What You Experience is What We Collect: User Experience Based Fine-Grained Permissions for Everyday Augmented Reality

Melvin Abraham
m.abraham.1@research.gla.ac.uk
University of Glasgow
Glasgow, United Kingdom

Mark McGill
Mark.McGill@glasgow.ac.uk
University of Glasgow
Glasgow, United Kingdom

Mohamed Khamis
Mohamed.Khamis@glasgow.ac.uk
University of Glasgow
Glasgow, United Kingdom

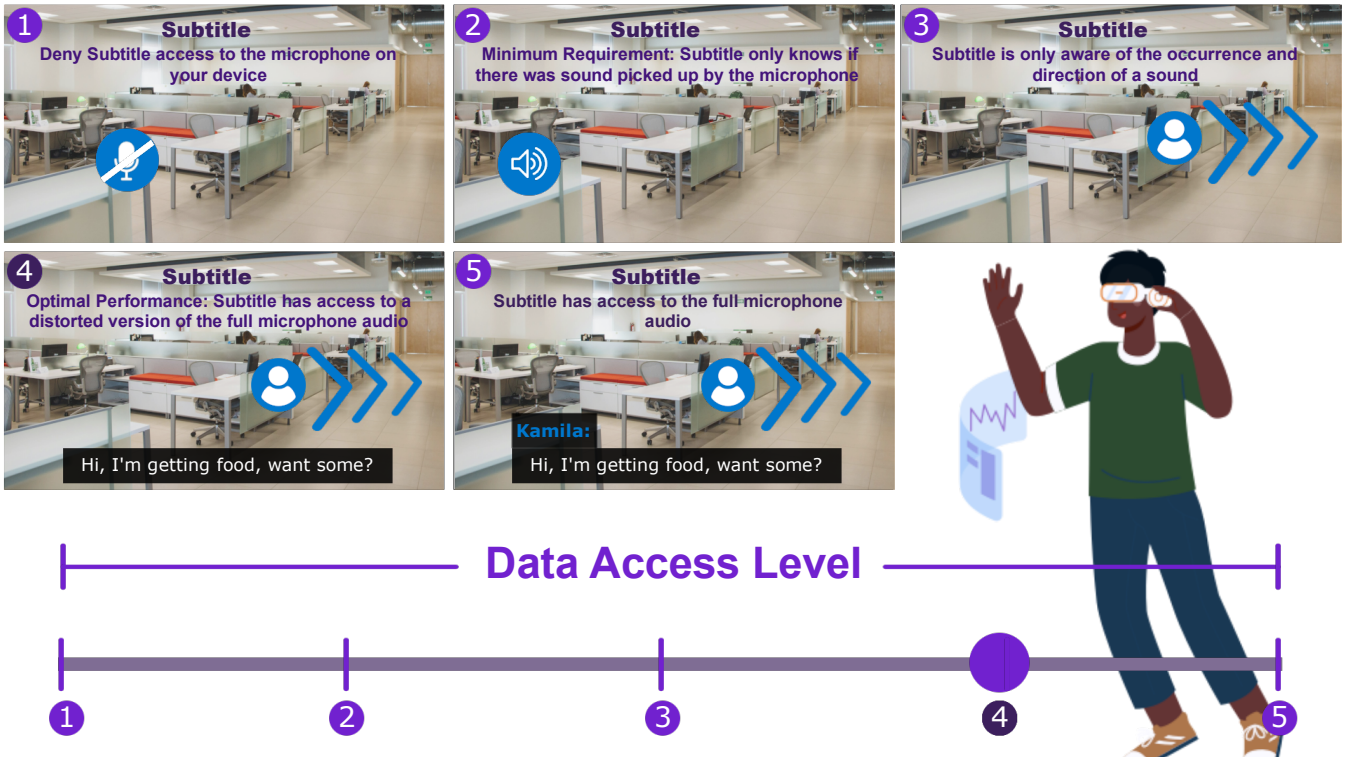

Figure 1: We propose a new permissions access control that accounts for the prolonged use of everyday AR headsets. Our prompt provides users variable control over the fidelity of data they provide to an application while showing a representation of the application user experience at the current data fidelity level. The current application is Subtitle: an app that provides situational awareness to AR users about the people in their surroundings. Images 1 to 5 show mockups we evaluated in AR; each image shows the different levels of data access that can be provided, along with how the application adapts to more data.

ABSTRACT

Everyday Augmented Reality (AR) headsets pose significant privacy risks, potentially allowing prolonged sensitive data collection of both users and bystanders (e.g. members of the public). While users control data access through permissions, current AR systems inherit smartphone permission prompts, which may be less appropriate for all-day AR. This constrains informed choices and risks over-privileged access to sensors. We propose (N=20) a novel AR permission control system that allows better-informed privacy decisions and evaluate it using five mock application contexts. Our system's novelty lies in enabling users to experience the varying impacts of permission levels on not only a) privacy, but also b) application functionality. This empowers users to better understand what data an application depends on and how its functionalities are impacted by limiting said data. Participants found that our method allows for making better informed privacy decisions, and deemed it more transparent and trustworthy than state-of-the-art AR and smartphone permission systems taken from Android and iOS. Our results offer insights into new and necessary AR permission systems, improving user understanding and control over data access.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; • **Security and privacy → Human and societal aspects of security and privacy**.

KEYWORDS

Augmented Reality, Privacy, AR sensing

ACM Reference Format:
Melvin Abraham, Mark McGill, and Mohamed Khamis. 2024. What You Experience is What We Collect: User Experience Based Fine-Grained Permissions for Everyday Augmented Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 24 pages. https://doi.org/10.1145/3613904.3642668

1 INTRODUCTION

Extended Reality (XR) is an umbrella term referring to Augmented (AR), Mixed (MR), and Virtual Reality (VR) [76, 108]. Significant growth in XR adoption has led people to take XR devices outside controlled lab settings. For example, using XR devices in the home [112], or for productivity [69, 70], and even in public [8, 40, 71]. AR devices, in particular, have the potential to see all-day, prolonged, everyday use as we transition towards spatial computing - with devices designed for fashionable, wearable form factors [22, 52, 84].

Everyday AR headsets have rich capabilities to sense [2, 84] the user [2, 20, 84, 94], bystanders around them [84, 86], and their surroundings [1–3, 20, 84]. Current AR headsets are equipped with 'always on' capabilities [2, 52] and will eventually be worn by users throughout the whole day. This long-term use will allow AR sensors capture and reveal a greater extent of information to platforms and applications, subjecting users to increased privacy risks [1, 2, 20, 84, 94]. This may enable AR headsets, platforms, and applications to capture even more data than users consented to in the first place [2, 3, 94] or were even aware was possible [2, 84, 94].

Permissions are the primary method of allowing users control over what data access can be accessed by applications. AR devices predominantly use binary permission prompts, due to reliance of Android permission architectures [4, 73]. Previous work has shown that users do not understand why applications requests permissions [59], what applications will do with the data if granted [27, 55, 56], and how long the data is stored for [27, 56]. Binary permissions bottleneck users into a decision to allow the permission or accept the app will not work as intended [27, 55].

Users face privacy issues when AR headsets only provide a binary decision as to whether or not full access to the headset's sensor data is granted. Providing AR applications with complete unfiltered access to a sensor can leak more information than the user was aware of when they first allowed data access. Sensitive information such as the users' sexuality [58], behaviour traits [10, 47], and emotions [46] can be inferred in real-time to varying levels of accuracy using AR sensing data.

Suppose an application augments the user's desk [12, 69]. The application does not need complete and unrestricted front RGB and depth camera access. The application only needs to know where the user's desk is in relation to the headset; thus, a black-and-white depth scan or mesh scan would suffice. Full front-facing RGB and depth camera access could reveal privacy-invasive insights such as information on surfaces or data about the user's context and those around them. Instead, we propose that AR headsets can use fine-grained permissions to compromise between a user's privacy and the application's functionality. Fine-grained permissions would enable applications to capture only what the application needs to work and nothing more. The majority of consumer AR devices do not come with any level of fine-grained permissions, only allowing users to provide applications with full access to sensor data or none at all.

Our work addresses these challenges by introducing novel fine-grained permission systems for AR devices. We conducted a within-subject study (N = 20) by presenting our participants with five permission control methods. Two of the permission controls were the *Slider w/o Image control* and *Slider w/ Image control*, which are two versions of our novel approach to set an allowable degree of sensing/data access for the given AR application. The permission controls allow users to experience how their chosen level of privacy impacts the application's functionality- enabling users to manage a trade-off between privacy and functionality by, for example, showing the user how the application augments their environment at different levels of data access. We compared the *Slider w/o Image control*, and *Slider w/ Image control* to three state-of-the-art permission controls that we use as baselines: *Binary control*, the default used within AR systems, which is taken from legacy Android smartphone permissions. *Android 11 control*, which is the current implementation for Android 11, which could be expected to be implemented into future Android-based AR devices. *iOS control*, which is the current iOS 16 implementation that could be expected to find its way into rumoured Apple MR headsets in the future [41]. We examined the permission controls by placing the participants in five different mock application contexts. Then participants were presented with the permission controls one at a time to configure and subsequently asked to complete questionnaires and, finally a post-study semi-structured interview.

Our results show the *Slider w/ Image control* performed the best in participants' perceived understanding of the potential effects on their privacy and what data the app will use. Both the *Slider w/ Image control* and the *iOS control* performed the best in participants' perceived understanding of what functionalities the app will provide, and why the permission was requested. Our participants preferred the *Slider w/ Image control* the most as the control was ranked first, followed by *iOS control*, *Slider w/o Image control*, *Android 11 control*, and *Binary control*. Our participants stated when using the *Binary control* and *Android 11 control* the level of data access they provide was based on how much they trust the application. Participants trusted the *Slider w/ Image control*'s information at face value as there was an image of the application provided with the prompt. Participants voiced the *Slider w/ Image control* allowed them more nuanced control of data access due to the slider and the image compared to the other permission controls that seemed to force participants to allow access due to not knowing how well the application would work, if at all.

The contributions of this work are threefold: First, we introduce a new user-experience-based fine-grained permission system explicitly designed for the increased scope and quality of AR data collection. Second, through a user study, we evaluate our proposed

AR permission system with current state-of-the-art permissions systems across different metrics. Third, our results offer insights into designing permissions systems that users perceive to be privacy-protecting.

2 RELATED WORK

We draw on prior work from permissions on smartphones as they are more mature than AR devices and can be seen as the predecessors of XR devices due to the similar sensors and ability to collect sensitive information. Both devices often also rely on the same underlying platforms (e.g. Android used on XReal, Meta Quest, and PICO). Moreover, the permissions used on modern XR systems are the same as those used on mobile phones.

2.1 AR Privacy Issues

Privacy issues caused by data collection within head-worn AR are not the same as those on smartphones [2, 20, 51, 94]. Both devices can use the same physical sensors, such as RGB cameras, depth sensors and microphones; AR devices can also incorporate additional sensors, such as eye tracking and positional tracking sensors. Yet, the data collected when using an AR headset has an increased scope and richness due to the headset's prolonged use and 'always on' capabilities continuously sensing data [2, 3, 95]. For example, AR applications can use behavioural data to disclose a user's sexual preferences [58], emotions [46], and mental state [54].

Prior research shows that AR can amplify existing privacy issues and known challenges [2]. AR applications have access to a more significant data collection that users may not fully comprehend the breadth of personal information that an application can infer by combining different AR sensors [2, 31, 94]. For example, a user's eye gaze hotspots and behavioural data can be integrated to build highly targeted advertisements[2, 74, 94], displaying specific adverts based on the user's emotions that only they can see [2, 74]. The privacy concerns of AR sensing branch further than solely the user [2, 84, 94]. Bystanders around the AR user are equally at risk of being sensed without knowing they are being sensed or providing any consent for the data collection in the first place. [84]. AR devices can have the capability to sense a range of bystander data, such as behaviour, biometric characteristics, and identity [84].

Developers must be cautious when accessing AR sensor data. AR applications requesting full access to raw sensor data provide the application with much more information than is potentially needed [2, 86]. The sensed data could contain sensitive details that the user and bystanders could be uncomfortable with or even unaware they are sharing [3, 84, 94]. Overprivileged access to data can occur when applications request more information than is required for the functionality [27], such as requesting full camera access to track users' hand movements when access to the hand's positional data would be sufficient [51]. Due to a lack of understanding by users of why an application requested full access to the sensor, applications force users to make predictions and assumptions about how their data will be processed, stored, and used [2, 56]. Raw data access from a headset, such as video, audio, and infrared data, can uncover potentially sensitive user information and personally identify users [79, 87]. For example, recent research has shown that users can be personally identified through basic positional data within a VR experience to an accuracy of 94.33% [79] using sensors also present on AR headsets.

In summary, users need to be made more knowledgeable of the extent of information that can be inferred from AR sensors, especially during prolonged headset use. Combining different AR sensors that are '*always-on*' can allow applications to influence and collect private information about a user to which they did not consent to be collected. Hence more work is needed to investigate and control the data flow between the user and the application.

2.2 Permissions

Applications use permissions to facilitate consent from a user to access a resource on a device [44], such as accessing the device's camera or photos. Users then can allow or deny the application access to the requested resource. There are two forms of permissions, install-time and run-time. Install-time permissions are presented to users at the start when first installing the application [28]. The application presents users with a list of permissions the application wants access to; users can typically grant access to all the requested permissions or deny the application access, subsequently stopping the application's installation [102]. In contrast, users are presented with run-time permissions only when the application wants access to a particular resource [5, 28, 102]. Asking users for information when needed helps prevent over-privileged applications, where applications can access data without a need [27]. Run-time permissions, in theory, should correspond clearly with an exact functionality [5]. A clear association between the data requested and the application's functionality should be evident to the user as to why the application wants access to the requested data [5, 13, 113]. Unlike install-time permissions that stop the user from using the application, if the user denies access, the best practice for run-time permissions suggests that the application should still work in general minus the functionality where the denied data is needed [5]. The predominant permission type on Android 6.0 and iOS 6 and above devices is run-time [63, 102], which also includes XR devices like the Meta Quest. Nevertheless, in practice, applications can still present run-time permissions the first time users open the application after installation, thus creating an evolved version of install-time permissions where users struggle to associate permissions with specific functionality.

In summary, permissions come in two forms, install-time and run-time. Install-time permissions, if denied, prevent the application from being installed. Each run-time permission should, in theory, be associated with a specific functionality, and if denied, the application should still be able to run minus the functionality.

2.3 Permissions Controls

Traditionally permissions controls are seen as a binary "*Allow*" or "*Deny*" decision. Users can either allow or deny applications full access to the requested resource. Binary permission controls are still in use today; however, from Android 11 onwards, applications requesting permissions to access the device's location, microphone, or camera can use one-time permissions. One-time permissions provide temporary access to information, allowing users to provide access "*While using the app*", "*Only at this time*", or "*Deny*" [5]. iOS

15 and above also uses both binary permissions of "*Don't Allow*" and "*Allow*"; and a similar temporary control used for location "*Allow while using App*", "*Allow Once*", and "*Don't Allow*" [50]. Currently, AR devices do not have such temporary access permission and only use the standard "*Allow*" and "*Deny*" binary options. However, such existing temporary access permission systems assume the application is not running indefinitely, failing to account for the 'always on' capabilities of everyday AR devices and the extended scope of data collection [2, 3, 20, 94].

Even though Android and iOS permissions have similar permissions controls, the method by which they are presented to the user differs. iOS permissions require developers to add an explanation for why the permission is needed, such as "The app records during the night to detect snoring sounds" [50]. Permissions can explain why the application needs access to specific data to increase users' understanding [59, 113, 114]. However, applications can implement malicious or misleading justifications to trick users into providing access when they might not have otherwise if the justification was accurate [61, 111]. Android permissions can also include explanations, but as an optional feature and not by default [5, 102]. The OS generates the Android prompt by default, hence why the messages are vague. The text is generic enough that developers can use the prompts in multiple situations and applications. On iOS, the application developers write the prompt specifically for the context and application; hence other applications cannot use the same prompt.

Previous work shows that such contextualised text-based explanations similar to the iOS permission prompt performed better than non-contextualised examinations similar to the Android permission prompt for Mobile Augmented Reality applications [44]. Nevertheless, textual and non-textual contextualised justifications are absent within current consumer AR devices. Textual justifications may be a first step to explaining AR permissions better. However, textual justifications do not take advantage of the capabilities of AR devices, such as novel visualisations, displaying non-textual information, and more expansive screen space. Therefore, there is a need for permission systems that leverage AR affordances.

Another method to control permissions is the use of Fine-grained permissions. Fine-grained permissions refer to how precise the data accessed by an application is. iOS and Android can allow users to provide different levels of data availability regarding location data. On Android, a user can select between providing approximate or precise location information to an application [6, 30]. The approximate location permissions allow the app to access the users' location within 3 square kilometres (1.2 square miles). As the name suggests, the precise location permission allows the application to access the users' location within 50 meters or less (160 feet) [6]. Apple implemented a similar fine-grained control for location through a toggle switch and provides visual feedback to the user showing the accuracy of both options [50]. Nevertheless, even when the applications can offer fine-grained control for the users' location, this permission presentation method is not the default prompt. Developers can even evade the option for users to provide approximate location data by requesting only the precise location permission [110] and bypassing interventions explicitly designed to protect user privacy-nudging users into providing more data than necessary.

In summary, permissions controls within AR headsets only give the user a binary decision to allow the application full data access or nothing at all. On both Android and iOS devices, when accessing sensitive sensors, the user can provide the application access based on the session length. However, such a solution would not work for AR headsets as sessions can run indefinitely, once again failing to account for '*always-on*' sensors. The iOS permission prompts are unique and written by developers, while Android permission prompts are generic and provided by the OS. Neither prompt takes advantage of the new capabilities provided by AR headsets, such as infinite screenspace and immersion.

2.4 AR Data Access and Control

There is limited literature on controlling data access within an AR context. Gallardo et al. [31] interviewed several AR users to investigate what privacy concerns may exist in an everyday AR context. The AR users stated that AR headsets will need to account for different contexts that users will be in, and data access controls may need to become context-dependent [31], motivating the need for AR data access controls to provide variable levels of permissions based on the current situation. This work shows that a one-size-fits-all permissions approach is inappropriate for everyday AR as it does not account for prolonged headset use [31].

Previous research has looked at potential OS-level abstractions to provide fine-grained data access for AR headsets [51]. The fine-grained permissions were implemented using recognisers. Recognisers allow applications to access finer-grained permissions rather than access to the complete sensor data [51]. For example, if an application wants to render filters onto a person's face, the recognisers only provide the application with where a person's face and facial features are [51]. Preventing the application from accessing the complete front camera data, which subsequently leaks information about the users' surroundings to the application, providing the application with overprivileged access [51]. Currently, recognisers work at varying accuracy levels based on different situations, thus impacting the users' experience when using the application [51]. Moreover, recognisers do not allow users to customise the level of data access an application has.

Recognisers have also been used as the basis of a world-based data access control [95]. The World-Driven Access Control uses '*passports*' to communicate to the AR headset that it has entered a sensitive location and must alter its data access settings [95]. This work once again motivates the need for AR permissions to be able to adapt and find new ways of providing data rather than providing full raw data access.

Outside of individual data access, previous work has looked at how information can be shared or protected from others in a multi-user AR context [93, 96]. For example, Rajaramet et al. [93] ran an elicitation study with AR and security and privacy experts to design different interaction techniques and how access control is managed between colocated and remote AR users. Ruth et al. [96] examined how augmentations and applications can be secured and shared between users. Both these works account for one type of context and do not explore when users move between locations or allow users to control the level of data access.

Due to a lack of data access guidance for XR developers [2, 3], over recent years, academia [3, 43], industry [49], and government [68] have developed potential security and privacy standards for

general XR. For example, XRSI developed a privacy and safety framework [49] highlighting the need for transparent data handling, user consent mechanisms, and ethical considerations in immersive experiences. XRSI's framework complements broader standards such as GDPR and NIST guidance [117]. Collectively, all these standards and frameworks represent a focused effort to build both adaptable and robust measures for ensuring future XR systems are secure and private.

Previous research [2, 20, 31, 94] motivates an investigation into how AR devices can present fine-grained permission systems that empower users to manage their privacy rather than becoming another hurdle. However, there is no work looking at how users may engage with such fine-grained permissions when given the chance to customise the level of data access over traditional permissions, if at all.

2.5 Usability of Permissions

Past research has shown that when users are presented with runtime permissions, they tend not to read permissions and grant the application access to all the requested data without understanding the consequences of their actions [27, 55]. When users do pay attention to permissions and read the provided information, the permission prompt can lack clarity and be challenging for the user to understand [27, 56]. Permissions are seen to be more complex to comprehend [75]. The lack of clarity is an issue, as previous work has shown that the main factor in when a user allows or denies a permission is based on the user's expectations, their understanding of the application's functionality, and their considerations of privacy consequences at that moment [23, 63, 75, 102, 113].

By requesting permissions to access information, it is clear to users what data is accessed by the application [44, 56]. However, a struggle when interacting with permissions is that the reason why the application requests access to specific data is not apparent to users [56]. Permissions that do not explicitly state a reason or are unclear from the context leave users confused about why the application needs access in the first place [59] — ultimately leading users not to allow the application access to the requested data [9].

Users prefer when permission systems clearly explain why the application has requested access to information. When applications present users with the reasoning behind the data access with transparency, the application is also perceived to be more trustworthy [39, 44, 59, 113, 114]. Building users' trust is vital for applications as users tend to grant permissions for applications they trust [110].

Applications can become overprivileged by requesting access to more resources than required [27]. Previous work has shown that developers tend to get confused over the scope of each permission and thus request more than they require to avoid their application crashing [110]. Users allowing more permissions than is needed can cause a privacy issue later on if the developer starts to collect the information previously granted without the user being aware of the change. Research has shown that users rarely remove permissions after they have been granted for reasons such as a lack of awareness, 'out of sight out of mind', or even believing they had no reason to [60, 64, 101, 110]. Overprivileged applications make it more important than ever to configure permissions correctly when provided with the opportunity to, as users tend not to remove previously granted permissions.

Current AR permission systems need more explicit justifications presented to users for why the applications requested access to data. AR users cannot make an informed privacy decision whether to allow or deny the permission due to the lack of information [2].

In summary, users tend not to read permission prompts, and when they do can easily get confused as many prompts lack clarity. Users must understand why the requested data is needed to make an informed privacy decision. Overprivilaged applications constitute a significant privacy concern as many users do not change their permission settings to revoke previously granted access at a later point.

3 AR FINE-GRAINED PERMISSION CONCEPT

We propose a user-experience-based fine-grained permission control explicitly built for AR. We wanted to explore a method that allows users to improve their privacy by trading parts of the application's functionality and user experience. The underlying concept of our permission control was to combat modern permission controls that ask users to provide applications with complete data access for a long or limited period of time. Our concept instead allows users to offer applications partial data access for an extended period of time. We take advantage of fine-grained permissions to control just how partial the data they provide is. In the following, we address some significant characteristics and requirements of our concept that informed our initial design.

Protects User Privacy: Modern-day privacy is about information [18]. Privacy, by its nature is individualistic, acting as a personal and social construct combining access, confidentiality, and context [65, 80, 81, 105]. Any implementation should be designed with 'Privacy by Design' [57] in mind. Any design, by default, should work to protect as much of the user's data as possible. Any implementation should provide mechanisms to only supply the application with what is needed and nothing more unless specified by the user. Deciding how much privacy or to what extent data is shared is where users should be given the control to configure fine-grained permissions accordingly.

Transparency: Transparency can build trust between an application and its users [2, 110], but greater transparency does not always equate to user trust [2]. Permission controls are the in-between for allowing or not allowing applications access to a resource [27, 44]. Any implementation should communicate with the user what will happen after they make a decision. Communicating the consequences of a decision could take place in the form of an explanation, a visual representation, or even a trial experience. We believe this area has a lot of creative and design space. Unlike mobile permission prompts, AR allows designers to make use of extra real estate outside of the dimensions of a mobile phone screen.

Functionality Specific and Functionality Enhancing: The best practice of standard permissions is to be linked with a functionality [5, 13, 113]. Clear links between permissions and functionality increase user understanding of permissions and why an application has requested it [5, 27, 28, 102]. Unlike standard permissions, if a user declines a request, the functionality will not work [5]. We suggest that a fine-grained permission request can be directly equated

to adding functionality to the application but, more importantly, increasing the quality of preexisting functionality. The aim is to make applications work at different data access levels, allowing users to increase functionality as more data is provided rather than creating an all-or-nothing stand-off between the user and the application.

4 DESIGN AND IMPLEMENTATION

The following permission controls were our interpretation of the concept outlined above. The proposed designs varyingly meet the design considerations noted around privacy, transparency, and functionality by providing a descriptive fine-grained slider UI for controlling the extent of allowable permissions granted to the application. These designs aim to explore whether users would make privacy-functionality trade-offs if given such control. As in our study there are 4-5 options for the user to choose from, we use a slider with a "magnet effect" allowing the user to slide only to the options rather than anything between. A different design of the permission control method showing a finite set of options like radio buttons would work but the advantage of discrete sliders is that it communicates to users a form of 'more or less' in this case of providing more or less data. Discrete sliders have also been used by previous work in XR devices to communicate steps of a complex continuum [38]. The implementations are only one version of how a user-experience-based fine-grained permission control may look and should not be mistaken as the best nor only implementation.

Slider w/ Image control This permission control is our novel proposed permission system that allows users to set their own level of access to the requested resource with a slider (see Figure 2A). The permission prompt allows users to experience how their chosen access level impacts the app's functionality and user experience—empowering users to customise their own trade-off between user privacy and functionality. The popup is titled with the app name and a sentence explaining the fidelity of the data provided at the current slider level. Optionally one sentence is labelled with "Minimum Requirement" as the lowest level of data fidelity for the app to work to some functional but impaired degree. Another sentence is labelled with "Optimal Performance" to represent the optimal trade-off between privacy and functionality where the level of data fidelity allows the app to work as intended. The "Optimal Performance" label is in place to communicate to the user when the segment on the slider provides all the base functionality, thus implying there may not be a need to provide full data access. Under the explanation sentence is an screenshot image of the actual application. The screenshot is there to show the user what the application and the functionalities will look and behave like at the current slider position. Below the image is a UI slider segmented to the number of data fidelity options available. The explanation sentence changes to describe the fidelity option when the user moves the slider.

Slider w/o Image control This permission control is the same as the *Slider w/ Image control* but without an image element to show the app's functionality at the current slider position (see Figure 2B). The image is removed to compare how users prefer the addition of a graphical explanation compared to only text.

4.1 Baselines

The following permission controls were used as baselines to compare against the permission controls we developed. We chose the *Binary control*, *Android 11 control*, and *iOS control* as they are the state-of-the-art permission systems that are available and openly used today. This study focuses on the evaluation of permission control methods, specifically looking at the prompt designs rather than the interaction methods. As most AR systems are based on Android and use the older *Binary control*, we included the *Android 11 control* as it is the updated version of how Android systems currently provide data access. The addition of the *iOS control* follows the same reasoning, as visionOS used in Apple's Vision Pro may base data access on iOS. It is important to note that in this study, the design of the baseline permission prompts mirror the actual designs in the real-world, thus only providing the exact same text and information, contributing to our studies ecological validity in comparing to real-world baselines.

Binary control This is the existing AR permission system used as a baseline (see Figure 2C). This permission system is the traditional permission prompt and the standard on today's AR glasses. This prompt states the specific permission requested and gives the user a binary decision to either decline or accept the permission by pressing the "Deny" or "Allow" buttons.

Android 11 control This is the standard system used in Android 11 [5] (see Figure 2D). The prompt is titled with the app's name and a sentence stating the permissions the app requests. The user has the decision to allow the app access to the requested resource when the app is in the foreground, only providing access to the requested resource once, or declining access by pressing the "While using the app", "Only at this time", or "Deny" buttons.

iOS control This control is based on the iOS 16 permission prompt [50] (see Figure 2E). The prompt is titled "Allow [app name] to access [permission name]" and an explanation sentence stating why the app has requested the permission below the title. The user has the decision to allow the app access to the requested resource when the app is in the foreground, only providing access to the requested resource once, or declining access by pressing the "Allow While Using App", "Allow Once", or "Don't Allow" buttons.

4.2 Technical Implementation

All the permission control methods were implemented using C# within Unity with the NRSDK 1.9.1 [83]. We loaded the permissions controls directly onto the XReal Light AR headset (1920x1080 px per eye, 60Hz refresh rate) [82] to use as a standalone headset, not to require a connection to external computing power. While users can interact with the permission control methods using both hand tracking and the Nreal controller, only the controller was used for this study due to the performance of the XReal hand tracking under variable lighting conditions.

5 STUDY DESIGN

The study was designed as within-subjects. There were two independent variables: the *permission control methods* (see Figure 4) and the *contexts* (see subsection 5.1). Both IVs had five levels, thus creating twenty-five unique conditions in total for all the participants to experience. The order of the contexts and permissions control

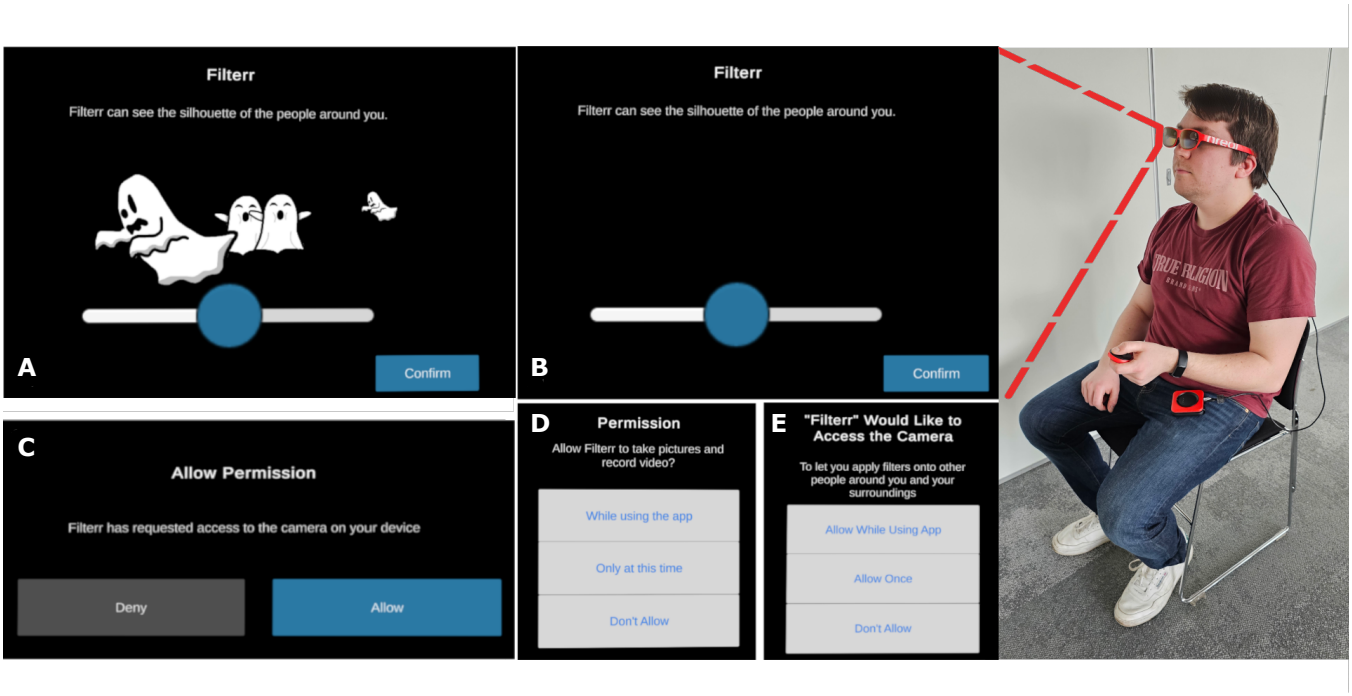

Figure 2: The participant on the right can be seen wearing a pair of XReal Light AR glasses. While wearing the AR glasses, the participant is presented with and asked to evaluate the different AR prompts one at a time for each of the five contexts, in this case, "*Filterr*". Filterr is a hypothetical app allowing AR users to apply face and body filters onto others around them. A) Shows the *Slider w/ Image control* providing the app access to only the silhouette of the people around them. The image of the ghosts represents a filter being applied on top of bystanders. The ghosts' size is based on how far away the bystanders are from the user. B) Is the *Slider w/o Image control*, which presents the same information minus the representation of the app's user experience at the current slider position. C) Is the *Binary control*, which is the current permission prompt used within AR devices. D) Presents the *Android 11 control*, which allows the user to provide data access for different durations. E) Is the *iOS control*, which provides the user with a sentence of textual explanation for why the permission was requested, along with allowing the user to provide data access for different durations.

methods presented to the participant were counterbalanced using a 5x5 Latin Square [115] to prevent learning effects.

5.1 Contexts

Five different contexts were formulated to evaluate our control methods so the participants could have enough *"hands-on"* time to judge and configure the permission control methods reasonably. Each context is an application archetype upon which we would test our envisaged permission prompts. **Subtitle:** This app provides situational awareness to AR users about the people in their surroundings. This application archetype is based on the work from Google [33, 85]. **Filterr:** This app allows AR users to apply face and body filters onto other people. This application archetype is based on Snapchat [107]. **Room Designer:** This is an interior design app that allows AR users to design the room they are currently in. This application archetype is based on the Ikea Place app [48]. **StreetNav:** This app provides AR users with directions to walk in order to navigate to a location. This application archetype is based on Google Maps [34]. **AR Health:** This app tracks AR users' mental and physical health, and its archetype is based on Apple Health [7]. Each of the contexts was a means of presenting different permission prompts for different application archetypes. Hence, we did not have a functional application for any context and did **not** collect any of the data the permissions requested.

5.2 Permission Control Methods

Five permission control variables were used for this study as presented in Figure 4. Two fine-grained permission methods ***Slider w/o Image control*** and ***Slider w/ Image control***. Three baselines from state-of-the-art permission methods ***Binary control***, ***Android 11 control***, and ***iOS control***. During the study the *Android 11 control* and *iOS control* were referred as "*Data Access Without Description Control*" and "*Data Access With Description Control*" as we did not want to bias the participants by showing brand names or collect results based on any preconceived notions of the two companies privacy practices.

5.2.1 Permission Controls Texts. For each of the contexts, the permission controls text changed. The slider segments for the *Slider w/o Image control* and *Slider w/ Image control* are associated with a specific AR API that currently has the capability to manipulate and process the sensor data in this way by default. The segments for Subtitle are derived from the Android Media and Audio API. The segments for Filterr are derived from the StereoLabs ZED SDK 3. The segments for Room Designer are derived from the Microsoft Mixed Reality API. The segments for StreetNav are derived from the Google Android Location API and Google's AR Core. Finally, the segments for AR Health are derived from both the Google Android and Apple iOS API's. The text on the *Binary control* is taken from the Meta store. The *Android 11 control* text is from the Android Play Store. The *iOS control* text is taken from the iOS App Store.

The *Binary control* and *Android 11 control* displays a message asking for access to a sensor. The *iOS control* displays a contextualised justification for a sensor. Both Sliders present a slider ranging from 1 (Deny) to 5 (Full sensor access). For a complete detailed view of the permission control text, see Table 4-5, in the Appendix.

5.3 Dependent Variables

After each condition, the participants were asked to complete unweighted NASA-TLX questionnaire [45] to measure perceived workload for each permission control method. Next, participants are asked questions on perceived understanding used in [44] and [53]. Once the participants have completed all twenty-five conditions at the end of the study, participants were asked to fill out 5-point Likert questions on perceived usability and trust and rank the permission control methods in order of preference. These questions were asked at the end so that the participants could have sufficient time to use each permission control method. The usability questions are based on the System Usability Scale (SUS) [15] and a question used in [44].

5.4 Post-Study Semi-Structured Interview

The post-study semi-structured interview was conducted to gain more in-depth insights into the decisions made by the participants about their mindset and thinking when based on the actions in the user study and questionnaire responses. The researcher followed an interview guide (see subsection A.1) to ensure consistency among all the participants while having the freedom to explore and discuss topics brought up during the interview. Interview questions examined comprehension and attitudes towards the permission control methods experienced, in particular contrasting the *Slider w/o Image control* and *Slider w/ Image control* with participant's prior experience both with mobile device permissions and the baselines presented in this study, before finally discussing enhancements to the fine-grained permission controls.

6 METHODOLOGY

A within-subject study was conducted to evaluate the different AR permission control methods. The study aimed to understand to what extent AR users would engage with fine-grained preferences, pictorial and textual descriptions' influence on permission configuration, and attitudes towards adopting fine-grained permission prompts. This study involved participants wearing an AR headset to interact with different permissions controls within different application contexts, followed by questionnaires to reflect on the overall conditions and ending with a semi-structured interview.

6.1 Recruitment and Demographics

Our Institution's Research Ethics Board granted ethics approval before conducting the research. The complete study took 60 minutes of the participants' time; the participants received £10 Amazon gift cards as reimbursement. We recruited 20 participants by sending invitations over a university mailing list, posting on social media, and word of mouth. All potential participants had to be aged 18+ to take part. Our participants were within the age range of 20-40 (M = 27.45, SD = 5.30). Nine participants self-identified as female and eleven self-identified as male. The option of non-binary and other was available yet was not selected. Zero participants declared they had not heard of AR; two participants declared they had never used AR but knew of it; nine declared they had used mobile AR infrequently (AR games such as Pokemon GO, AR applications); one declared they had used mobile AR frequently (AR games such as Pokemon GO, AR applications); and nine declare they have used an AR headset (Hololens, VR headset with passthrough).

We used the *Internet Users' Information Privacy Concerns questionnaire* (IUIPC) [62] to assess participants' general privacy attitudes. The IUIPC's score is between 1 and 7, where 7 represents a high privacy attitude. Participants rated their wish for control (M = 5.96, SD = 0.05), awareness (M = 6.43, SD = 0.30), and the perceived ratio between benefit and collection (M = 5.77, SD = 0.10). A limitation of privacy questionnaires is that they only provide the *theoretical* privacy attitudes of the user and do not provide insight into if the attitudes are present in practice [32]. Nonetheless, the mean scores implying the participants represent a high general privacy attitude.

6.2 Study Procedure

Once the participant was ready to begin the study, they wore a pair of XReal Light AR glasses [82]. The lead researcher explained to the participant how to use the AR headset and controller. The experiment had five different contexts, each with five permission control methods, allowing the participants to configure the requested permission(s). The order of the contexts and permissions controls presented to the participant was counterbalanced using a 5x5 Latin Square [115]. The participant started the study when presented with a context while wearing the AR headset. The researcher read a description of the context as stated in subsection 5.1. After confirming they understood the context and needed no further repetition, the participant was shown a permission control method presented in subsection 5.2. The participant was then asked to explore each option and make an informed choice based on if they were using this application in reality. After the participant configured the permissions, they were asked to complete a questionnaire on a tablet. The participant was asked not to take off the AR glasses while filling out the questionnaire so they could refer back to the permission control method at any time. Once the participant interacted with all five permissions control methods, this process was repeated for all the other contexts.

After completing all conditions, the participant removed the AR headset, completed another questionnaire, and ranked the permission control methods on a tablet. Printouts of the five permission control methods were given to the participant to aid their memory. Once the participant completed the questionnaires, a semi-structured interview was conducted (see subsection 5.4).

6.3 Limitations

Our sample size is small compared to a global sample size. However, a power analysis was conducted using R studio [91, 109] with the pwrss package [16]. Setting an alpha of 0.05 as the significance criterion and a power of 80% [29], a sample of 20 participants is sufficient to determine a medium effect size ($\eta_p^2 = 0.06$) or greater with a repeated measures ANOVA. The power analysis assumes a perfect sample of participants, which in practice is extremely difficult to recruit from a local sample [66]. Nevertheless, our within-subject study has more participants than the average accepted number of local participants for a within-subject study in the human-computer-interaction field [17]. While the pairwise comparisons were carefully conducted with corrections applied to mitigate potential biases, no additional corrections were applied on top of the already adjusted values within potential single-family hypotheses. Thus, care is required when generalising the results to broader contexts [21, 118].

We also cannot completely rule out that participants did not fully understand the context descriptions and may have configured the permissions in a way they might not have if their understanding was correct. However, as detailed in subsection 6.2, we took all precautions to ensure the participant understood the context, double-checking by asking if they understood the context and repeating the context when needed. Participants may not have understood a question or term before answering the question. However, to minimise confusion, the researcher was present in the room and made sure the participant knew they could ask questions or request clarification if needed.

6.4 Analysis

Quantitative: We performed an Aligned-Rank Transform (ART) [24, 116] using the R version of ARTool [26] to convert our non-parametric data into a format to conduct both a one or two-way repeated-measures ANOVA for statistical significance testing. ART enables parametric tests to be conducted on non-parametric data, in our case, Likert-scale responses or data that was non-normal distributions [100, 116]. The effect sizes are reported when the partial eta squared (η_p^2) values are as follows: **small** when $0.01 \geq \eta_p^2 < 0.06$, **medium** when $0.06 \geq \eta_p^2 < 0.14$, and **large** when $\eta_p^2 \geq 0.14$. These values were chosen based on previous work [29, 77]. We report significance values where p equals or is less than *0.05*. In cases of significant differences, we run ART-C *post hoc* contrasts [25] tests using Tukey corrections to correct the alpha value for multiple comparisons [25, 116]. We do not explicitly report *post-hoc* contrasts for the interaction effects between context and the permission control method as 300 pairwise comparisons are available hence, presenting meaningful results would not be feasible. However, we present the interaction graphs within the Appendix for by-eye comparisons and tables indicating significant differences (seesubsection A.2).

Qualitative: Data from the semi-structured interview were coded using inductive coding [67] to develop a qualitative codebook. A single coder iteratively coded the data to account for emerging codes. Quotes from the interview were grouped using codes. Codes were not defined before the coding process and emerged through patterns and similar answers within the data. The lead and secondary authors reviewed the final codes list and grouped similar codes into main themes.

7 RESULTS

7.1 Understanding Permissions Control

Figure 3: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on a boxplot. The *Slider w/ Image control* and *iOS control* performed the best in terms of perceived understanding of why the application requested the data, the effect providing the data has on the user's privacy, what functionality the app will provide with the data, and what data the app will use. The *Binary control* and *Android 11 control* performed poorly for user understanding.

Understanding of data used. Statistical tests for the participant's understanding of **what data the app will be able to use** showed significant differences (*"The permission control method gives me a good understanding of what data the app will be able to use: 1=Strongly disagree; 5=Strongly agree"*, see Figure 3 for distribution of participant responses). The ANOVA suggests that the main effect of Context was not statistically significant and **small** ($F(4) = 2.35$, $p = 0.054$; $\eta_p^2 = 0.02$). The main effect of Permission Control Methods was statistically significant and **large** ($F(4) = 32.07$, $p < 0.001$; $\eta_p^2 = 0.22$). The interaction between Context and the Permission Control Methods was statistically significant and **medium** ($F(16) = 2.49$, $p = 0.001$; $\eta_p^2 = 0.08$). *Post hoc* analysis revealed the *Binary control* performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p < 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *Android 11 control* also performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p = 0.0018$), and the *Slider w/ Image control* ($p < 0.0001$). The *iOS control* performed statistically

significantly better than the *Slider w/o Image control* ($p = 0.0001$) yet was still beaten by the *Slider w/ Image control* ($p < 0.0001$). The *Slider w/ Image control* went onto outperform the *Slider w/o Image control* ($p < 0.0001$). This shows that clear descriptions and images allow participants to understand better what data the app would be able to use.

Understanding of application functionality. Statistical tests for the participant's understanding of **the functionalities of the applications** showed significant differences (*"The permission control method gives me a good understanding of what data the app will be able to use: 1=Strongly disagree; 5=Strongly agree"*, see Figure 3 for distribution of participant responses). The ANOVA suggests that the main effect of Context is statistically significant and **small** ($F(4) = 3.23$, $p = 0.012$; $\eta_p^2 = 0.03$). The main effect of the Permission Control Methods is statistically significant and **large** ($F(4) = 133.88$, $p < 0.001$; $\eta_p^2 = 0.54$). The interaction between Context and the Permission Control Methods is statistically significant and **medium** ($F(16) = 3.11$, $p < 0.001$; $\eta_p^2 = 0.10$). *Post hoc* analysis for Context revealed that the Subtitle context performed statistically significantly worse than both the RoomDesigner ($p = 0.0248$) and StreetNav ($p = 0.0212$) contexts. *Post hoc* analysis for the Permission Control Methods revealed that the *Binary control* performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p < 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *Android 11 control* also performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p < 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *iOS control* performed statistically significantly better than the *Slider w/o Image control* ($p = 0.0001$). The *Slider w/ Image control* performed better than the *Slider w/o Image control* ($p < 0.0001$). These results show that the *Slider w/ Image control* and *iOS control* were the best at communicating the application's functionalities.

Understanding of potential effect on privacy. Statistical tests for the participant's understanding of **permission's potential effect on their privacy** (*"The permission control method gives me a good understanding of the potential effect to my privacy: 1=Strongly disagree; 5=Strongly agree"*, see Figure 3 for distribution of participant responses). The ANOVA suggests that the main effect of Context was not statistically significant **and small** ($F(4) = 0.62$, $p = 0.646$; $\eta_p^2 = 5.44e-03$). The main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 24.58$, $p < 0.001$; $\eta_p^2 = 0.18$). The interaction between Context and the Permission Control Method was statistically not significant **and small** ($F(16) = 1.33$, $p = 0.173$; $\eta_p^2 = 0.04$). *Post hoc* analysis revealed the *Binary control* performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p = 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *Android 11 control* also performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p < 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). Finally the *Slider w/ Image control* went onto outperformed both the *Slider w/o Image control* ($p = 0.0044$) and the *iOS control* ($p = 0.0018$). These results show that the *Slider w/ Image control* was the clearest method for the participants to understand what would happen to their privacy based on their permission choice.

Understanding of why data was requested. Statistical tests for the participant's understanding of **why the applications needed access** to the requested permission (*"I understood why the application needed access to the permissions that were being requested: 1=Strongly disagree; 5=Strongly agree"*, see Figure 3). The ANOVA suggests that the main effect of Context was statistically significant and **small** ($F(4) = 3.06$, $p = 0.017$;$\eta_p^2 = 0.03$). The main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 72.16$, $p < 0.001$; $\eta_p^2 = 0.39$). The interaction between Context and Type is statistically significant and **medium** ($F(16) = 1.97$, $p = 0.014$; $\eta_p^2 = 0.06$). *Post hoc* analysis for the Context revealed no significant differences between contexts. *Post hoc* analysis for the Permission Control Methods revealed the *Binary control* performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p = 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *Android 11 control* also performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p = 0.0003$), and the *Slider w/ Image control* ($p < 0.0001$). Finally, the *Slider w/o Image control* was outperformed by both the *iOS control* ($p < 0.0001$) and the *Slider w/ Image control* ($p = 0.0018$). These results show that the *Slider w/ Image control* and *iOS control* allowed the participants to see the correlation between the permission requested and why the app needed access to that data easier than the other permission control methods.

7.2 SUS Based Usability Scales

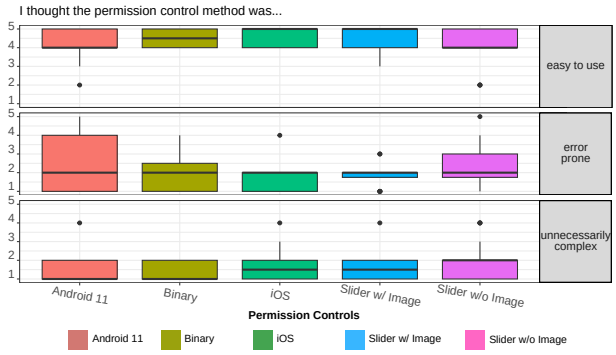

Figure 4: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on a boxplot. Each factor is taken from the SUS Usability Survey. All the permission controls scored similarly for being perceived to be error prone. The *Slider w/o Image control* was perceived to be more unnecessarily complex compared to the other permission controls. All the permission controls were perceived as easy to use with the *Slider w/ Image control* and *iOS control* scoring the highest.

The SUS based usability questions were asked after the participants completed all 25 unique conditions, hence the only factor is the Permission Control Method. See Figure 4 for distribution of participant responses).

Easy to Use. Statistical tests showed a significant difference between the Permission Control Methods in their perceived **ease of**

use (*"I thought the {control method name} was easy to use: 1=Strongly disagree; 5=Strongly agree"*). The ANOVA suggests that the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 4.85$, $p = 0.002$; $\eta_p^2 = 0.20$). *Post hoc* analysis revealed the *Slider w/o Image control* performed statistically significantly worse than the *iOS control* ($p = 0.0026$), and the *Slider w/ Image control* ($p = 0.0159$). These results show that the participants perceived the *Slider w/ Image control* and *iOS control* as easier to use than the *Slider w/o Image control*.

Error Proneness. There were no significant differences between the Permission Control Methods in perceived **error proneness** (*"I thought the {control method name} was error-prone: 1=Strongly disagree; 5=Strongly agree"*).

Unnecessary Complexity. Statistical tests showed a significant difference between the Permission Control Methods in their perceived **Unnecessary Complexity** (*"I found the {control method name} was unnecessarily complex: 1=Strongly disagree; 5=Strongly agree"*). The ANOVA suggests that the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 3.35$, $p = 0.014$; $\eta_p^2 = 0.15$). *Post hoc* analysis revealed the *Slider w/o Image control* performed statistically significantly worse than the *Binary control* ($p = 0.0049$). These results show that the *Slider w/o Image control* without an accompanying image may be considered overly complicated, showing how important the addition of the visual representations are.

7.3 Trust

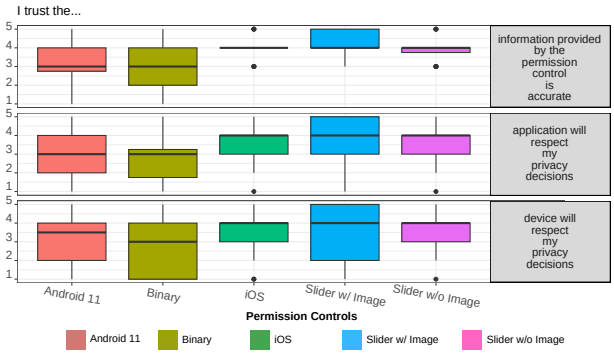

Figure 5: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on a boxplot. Participants trusted the *Slider w/ Image control*, *Slider w/o Image control*, and *iOS control* the most to provide accurate information. Leading them to perceive that both the app and device will respect their privacy decisions. The *Binary control* and *Android 11 control* both performed the worst regarding trust.

The Trust questions were asked after the participants completed all 25 unique conditions, hence the only factor is the Permission Control Method. See Figure 5 for distribution of responses.

Trust of Information Accuracy. Significant differences were found when the participants were asked if they trusted the **information provided by the permission methods was accurate** (*"I trust the information provided by the permission control is accurate"*). The ANOVA suggests that the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 11.96$, $p < 0.001$; $\eta_p^2 = 0.39$). *Post hoc* analysis revealed the *Binary control* performed statistically significantly worse than the *iOS control* ($p = 0.0016$), the *Slider w/o Image control* ($p = 0.0051$), and the *Slider w/ Image control* ($p < 0.0001$). As well as the *Android 11 control* also performing statistically significantly worse than the *iOS control* ($p = 0.0042$), the *Slider w/o Image control* ($p = 0.0125$), and the *Slider w/ Image control* ($p < 0.0001$). These results show that the participants trusted the information provided by the permission prompt was accurate more when the permission control method explained why the application requested the data along with the name of the sensor, either through text as seen on the *iOS control* and *Slider w/o Image control* or image-based like the *Slider w/ Image control*.

Trust of Device. After running statistical tests on how much the participants trusted the **AR device**, significant differences between the Permission Control Methods were found (*"I trust the device will respect my privacy decisions"*). The ANOVA suggests that the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 5.15$, $p = 0.001$; $\eta_p^2 = 0.21$). *Post hoc* analysis revealed the *Binary control* performed statistically significantly worse than the *iOS control* ($p = 0.0071$), the *Slider w/o Image control* ($p = 0.0296$), and the *Slider w/ Image control* ($p = 0.0015$).

Trust of Application. Significant differences were found for how much the participants trusted the **application** (*"I trust the application will respect my privacy decisions"*). The ANOVA suggests that the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 6.20$, $p < 0.001$; $\eta_p^2 = 0.25$). *Post hoc* analysis revealed the *Android 11 control* performed statistically significantly worse than the *iOS control* ($p = 0.0325$), and the *Slider w/ Image control* ($p = 0.0150$). The *Binary control* also performed statistically significantly worse than the *iOS control* ($p = 0.0050$), the *Slider w/o Image control* ($p = 0.0478$), and the *Slider w/ Image control* ($p = 0.0021$). These results show that the participants trusted the application more when the permission control method explained why the application requested the data, either through text as seen on the *iOS control* and *Slider w/o Image control* or image-based like the *Slider w/ Image control*.

7.4 Privacy Decision Opportunities

The Privacy Decision Opportunities questions were asked after the participants completed all 25 unique conditions, hence the only factor is the Permission Control Method.

When the participants were asked how well each permission control method provided them with the opportunity to set the perceived best privacy decision, significant differences were found after the 5-point Likert questions were analysed (*"The permission control method provided the opportunities to set the best privacy decision: 1=Strongly disagree; 5=Strongly agree"*, see Figure 6 for distribution of participant responses). The ANOVA suggests that the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 29.57$, $p < 0.001$; $\eta_p^2 = 0.61$). *Post hoc* analysis revealed the *Android 11 control* performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o*

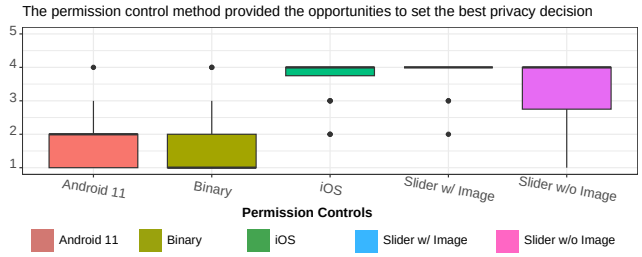

Figure 6: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on a boxplot. The *Slider w/ Image control*, *Slider w/o Image control*, and *iOS control* were perceived best by participants to provide the best opportunity to make the best privacy decisions. The *Binary control* and the *Android 11 control* performed poorly.

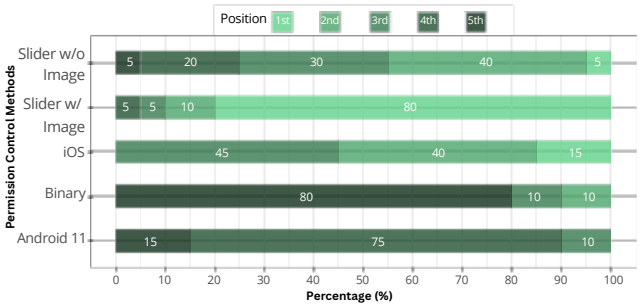

Figure 7: The figure shows the percentages of how many times the participants ranked a permission method at a particular position. The *Binary control* was ranked 5th (last) 80% of the time, while the *Slider w/ Image control* was ranked 1st 80% of the time.

Image control ($p = 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *Binary control* also performed statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p < 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). Finally, the *Slider w/ Image control* performed statistically significantly better than the *Slider w/o Image control* ($p = 0.0261$). The results show that the *Binary control* and the *Android 11 control* performed poorly compared to the rest of the control methods. The results indicate that the participants need more information than just the name of the sensor being requested.

7.5 Perceived Workload

The overall NASA-TLX scores for the *Android 11 control* was 8.65 (SD = 10.47), the *Binary control* was 9.92 (SD = 11.82), the *Slider w/ Image control* was 10.68 (SD = 9.41), the *Slider w/o Image control* was 12.32 (SD = 13.19), and finally, the *iOS control* was 12.89 (SD = 12.76). Statistical analysis showed the main effect of Permission Control Method is statistically significant and **large** ($F(4) = 3.38$, $p = 0.013$; $\eta_p^2 = 0.15$). *Post hoc* analysis showed statistically significant differences between the *iOS control* (Mdn= 8.66) and the *Android 11 control* (Mdn= 3.33, $p = 0.036$). These results show on the unweighted NASA-TLX only the *Android 11 control* scored low, and the rest scored medium for perceived workload [42, 45, 88]. Note that the medium level in the NASA-TLX is derived from the original weighted scale's 21/3 division.

7.6 Permission Control Method Rankings

Participants ranked the permission control methods in the order of most to least preferred (see Figure 7 for distribution of permission control method rankings). The preferred permission control method was the *Slider w/ Image control* (Mdn= 1st), then *iOS control* (Mdn= 2nd), then *Slider w/o Image control* (Mdn= 3rd), then *Android 11 control* (Mdn= 4th), and finally the *Binary control* (Mdn= 5th). Statistical analysis showed the main effect of the Permission Control Method was statistically significant and **large** ($F(4) = 46.21$, $p < 0.001$; $\eta_p^2 = 0.71$). *Post hoc* analysis revealed the *Binary control* ranked statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p < 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). The *Android 11 control* also ranked statistically significantly worse than the *iOS control* ($p < 0.0001$), the *Slider w/o Image control* ($p = 0.0001$), and the *Slider w/ Image control* ($p < 0.0001$). Finally the *Slider w/ Image control* ranked statistically significantly better than the *Slider w/o Image control* ($p < 0.0001$), and the *iOS control* ($p = 0.0058$)

7.7 Semi-structured Interview

We present qualitative data sectioned into three themes based on the clustering of individual qualitative codes.

7.7.1 Understanding information and making decisions. As reflected in subsection 7.1, participants found the *Binary control* difficult to understand. P11 voiced "*it doesn't really tell me anything, it does tell me what is being used but it gives me absolutely no indication for anything other than that and that's just not enough for me for anything AR related*". Participants also stated that the simplicity of the *Binary control* left them guessing "*what I'm actually agreeing to and when and how it's going to be used*" (P2).

Our participants brought up how they had to decide if they trusted the app first before allowing data access when presented with the *Binary control* and the *Android 11 control* due to the lack of information for why the data is needed: "*[My thoughts were] do you want to use this or not, and they didn't give you any reason why or what it was going to use, so it was very much do I trust this app?*".

Conversely, participants appreciated the inclusion of a visual representation of the application functionality as it allowed them to build a clearer mental modal and decide how much information to provide: "*it kind of yeah just helps you to better understand what and how much you give access to*" (P3). The combination of both text and image worked well together, stated by P16 "*So the text gave me like enough of an idea, but the image showed me like a practical scenario, so I guess the image or any kind of compliment to the text that was already there*".

7.7.2 Deciding levels of data access. Participants mentioned that when using the *Binary control*, *Android 11 control*, and *iOS control*, they felt forced to accept the permissions worrying the application would not work. P19 voiced "*it's slightly forcing you down the path*

of agreeing because you don't understand the consequences of saying no and if you can still use that same." In contrast, participants mentioned that they have more control over how much data they provide when using both the *Slider w/o Image control* and the *Slider w/ Image control*. P3 stated, "*it's not an all or nothing scenario so... it gives me more autonomy to choose what data you want to give away*". P12 voiced "*it felt like I had a lot more power in deciding what I would actually like to give up because I can see how it will affect the functionality*" when referring to the *Slider w/ Image control*.

7.7.3 Limitations of, and Alterations to, fine-grained permissions design. Some participants mentioned struggling to match the data being accessed with the functionality being provided based only on the visual representation of the application: "*most of the time, the picture put across the meaning of the description, but not always*" (P10), and "*The descriptions, in some cases were also very helpful and sometimes the image didn't quite convey everything as well as it could*" (P20). The same comments were mentioned for sensors that users may not be familiar with, such as eye tracking: "*I didn't actually understand that it was using eye tracking*" (P5).

Participants also proposed a number of variants to the *Slider w/ Image control* design, including having an application slider with optional sensor sliders for fine-tuning of permissions (P9); associating sliders with specific functionality exposed to applications (e.g. location, health) rather than overall application functionality (P8); and combining the slider approach with user-configurable session-based (P13) and location-based (P12,P16) data access constraints for more granular control of when a permission applies.

8 DISCUSSION

8.1 Evidencing the Need for Fine-Grained Permissions for AR

Our study showed that *Slider w/ Image control* performed the best for the user's perceived understanding of the permission control method. Participants had a good understanding of why data was being requested, what the app would do with the data, what functionalities would be available when the app had data access, and finally, what effects allowing data access has on the participant's privacy. The addition of the descriptive imagery over *Slider w/o Image control* also helped participants understand why the permission was requested compared to only providing textual explanations, with no perceived impact on usability, being just as easy to use as the current AR binary controls.

Information privacy in the literature is traditionally seen as personal and based highly on current contexts [18, 65, 80, 81, 105]. Viewed through such a lens, users are in control of their individual privacy when they engage with permission controls and make decisions on what private information to share. However, literature has also challenged the idea of individualism and privacy [11, 92, 99, 104], particularly where others privacy attitudes (or lack of) impact more than the user themselves [11, 92, 99, 104]. For example on social media, no matter how disciplined a users privacy attitudes are, it makes no difference if the users friend of posts, shares or leaks private information about them [11, 99].

In a future where everyday AR headsets that are packed with sensing capabilities are worn by multiple people at any given time, privacy could also be considered a collective problem. Literature has shown applications shift the privacy risk assessment of allowing permissions to their users [110]. Placing privacy decisions in the hands of the user requires the user to have a well-rounded view and access to all the information first. Hence, for users to even have a chance to make a fair and informed decision, users need to have a strong understanding of the implications of their choices in allowing or denying data access. Conversely, both *Binary control* and *Android 11 control* performed poorly in how well participants understood the permission control methods. *Binary control* was consistently last, and *Android 11 control*, when not tied last, came second last. Both the permission control methods present the least information, thus placing last reflects previous work that permission prompts that show very little information are more challenging for users to understand [27, 56, 75]. These results also show that the participants were cautious about their privacy when considering the new level of privacy invasion that AR can potentially bring [2, 94].

AR (and XR permissions, more generally) have thus far typically copied and pasted from smartphone permission architectures. We posit that this is a short-term solution, as AR has different privacy risks than smartphones [2]. Hence as AR devices develop and grow in consumer adoption, we argue that permissions controls such as the *Slider w/ Image control* built specifically for AR are needed to enhance the privacy of consumer AR devices and support users in determining an acceptable privacy/functionality trade-off. When users have better privacy controls, everyone benefits. Our results emphasise the utility of, and need for, such an approach.

8.1.1 Supporting Better Privacy Decisions from the Start. Past work shows that users rarely modify permissions after allowing access [60, 64, 101, 110]. Users treating permission setting as a 'one-and-done' procedure adds to the importance of providing users with the complete information to make informed decisions [23, 63, 75, 102, 113] early and when given the opportunity. Our results imply that users can make better privacy decisions the first time around when presented with a user-experience-based fine-grained permission system, and rankings and qualitative feedback affirm that participants preferred the permission control methods that gave them fine-grained controls over data access.

8.1.2 The Benefits of Transparency: Supporting Trust in AR Apps and Devices. The use of fine-grained permissions also has potentially beneficial consequences beyond supporting user privacy. In terms of trusting the accuracy of the information, the application, and the device, the *Slider w/ Image control*, *Slider w/o Image control*, and *iOS control* were tied for first place. In effect, participants' trust increased when the permission control transparently communicated the application functionality and the scope of data capture. This is important because previous research shows that when users trust the application, they are more likely to provide more data access [39, 44, 59, 113, 114]. If permission controls can inspire greater trust, they are likely to then encourage users to grant some (rather than no) data access, which could consequently increase the acceptance and adoption of consumer AR devices and applications in the near future.

8.2 Challenges in Supporting Fine-Grain Permissions for Everyday AR In Practice

Whilst our *Slider w/ Image control* design offers a usable, preferable means of supporting users in decision making around the privacy/functionality trade-off for AR apps, there are a number of challenges that would need to be addressed to put such a design into practice.

8.2.1 Creating and Validating Granular Permission Prompts. Writing effective permission descriptions such as the ones seen in the *iOS control* and *Slider w/o Image control* poses challenges for developers to implement. Writing a message that is both detailed and concise in a few sentences, and can accurately summarise the extent of data capture and the resultant functionality of the application, is an extra burden placed upon the shoulders of developers and copywriters [36]. Any such content would need to be validated by a trusted third party such as the AR platforms themselves, much as Apple for example currently validate app-specific messages for camera permissions[1]. The *Slider w/ Image control* could also be seen to increase the burden for developers and platforms to inform privacy decisions meaningfully - but by using screenshots, illustrations, or live previews, it can be feasible to envisage creating useful visualisations that could be accomplished with only modest effort/input from developers, and be standardised across platforms.

Leaving the application companies to create the segments will bring into question the validity of the segments themselves. Currently there is no infrastructure preventing developers from making incorrect segments or providing insufficient trade-offs for the user to decide from. A possible solution is to place the responsibility of checking the trade-offs to a third party such as the app store. Apple and Meta both already check applications data access before they can be added to their app stores [50, 90], checking the validity of the segments could be added to the existing checklist. Applications could provide the list of proposed segments and need to justify how each segment follows a sensible data access gradient.

8.2.2 Supporting Granularity in Existing Sensing APIs. The implementation of *Slider w/ Image control* also requires developing crucial supporting infrastructure. At a low level, platforms and SDKs could provide more granular APIs that discretize existing sensors' resolution, sample rate and accuracy [19, 78]. We could envisage for example, requesting low resolution, time-limited, or selectively obfuscated/censored RGBD, eye-tracking, physiological and microphone data that could enhance privacy whilst limiting an application's access only to what is justifiable. At a higher level, platforms could provide granular wrapper APIs that provide developers only with the computed results or data they need rather than the underlying raw sensor data, e.g. requesting tracking of nearby planar surfaces rather than receiving an RGBD feed. Given such granularity, we could envisage having standardised descriptions, imagery and visualisations that map to defined granular low/high-level APIs, further easing the burden on developers around defining understandable permission prompts. The benefit of the *Slider w/ Image control* design lies in the ability to adapt to permissions evolving towards increasing granularity. The design can incorporate further discrete points based on the needs of the application or the granularity of the underlying APIs. This also means that the slider could potentially set data access levels on a continuous scale if appropriate e.g. to account for the large data capture abilities of everyday AR headsets, and extent of control users might wish to exert over consequent application permissions and resulting functionality. For example, consider an application where proximity dictates what it senses of reality - in such a case, a near-continuous slider would be preferable.

Beyond individual sensor access control, research has also shown that sensitive information can be inferred by combining data from multiple AR sensors [2, 87, 103]. Hence it is paramount that developers and platforms have a basis for understanding which sensor combinations represent more invasive and potentially privacy-sensitive decisions, and communicate this to users. Subsequently, developers and/or platforms will need to be able to define and describe the data access levels for their applications fairly to users. The data access levels must enable users to make informed, meaningful trade-offs between their privacy and the application's functionality. As presented in our implementation of the *Slider w/ Image control*, developers could provide users with an "Optimal" data access level that ideally provides almost perfect performance and functionality while still providing users with a degree of privacy.

8.2.3 Vulnerabilities in Proposed Slider-Based Permission Prompts. We note that our proposed *Slider w/ Image control* design could be exposed to potential misuse or attack through deceptive designs/dark patterns, designs implemented to deceive the user into doing something that is not in their best interest [14, 35, 37]. The developed-defined content of the Slider prompt is an obvious vulnerability. Developers could make images for the settings that provide the most sensor data more appealing to users than lower data access levels. Creating misleading images can act as clickbait, showing functionality too good to be true or of unrealistic performance degradation in lower data access levels. Developers could also state that the application's functionality cannot be implemented with less data access. A potential solution to combat *'clickbait'* sliders could be allowing the image element to only show a screenshot of the actual application without any supplementary text or alteration.

Outside of the images being used to mislead users, malicious applications could use the "Optimal Performance" tag to nudge users to a certain level of data access. As mentioned, the spirit of the tag is to represent a data access level that provides the user with most or all functionality without providing full data access. As there is currently no way of regulating such a rule, developers could place the tag on the highest level of data access. Such a tag, in theory would be effective to potentially speed through a permission prompt while protecting user privacy. Yet, if used dishonourably in practice, this may cause more harm than good. Meaning this should be monitored and moderated in any eventual real-world implementation.

Another potential misuse of the *Slider w/ Image control* is that applications bypass this permission method entirely by setting the slider prompt text to "Move the slider to the right" or "Move the slider to here", indicating the users to move the slider to the maximum point. Dark Patterns such as this would allow an application to access over-privileged data and go against the spirit of the *Slider*

[1] https://developer.apple.com/documentation/avfoundation/capture_setup/requesting_authorization_to_capture_and_save_media, last accessed 29/08/2023.

w/ Image control. iOS had a similar problem with the background of an application behind the permission prompt asking users to allow full permission and displaying an arrow to users pointing at the "Allow" button [50]. Apple has since banned such applications' access to the Apple app store and provided clear rules for how applications can communicate to users regarding permissions to data [50]. A similar rule would be a simple method of stopping the *Slider w/ Image control* from being misused. As previously discussed, we see platform validation of such descriptions as being an integral part of safeguarding users.

8.3 Future Directions

8.3.1 Transitioning from Smartphone to AR-Specific Permission Prompts. We recommend that AR devices transition away from the traditional binary permissions currently in use due to privacy concerns, lack of user understanding, and trust issues, as found in our research. Our findings open the door to further exploring AR-specific permission control methods rather than moving systems from one medium (smartphones) to another (AR/XR headsets). When designing AR-specific permission controls, utilising AR features such as immersiveness would aid in moving away from the reliance on text-based permission prompts that are present on other devices. For example, as shown in Prange et al. [89], various AR elements were placed in the user's space to communicate the presence of any IoT devices and potential privacy intrusions. As we mentioned in section 3, AR headsets could offer a multitude of approaches for immersively visualising and explaining the permission decisions being requested from users. Augmenting the user's surroundings could go further and replace the images used in our *Slider w/ Image control* implementation altogether. For example, if the user was placed in the AR app and allowed to interact with the permission to change the data access levels, they could potentially see the differences visualised in real-time.

8.3.2 Towards Usable, Automated Data Access Controls. Our findings also motivate investigating different methods to handle data access control outside or complementary to permissions. New AR permission control methods need to consider that everyday AR headsets will be equipped with '*always-on*' sensors that are continuously sensing information. AR specific permission prompts need to consider different approaches to allow users to provide temporary, session-based and location-based access to data. Current smartphone methods to provide temporary access commonly rely on an active session, such as "Only at this time" or "Only while using" which can get blurred in an AR context when everyday AR headsets run multiple applications for long extended periods. To account for headsets being in multiple locations in one wear session, AR headsets could dynamically configure permissions on the fly. Roesner et al. [95] proposed a permission system that used digital '*passports*' to disable or alter AR sensors when the headset was in a sensitive location. Setting digital passports to dynamically configure permission in line with what the user is comfortable being collected could be introduced using '*user privacy profiles*'. Users could set a privacy profile at the headset's first installation or start-up, similar to VR headsets asking users to set a play boundary [72]. If a user enters a location that does not have permissions configured already, the user should not be burdened again to set and alter their profiles manually. Adding the *Slider w/ Image control* on top of the Roesner et al.'s [95] passports could enable a clearer understanding for users of how the application will work within a location's determined data access level or the user's preset privacy profile.

An example of how an AR permission system can account for the prolonged use of AR headsets is by implementing a '*least privilege needed until told otherwise*' [98] approach. Hence the headset restricts all data access in a new location until the user explicitly grants permission. However, such an approach is naive, as functionality could be blocked or degraded indiscriminately regardless of the user's comfort and awareness of the risks. Moreover, removing data access whenever the user enters a new location would be a usability nightmare, as the user would need to manually reset their permissions multiple times a day. A more promising approach may be to implement a '*context privacy manager*' that predicts settings based on past behaviours from the user or incorporates different baseline permission configurations based on the user's context. Such a manager would remove the burden from the user of manually re-configuring their permissions continuously throughout the day. An AR context privacy manager should go beyond the singular dimension of location and account for more nuanced information, such as *why* the context requires a specific level of privacy, e.g. the user is in a vulnerable state or dealing with confidential information. When combined with the *Slider w/ Image control*, the context privacy manager could move the controls slider to different positions rather than the binary choice of allowing or blocking data access. Consequently, the only impact the user faces is the level of functionality available at a given point in time.

8.4 Towards Fine-Grained Permissions for Everyday AR

Our paper has outlined and evidenced the need for permission control methods that can better address the privacy challenges posed by everyday AR in the near future, proposing that platforms move away from binary permission prompts towards fine-grain permissions that enable users to make informed decisions in balancing their privacy versus the AR application functionality. For such a vision to become a reality, collaboration between developers, academia, and platform creators is needed. All parties must put time and effort into further developing privacy protections that allow applications access to the data they need, such as fine-grained permissions or the *Slider w/ Image control*. While challenging, such initiatives are still possible with time. For example, fine-grained location permissions are now readily available for both Android and iOS [6, 30, 50] and provide precedent to be extended for other AR sensors. In this work, we placed the *Slider w/ Image control* within five contexts based on realistic apps that could be seen within everyday AR headsets. A direction that should be evaluated is how well the *Slider w/ Image control* performs given the nuances of the user's data access behaviours within more specific *high-value* or *high-impact* application archetypes, considering the predominant use cases that eventually emerge around consumer everyday AR such as immersive video streaming [97] or augmented productivity [71].

Moreover, permission systems built specifically for AR should carefully consider the burden on AR users to engage with fine-grained permissions. Whilst our proposed designs did not impact perceived user workload to a problematic degree, there is significant scope for variation here, in terms of the complexity of the prompt information and how it is visually conveyed, that would necessitate carefully assessing the privacy-utility trade-off [106] and the consequent burden placed on users in any eventual real-world implementation. We argue that such an effort must be made to safeguard user privacy before we see consumers' anticipated mass adoption of everyday AR.

9 CONCLUSION

In this paper, we explored permission methods to allow AR users to control their privacy effectively. We presented and evaluated (N=20) five different AR permission control methods across five different usage contexts. We introduced a novel permission control method to explore how users customise the trade-off between their privacy and application functionality. We found that our participants prefer permission control methods that allow them fine-grained controls over the data they provide compared to prompts that request full access, even when users can provide temporary access, such as "Allow once". Our participants preferred permission controls that allowed them to experience why the data was requested and how the data would add to the app's functionality through graphical and textual explanations. Moreover, we found that our participants were more likely to trust the app when it is transparent with users about requesting data. Our work creates space for new permission control methods built explicitly for head-worn AR to be explored - methods that consider the advantages, challenges and differences of AR compared to other devices, such as smartphones, in order to protect user privacy.

ACKNOWLEDGMENTS

This research is supported by an EPSRC DTP studentship (EP/T517896/1).

REFERENCES

[1] Melvin Abraham and Mohamed Khamis. 2021. Communicating Security & Privacy Information in Virtual Reality. *1st International Workshop on Security for XR and XR for Security* (2021).
[2] Melvin Abraham, Pejman Saeghe, Mark McGill, and Mohamed Khamis. 2022. Implications of XR on Privacy, Security and Behaviour: Insights from Experts. In *NordiCHI '22: Proceedings of the 12th Nordic Conference on Human-Computer Interaction: Participative computing for sustainable futures*.
[3] Devon Adams, Alseny Bah, Catherine Barwulor, Nureli Musaby, Kadeem Pitkin, and Elissa M Redmiles. 2018. Ethics emerging: the story of privacy and security perceptions in virtual reality. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. 427–442.
[4] Google Android. 2022. Request app permissions. https://developer.android.com/guide/topics/permissions/overview#runtime
[5] Google Android. 2022. Request app permissions. https://developer.android.com/training/permissions/requesting
[6] Google Android. 2022. Request location permissions. https://developer.android.com/training/location/permission
[7] Apple. 2022. Apple Health. https://www.apple.com/uk/ios/health/
[8] L. Bajorunaite, S. Brewster, and J.R. Williamson. 2021. Virtual Reality in transit: how acceptable is VR use on public transport?. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*.
[9] Rebecca Balebako, Florian Schaub, Idris Adjerid, Alessandro Acquisti, and Lorrie Cranor. 2015. The impact of timing on the salience of smartphone app privacy notices. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*. 63–74.
[10] Shlomo Berkovsky, Ronnie Taib, Irena Koprinska, Eileen Wang, Yucheng Zeng, Jingjie Li, and Sabina Kleitman. 2019. Detecting personality traits using eye-tracking data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[11] Andrew Besmer and Heather Richter Lipford. 2010. Moving beyond untagging: photo privacy in a tagged world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1563–1572.
[12] Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Magic desk: bringing multi-touch surfaces into desktop work. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2511–2520.
[13] Bram Bonné, Sai Teja Peddinti, Igor Bilogrevic, and Nina Taft. 2017. Exploring decision making with {Android's} runtime permission dialogs using in-context surveys. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. 195–210.
[14] Harry Brignull, Marc Miquel, Jeremy Rosenberg, and James Offer. 2015. Dark Patterns-User Interfaces Designed to Trick People. *Retrieved November* 30 (2015), 2021.
[15] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
[16] Metin Bulus. 2023. *pwrss: Statistical Power and Sample Size Calculation Tools*. https://CRAN.R-project.org/package=pwrss R package version 0.3.1.
[17] Kelly Caine. 2016. Local standards for sample size at CHI. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 981–992.
[18] Fred H Cate. 1997. Privacy in the information age. (1997).
[19] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. 2021. A privacy-preserving approach to streaming eye-tracking data. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2555–2565.
[20] Jaybie A De Guzman, Kanchana Thilakarathna, and Aruna Seneviratne. 2019. Security and privacy approaches in mixed reality: A literature survey. *ACM Computing Surveys (CSUR)* 52, 6 (2019), 1–37.
[21] Pierre Dragicevic. 2016. Fair statistical communication in HCI. *Modern statistical methods for HCI* (2016), 291–330.
[22] Chloe Eghtebas, Francisco Kiss, Marion Koelle, and Paweł Woźniak. 2021. Advantage and Misuse of Vision Augmentation–Exploring User Perceptions and Attitudes using a Zoom Prototype. In *Augmented Humans Conference 2021*. 77–85.
[23] Yusra Elbitar, Michael Schilling, Trung Tin Nguyen, Michael Backes, and Sven Bugiel. 2021. Explanation beats context: The effect of timing & rationales on users' runtime permission decisions. *USENIX Security'21* (2021).
[24] Lisa A Elkin, Matthew Kay, James J Higgins, and Jacob O Wobbrock. 2021. An aligned rank transform procedure for multifactor contrast tests. In *The 34th annual ACM symposium on user interface software and technology*. 754–768.
[25] Lisa A. Elkin, Matthew Kay, James J. Higgins, and Jacob O. Wobbrock. 2021. An Aligned Rank Transform Procedure for Multifactor Contrast Tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 754–768. https://doi.org/10.1145/3472749.3474784
[26] Lisa A Elkin, Matthew Kay, James J Higgins, and Jacob O Wobbrock. 2023. ARTool- Align-and-rank data for nonparametric factorial ANOVA. https://depts.washington.edu/acelab/proj/art/
[27] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. 2011. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*. 627–638.
[28] Adrienne Porter Felt, Kate Greenwood, and David Wagner. 2011. The effectiveness of application permissions. In *2nd USENIX Conference on Web Application Development (WebApps 11)*.
[29] Andy P. Field. 2017. *Discovering statistics using IBM SPSS statistics* (5th ed.). SAGE Publications, London.
[30] Huiqing Fu and Janne Lindqvist. 2014. General area or approximate location? How people understand location permissions. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. 117–120.
[31] Andrea Gallardo, Chris Choy, Jaideep Juneja, Efe Bozkir, Camille Cobb, Lujo Bauer, and Lorrie Cranor. 2023. Speculative Privacy Concerns About AR Glasses Data Collection. *Proceedings on Privacy Enhancing Technologies* 4 (2023), 416–435.
[32] Nina Gerber, Paul Gerber, and Melanie Volkamer. 2018. Explaining the privacy paradox: A systematic review of literature investigating privacy attitude and behavior. *Computers & security* 77 (2018), 226–261.
[33] Google. 2022. Google Keynote (Google I/O '22). https://www.youtube.com/watch?v=nP-nMZpLM1A
[34] Google. 2022. Google Maps. https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en_GB&gl=US&pli=1
[35] Colin M Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L Toombs. 2018. The dark (patterns) side of UX design. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–14.
[36] Matthew Green and Matthew Smith. 2016. Developers are not the enemy!: The need for usable security apis. *IEEE Security & Privacy* 14, 5 (2016), 40–46.

[37] Saul Greenberg, Sebastian Boring, Jo Vermeulen, and Jakub Dostal. 2014. Dark patterns in proxemic interactions: a critical perspective. In *Proceedings of the 2014 conference on Designing interactive systems*. 523–532.

[38] Uwe Gruenefeld, Jonas Auda, Florian Mathis, Stefan Schneegass, Mohamed Khamis, Jan Gugenheimer, and Sven Mayer. 2022. VRception: Rapid prototyping of cross-reality systems in virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–15.

[39] Jie Gu, Yunjie Calvin Xu, Heng Xu, Cheng Zhang, and Hong Ling. 2017. Privacy concerns for mobile app download: An elaboration likelihood model perspective. *Decision Support Systems* 94 (2017), 19–28.

[40] Jan Gugenheimer, Christian Mai, Mark McGill, Julie Williamson, Frank Steinicke, and Ken Perlin. 2019. Challenges using head-mounted displays in shared and social spaces. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–8.

[41] Hector Hamish. 2023. Apple VR headset news, leaks, and what we want to see. https://www.techradar.com/news/apple-vr-headset

[42] Peter A Hancock and Najmedin Meshkati. 1988. *Human mental workload*. North-Holland Amsterdam.

[43] Jassim Happa, Anthony Steed, and Mashhuda Glencross. 2021. 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, 397–398.

[44] David Harborth and Alisa Frik. 2021. Evaluating and Redefining Smartphone Permissions with Contextualized Justifications for Mobile Augmented Reality Apps. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 513–534.

[45] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[46] Steven Hickson, Nick Dufour, Avneesh Sud, Vivek Kwatra, and Irfan Essa. 2019. Eyemotion: Classifying facial expressions in VR using eye-tracking cameras. In *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 1626–1635.

[47] Sabrina Hoppe, Tobias Loetscher, Stephanie A Morey, and Andreas Bulling. 2018. Eye movements during everyday behavior predict personality traits. *Frontiers in human neuroscience* (2018), 105.

[48] Ikea. 2022. Ikea Places App. https://apps.apple.com/ie/app/ikea-place/id1279244498

[49] XR Safety Initiative. 2020. The XRSI privacy framework.

[50] Apple IOS. 2022. Accessing private data. https://developer.apple.com/design/human-interface-guidelines/patterns/accessing-private-data/

[51] Suman Jana, David Molnar, Alexander Moshchuk, Alan Dunn, Benjamin Livshits, Helen J Wang, and Eyal Ofek. 2013. Enabling {Fine-Grained} Permissions for Augmented Reality Applications with Recognizers. In *22nd USENIX Security Symposium (USENIX Security 13)*. 415–430.

[52] Ross Johnstone, Neil McDonnell, and Julie R Williamson. 2022. When virtuality surpasses reality: possible futures of ubiquitous XR. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–8.

[53] Ankit Kariryaa, Gian-Luca Savino, Carolin Stellmacher, and Johannes Schöning. 2021. Understanding users' knowledge about the privacy and security of browser extensions. USENIX.

[54] Christina Katsini, Yasmeen Abdrabou, George E Raptis, Mohamed Khamis, and Florian Alt. 2020. The role of eye gaze in security and privacy applications: Survey and future HCI research directions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–21.

[55] Patrick Gage Kelley, Sunny Consolvo, Lorrie Faith Cranor, Jaeyeon Jung, Norman Sadeh, and David Wetherall. 2012. A conundrum of permissions: installing applications on an android smartphone. In *International conference on financial cryptography and data security*. Springer, 68–79.

[56] Patrick Gage Kelley, Lorrie Faith Cranor, and Norman Sadeh. 2013. Privacy as part of the app decision-making process. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 3393–3402.

[57] Marc Langheinrich. 2001. Privacy by design—principles of privacy-aware ubiquitous systems. In *Ubicomp 2001: Ubiquitous Computing: International Conference Atlanta Georgia, USA, September 30–October 2, 2001 Proceedings*. Springer, 273–291.

[58] Daniel J Liebling and Sören Preibusch. 2014. Privacy considerations for a pervasive eye tracking world. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. 1169–1177.

[59] Jialiu Lin, Shahriyar Amini, Jason I Hong, Norman Sadeh, Janne Lindqvist, and Joy Zhang. 2012. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the 2012 ACM conference on ubiquitous computing*. 501–510.

[60] Fen Liu, Guanghui Xu, Qi Wu, Qing Du, Wei Jia, and Mingkui Tan. 2020. Cascade reasoning network for text-based visual question answering. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4060–4069.

[61] Xueqing Liu, Yue Leng, Wei Yang, Wenyu Wang, Chengxiang Zhai, and Tao Xie. 2018. A large-scale empirical study on android runtime-permission rationale messages. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 137–146.

[62] Naresh K Malhotra, Sung S Kim, and James Agarwal. 2004. Internet users' information privacy concerns (IUIPC): The construct, the scale, and a causal model. *Information systems research* 15, 4 (2004), 336–355.

[63] Nathan Malkin, David Wagner, and Serge Egelman. 2022. Runtime Permissions for Privacy in Proactive Intelligent Assistants. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. 633–651.

[64] Prashanthi Mallojula, Javaria Ahmad, Fengjun Li, and Bo Luo. 2021. You Are (not) Who Your Peers Are: Identification of Potentially Excessive Permission Requests in Android Apps. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 114–121.

[65] Kirsten Martin. 2016. Understanding privacy online: Development of a social contract approach to privacy. *Journal of business ethics* 137 (2016), 551–569.

[66] Florian Mathis, Kami Vaniea, and Mohamed Khamis. 2022. Prototyping usable privacy and security systems: Insights from experts. *International Journal of Human–Computer Interaction* 38, 5 (2022), 468–490.

[67] Philipp Mayring. 2014. Qualitative content analysis: theoretical foundation, basic procedures and software solution. (2014).

[68] Mark McGill. 2021. White Paper-The IEEE Global Initiative on Ethics of Extended Reality (XR) Report–Extended Reality (XR) and the Erosion of Anonymity and Privacy. *Extended Reality (XR) and the Erosion of Anonymity and Privacy-White Paper* (2021), 1–24.

[69] Mark McGill, Stephen Brewster, Daniel Pires De Sa Medeiros, Sidney Bovet, Mario Gutierrez, and Aidan Kehoe. 2022. Creating and Augmenting Keyboards for Extended Reality with the K eyboard A ugmentation T oolkit. *ACM Transactions on Computer-Human Interaction* 29, 2 (2022), 1–39.

[70] Mark Mcgill, Aidan Kehoe, Euan Freeman, and Stephen Brewster. 2020. Expanding the bounds of seated virtual workspaces. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 3 (2020), 1–40.

[71] Mark McGill, Gang Li, Alex Ng, Laura Bajorunaite, Julie Williamson, Frank Pollick, and Stephen Brewster. 2022. Augmented, Virtual and Mixed Reality Passenger Experiences. In *User Experience Design in the Era of Automated Driving*. Springer, 445–475.

[72] Meta. 2023. Set up your boundary for Meta Quest. https://www.meta.com/en-gb/help/quest/articles/in-vr-experiences/oculus-features/boundary/

[73] Oculus Meta. 2022. Meta Quest Apps Must Target Android 10 Starting September 29. https://developer.apple.com/design/human-interface-guidelines/patterns/accessing-private-data/

[74] Abraham Hani Mhaidli and Florian Schaub. 2021. Identifying manipulative advertising techniques in xr through scenario construction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–18.

[75] Kristopher Micinski, Daniel Votipka, Rock Stevens, Nikolaos Kofinas, Michelle L Mazurek, and Jeffrey S Foster. 2017. User interactions and permission use on android. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 362–373.

[76] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. 1995. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, Vol. 2351. Spie, 282–292.

[77] Peter E Morris and Catherine O Fritz. 2013. Effect sizes in memory research. *Memory* 21, 7 (2013), 832–842.

[78] Vivek Nair, Gonzalo Munilla Garrido, and Dawn Song. 2022. Going incognito in the metaverse. *arXiv preprint arXiv:2208.05604* (2022).

[79] Vivek Nair, Wenbo Guo, Justus Mattern, Rui Wang, James F O'Brien, Louis Rosenberg, and Dawn Song. 2023. Unique identification of 50,000+ virtual reality users from head & hand motion data. *arXiv preprint arXiv:2302.08927* (2023).

[80] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.

[81] Helen Nissenbaum. 2011. A contextual approach to privacy online. *Daedalus* 140, 4 (2011), 32–48.

[82] Nreal. 2022. NRSDK Light. https://www.nreal.ai/light/

[83] Nreal. 2022. NRSDK Overview. https://nreal.gitbook.io/nrsdk/nrsdk-fundamentals/core-features

[84] Joseph O'Hagan, Pejman Saeghe, Jan Gugenheimer, Daniel Medeiros, Karola Marky, Mohamed Khamis, and Mark McGill. 2023. Privacy-Enhancing Technology and Everyday Augmented Reality: Understanding Bystanders' Varying Needs for Awareness and Consent. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–35.

[85] Alex Olwal, Kevin Balke, Dmitrii Votintcev, Thad Starner, Paula Conn, Bonnie Chinh, and Benoit Corda. 2020. Wearable subtitles: Augmenting spoken communication with lightweight eyewear for all-day captioning. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1108–1120.

[86] Joseph O'Hagan, Julie R Williamson, Mark McGill, and Mohamed Khamis. 2021. Safety, power imbalances, ethics and proxy sex: Surveying in-the-wild interactions between vr users and bystanders. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 211–220.
[87] Ken Pfeuffer, Matthias J Geiger, Sarah Prange, Lukas Mecke, Daniel Buschek, and Florian Alt. 2019. Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[88] Atyanti Dyah Prabaswari, Chancard Basumerda, and Bagus Wahyu Utomo. 2019. The mental workload analysis of staff in study program of private educational organization. In *IOP Conference Series: Materials Science and Engineering*, Vol. 528. IOP Publishing, 012018.
[89] Sarah Prange, Ahmed Shams, Robin Piening, Yomna Abdelrahman, and Florian Alt. 2021. Priview–exploring visualisations to support users' privacy awareness. In *Proceedings of the 2021 chi conference on human factors in computing systems*. 1–18.
[90] Meta Quest. 2023. Meta Quest Store - App Submission Guide. https://developer.oculus.com/resources/app-submission-success/
[91] R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
[92] Emilee Rader. 2022. Normative and {Non-Social} Beliefs about Sensor Data: Implications for Collective Privacy Management. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. 653–670.
[93] Shwetha Rajaram, Chen Chen, Franziska Roesner, and Michael Nebeling. 2023. Eliciting Security & Privacy-Informed Sharing Techniques for Multi-User Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.
[94] Franziska Roesner, Tadayoshi Kohno, and David Molnar. 2014. Security and privacy for augmented reality systems. *Commun. ACM* 57, 4 (2014), 88–96.
[95] Franziska Roesner, David Molnar, Alexander Moshchuk, Tadayoshi Kohno, and Helen J Wang. 2014. World-driven access control for continuous sensing. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1169–1181.
[96] Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. 2019. Secure {Multi-User} content sharing for augmented reality applications. In *28th USENIX Security Symposium (USENIX Security 19)*. 141–158.
[97] Pejman Saeghe, Gavin Abercrombie, Bruce Weir, Sarah Clinch, Stephen Pettifer, and Robert Stevens. 2020. Augmented reality and television: Dimensions and themes. In *ACM International Conference on Interactive Media Experiences*. 13–23.
[98] Jerome H Saltzer and Michael D Schroeder. 1975. The protection of information in computer systems. *Proc. IEEE* 63, 9 (1975), 1278–1308.
[99] Emre Sarigol, David Garcia, and Frank Schweitzer. 2014. Online privacy as a collective phenomenon. In *Proceedings of the second ACM conference on Online social networks*. 95–106.
[100] Stefan Schneegass, Romina Poguntke, and Tonja Machulla. 2019. Understanding the impact of information representation on willingness to share information. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.
[101] Gian Luca Scoccia, Anthony Peruma, Virginia Pujols, Ivano Malavolta, and Daniel E Krutz. 2019. Permission issues in open-source Android apps: An exploratory study. In *2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, 238–249.
[102] Bingyu Shen, Lili Wei, Chengcheng Xiang, Yudong Wu, Mingyao Shen, Yuanyuan Zhou, and Xinxin Jin. 2021. Can Systems Explain Permissions Better? Understanding Users' Misperceptions under Smartphone Runtime Permission Model. In *30th USENIX Security Symposium (USENIX Security 21)*. 751–768.
[103] Manimaran Sivasamy, VN Sastry, and NP Gopalan. 2020. VRCAuth: continuous authentication of users in virtual reality environment using head-movement. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 518–523.
[104] Robert H Sloan and Richard Warner. 2018. Why Are Norms Ignored? Collective Action and the Privacy Commons. *Collective Action and the Privacy Commons (February 18, 2018)* (2018).
[105] HJ Smith, T Dinev, and H Xu. 2011. Information privacy research: an interdisciplinary review. MIS Q. 35 (4): 989–1015.
[106] Daniel Smullen, Yuanyuan Feng, Shikun Zhang, and Norman M Sadeh. 2020. The Best of Both Worlds: Mitigating Trade-offs Between Accuracy and User Burden in Capturing Mobile App Privacy Preferences. *Proc. Priv. Enhancing Technol.* 2020, 1 (2020), 195–215.
[107] Snapchat. 2022. Snapchat. https://www.snapchat.com/l/en-gb/create
[108] Maximilian Speicher, Brian D Hall, and Michael Nebeling. 2019. What is mixed reality?. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.
[109] R Studio. 2023. Download RStudio IDE. https://posit.co/downloads/
[110] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. 2023. Stuck in the Permissions With You: Developer & End-User Perspectives on App Permissions & Their Privacy Ramifications. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*.
[111] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negrón-Arroyo, Christopher Thompson, Serge Egelman, and David Wagner. 2014. The effect of developer-specified explanations for permission requests on smartphone user behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 91–100.
[112] Radu-Daniel Vatavu, Pejman Saeghe, Teresa Chambel, Vinoba Vinayagamoorthy, and Marian F Ursu. 2020. Conceptualizing augmented reality television for the living room. In *ACM International Conference on interactive media experiences*. 1–12.
[113] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. 2015. Android permissions remystified: A field study on contextual integrity. In *24th USENIX Security Symposium (USENIX Security 15)*. 499–514.
[114] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. 2017. The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1077–1093.
[115] Evan James Williams. 1949. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Chemistry* 2, 2 (1949), 149–168.
[116] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 143–146.
[117] XRSI. 2020. The XRSI Privacy and Safety Framework. https://xrsi.org/publication/the-xrsi-privacy-framework
[118] Emanuel Zgraggen, Zheguang Zhao, Robert Zeleznik, and Tim Kraska. 2018. Investigating the effect of the multiple comparisons problem in visual analysis. In *Proceedings of the 2018 chi conference on human factors in computing systems*. 1–12.

A APPENDICES

A.1 Post-Study Semi-Structured Interview Script

(1) Can you explain to me why you ranked the permission control methods the way you did?
(2) Can you reflect on how well you understood the permission control methods?
(3) What was your thought process when interacting with the five permission controls (order of permission prompts were randomised)?
(4) Contrasting the *Slider w/o Image control* and *Slider w/ Image control* with what you have used before, were there any changes to your mindset when you were presented with fine grain control of your data?
(5) Do you feel the permission controls were enough to protect your privacy when using the applications?
(6) Is there anything we have missed regarding the permission control methods or something you feel I haven't asked you about yet?
(7) Considering the implementation of the *Slider w/o Image control* and *Slider w/ Image control*, would you prefer if there was one slider for each sensor the application wants data from or would you prefer if there was one slider in total for the whole app that controlled every sensor the application wanted data from?
(8) Is there anything you feel the *Slider w/o Image control* and *Slider w/ Image control* stopped you from doing or was missing?
(9) Are there any final thought and opinions you want to share about your experience?

A.2 Interaction Effects Between Context and Permission Control Methods

A.2.1 Understanding of What Data the Application Will be Able to Use. Table 1 contains the significant *post-hoc* ART-C contrasts for the interactions effects between Context and Permission Control Methods, the P values presented are corrected using Tukey corrections. Figure 8 shows the interaction plot of the two factors.

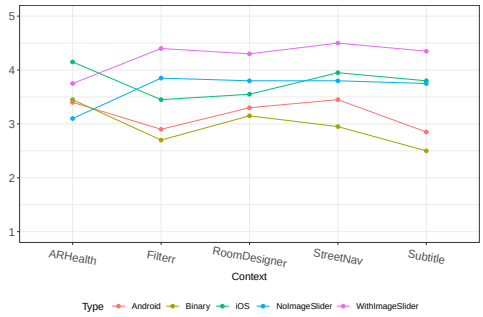

Figure 8: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on an interaction plot showing the changes in mean Likert Scores of understanding what data they application will be able to use for each permission control method across all of the contexts.

A.2.2 Understanding the Applications Functionality. Table 2 contains the significant *post-hoc* ART-C contrasts for the interactions effects between Context and Permission Control Methods, the P values presented are corrected using Tukey corrections. Figure 9 shows the interaction plot of the two factors.

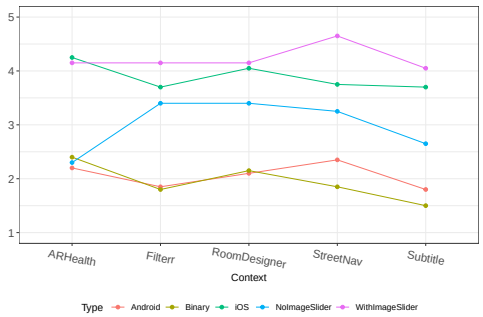

Figure 9: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on an interaction plot showing the changes in mean Likert Scores of understanding what functionality the application will be able to provide for each permission control method across all of the contexts.

Table 1: Degrees-of-freedom method: kenward-roger. P value adjustment: tukey method for comparing a family of 25 estimates

Significant Values of the Interaction Effects Between Context and Permission Control Methods				
Contrasts	SE	df	t.ratio	p.value
ARHealth,Android - StreetNav,WithImageSlider	33.01	456.00	-4.00	0.02
ARHealth,Binary - StreetNav,WithImageSlider	33.01	456.00	-3.95	0.02
ARHealth,iOS - ARHealth,NoImageSlider	33.01	456.00	3.86	0.03
ARHealth,iOS - Filterr,Android	33.01	456.00	4.67	0.00
ARHealth,iOS - StreetNav,Binary	33.01	456.00	4.13	0.01
ARHealth,iOS - Subtitle,Android	33.01	456.00	4.57	0.00
ARHealth,NoImageSlider - RoomDesigner,WithImageSlider	33.01	456.00	-4.34	0.00
ARHealth,WithImageSlider - Filterr,Binary	33.01	456.00	3.84	0.03
ARHealth,WithImageSlider - Subtitle,Binary	33.01	456.00	4.30	0.01
Filterr,Android - StreetNav,iOS	33.01	456.00	-3.85	0.03
Filterr,Binary - StreetNav,iOS	33.01	456.00	-4.06	0.01
Filterr,iOS - Filterr,WithImageSlider	33.01	456.00	-3.84	0.03
Filterr,iOS - StreetNav,WithImageSlider	33.01	456.00	-4.27	0.01
Filterr,iOS - Subtitle,WithImageSlider	33.01	456.00	-3.70	0.05
Filterr,NoImageSlider - Subtitle,Binary	33.01	456.00	4.12	0.01
Filterr,WithImageSlider - RoomDesigner,Android	33.01	456.00	4.12	0.01
Filterr,WithImageSlider - RoomDesigner,Binary	33.01	456.00	4.32	0.00
Filterr,WithImageSlider - StreetNav,Android	33.01	456.00	3.81	0.03
RoomDesigner,Android - StreetNav,WithImageSlider	33.01	456.00	-4.55	0.00
RoomDesigner,Android - Subtitle,WithImageSlider	33.01	456.00	-3.98	0.02
RoomDesigner,Binary - RoomDesigner,WithImageSlider	33.01	456.00	-3.81	0.03
RoomDesigner,Binary - Subtitle,WithImageSlider	33.01	456.00	-4.18	0.01
RoomDesigner,iOS - StreetNav,WithImageSlider	33.01	456.00	-3.93	0.02
RoomDesigner,WithImageSlider - StreetNav,Binary	33.01	456.00	4.61	0.00
StreetNav,Android - StreetNav,WithImageSlider	33.01	456.00	-4.25	0.01
StreetNav,iOS - Subtitle,Android	33.01	456.00	3.75	0.04
StreetNav,iOS - Subtitle,Binary	33.01	456.00	4.53	0.00
StreetNav,NoImageSlider - Subtitle,Binary	33.01	456.00	3.71	0.05
Subtitle,Binary - Subtitle,iOS	33.01	456.00	-3.71	0.05
Subtitle,Binary - Subtitle,NoImageSlider	33.01	456.00	-4.10	0.01

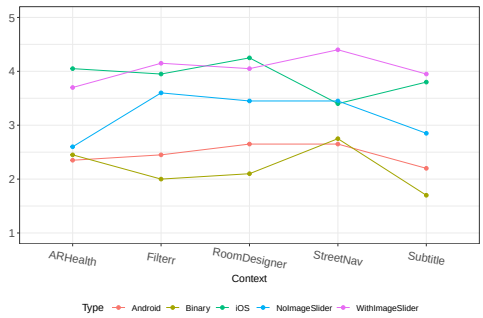

Figure 10: 5-point Likert scales 1=Strongly disagree to 5=Strongly agree plotted on an interaction plot showing the changes in mean Likert Scores of understanding why the application requested the data it did for each permission control method across all of the contexts.

A.2.3 Understanding Why the Application Requested the Data. Table 3 contains the significant *post-hoc* ART-C contrasts for the interactions effects between Context and Permission Control Methods, the P values presented are corrected using Tukey corrections. Figure 10 shows the interaction plot of the two factors.

A.3 Detailed Permission Prompts

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Table 2: Degrees-of-freedom method: kenward-roger. P value adjustment: tukey method for comparing a family of 25 estimates

Significant Values of the Interaction Effects Between Context and Permission Control Methods				
Contrasts	SE	df	t.ratio	p.value
ARHealth,Android - Filterr,NoImageSlider	27.64	456.00	-4.21	0.01
ARHealth,Android - RoomDesigner,NoImageSlider	27.64	456.00	-4.12	0.01
ARHealth,Binary - Filterr,iOS	27.64	456.00	-4.56	0.00
ARHealth,Binary - Subtitle,iOS	27.64	456.00	-4.61	0.00
ARHealth,iOS - StreetNav,NoImageSlider	27.64	456.00	3.78	0.04
ARHealth,NoImageSlider - Filterr,NoImageSlider	27.64	456.00	-3.91	0.02
ARHealth,NoImageSlider - RoomDesigner,NoImageSlider	27.64	456.00	-3.83	0.03
Filterr,iOS - StreetNav,WithImageSlider	27.64	456.00	-3.77	0.04
Filterr,iOS - Subtitle,NoImageSlider	27.64	456.00	3.82	0.03
Filterr,NoImageSlider - RoomDesigner,Android	27.64	456.00	4.58	0.00
Filterr,NoImageSlider - RoomDesigner,Binary	27.64	456.00	4.49	0.00
Filterr,NoImageSlider - StreetNav,Android	27.64	456.00	3.73	0.04
RoomDesigner,Android - RoomDesigner,NoImageSlider	27.64	456.00	-4.49	0.00
RoomDesigner,Android - StreetNav,NoImageSlider	27.64	456.00	-4.05	0.01
RoomDesigner,Binary - RoomDesigner,NoImageSlider	27.64	456.00	-4.40	0.00
RoomDesigner,Binary - StreetNav,NoImageSlider	27.64	456.00	-3.95	0.02
StreetNav,iOS - Subtitle,NoImageSlider	27.64	456.00	4.11	0.01
StreetNav,WithImageSlider - Subtitle,iOS	27.64	456.00	3.72	0.04
Subtitle,Binary - Subtitle,NoImageSlider	27.64	456.00	-4.06	0.01
Subtitle,iOS - Subtitle,NoImageSlider	27.64	456.00	3.87	0.03

Table 3: Degrees-of-freedom method: kenward-roger. P value adjustment: tukey method for comparing a family of 25 estimates

Significant Values of the Interaction Effects Between Context and Permission Control Methods				
Contrasts	SE	df	t.ratio	p.value
ARHealth,Android - ARHealth,WithImageSlider	31.61	456.00	-4.36	0.00
ARHealth,Android - Filterr,NoImageSlider	31.61	456.00	-3.78	0.04
ARHealth,Android - Subtitle,iOS	31.61	456.00	-4.59	0.00
ARHealth,Binary - ARHealth,WithImageSlider	31.61	456.00	-4.05	0.01
ARHealth,Binary - Subtitle,iOS	31.61	456.00	-4.28	0.01
ARHealth,iOS - RoomDesigner,Android	31.61	456.00	4.67	0.00
ARHealth,iOS - StreetNav,Binary	31.61	456.00	4.30	0.01
ARHealth,iOS - Subtitle,NoImageSlider	31.61	456.00	4.00	0.02
ARHealth,NoImageSlider - Filterr,iOS	31.61	456.00	-4.38	0.00
ARHealth,NoImageSlider - RoomDesigner,WithImageSlider	31.61	456.00	-4.65	0.00
ARHealth,NoImageSlider - Subtitle,iOS	31.61	456.00	-3.84	0.03
ARHealth,NoImageSlider - Subtitle,WithImageSlider	31.61	456.00	-4.50	0.00
ARHealth,WithImageSlider - Filterr,Android	31.61	456.00	4.06	0.01
Filterr,Android - Subtitle,iOS	31.61	456.00	-4.29	0.01
Filterr,Binary - RoomDesigner,NoImageSlider	31.61	456.00	-4.38	0.00
Filterr,Binary - StreetNav,iOS	31.61	456.00	-4.35	0.00
Filterr,Binary - StreetNav,NoImageSlider	31.61	456.00	-4.34	0.00
Filterr,iOS - RoomDesigner,Android	31.61	456.00	4.21	0.01
Filterr,iOS - StreetNav,Android	31.61	456.00	4.27	0.01
Filterr,iOS - StreetNav,Binary	31.61	456.00	3.84	0.03
Filterr,NoImageSlider - RoomDesigner,Binary	31.61	456.00	4.55	0.00
Filterr,NoImageSlider - Subtitle,Android	31.61	456.00	4.34	0.00
Filterr,WithImageSlider - StreetNav,Binary	31.61	456.00	4.55	0.00
Filterr,WithImageSlider - Subtitle,NoImageSlider	31.61	456.00	4.26	0.01
RoomDesigner,Android - RoomDesigner,WithImageSlider	31.61	456.00	-4.48	0.00
RoomDesigner,Android - Subtitle,WithImageSlider	31.61	456.00	-4.32	0.00
RoomDesigner,Binary - RoomDesigner,NoImageSlider	31.61	456.00	-4.07	0.01
RoomDesigner,Binary - StreetNav,iOS	31.61	456.00	-4.04	0.01
RoomDesigner,Binary - StreetNav,NoImageSlider	31.61	456.00	-4.03	0.01
RoomDesigner,iOS - Subtitle,NoImageSlider	31.61	456.00	4.52	0.00
RoomDesigner,NoImageSlider - Subtitle,Android	31.61	456.00	3.86	0.03
RoomDesigner,WithImageSlider - StreetNav,Android	31.61	456.00	4.54	0.00
RoomDesigner,WithImageSlider - StreetNav,Binary	31.61	456.00	4.11	0.01
RoomDesigner,WithImageSlider - Subtitle,NoImageSlider	31.61	456.00	3.81	0.03
StreetNav,Android - Subtitle,iOS	31.61	456.00	-3.73	0.04
StreetNav,Android - Subtitle,WithImageSlider	31.61	456.00	-4.39	0.00
StreetNav,Binary - Subtitle,WithImageSlider	31.61	456.00	-3.95	0.02
StreetNav,iOS - Subtitle,Android	31.61	456.00	3.82	0.03
StreetNav,NoImageSlider - Subtitle,Android	31.61	456.00	3.82	0.03

Table 4: *Binary control* prompts per context

Context Name	Binary Text
Subtitle	Subtitle has requested access to the microphone on your device
Filterr	Filterr has requested access to the camera on your device
Room Desinger	Room Designer has requested access to the camera on your device
StreetNav	1- StreetNav has requested access to the camera on your device 2- StreetNav has requested access to your device's GPS location
AR Health	1- AR Health has requested access to the Pedometer on your device 2- AR Health has requested access to your device's GPS location 3- AR Health has requested access to your device's Heart Rate sensor 4- AR Health has requested access to eye tracking

Table 5: *iOS control* prompts per context

Context Name	Permission Title	Permission Description
Subtitle	"Subtitle" Would Like to Access the Microphone	To let you know who else is in the room with you and what they are saying
Filterr	"Filterr" Would Like to Access the Camera	To let you apply filters onto other people around you and your surroundings
Room Designer	"Room Designer" Would Like to Access the Camera	To let you place objects in your space and recommend matching items
StreetNav	1- "StreetNav" Would Like to Access the Camera	1- To let you see directions augmented onto your peripheral vision
	2- Allow "StreetNav" to use your location?	2- To see turn-by-turn directions, search nearby locations and get traffic updates, allow StreetNav to find your location
AR Health	1- "AR Health" Would Like to Access Pedometer	1- To allow you to count and review how many steps you have taken
	2- Allow "AR Health" to use your location	2- To let you see where you have been and how many calories you burned
	3- "AR Health" Would Like to Access HeartRate	3- To let you see what heart rate information such as resting and current heart rate
	4- "AR Health" Would Like to Access Eye Tracking	4- To let you see how your mental health, such as your mood and illnesses

Table 6: *Android 11 control* prompts per context

Context Name	Permission Text
Subtitle	Allow Subtitle to record audio?
Filterr	Allow Filterr to take pictures and record video?
Room Desinger	Allow Room Desinger to take pictures and record video?
StreetNav	1- Allow StreetNav to take pictures and record video? 2- Allow StreetNav to access this device's location?
AR Health	1- Allow AR Health to access your physical activity? 2- Allow AR Health to access this device's location? 3- AR Health has requested access to your devices Heart Rate sensor 4- AR Health has requested access to eye tracking

Table 7: *Slider w/o Image control* and *Slider w/ Image control* prompts per context

Context Name	Slider Text
Subtitle	1- Deny Subtitle access to the microphone on your device 2- Minimum Requirement: Subtitle only knows if there was sound picked up by the microphone 3- Subtitle is only aware of the occurrence and direction of a sound 4- Optimal Performance: Subtitle has access to a distorted version of the full microphone audio 5- Subtitle has access to the full microphone audio
Filterr	1- Deny Filterr access to the camera on your device 2- Minimum Requirement: Filterr can only see where a person is, and how far away they are 3- Filterr can see the silhouette of the people around you 4- Optimal Performance: Filterr can see a full 3D model of the person 5- Filterr has full access to the front camera
Room Designer	1- Minimum Requirement: Deny Room Designer access to the camera on your device 2- Room Designer only has access to the dimensions of your space 3- Room Designer has access to 3D structural data of your space 4- Optimal Performance: Room Designer has access to a full colour and texture 3D scan of your space
StreetNav	1- Minimum Requirement: Deny StreetNav access to both camera and locational data from your headset 2- StreetNav can only see the route of your journey 3- StreetNav has access to your headset's approximate GPS location 4- Optimal Performance: StreetNav has precise access to your headset's GPS location 5- StreeNav can determine your exact location using full GPS and camera data
AR Health	1- Deny AR Health access to the health sensors on your device 2- Minimum Requirement: AR Health can only see an average of your health sensors data 3- AR Health receives health data from your device every minute 4- AR Health receives health data from your device every 10 seconds 5- AR Health receives full real time health data from your device