specificity, we search the policy for the name of

[2]The first row of the cell represents the number of privacy policies in the corresponding language, the second row indicates the number of apps whose supported languages match their privacy policies, and the third row shows the proportion.

TABLE IV: The status of child-related privacy policies across app categories[3]

Category	Oculus					Viveport					Pico					Microsoft					PlayStation				
	App	I	%	D	%	App	I	%	D	%	App	I	%	D	%	App	I	%	D	%	App	I	%	D	%
Shooter	76	-	-	4	5.3	90	40	44.4	0	0.0	51	5	9.8	5	9.8	12	6	50.0	0	0.0	22	18	81.8	0	0.0
Action	646	-	-	49	7.6	326	144	44.2	0	0.0	87	9	10.3	4	4.6	55	27	49.1	1	1.8	81	64	79.0	2	2.5
Simulation	156	-	-	5	3.2	238	208	87.4	0	0.0	15	2	13.3	1	6.7	18	13	72.2	0	0.0	14	9	64.3	0	0.0
Adventure	259	-	-	15	5.8	639	286	44.8	5	0.8	27	1	3.7	2	7.4	-	-	-	-	-	50	35	70.0	4	8.0
Horror	60	-	-	5	8.3	19	1	5.3	0	0.0	5	1	20.0	0	0.0	-	-	-	-	-	10	7	70.0	0	0.0
Education	190	-	-	3	1.6	89	73	82.0	0	0.0	13	10	76.9	0	0.0	98	95	96.9	0	0.0	-	-	-	-	-
Music	76	-	-	2	2.6	36	34	94.4	0	0.0	21	4	19.0	0	0.0	3	1	33.3	0	0.0	42	39	92.9	0	0.0
Social	88	-	-	4	4.5	3	3	100.0	0	0.0	18	11	61.1	0	0.0	-	-	-	-	-	-	-	-	-	-
Puzzle	102	-	-	1	1.0	-	-	-	-	-	28	2	7.1	0	0.0	17	12	70.6	0	0.0	12	8	66.7	1	8.3
Sport	151	-	-	2	1.3	43	31	72.1	0	0.0	41	6	14.6	1	2.4	14	13	92.9	0	0.0	8	7	87.5	0	0.0
Casual	306	-	-	4	1.3	721	584	81.0	2	0.3	51	18	35.3	1	2.0	-	-	-	-	-	8	5	62.5	0	0.0
Strategy	29	-	-	1	3.4	11	5	45.5	0	0.0	11	1	9.1	0	0.0	8	5	62.5	0	0.0	4	4	100.0	0	0.0
Media	23	-	-	0	0.0	581	507	87.3	1	0.2	45	35	77.8	0	0.0	10	9	90.0	0	0.0	-	-	-	-	-
Business	20	-	-	0	0.0	26	21	80.8	0	0.0	-	-	-	-	-	33	24	72.7	0	0.0	-	-	-	-	-
Total	-	-	-	-	3.3	-	-	66.7	-	0.1	-	-	27.5	-	2.5	-	-	69.0	-	0.2	-	-	77.5	-	1.9

I (Inconsistency): the app is for general user groups, but its privacy policy does not include a children's data protection regulation.
D (Discrepancy): the app's age restriction exceeds the age limitation stated in its privacy policy.

the VR app and VR-related terms, such as "*virtual reality*", "*VR*", "*immersive*", and "*head-mounted display*". We conduct a manual check on the identified cases for confirmation and find that most cases are with PlayStation and Microsoft, and those privacy policies are for console and PC games. This is likely because both stores develop console and PC games before venturing into VR.

Finding 5: Almost 1/3 privacy policies are not specific to their VR apps. Existing VR apps significantly suffer from this issue, with 92.67% in PlayStation, 71.47% in Microsoft, 68.64% in Viveport, 29.27% in Pico and 18.61% in Oculus. The reason for this situation is that before developing VR app stores, PlayStation, Microsoft, and Viveport had other app stores such as console games and PC apps. Therefore, many developers simply transfer the privacy policies from their previous apps to VR apps. Pico and Oculus, only provide VR app stores, which results in better performance in this regard.

We also analyze the readability of the privacy policies across apps in five VR app stores. The details are provided in Appendix A1.

VI. Analyzing Contents of Declarative Documents (RQ2)

In this section, we examine the content of VR apps' declarative documents from two angles, focusing on the consistency of age restrictions for child users (**Concern #2**) and cross-store permission misuse (**Concern #3**), to address RQ2.

A. Protection for Children

Due to the immersive nature of VR apps, which can lead to issues such as overstimulation and reduced awareness of the real world for users [58], especially for children who may be more vulnerable to these effects, VR stores require developers to categorize apps based on their content and apply age restrictions [59]. For example, a shooting game is categorized as *shooter* and *action*, suitable for users above 13 years old, and a drawing app is categorized as *education* and *casual*, suitable for all user groups. We explore two questions in the assessment of child user protection, including **Q1)** for apps suitable for all user groups, whether child data protection clauses are included in their privacy policies, and **Q2)** for apps with age restrictions, whether the age limitations are consistent with those stated in their privacy policies. An example of violating Q2 is that a VR app in the *shooter* category prohibits users under the age of 13, but its privacy policy defines the age limitation as 11, causing disputes when it collects data of a 12-year-old user.

For Q1, we use QuPer [14] to detect the component of `CHILDREN` when the app has no restriction on the user groups. For Q2, we first locate sentences in the `CHILDREN` section and determine whether they include numbers that signify ages. To do this, we employ SpaCy [60] to examine the tokens preceding or following the numbers for specific terms like "*years*", "*age*", and "*old*". We manually select 20 privacy policies containing the children component for confirmation, and all ages specified by them can be extracted using our method. The results of Q2 are summarized in Table IV.

Finding 6: Existing VR apps fall short in enforcing age restrictions and disclosing specific privacy policies on child users' data. Only apps in Oculus have clearly defined age restrictions. The other four app stores have an average of 49.1% of apps suitable for users of all age groups, but many of them fail to include specific children's privacy clauses in their privacy policies, particularly, 66.7% in Viveport, 27.5% in Pico, 69.0% in Microsoft, and 77.6% in PlayStation.

Finding 7: The discrepancy between the age restriction disclosed through the app store and the age limitation defined in the privacy policy is not uncommon. The discrepancy occurs in all app stores, with Oculus (3.3%), Pico (2.5%) and PlayStation (1.9%) suffering more than the other two. Notably, in the *shooter* category of Pico and *horror* category of Oculus, this discrepancy reaches 9.8% and 8.3%, respectively.

B. Cross-store Permission Misuse

The permission system plays a central role in resource access control in Android-based systems. The two Android-based platforms, i.e., Pico and Oculus, extend the native Android permission system with their custom permission labels for VR-specific data, e.g., the facial tracking permission `com.oculus.permission.FACE_TRACKING` in Oculus, and `com.picovr.permission.FACE_TRACKING` in Pico. These permission labels are specific to their respective platforms, so when an app attempts to use a permission from one platform on another, the compiler may fail to match or recognize these permissions, potentially leading to compilation errors or causing the permission system to malfunction at runtime. However, we discover that some Oculus permissions in Pico *.apks*, including `HAND_TRACKING`, `FACE_TRACKING` and `EYE_TRACKING`, etc.

> **Finding 8: There is a discrepancy between requested permissions and declared permissions in the Pico app store.** Pico outperforms Oculus by mandating the disclosure of permissions. Nonetheless, a notable proportion (7.2%) of VR apps on Pico have requested at least one permission that is not indicated on their homepage. Such permissions include `READ_EXTERNAL_STORAGE`, `CAMERA` and `READ_PHONE_STATE`.
>
> **Finding 9: There are cross-platform declarations for VR-specific permissions present in Pico.** We discover a total of 27 Oculus VR-specific permission declarations across the *.apks* of 12 Pico apps, highlighting unreasonable compatibility issues between VR platforms and the "wrapper app." This might cause potential permission issues, and we will conduct more studies in the future to further explore this observation.

We also examine the completeness of the privacy policies across apps in five VR app stores. The details are provided in Appendix A2.

VII. Analyzing Privacy Compliance (RQ3)

In this section, we evaluate apps' data access (**Concern #4**) and behavior-policy compliance (**Concern #5**), by analyzing the code and privacy policies of Android VR apps.

A. API Invocations Extraction

We design different methods for extracting APIs related to VR-specific privacy-sensitive data based on different game engines and compilation modes. The differences in compilation language and file structure make the analysis of different VR applications very challenging. This also prompts us to find methods to analyze VR apps across different development environments. To solve this challenge, we summarize the APIs used to access VR-specific privacy-sensitive data from the official SDK documentation of Pico and Oculus. The mapping is listed in Appendix B. We take the invocations of these APIs as indicators of an app's behavior in accessing sensitive data. Moreover, by creating a large corpus of VR-specific data types using domain-specific knowledge, we design a comprehensive and adaptive analysis framework that supports compliance checking for VR applications across different stores. In the remainder of this section, we brief the mechanisms of these two engines and outline our analysis.

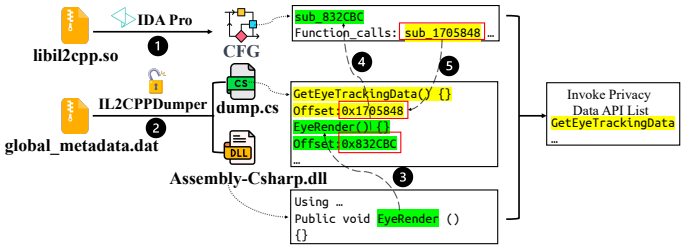

Fig. 4: A demonstration of IL2CPP native code analysis

1) Unity-based Apps: Unity uses two modes of compiling and executing C# scripts, i.e., *Mono* and *Intermediate Language to C++ Compiler (IL2CPP)*. As discussed in Section IV-B, in the former, the bytecode can be obtained, and in the latter, only native code is available for our analysis. Below, we describe our approach to analyzing apps in these two scenarios.

Mono. Mono is an open-source implementation of the .NET Framework. The Mono-oriented apps are initially compiled from C# code into a universal Intermediate Language (IL). At runtime, the IL code is Just-In-Time (JIT) compiled into machine code for execution. We employ *dotPeek* [61] to translate the IL code back into C# code, on which our static analysis is conducted. Next, we construct the Abstract Syntax Tree (AST) of the code using *Roslyn* [62], and traverse through its nodes to search for the sensitive API calls.

IL2CPP. IL2CPP compiles C# code into IL code, which is then converted into C++ code. The C++ code is compiled into native binary code for the target platform, resulting in a file named `libil2cpp.so`. Meanwhile, the meta-information is kept in another file named `global-metadata.dat`. This file can be decoded with an open-source tool named IL2CPPDumper [63], and the following two files can be derived.

- `dump.cs`, which is the symbol table that contains classes, functions, fields, and their offsets.
- `DummyDll/Assembly-CSharp.dll`, which includes all function declarations of the main app.

Figure 4 illustrates our approach of extracting API calls. We first decompile `libil2cpp.so` and construct the control flow graph (CFG) using IDA Pro (❶). An intuitive way of extracting API calls is through searching the CFG, but the CFG is too huge to accommodate this. To solve this challenge, we start with the functions declared in `Assembly-CSharp.dll`. Taking `EyeRender()` shown in Figure 4 as an example, we get its offset from the symbol table (❸), i.e., *832CBC*. With this offset, we locate the subgraph of `EyeRender()` in the CFG (❹), where we find that another function with offset *1705848* is invoked. Through looking up the symbol table (❺), this offset corresponds to

[3]We only consider categories that are present in each VR app store and have a substantial number of apps.

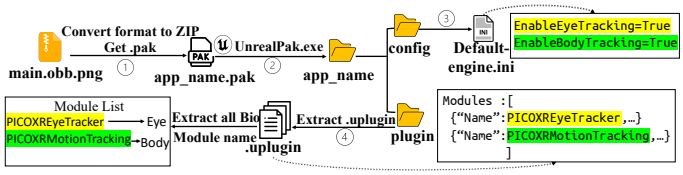

Fig. 5: A demonstration of Unreal-based app analysis

the API `GetEyeTrackingData()` (used for collecting eye tracking data). By recursively executing this process, we can identify all sensitive data-related APIs invoked by the app.

2) Unreal-based Apps: Unreal includes richer information in the *.apk* file than Unity does, including the configurations app developers provide to the development engine regarding their dependence on sensitive data. We thus resort to these configuration items to extract the data types the app requests from Unreal.

Figure 5 illustrates the steps conducted in our analysis. Unreal stores all app assets (e.g., models, scenes, game objects, and all plugins from the SDK) in a package named `main.obb.png`, which is in the zip archive format. Unpacking it, we get a *.pak* file (①). We then use an Unreal-developed tool UnrealPakTool [64] to decode the *.pak* file, resulting in a folder named `app_name` (②). Inside this folder, some apps have a `Defaultengine.ini` file (③), which stores app requested data in the format of key-value pairs. For example, the configuration item `EnableEyeTracking=True` in Figure 5 indicates that the app has activated the eye-tracking feature. Other apps have a few *.uplugin* files (④), which contains all the modules called by the app. For example, the module *PICOXRMotionTracking* indicates that the app has requested body movement data. From these configuration items, we can derive the sensitive data types the app requests.

B. Policy Data Types Extraction

Next, we proceed with extracting data types that the app declares to collect in the privacy policy. As several existing studies involve the extraction of entities from privacy policies [65], [41], [66], [67], we explore reusing them in our study. For instance, PolicyLint [65] generates an ontology and uses a name entity recognition (NER) model along with specific sentence patterns to accurately extract entities from privacy policies. However, in the domain of VR, there is a lack of extensive labeled data and specific sentence patterns needed to train a NER model. We thus apply the lightweight method proposed in Skipper [41].

Skipper has paragraph and sentence classifiers to obtain sentences related to user data types in the COLLECT component of the privacy policy, such as *"We collect your information include name, email address, state and technical information such as estimated hand size and hand pose data."* After obtaining these sentences, it uses natural language process techniques to parse them to obtain the data types. To adapt Skipper for our analysis, we construct a corpus of VR-specific data types from the official documentation of VR platforms and engines. Three authors of this paper independently find and label data types into the categories discussed in Section III-A1. Then they discuss each collected data type and merge them, leading to the corpus listed in Table V. We replace Skipper's original general data dictionary with this VR-specific data corpus, thereby empowering Skipper with the capability to extract VR-specific data from privacy policies. To check the precision of Skipper, we randomly select 20 privacy policies that contain a declaration of VR-specific data for manual confirmation. Three fails because Skipper mis-labels sentences into `COLLECT`.

TABLE V: VR-specific data types described in privacy policies

VPD‡	Related Data Types
Body	Body Tracking, Motion Capture Data Physical Interaction Data, User Posture and Movement
Face	Facial Recognition, Facial Mapping Emotion Detection, Facial Geometry Data, Camera
Eye	Eye Tracking, Gaze Detection Eye Movement Metrics, Pupil Dilation Data, Iris Scan
Hand	Hand Tracking, Hand Size, Hand Pose Data Touch Interaction, Hand Movement Data

‡ VPD: VR-specific Privacy-sensitive Data Types

C. Data Access and Privacy Compliance Assessment

We compare the data types extracted from the *.apk* files and the privacy policies. Table VI shows the use of privacy-sensitive data by apps across Pico and Oculus stores. On the Oculus platform, 331 apps are developed using Unity, 23 with Unreal, and 11 with other game engines. In Pico, 346 apps are developed with Unity, 78 with Unreal, and 13 with other game engines. Except for the case of using Unity for development on the Pico platform, there is a lack of resource files in other cases, like `global_metadata.dat` in Unity and `main.obb.png` in Unreal.

We display the frequently occurring VR privacy-sensitive data APIs in Appendix B. Among the 4 VR-specific data types, we find that hand-tracking APIs are among the most frequently used, such as `GetJointLocations` and `Pico.Get_Handness`. These APIs capture hand movements and joint states, enabling gesture control and interactions within the VR environment. Additionally, eye-tracking APIs like `UPvr_getEyeTrackingPos` and `OculusEyeTracker` are widely employed. These APIs track users' gaze directions in real-time, allowing for more natural interactions, such as adjusting perspectives and selecting objects. Similarly, face-tracking APIs, like `WantFaceTrackingService`, is often used to capture facial expressions, particularly in social VR or virtual character interactions where realistic emotional expression is crucial.

> **Finding 10: Both Pico and Oculus each have 18.3% and 17.5% apps that have misalignment between the declarations in privacy policies and actual data use behaviors.** Current VR apps often fail to declare their practices regarding the use of user VR-specific privacy-sensitive data in their privacy policies. Furthermore, even

TABLE VI: The status of VR-specific privacy data access and policy compliance

Store	Non-compliance Case	Engine	LF†	Access to VR-specific Privacy-sensitive Data§							
				Body		Face		Eye		Hand	
Pico	80 (18.3%)	Unity	0	17	(4.9%)	8	(2.3%)	6	(1.7%)	22	(6.4%)
				1	(0.3%)	0	(0.0%)	2	(33.3%)	3	(13.6%)
		Unreal	23	7	(12.7%)	7	(9.0%)	0	(0.0%)	21	(27.0%)
				0	(0.0%)	1	(14.3%)	0	(0.0%)	1	(4.8%)
Oculus	64 (17.5%)	Unity	13	12	(3.6%)	20	(6.0%)	9	(2.7%)	18	(5.4%)
				2	(16.7%)	2	(10.0%)	2	(22.2%)	2	(11.1%)
		Unreal	8	2	(8.7%)	2	(8.7%)	4	(17.4%)	5	(21.7%)
				0	(0.0%)	0	(0.0%)	0	(0.0%)	0	(0.0%)

† LF: The *.apk* lacks resource files.
§ The upper statistics represents the number of apps that are detected using the corresponding data, and the lower data represents the number of apps that make correct declaration in their privacy policies.

when such declarations are made, we find that most privacy policies use vague terms like "*biometric data*", "*game interaction data*" or "*sensor data*", without specifying the exact types of data involved.

VIII. Discussion

A. Implications

The result of our cross-store analysis underscores the existence of significant privacy concerns within the current VR ecosystem. More specifically, the lack of a unified approach to privacy disclosures has led to a fragmented VR ecosystem, with significant variations in user experience and data protection standards. This fragmentation can hinder the overall development of the VR industry, as it complicates efforts to establish universal standards and best practices. Additionally, the VR privacy compliance issues identified in this study result in VR app stores violating data protection regulations, leading to fines or sanctions. In this section, we will further discuss the implications of our research for three stakeholders - VR app developers, store operators, and users.

For store operators. Store operators must enforce a stringent information disclosure process, enabling users to effectively understand the privacy-related information of VR apps. They should enhance their vetting process to ensure the application is compliant with data protection regulations like GDPR and their privacy policies. Platforms should also encourage developers to create specific privacy policies for their apps, rather than relying on generic policies or policies for their other apps, ensuring greater transparency and user trust.

For app developers. We emphasize the significance of VR app developers prioritizing privacy within the VR ecosystem. In summarizing our findings, we propose the following four guidelines for VR app developers: (1) ***Providing high-quality privacy policies.*** Developers should create clear and explicit privacy policies for their apps, ensuring that the policies comprehensively outline how data is collected, processed, stored, and shared, including specific details pertaining to VR, such as biometric data and ambient information. (2) ***Providing comprehensive app information.*** This includes offering comprehensive privacy-relevant details on the app's homepage, covering permissions, play style, environmental requirements, and so on. Such information can enhance users' comprehension of the app's data processing practices. Moreover, it is crucial to ensure that the information provided aligns with the actual behavior of the app. (3) ***User authorization and option.*** The app should provide users with a clear authorization mechanism that allows them to selectively consent or decline data collection and sharing. Prioritize respecting user choices and safeguarding their privacy right. (4) ***Special attention to children's privacy protection.*** When the developed app includes children in its user group, the privacy policy should outline a detailed process for handling children's data. Otherwise, explicit age restrictions should be provided along with an age verification mechanism to ensure access to the VR app is restricted to users of the appropriate age.

For users. As users explore the immersive world of VR technology, it is crucial to remain vigilant about personal data security, particularly biometric information. Before using any VR app, a thorough review of its privacy policy is imperative. This step ensures a clear understanding of the app's practices in collecting, processing, and utilizing personal information. Additionally, during the installation of VR apps, it is important to scrutinize the permission requests. Users should judiciously consider whether to grant permissions, especially when sensitive information is involved.

B. Limitations

To the best of our knowledge, we are the first comprehensive multi-store study of privacy practices of the current VR app ecosystem. However, several limitations exist that could be addressed in future work.

Reverse engineering and static analysis. We rely on static analysis to obtain the behavioral profiles of VR apps. It demonstrates the efficiency in finding API invocations to access privacy-sensitive data. Additionally, it supports multiple development environments, facilitating the analysis of *.apk* from different manufacturers and engines. However, this type of analysis may cause false positives, as certain code may not be reachable at runtime. This can be enhanced by designing strategies for dynamic analysis in our future work.

Analysis of Android-based apps. We collect a total of 6,565 VR apps from five VR app stores. Currently, we only analyze the free-download apps from two Android system app stores (Pico and Oculus) for their behavior due to the limitation in the app availability and toolchains.

Single privacy policy language analysis. As shown in Figure 3, current VR apps offer privacy policies in many languages. For instance, Pico provides a substantial number of privacy policies in Chinese. However, our analysis currently focuses only on the English versions of these policies. In future work, we plan to include privacy policies in different languages in our analysis.

IX. Related work

Previous research has focused on the analysis of VR technology's access to user privacy data [6], [7], [38], [24], [8]. However, few studies in the academic community regarding the privacy compliance of various VR platform apps. To the best of our knowledge, our work represents the largest multi-store study of information and behavior analysis in VR apps.

Privacy concerns in virtual reality. A series of studies have been dedicated to investigating privacy concerns arising from VR technology [32], [68], [7]. Adams et al. [39] present the first work on VR security and privacy perceptions, a mixed-method study involving semi-structured interviews with 20 VR users and developers, a survey of VR privacy policies, and an ethics co-design study with VR developers. Zhang et al. [69] propose a multidimensional peer-related privacy concern that focuses on privacy violations from online peers. Buck et al. [6] examine the ethical and privacy concerns arising from interactions and the availability of personal data. Building upon these previous works, we further refine declarative and behavioral concerns to comprehensively assess existing privacy concerns in the current VR ecosystem.

Privacy declaration analysis for applications in different domains. The emergence of data protection regulations such as GDPR and CCPA, has prompted various application domains to prioritize compliance assessments of privacy declaration information [70], [71], [72], [73], [74], [75], [76]. Andow et al. [65] present PolicyLint, an automated tool for identifying contradictions within Android application privacy policies. PolicyLint employs sentence-level NLP to comprehend the stated declarations regarding data collection and sharing. In the Virtual Personal Assistant (VPA) domain, Xie et al. [41] developed Skipper to identify the noncompliance between Alexa skills' behaviors and their declared information in the privacy policy. Yan et al. [14] introduce Quper to perform a comprehensive analysis of the privacy policy quality of VPA applications from four metrics according to data protection regulations. Drawing from specific terms in VR policies, such as "eye-tracking" and "hand-tracking" data, we conduct a method for evaluating privacy policies of VR apps based on previous work, ensuring comprehensive protection and compliance for VR user data.

X. Conclusion

In this work, we conduct comprehensive large-scale measurements on the privacy practices of the current VR app ecosystem. Our study covers 6,565 VR apps obtained from five major VR app stores, focusing on their declarative and behavioral privacy profiles. We explore the status of VR apps in declaring privacy-related information, evaluate the contents of the declarative documents, and assess the compliance between the app's declarative and behavioral privacy practices. To our best knowledge, our study represents the inaugural systematic investigation into the *status quo* of privacy practices within VR app ecosystems. Our research aims to serve as a crucial alert for VR app developers and a compelling catalyst for VR app store operators to implement robust and effective privacy oversight mechanisms.

References

[1] GlobelNewswire. (2023) Virtual reality market size 2032 valuation. https://www.globenewswire.com/en/news-release/2023/05/19/2672447/0/en/Virtual-Reality-Market-Size-to-Worth-Around-USD-113-59-BN-by-2032.html.

[2] Counterpoint, "Global xr (ar & vr headsets) market share (q1 2024 - q2 2025)," https://counterpointresearch.com/en/insights/global-xr-ar-vr-headsets-market-share-quarterly, 2025.

[3] C. H. Heruatmadja, A. N. Hidayanto, H. Prabowo *et al.*, "Biometric as secure authentication for virtual reality environment: A systematic literature review," in *2023 International Conference for Advancement in Technology (ICONAT)*. IEEE, 2023, pp. 1–7.

[4] V. Nair, G. M. Garrido, D. Song, and J. F. O'Brien, "Exploring the privacy risks of adversarial VR game design," *Proc. Priv. Enhancing Technol.*, vol. 2023, no. 4, pp. 238–256, 2023. [Online]. Available: https://doi.org/10.56553/popets-2023-0108

[5] C. B. Fernandez and P. Hui, "Life, the metaverse and everything: An overview of privacy, ethics, and governance in metaverse," in *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2022, pp. 272–277.

[6] L. E. Buck and B. Bodenheimer, "Privacy and personal space: Addressing interactions and interaction data as a privacy concern," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2021, pp. 399–400.

[7] J. Happa, A. Steed, and M. Glencross, "Privacy-certification standards for extended-reality devices and services," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2021, pp. 397–398.

[8] R. Trimananda, H. Le, H. Cui, J. T. Ho, A. Shuba, and A. Markopoulou, "{OVRseen}: Auditing network traffic and privacy policies in oculus {VR}," in *31st USENIX security symposium (USENIX security 22)*, 2022, pp. 3789–3806.

[9] H. Guo, H.-N. Dai, X. Luo, Z. Zheng, G. Xu, and F. He, "An empirical study on oculus virtual reality applications: Security and privacy perspectives," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[10] Y. Zhang, Z. Hu, X. Wang, Y. Hong, Y. Nan, X. Wang, J. Cheng, and L. Xing, "Navigating the privacy compliance maze: Understanding risks with privacy-configurable mobile sdks," in *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, 2024.

[11] S. Zimmeck, Z. Wang, L. Zou, R. Iyengar, B. Liu, F. Schaub, S. Wilson, N. Sadeh, S. Bellovin, and J. Reidenberg, "Automated analysis of privacy requirements for mobile apps," in *2016 AAAI Fall Symposium Series*, 2016.

[12] E. Union, "General data protection regulation(gdpr)," https://gdpr-info.eu/, 2023.

[13] O. of the Attorney General, "California consumer privacy act (ccpa)," https://oag.ca.gov/privacy/ccpa, 2023.

[14] C. Yan, F. Xie, M. H. Meng, Y. Zhang, and G. Bai, "On the quality of privacy policy documents of virtual personal assistant applications," in *24th Privacy Enhancing Technologies Symposium (PETS 2024)*, 2024.

[15] N. Samarin, S. Kothari, Z. Siyed, O. Bjorkman, R. Yuan, P. Wijesekera, N. Alomar, J. Fischer, C. Hoofnagle, and S. Egelman, "Lessons in vcr repair: Compliance of android app developers with the california consumer privacy act (ccpa)," in *23th Privacy Enhancing Technologies Symposium (PETS 2023)*, 2023.

[16] M. Fan, L. Yu, S. Chen, H. Zhou, X. Luo, S. Li, Y. Liu, J. Liu, and T. Liu, "An empirical evaluation of gdpr compliance violations in android mhealth apps," in *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*. IEEE, 2020, pp. 253–264.

[17] Z. Su, K. Cai, R. Beeler, L. Dresel, A. Garcia, I. Grishchenko, Y. Tian, C. Kruegel, and G. Vigna, "Remote keylogging attacks in multi-user vr applications," *arXiv preprint arXiv:2405.14036*, 2024.

[18] V. Nair, L. Rosenberg, J. F. O'Brien, and D. Song, "Truth in motion: The unprecedented risks and opportunities of extended reality motion data," *IEEE Security & Privacy*, 2023.

[19] J. S. Spiegel, "The ethics of virtual reality technology: Social hazards and public policy recommendations," *Science and engineering ethics*, vol. 24, no. 5, pp. 1537–1550, 2018.

[20] A. Plechatá, V. Sahula, D. Fayette, and I. Fajnerová, "Age-related differences with immersive and non-immersive virtual reality in memory assessment," *Frontiers in psychology*, vol. 10, p. 1330, 2019.

[21] D. Maloney, S. Zamanifard, and G. Freeman, "Anonymity vs. familiarity: Self-disclosure and privacy in social virtual reality," in *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*, 2020, pp. 1–9.

[22] J. Lim, H. Yun, A. Ham, and S. Kim, "Mine yourself!: A role-playing privacy tutorial in virtual reality environment," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022, pp. 1–7.

[23] X. Huang, D. Zou, G. Cheng, and H. Xie, "A systematic review of ar and vr enhanced language learning," *Sustainability*, vol. 13, no. 9, p. 4639, 2021.

[24] J.-M. López-Gil, J. Virgili-Gomá, R. Gil, T. Guilera, I. Batalla, J. Soler-González, and R. García, "Method for improving eeg based emotion recognition by combining it with synchronized biometric and eye tracking technologies in a non-invasive and low cost way," *Frontiers in computational neuroscience*, vol. 10, p. 85, 2016.

[25] R. Menges, C. Kumar, and S. Staab, "Improving user experience of eye tracking-based interaction: Introspecting and adapting interfaces," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 26, no. 6, pp. 1–46, 2019.

[26] N. Stein, G. Bremer, and M. Lappe, "Eye tracking-based lstm for locomotion prediction in vr," in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022, pp. 493–503.

[27] H. Singh, "Unreal engine vs. unity 3d games development," https://theninehertz.com/blog/unreal-engine-vs-unity-3d-games-development, 2023.

[28] 6sense, "Unity market share and competitor report," https://6sense.com/tech/game-development/unity-market-share, 2024.

[29] Unity, "Intermediate language to c++ unity overview," https://docs.unity3d.com/Manual/IL2CPP.html, 2023.

[30] ——, "The mono scripting backend unity overview," https://docs.unity3d.com/Manual/Mono.html, 2023.

[31] Y. Wu, C. Shi, T. Zhang, P. Walker, J. Liu, N. Saxena, and Y. Chen, "Privacy leakage via unrestricted motion-position sensors in the age of virtual reality: A study of snooping typed input on virtual keyboards," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2023, pp. 3382–3398.

[32] A. Ginosar and Y. Ariel, "An analytical framework for online privacy research: What is missing?" *Information & Management*, vol. 54, no. 7, pp. 948–957, 2017.

[33] F. O'Brolcháin, T. Jacquemard, D. Monaghan, N. O'Connor, P. Novitzky, and B. Gordijn, "The convergence of virtual reality and social networks: threats to privacy and autonomy," *Science and engineering ethics*, vol. 22, pp. 1–29, 2016.

[34] V. Nair, W. Guo, J. Mattern, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song, "Unique identification of 50,000+ virtual reality users from head & hand motion data," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 895–910.

[35] M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson, "Personal identifiability of user tracking data during observation of 360-degree vr video," *Scientific Reports*, vol. 10, no. 1, p. 17404, 2020.

[36] V. Biener, S. Kalamkar, N. Nouri, E. Ofek, M. Pahud, J. J. Dudley, J. Hu, P. O. Kristensson, M. Weerasinghe, K. Č. Pucihar *et al.*, "Quantifying the effects of working in vr for one week," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3810–3820, 2022.

[37] C. Kim, H.-S. Cha, J. Kim, H. Kwak, W. Lee, and C.-H. Im, "Facial motion capture system based on facial electromyogram and electrooculogram for immersive social virtual reality applications," *Sensors*, vol. 23, no. 7, p. 3580, 2023.

[38] H. Lee and Y. Hwang, "Technology-enhanced education through vr-making and metaverse-linking to foster teacher readiness and sustainable learning," *Sustainability*, vol. 14, no. 8, p. 4786, 2022.

[39] D. Adams, A. Bah, C. Barwulor, N. Musaby, K. Pitkin, and E. M. Redmiles, "Ethics emerging: the story of privacy and security perceptions in virtual reality," in *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, 2018, pp. 427–442.

[40] R. Di Pietro and S. Cresci, "Metaverse: security and privacy issues," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2021, pp. 281–288.

[41] F. Xie, Y. Zhang, C. Yan, S. Li, L. Bu, K. Chen, Z. Huang, and G. Bai, "Scrutinizing privacy policy compliance of virtual personal assistant apps," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, 2022, pp. 1–13.

[42] K. Liu, G. Xu, X. Zhang, G. Xu, and Z. Zhao, "Evaluating the privacy policy of android apps: a privacy policy compliance study for popular apps in china and europe," *Scientific Programming*, vol. 2022, 2022.

[43] M. Srinath, S. Sundareswara, P. Venkit, C. L. Giles, and S. Wilson, "Privacy lost and found: An investigation at scale of web privacy policy availability," in *Proceedings of the ACM Symposium on Document Engineering 2023*, 2023, pp. 1–10.

[44] S. I. Becher and U. Benoliel, "Law in books and law in action: the readability of privacy policies and the gdpr," in *Consumer law and economics*. Springer, 2021, pp. 179–204.

[45] S. Liu, B. Zhao, R. Guo, G. Meng, F. Zhang, and M. Zhang, "Have you been properly notified? automatic compliance analysis of privacy policy text with gdpr article 13," in *Proceedings of the Web Conference*, 2021, pp. 2154–2164.

[46] O. of the Attorney General, "Children's online privacy protection act (coppa)," https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa, 2023.

[47] D. Maloney, G. Freeman, and A. Robb, "It is complicated: Interacting with children in social virtual reality," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 343–347.

[48] M. Stoilova, S. Livingstone, and R. Nandagiri, "Digital by default: Children's capacity to understand and manage online data and privacy," *Media and Communication*, vol. 8, no. 4, pp. 197–207, 2020.

[49] K. A. Behnke, M. L. Ambrose, and J. K. Bennett, "Must be 13 to play: Addressing children participation in networked games." in *FDG*, 2014.

[50] F. Nusrat, F. Hassan, H. Zhong, and X. Wang, "How developers optimize virtual reality applications: A study of optimization commits in open source unity projects," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 473–485.

[51] J. Huggins, "Selenium automates browsers," https://www.selenium.dev/, 2023.

[52] Y. Nayanajith, "App apk extractor & analyzer," https://play.google.com/store/apps/details?id=braveheart.apps.apkextract, 2024.

[53] X. S. M. ibotpeaches, "Apktool:a tool for reverse engineering android apk files," https://apktool.org/, 2024.

[54] G. M. Garrido, V. Nair, and D. Song, "Sok: Data privacy in virtual reality," in *24th Privacy Enhancing Technologies Symposium (PETS 2024)*, 2024.

[55] Y. Meier, J. Schäwel, and N. Krämer, "The shorter the better? effects of privacy policy length on online privacy decision-making. media and communication, 8 (2), 291–301," 2020.

[56] European Data Protection Board. (2018) Article 29 working party guidelines on transparency under regulation 2016/679. [Online]. Available: https://edpb.europa.eu/about-edpb/more-about-edpb/legacy-art-29-working-party_en

[57] Michal.Danilk, "Language detection library ported from google's language-detection," https://pypi.org/project/langdetect/, 2021.

[58] J. Li, F. Frulli, S. Clarke, and A. Butz, "Towards balancing real-world awareness and vr immersion in mobile vr," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022, pp. 1–6.

[59] S. Vlahovic, L. Skorin-Kapov, and Z. Car, "An initiative toward an enhanced industry-reported comfort, accessibility, and safety rating system for vr applications," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2023, pp. 95–98.

[60] M. Honnibal, I. Montani, S. V. Landeghem, and A. Boyd, "spacy: Industrial-strength natural language processing in python," 2020.

[61] JetBrains, "dotpeek, free.met decompiler and assembly browser," https://www.jetbrains.com/decompiler/, 2024.

[62] Microsoft, "The roslyn .net compiler provides c# and visual basic languages with rich code analysis apis," https://github.com/dotnet/roslyn, 2024.

[63] Jumboperson, "Il2cppdumper: il2cpp reverse engineer," https://github.com/Perfare/Il2CppDumper?tab=readme-ov-file, 2023.

[64] Unreal, "Unrealpak: An official tool created by epic as part of unreal engine 4," https://github.com/allcoolthingsatoneplace/UnrealPakTool, 2024.

[65] B. Andow, S. Y. Mahmud, W. Wang, J. Whitaker, W. Enck, B. Reaves, K. Singh, and T. Xie, "{PolicyLint}: investigating internal privacy policy contradictions on google play," in *28th USENIX security symposium (USENIX security 19)*, 2019, pp. 585–602.
[66] D. Bui, K. G. Shin, J.-M. Choi, and J. Shin, "Automated extraction and presentation of data practices in privacy policies." *Proc. Priv. Enhancing Technol.*, vol. 2021, no. 2, pp. 88–110, 2021.
[67] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 531–548.
[68] 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)*. IEEE, 2018, pp. 392–408.
[69] N. A. Zhang, C. A. Wang, E. Karahanna, and Y. Xu, "Peer privacy concern: Conceptualization and measurement." *MIS Quarterly*, vol. 46, no. 1, 2022.
[70] C. Yan, R. Ren, M. H. Meng, L. Wan, T. Y. Ooi, and G. Bai, "Exploring chatgpt app ecosystem: Distribution, deployment and security," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 1370–1382.
[71] C. Yan, M. H. Meng, F. Xie, and G. Bai, "Investigating documented privacy changes in android os," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 2701–2724, 2024.
[72] C. Yan, B. Guan, Y. Li, M. H. Meng, L. Wan, and G. Bai, "Understanding and detecting file knowledge leakage in gpt app ecosystem," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 3831–3839.
[73] C. Yan, L. Wan, B. Guan, F. Yu, and G. Bai, "Tracking gpts third party service: Automation, analysis, and insights," in *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 2025, pp. 1602–1606.
[74] Z. Wang, Z. Ma, X. Feng, R. Sun, H. Wang, M. Xue, and G. Bai, "Corelocker: Neuron-level usage control," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 2497–2514.
[75] F. Xie, C. Yan, M. H. Meng, S. Teng, Y. Zhang, and G. Bai, "Are your requests your true needs? checking excessive data collection in vpa app," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–12.
[76] L. Wan, C. Yan, M. H. Meng, K. Wang, and H. Wang, "Analyzing excessive permission requests in google workspace add-ons," in *International Conference on Engineering of Complex Computer Systems*. Springer, 2024, pp. 323–345.
[77] H. Choi, J. Park, and Y. Jung, "The role of privacy fatigue in online privacy behavior," *Computers in Human Behavior*, vol. 81, pp. 42–51, 2018.
[78] Niharika Guntamukkala, Rozita Dara, and Gary Grewal, "A machine-learning based approach for measuring the completeness of online privacy policies," in *In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE*, 2015, pp. 289–294.

APPENDIX

A. Other privacy policy analysis

1) Readability of Privacy Policies: Privacy policies often lead to *privacy policy fatigue* [77] among users due to their complexity and time-consuming nature, necessitating highly readable privacy policies. In addition, given that VR apps have a large child user base, it is crucial to develop privacy policies that are easy to understand and suitable for young age groups. These policies need to be concise and clear, considering the comprehension ability of child users, to ensure they are directly relevant and easily acceptable to them.

We assess the readability from the word/sentence complexity and the document structure. For the former, we employ three widely recognized metrics, i.e., the *automated readability index* (ARI) that is designed to determine the education grade level needed for document comprehension, *Flesch reading ease score* (FRES) that evaluates how easy a text is to understand based on sentence length and word complexity, and *Lesbarhetsindex* (LIX) that assesses the complexity of a text based on the percentage of long words used. For the latter, we use seven metrics that effectively capture the essence of a document's structure, including letters per word (LPW), syllables per word (SPW), words per sentence (WPS), sentence count (SC) , word count (WC), reading time (RT) and speaking time (ST). The results are presented in Figure 6.

2) Completeness of Privacy Policies: The completeness of a privacy policy pertains to how thoroughly its disclosed information meets the requirements set by regulations, store operators, and users. We take as reference GDPR Article 13 [12], which outlines the necessary components of a privacy policy. Moreover, recent studies [65], [78], [14], [45] have summarized the types of content derived from data regulations or user studies.

We utilize QuPer [14], a tool that automatically labels paragraphs within the privacy policies into components. It supports eleven components, including `COLLECT`, `SHARE`, `SECURITY`, `RIGHT`, `CHILDREN`, `REGION`, `UPDATE`, `PROVIDER`, `RETENTION`, `DATA_USE` and `COOKIE`, which are shown in Table IX. We remove the `COOKIE` component, as VR apps are not relevant to browser cookies. Quper is claimed to achieve an F1-score of 82%. We also randomly select fifty privacy policies and manually confirm the identified components by Quper. In Table VII, we summarize the coverage status of the privacy policies among the five app stores.

B. Mapping between the privacy-sensitive data and APIs

Table VIII shows the summary of APIs used to access VR-specific privacy-sensitive data from the official SDK documentation of Pico and Oculus.

TABLE VII: A summary of the status of component coverage in privacy policies among different stores

Component (%) / Store	COLLECT	SHARE	SECURITY	RIGHT	CHILDREN	REGION	UPDATE	PROVIDER	RETENTION	DATA_USE
Oculus	96.3	76.7	61.9	88.0	76.0	32.7	79.6	79.6	30.5	94.6
Viveport	96.7	77.6	60.1	93.2	79.7	26.9	70.8	73.1	21.7	92.5
Pico	91.8	64.3	47.9	85.6	69.2	28.9	66.6	61.3	19.3	88.2
Microsoft	92.9	68.7	52.1	84.8	64.0	34.1	58.8	53.1	24.2	87.2
PlayStation	90.6	68.8	56.3	87.5	71.9	40.6	87.5	90.6	28.1	90.6
Total	93.7	71.2	55.7	87.8	72.2	32.6	72.7	71.5	24.8	90.6

TABLE VIII: The mapping between the privacy-sensitive data (Section III-A1) and APIs

Store	Engine	VPD‡	APIs/Classes
Pico	Unity	Body	`GetBodyTrackingPose, BodyTrackerRole, BodyTrackerResult, BodyTrackerTransform`
		Face	`WantFaceTrackingService, GetFaceTrackingSupported, StartFaceTracking`
		Eye	`UPvr_getEyeTrackingPos, UPvr_getEyeTrackingData, UPvr_getEyeTrackingGazeRay`
		Hand	`GetHandScale, GetJointLocations, GetSettingState`
	Unreal	Body	`PXR.Get_Body_Tracking_Pose, PXR.Set_Swift_Mode, PICOXRMotionTracking`
		Face	`Pico.Get_Face_Tracking_State, Pico.Start_Face_Tracking, EnableFaceTracking`
		Eye	`Pico.Get_Eye_Tracking_Gaze_Ray, Pico.Set_Boundary_Visible, PICOXRHMD,OpenXREyeTracker`
		Hand	`Pico.Get_Handness, PicoMobileController, PICOXRHMD, OpenXRHandTracking`
Oculus	Unity	Body	`OVRBody, OVRBone, OVRCustomSkeleton`
		Face	`OVRCustomFace, OVRCustomFaceExtensions, OVRFace`
		Eye	`OVREyeGaze`
		Hand	`OVRHand`
	Unreal	Body	`OpenXRHMD, OpenXREditor, OpenXR`
		Face	`FacialAnimation`
		Eye	`OculusEyeTracker`
		Hand	`GetHandJointTransform, GetHandJointTransform`

‡ VPD: VR-specific Privacy-sensitive Data Types

TABLE IX: Main components in VR privacy policy

	Component	Description	Example
1	COLLECT	What types of data does the app collect from the user	We collect your information include name, email address, state and technical information such as estimated hand size and hand pose data.
2	SHARE	How the app shares user information	We may share your personal data with third parties.
3	SECURITY	How the app protects user information	These security measures encompass password-protected directories and databases to secure your information, along with SSL (Secure Sockets Layer) technology to guarantee full encryption of your data.
4	RIGHT	Users' rights to their own data	You have the right to stop the advertising messages that we send to you at any time.
5	CHILDREN	Privacy Policy for Child Protection Mechanisms	We do not collect any information from anyone under 13 years of age.
6	REGION	Protection Mechanisms for Some Special Regions	If you are a resident of the state of California, we will abide by the regulations of CalOPPA when handling your information.
7	UPDATE	Information about privacy policy updates	Be aware that this Privacy Policy is subject to periodic updates. For the most current version in effect, please consult our website.
8	PROVIDER	Contact information of the privacy policy provider	If you have questions of your personal data, please send email to us at: xxx@.com.
9	RETENTION	How long will the app keep user data	We will retain your data for 12 months.
10	DATA_USE	How the app will use user data	The data we gather is utilized to enhance our website, aiming to provide you with improved service.

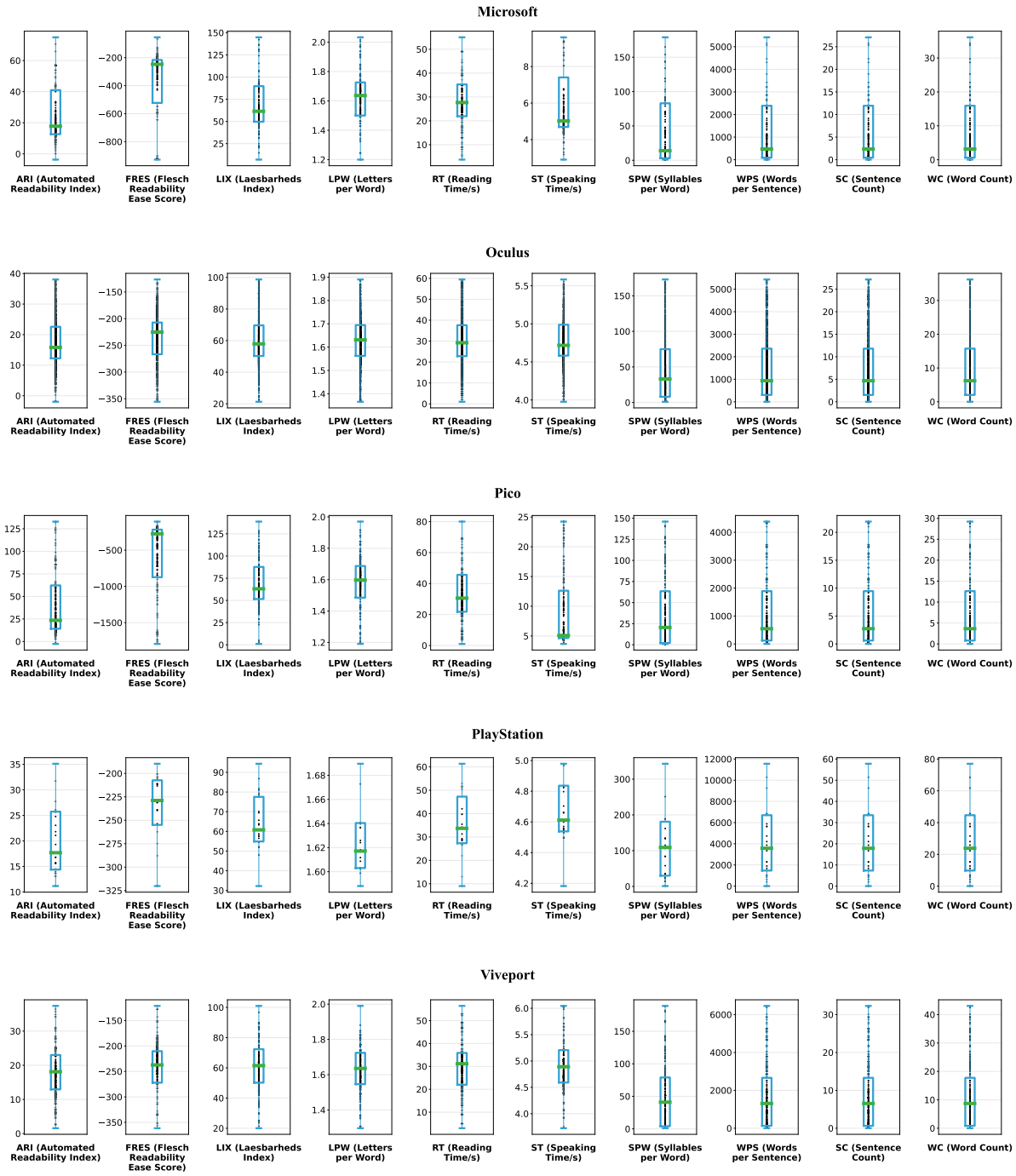

Fig. 6: Result of readability metrics in 5 VR app stores privacy policy