VRIJE
UNIVERSITEIT
BRUSSEL

Graduation thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Applied Sciences and Engineering: Computer Science

TOWARDS INTELLIGIBILITY IN MULTI-USER IOT ENVIRONMENTS

Artyom Kuznetsov

2021-2022

Promotor: Prof. Dr. Beat Signer
Supervisor: Ekene Attoh
Sciences and Bioengineering Sciences

Abstract

The Internet Of Things paradigm allows the interconnection of an multiple devices by the use of internet technologies. Frequently, task automation systems such as HomeAssistant[1], OpenHAB[2], Google Home[3] are used to enable interaction between smart-devices. One of the popular ways to configure smart devices is to use *If This Then That (IFTTT)* rules. An environment with a limited number of devices that requires only a few rules is an example where the construction of *IFTTT* rules is a relatively easy task. Users without technical backgrounds are able to easily create rules by following instructions. However, the construction of *IFTTT* rules is a challenging task when it involves many devices and rules. Eventually, rules become more complex, therefore it becomes harder for users to understand whether rules operate correctly or even whether some rules have conflicts. Intelligibility mechanisms are required to properly handle this sort of complexity by giving users context and a proper explanation of what is happening. The complexity increases even more when multiple users are involved. Many existing IoT task automation systems (e.g HomeAssistant) consider only one user in mind. Different users may construct automation rules that conflict with the rules of other users. Without proper intelligibility mechanisms, it would be hard for users to understand why the system does not work as expected.

To address the intelligibility problems present in multi-user environments, we need a system with intelligibility mechanisms that consider multiple users. Some literature [1] propose using of notification mechanisms to improve users' understanding of the system's operations. More specifically, the use of active notifications (e.g mobile app notifications) to let users know that something has changed in the system or someone used their smart device. The authors believe that notification mechanisms could improve system transparency. Moreover, the authors conducted a user survey which has shown that not all intelligibility mechanisms are useful in certain environments. Many research papers [2, 3, 1] consider the home environment which already has some limitations in that there are usually already implicit rules followed users which mask apparent intelligibility problems. In our research, we chose to consider an office environment instead. An office environment implies many smart devices and multiple users who use the system. The office environment also implies that there will be less of the possibility of the familiarity which is present in household environments, which may breed implicit trust relationships among the users.

[1] https://www.home-assistant.io/

[2] https://www.openhab.org/

[3] https://home.google.com/welcome/

Finally, we consider intelligibility during the conflict resolution stage. Some research [4, 5] propose conflict resolution algorithms that can help resolve conflicting rules among users. However, they usually do not consider intelligibility mechanisms that explain to users the reason behind their conflict resolution actions. We conducted a user survey to help us understand what users' needs in multi-user environments are. The results of the survey show us that users want to have system-based and admin-based conflict resolution, as well as intelligibility on the conflict resolution. The participants have also highlighted the importance of intelligibility in multi-user environments. Based on prior work and our survey, we were able to develop design guidelines that propose ways to handle intelligibility in multi-user environments. Likewise, we implemented a proof-of-concept application that implements the intelligibility features that we proposed in the design guidelines.

Acknowledgement

I would like to express my gratitude to my supervisor, *Ekene Attoh*, who guided me throughout this project. He gave me outstanding support during the thesis writing and taught me how to do proper research.

I would also like to thank my promoter *Prof. Dr. Beat Signer* for the opportunity to do this thesis, his insightful comments, proofreading and guidance.

Finally, I would thank my family, my mother *Yelena* and father *Leonid* for always supporting and inspiring me.

Contents

1 Introduction **4**
- 1.1 Problem Statement . . . 7
- 1.2 Contributions . . . 9
  - 1.2.1 User Survey . . . 9
  - 1.2.2 Design Guidelines . . . 9
  - 1.2.3 Proof of Concept Application . . . 9
- 1.3 Methodology . . . 10

2 Background **11**
- 2.1 The Internet of Things: A survey . . . 11
- 2.2 Intelligibility mechanisms in smart homes . . . 11
- 2.3 Human consideration in context-aware systems . . . 12
- 2.4 Design Space . . . 13

3 Related Work **14**
- 3.1 Debugging IF-THEN rules . . . 14
- 3.2 PervasiveCrystal . . . 15
- 3.3 Feedforward Torch . . . 15
- 3.4 Studying breakdowns in interactions . . . 17
- 3.5 Multi-User security and privacy in smart homes . . . 17
- 3.6 Algorithms for Conflicts Resolution . . . 19
- 3.7 VA and Conflict Resolution in Multi-User Environment . . . 20
- 3.8 Kratos: Access Control System . . . 21
- 3.9 Evolving needs in IoT control and accountability . . . 22
- 3.10 Intelligibility and control for context-aware IoT . . . 23

4 User Study - Office Environment **25**
- 4.1 Environment Definition . . . 26
- 4.2 Survey Scenario . . . 26
- 4.3 Survey Content . . . 28
- 4.4 Survey Results . . . 30

4.4.1 Users' Remarks 43
4.4.2 Survey Conclusion 47

5 Design Guidelines **48**
5.1 Data Ownership 48
5.2 Users Rules and Conflicting Rules Visualisation 49
5.3 Limited Sharing Of Rule Information 49
5.4 System-based and Admin-based Conflict Resolution 49

6 Solution **50**
6.1 Multi-User IoT Dashboard 50
6.1.1 MU-IoT Dashboard Architecture 51
6.1.2 Design Space Support 57

7 Conclusion and Future Work **59**

A Survey Questions **61**

B MU-IoT Dashboard Installation Guide **79**

Chapter 1

Introduction

The Internet of Things (IoT) interconnects devices, people and data allowing them to communicate seamlessly [6]. It is estimated that around 500 billion devices will be connected to the internet by the 2030 year [7]. IoT is applied in different domains such as home automation, industrial automation, healthcare and more. The internet of things requires the establishment of best practices for further acceleration and adoption [8].

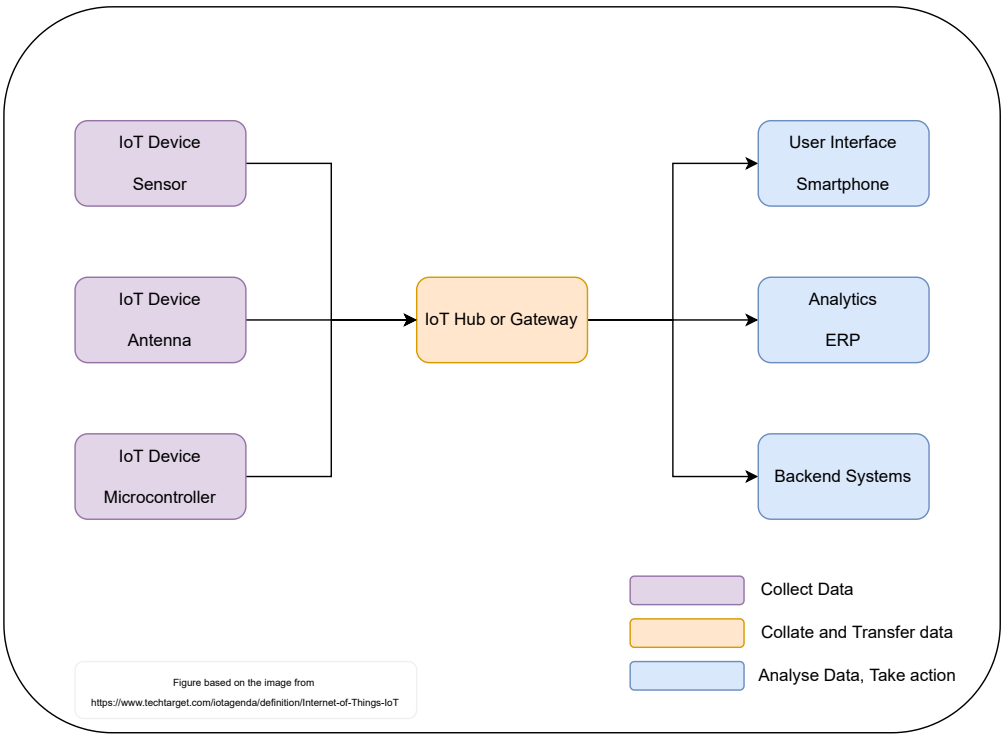


Figure 1.1: IoT Representation

Figure 1.1 demonstrates an example of a typical IoT system. Here we see three main stages in the functioning of IoT systems. IoT systems collect data through various devices and sensors, for example, smart cameras that collect data about people who enter or leave certain rooms. This data is usually not processed by the IoT devices themselves but is transferred to a location with more computational power. The IoT gateway[1][2] is a physical device with software that connects IoT devices and sensors to cloud-based computing and data processing solutions where the data from those devices can be processed further. After the data is processed, it is analysed in the third stage. Usually, based on the analysis, the data can then become actionable and be used further.

According to [9], there is an overlap in research in IoT systems and context-aware systems. The term context-aware system was coined by Schilit et al. [10]. Context-aware systems are the systems that sense surrounding context by use of sensors and can react to changes happening in their environment [9]. IoT systems use context based on raw data from sensors to make decisions.
Chegini et al. [11] provide several examples which demonstrate how IoT systems use context and how they react to changes. An example is a system that can be used in hospital to help monitor a patient's health. The system monitors patients by consuming sensor data and the data can be used to identify if something is wrong with the patient. In a scenario when a certain monitored value drops below a specified threshold (e.g. high resting heart rate below 100 BPM) the system can inject medicines or inform medical staff regarding the changed situation. Perrera et al. [12] also identified the features of context-awareness which relate to IoT: *presentation*, *execution* and *tagging*. As a presentation feature, the IoT system within context-awareness should perceive the user (analyse user's actions, track user's actions) and present actual information that the user would need to see. As an example, when a user comes to the grocery store and starts using his phone, they would probably want to see a shopping list. The second feature is execution, where the IoT system should be able to automatically execute certain actions based on the context. For example, all lights should be turned off in the office when all employees left the building. Finally, as tagging feature, the IoT system should be able to tag devices and sensors that can be used to interpret the changing context. That is important because data from one sensor may not be sufficient to correctly interpret the situation. Hence the data from several sensors and devices that collect similar information should be used instead.

[1]https://www.checkpoint.com/cyber-hub/network-security/what-is-iot/what-is-an-iot-gateway/

[2]https://www.techtarget.com/iotagenda/definition/IoT-gateway

An IoT system should also work autonomously and react on changes happening around by use of sensors data. Finally, an IoT system should interpret and analyse data produced by different sensors and provide meaningful results [9].

Bellotti et al. investigated human considerations in context-aware systems [13]. They coined the term *intelligibility* which entails informing users of a system's capabilities and understandings, providing feedback so that users may become aware of what the system knows, how it knows it, what is it doing about it and ensuring that users are still in control over a system's actions.

Different research has been undertaken in the area of intelligibility. In their study, Jakobi et al. [14] seek to determine users' needs in IoT. The authors used a living lab approach in order to understand users and their context and determine how users use IoT devices in a home environment. The study was conducted with 12 households over period of 26 months. It revealed that users have less trust in a system if they are unable to fully grasp what system is doing and why. The participants stated the need for features that enable them to understand how their systems are functioning. As an example, one of the participants stated that they would like to have a function that allows them to remotely verify the current state of the system i.e to verify that their rules work as they were configured to work. Another participant stated that some light bulbs were turning on and off during the mornings and that they were not able to easily identify the reason for such behaviour. The participant stated that they would prefer to have a mechanism that could explain the reason for such behaviour. As the participants got more used to their system, they however stated the need to have more control over the amount of feedback information they received from their systems. Participants stated that they would like a feature which allows them to get feedback from their system only when something was wrong. Similarly, Zheng et al [1] sought to investigate intelligibility in a home environment where multiple users use the IoT system. They noted that in prior work[2, 3, 15], smart devices in smart home environments have conflicts. The nature of those conflicts is due to the use of the system by multiple users. As an example, automation rules established by one user can contradict (be in conflict with) the automation rules of other users. Another problem is that more experienced users may oppress less experienced users by using mechanisms (e.g. remote control) or devices which are less known by less experienced users. The authors proposed design guidelines that were focused on providing mechanisms that could ensure the privacy and security of each user. To validate their research, they conducted a user study survey in multiple households. The survey has shown that almost all households were

not using the proposed features. The authors concluded that members of participating families have a high level of trust in each other. That implies less need for setting up privacy policies.

In [16], Aljawarneh et al. defined a *smart environment* as a place which comprises of interconnected devices, sensors, appliances and services that adapt themselves according to context to improve comfort, safety and security of its users. They stated that the introduction of multiple users in smart environments is a challenging task because multiple users have to share the same place, devices and computational resources in the same environment. Thus the system is expected to fulfil the needs of all users in an effective manner.

In this thesis, we aim to determine what intelligibility problems are present in multi-user environments and identify possible solutions to these problems.

A first step is to analyse the existing literature on intelligibility and identify research which focus on intelligibility in multi-user IoT environments. By carefully reviewing the literature, we hope to get a better understanding of existing problems and the proposed solutions in this space.

Having identified the existing problems from the literature, we aim to conduct a user survey introducing a realistic multi-user scenario. We aim to conduct an analysis of the survey results and extract users' intelligibility needs in multi-user environments. Having extracted these needs from our analysis, we will then propose design guidelines for IoT systems to support these needs.

As the final step, we are going to implement a proof of concept application based on the proposed design guidelines.

1.1 Problem Statement

Jakobi et al. [14], conducted user study where several households were provided with IoT systems. The authors were examining how these households interact with IoT systems at their houses. Results show that participants want a better understanding of how a context-aware system works. Participants want to have access to logs, would like to have features that can explain how the system operates and why it performs certain operations and actions. From the study, it has become clear that users do not explicitly trust context-aware IoT system. In they study, the participants were constantly using a dashboard to verify that system works according to how they configured it. The participants wanted to have more control mechanisms over the system. At the same time, participants noted that a large amount of information could negatively impact understanding.

Other studies [13, 17] also demonstrate the requirement for context-aware systems to have a mechanism that helps users to understand how a system operates and mediate erroneous implicit decisions.

Zheng et al. [1] indicated that most IoT solutions are not intended for interaction between multiple users. They conducted a study of the intelligibility of IoT systems in a multi-user environment in a home scenario where participants are family members or (close) acquaintances. Based on the analysis of the survey results, they proposed some guidelines that can improve user intelligibility such as *access controls* and *activity notifications*. The authors then implemented the proposed guidelines as features in a proof-of-concept application. They however found that most participants were not using the provided features related to multi-user intelligibility, such as role-based access control. This was due to the fact that users in such environments already rely on well established norms in the household and thus did not need the application's features in order to properly govern smart device usage. A similar study but with a different participant composition could possibly produce different results for example in environments where people are less familiar with one another. Finally, the proposed guidelines by Zheng et al. do not consider conflict resolution in multi-user environments and the Intelligibility related to conflict resolution.

Coppers et. al. determined that despite the simplicity of *trigger-action* rules, the implementation of complex rules is a rather hard task for users with non-programming background [4]. They stated that we can improve the intelligibility of a system that is based on trigger-action rules by providing visualisations that can demonstrate conflicts between different rules on a timeline. In addition, they provided a method to resolve conflicts temporarily by allowing the user to suppress certain conflicting rules for a certain period. Unfortunately, this approach does not take into account real multi-user scenarios.

Caivano et. al. stresses the importance of collaboration in multi-user environments [18]. They indicate that most existing IoT solutions are not capable of supporting collaboration among users. They also emphasise that existing IoT solutions use *IFTTT* rules to coordinate the operation of IoT devices. The authors propose to use gamification techniques to improve collaboration in IoT systems. Collaborative functionality can be used as a mechanism that can improve intelligibility among users. As the authors indicated, the collaborative features can be used in *IFTTT* construction, where multiple users play a role in constructing a single rule, e.g in situations when help is needed.

Sikder et al. indicated that access control mechanisms in existing solutions pose challenges in terms of sharing capabilities and control resolution [19]. Most existing home automation solutions have binary access control capabilities. It means that you either give full control to the user or none at all.

They propose to consider more granular access control mechanisms, such as location-based, restriction-based and demand-based access controls. The first type gives control to users if their location corresponds to the allowed one. The second type gives limited control to the system (e.g by allowing to use of certain devices). Finally, the third type gives control to the system if a user requests to use the system. Afterwards, the administrator or other users can decide whether to give permission or not.

Based on our literature review we see the importance of intelligibility mechanisms in IoT solutions. Participants in most of the studies have shown less trust in the systems which operate implicitly. At the same time, proper intelligibility mechanisms improve the user experience. We noticed that most of the research in the intelligibility domain do not cover multi-user scenarios. In our work, we focus on intelligibility in multi-user environments. Multi-user environments create additional intelligibility questions which are not handled in the reviewed literature.

1.2 Contributions

1.2.1 User Survey

The first contribution is the user study which we conducted to have a better understanding of the intelligibility of users in multi-user IoT environments. We used the results of the survey to construct design guidelines for intelligibility in multi-user IoT environments.

1.2.2 Design Guidelines

The second contribution are some design guidelines. These guidelines are built based on reviewed literature and the results of the user survey.

1.2.3 Proof of Concept Application

The third contribution is a proof-of-concept implementation of the design guidelines. We were able to implement a timeline dashboard application that covers all proposed design guidelines.

1.3 Methodology

We have chosen *Design Science Research Methodology (DSRM)* for our research. The DSRM research has six main activities, which are *the problem identification and motivation*, *the definition of objectives for a solution*, *the design and development*, *the demonstration*, *the evaluation*, and *the communication*. We decided to go with this methodology since we build our research on prior work. We used prior work to identify the problem and motivation. Additionally, we conducted a user survey based on prior work to identify users' needs in multi-user intelligibility. During the design and development step, we came up with design guidelines which are based on prior work and conducted a user study. that became the core of the architecture. These guidelines were used to come up with the architecture of our solution. We implemented a proof-of-concept application that addresses the proposed design guidelines. The research and all findings are a part of the master thesis.

Chapter 2

Background

2.1 The Internet of Things: A survey

Atzori et. al. [20] conducted a survey in the IoT domain. The authors described the Internet of Things paradigm and how historically the domain had several visions of the same problem. The authors indicate that three main visions exist: *internet-oriented*, *things-oriented* and *semantic-oriented* visions. Things paradigm was the first one and the simplest one. It considers devices interconnected with *RFID* tags. However such a paradigm is not enough, it was quite limited in nature. That is why The Internet of Things paradigm became more prevalent since it allowed us to consider devices that are located in different places and use the internet as connection mechanism.

Atzori et. al. also talked about the application of IoT systems. They have shown examples of IoT applications in domains such as transportation and logistics, and the health domain. The authors also talked about open issues that IoT have at the moment. They mentioned privacy, security, transportation protocol, digital forgetting, and data integrity as challenges in this domain that need to be properly tackled.

2.2 Intelligibility mechanisms in smart homes

Anind Dey and Barkhuus [21] discuss the question of whether context-aware computing takes control away from the users. They believe that *passive* and *active context-aware* applications have the problem of decreased trust in the systems. The authors defined an *active system* as a system that can change behaviour based on context without user intervention. The *passive system* on the contrary, requires a user to make the final decision. They define *personalization interaction* as a feature of a system that allows users to customise certain func-

tionality, hence improving user experience and interaction. As an example, users may refine the dashboard page of the automation system, where they select only functionality they are interested in. The authors conducted a case study with 23 participants where they ask participants to imagine that they have a choice between three interaction levels: *personalization*, *passive context-awareness* and *active context-awareness* in six different services (lunch service, class slides search and others). The results of the experiment demonstrate that participants feel the loss of control over the system in passive and active context-aware systems. Nevertheless, participants were in favour of using passive and active context-awareness systems over systems with personalization interaction levels. It sounds contradictory, however, participants may think that the advantages of using context-aware systems overweight the disadvantages.

2.3 Human consideration in context-aware systems

Bellotti et al. [13] analyse human considerations in context-aware systems. In contrast to computers, people make unpredictable judgements about context. People have intentions, emotions, and other factors that make it impossible to create an accurate prediction model. One example was given with an air-cooling solution based on sensors. Air conditioners control temperatures according to some rules and standards. Those standards are not appropriate for all people. Hence an air conditioner is not able to tell whether the temperature that was set is okay for a certain user or not. The authors state that systems should not take actions based solely on context-awareness but need to involve users in the decision making process. Users should be aware of what the system is doing and the outcomes of the actions. Hence users need to have understanding of how system perceives the world around itself.

The authors emphasise that users need to be able to understand how a system is interpreting the state of the world. They state that context-aware systems must be intelligible as to their states, "beliefs" and "initiatives" [13]. Context-aware systems should know their own responsibilities and acknowledge the context to provide the best result for end users.

The authors proposed key features which a context-aware system should have:

- **Intelligibility** "Context-aware systems that seek to act upon what they infer about the context must be able to represent to their users what they know, how they know it, and what they are doing about it" [13]
- **Accountability** "Context-aware systems must enforce user accountability when based upon their inferences about the social context, they seek to

mediate user actions that impact others" [13]

2.4 Design Space

Vermeulen et al. [17] proposed a design space that differentiates and identifies intelligibility mechanisms in different solutions. The authors propose that context-aware systems have six main differentiating factors:

- **Timing** - "intelligibility and control can be supported at different times (before action, after action)" [17]
- **Generality** - "interaction techniques for intelligibility and visualisation can be general or domain specific" [17]
- **Degree of co-location** - "intelligibility mechanisms can be embedded to the UI of the system or can be external" [17]
- **Initiative** - "intelligibility mechanisms can be shown by user request or being shown on system decision" [17]
- **Modality** - "different ways of explanation can be applied (text, speech, graphical visualisation)" [17]
- **Level of Control** - "level of control on decisions made by a system. Starting from no additional control is possible to full control over the system operation" [17]

The authors conducted several studies where it is shown that systems with different combinations of intelligibility factors have large impact on intelligibility experience. It becomes clear that optimal combination of factors depends on the domain of the system and they have to be chosen carefully during system development.

Chapter 3

Related Work

3.1 Debugging IF-THEN rules

Corno et. al. [22] discuss a way of debugging **if-then** rules that are widely used in IoT applications through the **Jigsaw metaphor**. The *Jigsaw metaphor* is a way to develop applications which consider components as puzzle pieces and hides the internal implementation from the user. It allows for building interactive applications that do not require low-level programming skills. Therefore it allows users to focus on the things they want to achieve, not internal implementation details [23].

Implementation of rules is not an easy task, especially for people with non-programming experience and when rules become more complex. That is the reason why we need to have a way to properly debug if-then rules. The authors had two research questions: *What information is required for a user to correct if-then rules?* and *Which visual language is better suitable for debugging trigger-action rules?*. To answer these questions the authors proposed the following guidelines [22]:

- ”Real-time feedback needs to be provided during the composition of trigger-action rules” [22]
- ”Allow update of problematic rules on-the-fly” [22]
- ”Textual and graphical explanations should be used to represent the run-time behaviour of programs” [22]
- ”’Why Did?’ and ’Why did not?’ questions at failure time can help users solve problems during debugging” [22]
- ”Block programming can be adapted to compose if-then rules for IoT” [22]

- "Data-flow visualisation could be used for the representation of multiple trigger-action rules to help users understand unwanted run-time behaviours" [22]

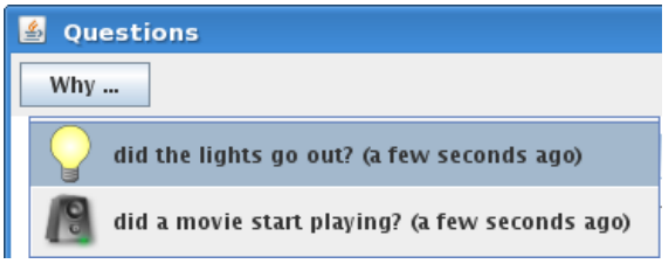
To validate the guidelines **My IoT Puzzle** tool was implemented. The tool is used for composing and debugging IF-THEN rules. The tool can identify loops, redundancies and inconsistencies. The evaluation demonstrates that participants who had limited or no experience in programming were able to identify and fix problems during rules composition. This is an intelligible way of debugging rules.

3.2 PervasiveCrystal

Vermeulen et. al. propose a system, called *PervasiveCrystal*, that asks and answer why and why not questions in a pervasive computing environment [24]. The authors state that *Pervasive computing* is computational capability that embedded into objects that allows the objects effectively communicate and perform required tasks in a way that minimises the user's involvement in the process. Because pervasive systems often work without user involvement, users may be surprised by certain actions that happen since they are not able to understand the complex reasoning of the system. It is important to note that the difficulty of understanding is not the only problem. These complex systems may conclude and act wrongly and we need to have a mechanism that allows users to identify and correct the system's mistakes. In the *PervasiveCrystal*, all events, actions and conditions are expected to have a descriptive label. It allows to explain why certain event did or did not certain action. Figure 3.1 demonstrates the *why and why not* user interface popup menu [24]. In the user study, all participants agreed that the system indeed helps to improve understanding of the system and is useful. Some participants encountered a problem when many events happen at the same time they get spammed with a why-menu. Besides, it is not clear whether the system is scalable and will be suitable for complex systems. Moreover, the system does not consider multi-user usage scenarios.

3.3 Feedforward Torch

Vermeulen et al. [25] write about a system that uses a feed-forward torch approach for intelligibility. A feed-forward torch is an opposite approach to feedback, meaning that the system gives information regarding the outcome of certain actions.



(a) The why menu.

(b) An answer to a why question.

Figure 3.1: A context menu of Pervasive Crystal

The solution has an approach that helps to identify the position of objects in 3D space. The solution allows users to point to objects in space and display feed-forward information as a projection screen. Feed-forward information was displayed via visualisations or text explanations. The authors conducted a user study to validate whether approach helps to improve understanding of complex systems. They prepared three scenarios where users need to do several steps to achieve requirements. One of the scenarios was auditorium, where users need to prepare auditorium for presentation. To prepare for the presentation they would need to turn on projector, lower the projection screen and turn off lights. Results of the survey shows that all participants were satisfied with proposed approach and it helped them to improve understanding in complex scenarios. Moreover, participants stated that they prefer information to be display via visualisations, not a plain text explanation, since it is an easier way to understand the explanation.

The authors have not mentioned multi-user handling in the paper.

3.4 Studying breakdowns in interactions

Avdic et. al. [26] study breakdowns in interaction with smart speakers. The authors studied how users address their problem to smart speakers and how they handle mistakes or system breakdowns. They conducted study to understand the users' mental model of smart speakers, how users use smart speakers and which recovery mechanisms they use when their request fails. Results show that participants had different view on how smart speakers operate. They preferred to use voice to control smart speakers. In situations when mistakes happen participants tend to physically come to the speaker in order to solve the problem or to use a smartphone.

3.5 Multi-User security and privacy in smart homes

Zheng et. al. write about understanding and improving security and privacy in multi-user smart home environment [1]. The authors indicate that modern smart home solutions are not intended for interaction between multiple users. They come up with two research questions:

- ″How should a smart home be designed to address multi-user security and privacy challenges?″
- ″What security and privacy behaviours and needs do these smart home users exhibit in practice?″

The authors proposed design principles that should address the above questions. The authors also created a mobile application which implements the design principles and designed a user study in order to validate the proposed design principles. The initial design principles consists of:

- **Access Control Flexibility** Support of a variety of roles and support of contextual factors, such as location. These factors can be combined.
- **User Agency** The system has to be built in such a way that non-technical users have a good understanding of how to use the system. The smart home has to be accessible and discoverable. Users have the ability to ask for permission instead of being completely locked out. On-boarding of new users has to be an easy process.
- **Respect Among Users** Each user, regardless of the role within the system should have a feeling of control and security. Users should not have the ability to remotely control or automate devices that could surprise or disturb other users.

- **Transparency of Smart Home Behaviours** Any action or behaviour should be notified if it can affect other users. The smart home system should not allow users to take actions that can affect other users without notice.

To achieve **access control flexibility** and **respect among users** principles the authors designed the following access controls

- **Role-based Access Control** Control over specific devices is given to specific roles (neighbour, guest, child and so on).
- **Location-Based Control** Control over devices can be done only if a user is physically near the device. As an example: a guest can control the media system only if they are in your house.
- **Supervisory Access Control** Control over devices is given by other users that have a higher role. As an example supervisor is a parent of a child.
- **Reactive Access Control** This is a flexible approach where one user can ask permission from the user that has the required permission. As an example a children ask permission to use TV from their parents.

To achieve **transparency of smart home behaviours** and **user agency activity principles** notifications were introduced. Notifications can be used to notify other users about certain actions being triggered or give alerts about new changes that are happening in the environment. Users should have the ability to disable notifications for certain devices or actions.

The study however revealed that many participants opted not to use the access control functionality in their home systems as a result of the (social) norms, trust and respect already present in many families. In all families, there were no incidents when one user tried to control the devices of others. Some participants stated that access controls are way too restrictive and they want more control. Several participants complained about being overwhelmed by the amount of notifications they received. In smart systems, users should be able to control what kind and how often they receive notifications.

Finally, even though many users are not in favour of access controls it does not mean they are useless. Access controls are powerful and can be used in IoT systems outside of the home environment, for instance in the office, where privacy and safety should be ensured for all employees. This paper gives a lot of insights about access controls for users and how we can use notifications to alert users about changes in smart home environment.

3.6 Algorithms for Conflicts Resolution

Thyagaraju et. al. proposed several algorithms for conflict resolution in a multi-user environment [5]. The authors think that ideally conflict resolution should be done without user intervention. However, it is not always possible. The first proposed algorithm is a *Preemptive Priority-Based Conflict Resolution Algorithm*. The authors propose an algorithm that resolves conflicts based on priority. They gave an example of a family and their children. Father and Mother have higher priority over their kids. Hence, all conflicts where higher priority users are involved are resolved in the favour of high priority users. As a limitation, it does not cover the resolution of conflicts among users with similar priorities. The second algorithm is a *Preemptive Priority Based Conflict Resolution Algorithm*. This is an extension of the first algorithm which has an additional role - an admin. An admin can modify users' priorities per their interests or pattern. The third algorithm is a *Non-Preemptive Priority Based Conflict Resolution Algorithm*. This algorithm prefers not to disturb users with lower priority even when the higher priority is in the same room. In short, it means that if low priority user uses some devices, then high priority user cannot take over the devices until low priority user stops using those devices. The fourth algorithm is a *Time Slice Based Preemptive Priority Conflict Resolution Algorithm*. This algorithm is similar to the third one. The main difference is that this algorithm introduces a time duration after which high-priority users can take over the devices that are being used by low priority users. The fifth algorithm is a *Interactive Preemptive Priority Based Conflict Resolution Algorithm*. In this algorithm, users prefer to have consent whether they want to take over devices with lower priority or allow low priority users to continue using the device according to their preference. All the above-mentioned algorithms are better suited to multi-user environments where the number of people is less than 6. Further, we continue with algorithms that consider a higher number of people. The sixth algorithm is a *Democratic Group Preference Conflict Resolution Algorithm*. In this algorithm, every user provides their interests through individual profiles. Each profile can have a preference in different categories, such as sport, entertainment, security and so on. To resolve conflicts, the algorithm computes the group preferences for every category and considers that these people are in the same room. Summarising all preferences the algorithm receives the statistics that then can be used for making decisions. The seventh algorithm is a *Role and Age Factor-based Group Preference Conflict Resolution Algorithm*. In this approach role and age is considered in the conflict resolution algorithm. The seniority of the user matters and the higher the age, the higher the priority the user has. Moreover, this approach uses categories to calculate preference. After the system analyses the age, role and preferences the system can provide conflict resolution.

We want to emphasise that different conflict resolution algorithms were proposed. At the same time, the paper does not provide validation of the algorithms. Currently, it is not clear which algorithms are better in certain scenarios and which are not. Finally, the authors proposed only conflict resolution algorithms and have not mentioned intelligibility to aid users in understanding the reasons behind the conflict resolution.

3.7 VA and Conflict Resolution in Multi-User Environment

Bauyrzhan Ospan et al., describe a virtual assistant that is used as a control interface of smart home multi-user environment [27]. The authors introduced a conflict resolution system that uses Case-Based Reasoning to decide conflicts between several users. We see the similarity of the algorithm to the *Democratic Group Preference Conflict Resolution Algorithm* that was discussed in the previous paper [**4809007**]. The following system uses the argumentation of users to determine how conflict has to be resolved. The system compares the user's request with previous requests by use of the k-nearest neighbour algorithm. Then it uses best-fit cases. The system uses the user's explanation for use of devices that have conflict to determine which user should take over the device. The system makes prioritisation based on preference profiles that the user has in the system. The authors introduced the following preferences:

- Young (up to 25) - Security, Work, Entertainment, Health, Food, Energy
- Adult (26-79) - Security, Health, Work, Food, Energy, Entertainment
- Elderly (older 70) - Health, Security, Food, Energy, Entertainment, Work

Based on the user type, system decides who has priority in the conflict resolution. In the evaluation step, we noticed two negative responses regarding conflict resolution. Users disagree with the priority policy that the authors introduced. One of the participants noted one scenario, where the younger user wanted to turn on the lamp to do homework, at the same time rule of adult user had a preference to have the light turned off because of the entertainment time. The system decided to keep the lamp turned off and did not count homework as an exceptional case. We would agree that the system should have determined such a scenario as an exceptional case and it should have turned on the light in the room. Another negative response was related to the fact that the system encountered an unknown error and it was possible to solve the issue only by looking at the system logs, which of course is not

ideal for the user, since they probably do not have access to the system logs or they could have a non-technical background which could make it impossible or hard to analyse the problem. The authors' approach makes use of a machine learning algorithm and possibly could be improved in some areas. The system decides who has priority to solve the conflicts. However we think that such approach where age has importance in decision factor cannot be applied in all environments (e.g in working office age should not be important). We want to mention that the authors provide the full source code of their implementation with instructions on how to run and be able to reproduce the results. We also note that intelligibility mechanisms for conflict resolution are not considered in this work.

3.8 Kratos: Access Control System

Sikder et al., proposed a Kratos [19], a multi-user multi-device-aware access control system for smart homes. Kratos introduces formal language for the expression of policies (rules) and a policy negotiation algorithm that resolves and optimises conflicts. The authors indicate that existing solutions are vendor and device-specific and they fail to deliver the diverse and complex user demands in a multi-device multi-user setting. Their work is based on existing user studies [28, 1] among smart home studies. Based on those studies the authors concluded that users need fine-grained access control capabilities. The users suggested having role-based access controls in a home environment. It is important to note that participants demonstrated a willingness to have a system that can resolve conflicts without user intervention. Regarding terminology the authors introduced **Policy**, **Priority** and **Conflict**. A policy is a request made by a user to control device usage in a multi-user smart environment. The authors introduced three categories of policies:

- "**Demand Policy** is a request made by a user that defines the control rules for a specific device or group of devices in the smart home system" [19]
- "**Restriction Policy** is the set of rules that defines a level of control of a user or group of users to a certain device or group of devices in the smart home system" [19]
- "**Location-based Policy** is a set of automation rules enforced by the user that are only applicable if the user is connected to the system network" [19]

The authors describe *Priority* as the importance level of a user that is used to create preferences for users of a higher category over users with lower priority.

Furthermore, they defined *Conflict* as the conflicting process that is created from two or more demand policies that interfere with or contradict based on the specific requests of the policies. Three types of conflicts are defined:

- "**Hard Conflict**. A conflict occurs when *demand policies* of a specific device are enforced by users that do not have *overlapping policies*" [19]
- "**Soft Conflict**. A conflict occurs when *demand policies* of a specific device are enforced by users that have *overlapping policies*" [19]
- "**Restriction Conflict**. A conflict happens when the *restriction policy* is disputed by a *restricted user*" [19]

The authors proposed to assign importance of the users by setting priority score to each user. Upon registration, the user receives the lowest priority score. Other users can assign a priority score to other users if they have higher priority over the one they want to assign. Besides they can only assign priority up to the equal number they are assigned. A conflict resolution algorithm can resolve conflicts when users have different priorities. If two users have the same priority level and have conflict then a negotiation algorithm comes into place. The algorithm calculates a medium value for policy and sends a suggestion to the users who have conflicts. Therefore this system provides a form of intelligibility. If users accept the negotiated policy then the policy is set otherwise, the system sends decision requests to users with higher priority or to administrator users. This is also a form of intelligibility. The authors conducted several experiments that demonstrated the efficiency of the proposed approach. The system was able to resolve all conflicts in all scenarios. However, the authors did not conduct any user study to validate the proposed solution among users. Furthermore, the system is intended for home automation, hence for the limited number of users and policies. Therefore the intelligibility mechanisms they provided are limited to home automation.

3.9 Evolving needs in IoT control and accountability

Jakobi et al. discuss evolving needs in IoT Control and Accountability [14]. The authors conducted social research among 12 families to investigate how user demands evolve with time regarding smart home feedback for supporting the maintenance of the system. Smart hardware was provided to the participants. Devices that were taken by participants are thermostats, motion sensors, brightness detectors, door/windows sensors, smart plugs, remote controls and

smoke detectors. The list is not exhaustive since participants could add more devices during the experiment. In terms of software, their setup consisted of *if-then-else* rules. For visualisation purposes, web and mobile clients were provided. During analysis of the feedback from participants, it became clear that most of the participants do not trust the system and tend to verify how the system is working occasionally by use of a dashboard. Furthermore, participants wanted to have more useful information about how the system is operating, but at the same time without being overwhelmed with unnecessary information. It is important to note that the solutions that were proposed in this experiment do not cover *true* multi-user scenarios. To improve intelligibility the authors suggested that the system that collects log information should identify routine tasks and only give information about deviations. In addition, similar events can be grouped and it would help reduce the amount of information that users need to verify. Finally, the system should continuously provide information about its state upon users' request.

3.10 Intelligibility and control for context-aware IoT

Coppers et al. proposes *FORTNIoT-FortClash* system. This system is used to mediate conflicts in the near-future in smart homes [4]. Existing conflict resolution solutions require modifying rules to avoid conflicts. There are some cases when conflict occurs occasionally or rarely and a user would want to manage conflicts manually. The authors proposed an approach called *FORTNIoT-FortClash*. They describe it as:

> An approach to predict many different types of conflicts in the near future, detect and provide a mediation mechanism that can suppress a rule's actions at specific moments.

Existing solutions require a user to compose rules in such a way that they do not cause conflicts with each other otherwise the user needs to modify the rules. This is not optimal because you need to compose flawless rules which do not conflict with each other. As alternative solution, the authors proposed to allow a user to temporarily suppress a rule's action, hence you do not need to manage all conflicts at the same time. They proposed three contributions:

- An approach to predict and detect conflicts in the near future in a smart home.
- A mechanism to mediate conflicts by suppressing the execution of actions at a specific moment.
- An evaluation of the extent to which *FORTNIoT-FortClash* is suitable for detecting and mediating different kinds of conflicts.

FortClash has full support for the most known conflict types, such as inconsistent actions, redundant actions, race conditions and loops. In addition, **FortClash** supports mediation of redundant triggers, condition bypass, and automatic action reversal. The limitation factor for **FORTNIoT-FortClash** is the incapacity to resolve conflicts that can only be mediated by modifying the condition itself. Another limitation is that *FortClash* does not consider multi-user scenarios, instead they focus on single user scenarios.

Chapter 4

User Study - Office Environment

We conducted a user study via an online questionnaire to determine participants preferences in multi-user environment scenario. We have selected office environment since such an environment may have different people who want to use IoT devices and people may have limited trust among each other.

We also created various visualisations examples in order to validate user preferences in how user rules, its conflicts and notifications should be visualised.

The survey was created in *Qualtrics* and consists of 47 questions. The *Likert Scale* is used to evaluate participants preferences for the proposed features.

The survey received responses from 24 participants. Most of the participants are male (75%). 88% of participants are in the range between 20-39, the rest are in the range of 40-59. The majority have a bachelor's degree (62%), while the rest are holders of a master's degree. It should be noted however that over 87% of participants have programming experience.

In our survey we have both domain-specific questions and questions to determine the demographics of our participants, such as gender and age. Additionally we ask participants about their programming experience and whether they have experience with IoT systems. Overall we have *17 main domain-specific questions*.

Demographically we have participants from several countries. Here is the list of countries that indicates participants location:

- Azerbaijan - 4 votes
- Belgium - 3 votes
- Russian Federation - 2 votes
- Turkmenistan - 2 votes

- Bulgaria - 1 vote
- The Czech Republic - 1 vote
- Brazil - 1 vote
- Germany - 1 vote

4.1 Environment Definition

In our user study, we consider the office as a *true* shared multi-user environment as opposed to the home environment which is chosen in most related work. In the solution proposed by Zeng et al [1], the participants opted not to use proposed features such as *Reactive Access Control* and *Supervisory Access Control*. One of the reasons cited for this is the fact that most users in the home environment have an implicit trust towards other users since they are usually family members or people who have been living together for some time already. We thus selected the office environment which reduces the chance of such familiarity-bred trust since one colleague may not necessarily know or trust another colleague as much as family members know or trust one another.Therefore, users within this environment can have different levels of trust. The trust between colleagues in *team A* may be higher than the trust colleagues in *team A* have between the colleagues in *team B*. This would thus lead to a need for having intelligibility mechanisms which could help to improve the overall user experience.

4.2 Survey Scenario

Consider you and a user called Aleksandra are colleagues at *VerySerious* Company. You both work from home Monday to Thursday and come to the office on Friday in order to collaborate and have debriefing meetings with other colleagues before the weekend. You are also both avid IoT users and have recently bought and installed new smart devices and configured rules to manage your homes.

Both you and Aleksandra store your rules in *private Solid Pods*. Solid is a storage architecture that enables decentralised storage of data. It enables you and Aleksandra to always maintain ownership of your rules (data) and to give access to applications that wish to make use of those rules, as opposed to applications owning the rules (data). Therefore, you and Aleksandra have

granted the IoT systems in your respective homes *READ access* to your IoT rules.

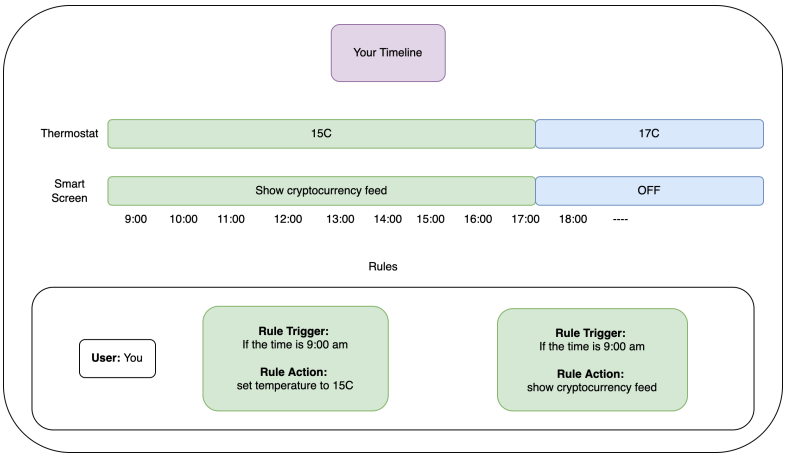


Figure 4.1: Your Rules

You have recently bought a smart thermostat to control the temperature in your home as well as a smart TV. You recently configured the following rules for use in your home:

If the time is *9:00 am*, then set the temperature to *15 degree Celsius*

If the time is *9:00 am*, then *show the current cryptocurrency trading prices and my cryptocurrency portfolio on the smart TV screen*

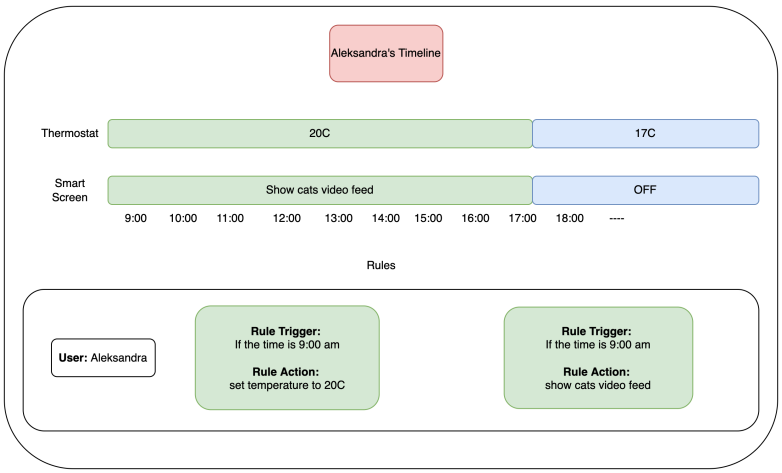


Figure 4.2: Aleksandra Rules

Aleksandra has also recently bought a smart thermostat to control the temperature in her home, as well a smart TV and camera and has configured the following rules for use at her house:

If the time is *9:00 am*, *then set the temperature to 20 degree Celsius*

If the time is *9:00 am*, *then show a video feed of my cats on the smart TV screen*

VerySerious company has informed its employees that it has recently installed a smart thermostat and smart TV in the brainstorming room on every floor and has granted its employees free access to these devices. Employees can give the IoT system in *VerySerious* company *READ access* to the rules in their SOLID pods and have the rules be executed by the system. All employees have equal rights in the office and equal access to smart devices but the use of the smart devices currently works on a first-come-first-served basis. The workday starts at *9:00 am* and ends at *5:00 pm*. The company has implemented the rules:

If the time is *5:00 pm*, *then set the thermostat to 17°C*

If the time is *5:00 pm*, *then set the smart screen to OFF*

As can be seen from the given scenario, both your rules and Aleksandra's rules have conflicting actions at 9 am. Your preferred temperature setting at *9:00 am* is *15 degree Celsius*, while Aleksandra's is *20 degree Celsius*. You also expect to see the current cryptocurrency trading prices and your cryptocurrency portfolio on the screen in the brainstorming room at *9:00 am*, while Aleksandra expects to see a video feed of her cats. You and Aleksandra come to the office on a busy Friday with other colleagues and both expect your configured home rules to keep working in the office. You are both unaware of each other's rules. You arrive at the office before everyone else, therefore the current temperature is set to *15 degree Celsius* and the smart screen is currently displaying the current cryptocurrency trading prices and your cryptocurrency portfolio. Aleksandra does not understand why her rules are not being executed by the system at the office when she arrives and is worried about the welfare of her cats.

4.3 Survey Content

In this section we demonstrate main questions we asked from participants and at least one response was received.

- Q1: How would you rate a feature that allows you and Aleksandra to view each other's rules on your respective timeline apps including possible conflicts?

- Q2: How would you like these possible conflicts to be depicted on the timeline application?
- Q3: How would you rate a feature that allows you to only reveal information which helps Aleksandra to see if her rules will be in conflict with yours?
- Q4: Which of your information do you consider is necessary to display on the timeline app to only help Aleksandra to see if her rules will be in conflict with yours?
- Q5: Do you consider this rule suppression feature to be appropriate?
- Q6: How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours?
- Q7: How would you prefer this information be presented to you on the timeline application?
- Q8: What kind of information do you require from Aleksandra in order to be able to make a decision on whether or not to let her rule take priority over yours? Keep in mind that this choice will also be applicable to your rules.
- Q9: Would you prefer that the timeline application automatically determines which rules should be suppressed or executed?
- Q10: What kind of information do you think this decision should be based on?
- Q11: Would you like to receive information from the system about why your rule was not executed if the system determines it should be suppressed?
- Q12: How would you like the reason for why the system determined your rule should be suppressed to be displayed on the timeline application?
- Q13: Would you prefer that the timeline lets an administrator determine which rules should be suppressed or executed?
- Q14: What kind of information do you think this decision should be based on?
- Q15: Would you like to receive information from the administrator about why they suppressed your rule?

- Q16: How would you like the reason for why the administrator determined your rule should be suppressed to be displayed on the timeline application?
- Q17: How would you rate a feature that suggests a compromise rule that is based on two conflicting rules? As an example, it could suggest 18C temperature for both users

4.4 Survey Results

In this section, we present the results gathered from the user survey. Based on the answers to question Q1 users would like to have a feature that allows viewing their rules and the rules of other users as well as possible conflicts between rules. As shown in Figure 4.3, all participants (15 out of 15) found it 'Important' or 'Very Important' to have a feature that shows the rules of other users as well as which rules are in conflict with theirs.

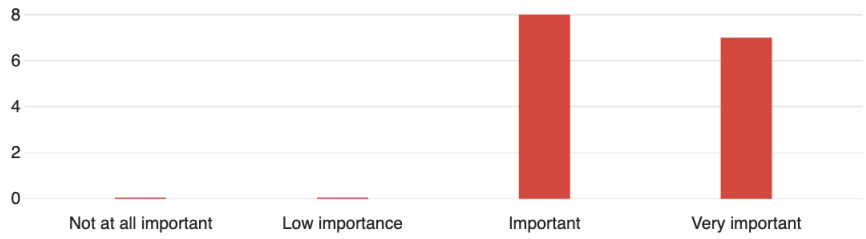


Figure 4.3: Question 1 Responses

Finding 1: *"Users want to be aware of the rules of other users in their environment and if those rules conflict with theirs"*

The positive response to the need for the above-mentioned feature enables us to ask a question regarding which visualisation is preferable for depicting rules and conflicts. Figure 4.4 depicts a visualisation example where conflicts are depicted on timeline with the use of exclamation (!) red mark icon that show conflicting rule. While figure 4.5 depicts a visualisation example where conflicts depicted on the rules with the use of exclamation (!) mark icon where red icon means that rule has conflict. Additionally the orange icon depicts rules of other users that have conflict. Based on the responses of the participants, figure 4.6 is the preferred visualisation for 8 out of 15 participants.

Figure 4.4: Question 1 - Visualisation Example 1

Figure 4.5: Question 1 - Visualisation Example 2

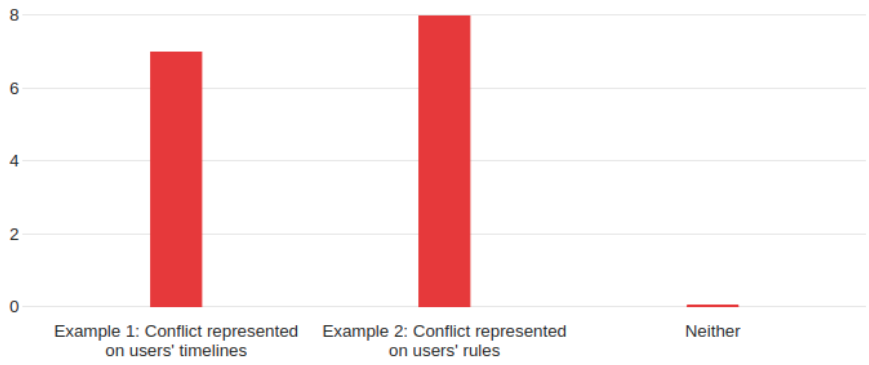


Figure 4.6: Question 2 Responses

According to the responses to question Q3 all participants would prefer to limit the information shown to other users when informing them about rules which may be in conflict. As shown in Figure 4.7 majority (10 out of 16) of participants identified the limitation of the required information as 'Important', while others (6 out of 16) identified it as 'Very Important'.

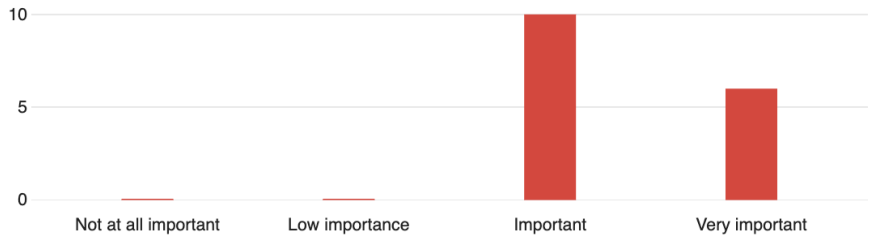


Figure 4.7: Question 3 Responses

In question Q4 we asked participants to choose what kind of information they are open to sharing with others users to help determine when rules may be in conflict. Participants were able to select several responses if they had multiple preferences. Figure 4.8 shows the majority of participants (10 out of 24) show a preference for showing only the conflicting device. This is followed by a preference for showing the complete rule which was selected by 6 out of 24 participants and finally a preference for showing only the rule trigger which was selected by 5 out of 24 participants. The least preferred option was showing *only a username*. Based on this result we assume that some participants

want to limit the information shared with other users as much as possible, while others are more open for sharing information. We assume that showing only conflicting rules may not be sufficient to identify the reason why a certain rule is being suppressed. This is the reason why we think that showing a combination of *conflicting devices* and *rule triggers* should be used in a system.

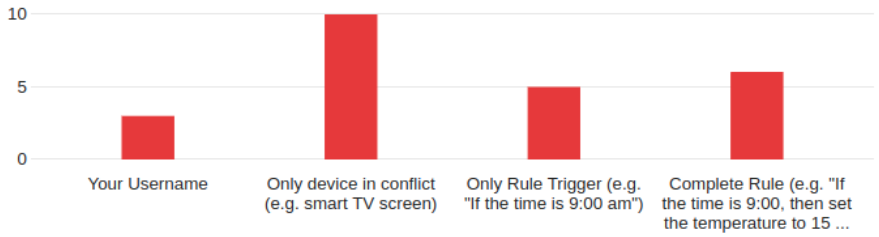


Figure 4.8: Question 4 Responses

Finding 2: *"Users would like to have intelligibility on possible conflicts with other users' rules but want to limit the information the information that is shared in order to provide this"*

Next, based on the related work [4] we asked participants to imagine that the timeline application received a new feature update which allows a user to suppress the rules of other users. This feature would help the user to make their rule take priority over the rule of another user. Based on the answers to question Q5 majority of participants (12 out of 17) consider rule suppression to be appropriate. Figure 4.9 demonstrates that 12 participants responded 'Yes' while 5 participants responded 'No'.

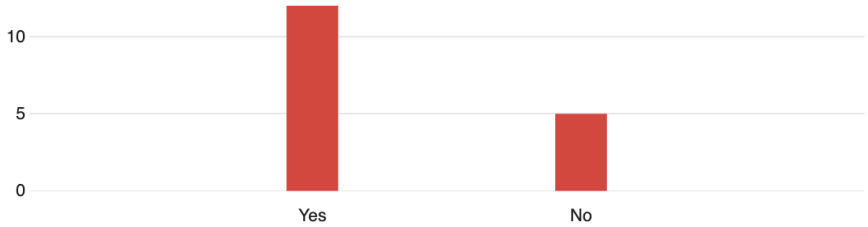


Figure 4.9: Question 5 Responses

Participants who responded 'No' on the previous question received an additional question Q6 where we ask if they would like a feature that enables users to ask other users for permission to suppress their rules. This question was based on the work of Zheng et. al. [1] where they proposed reactive access control mechanism. Figure 4.10 demonstrates that the majority of participants (3 out of 5) voted for having such a feature.

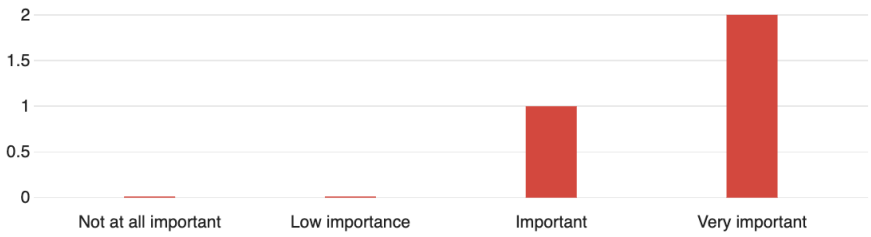


Figure 4.10: Question 6 Responses

We also asked the participants in question Q7 which visualisation they prefer to show suppression request notifications. Figure 4.11 shows a visualisation example of a notification popup on the dashboard. Figure 4.12 shows a notification popup on the timeline. Figure 4.13 show that participants have a slight preference for having a notification popup window on the dashboard instead of showing it on the timeline itself.

Based on the related work [19, 27, 5], in order to understand what kind of information should be used as a decision basis for rule suppression, we asked participants question Q8. Figure 4.14 shows that 3 participants (3 of 11) selected *rule category* as decision factor. *Rule Category* shows to which category falls the rule. *Security*, *Health* are examples of *rule category*. *Username* was chosen by two participants (2 of 11). 2 out of 11 participants chose: *A simple message with the reason*, *time duration of rule execution* and *user category*.

Based on the related work [21], in order to validate whether participants prefer to have automatic conflict resolution we asked question Q9. Figure 4.15 demonstrates that the majority (14 out of 17) would prefer to have automatic conflict resolution.

Finding 3: *"Most users show a preference for system-based rule suppression"*

In question Q10, based on the related work [27] we asked participants which factors should be used as a decision basis for automatic suppression.

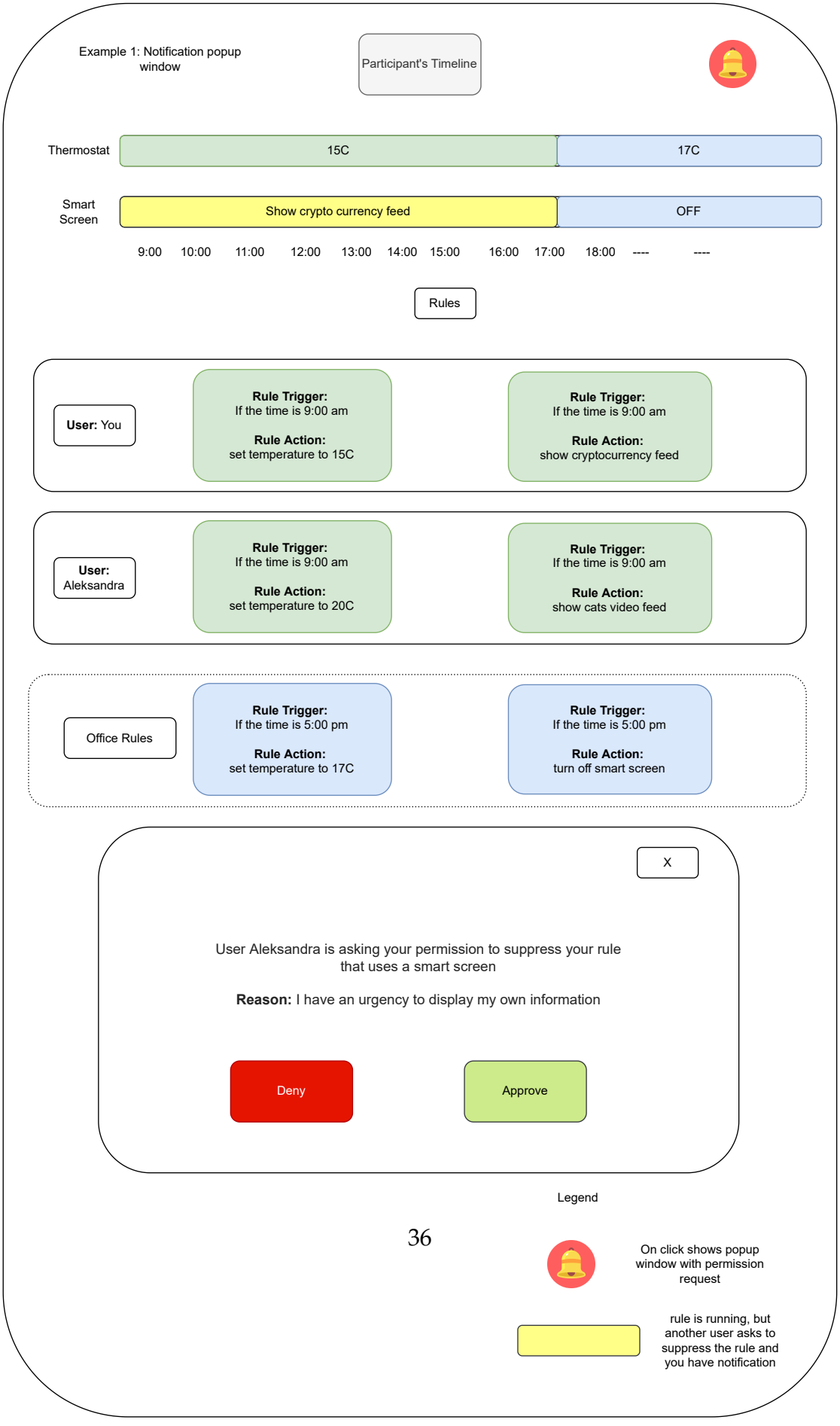


Figure 4.11: Question 7 - Visualisation Example 1

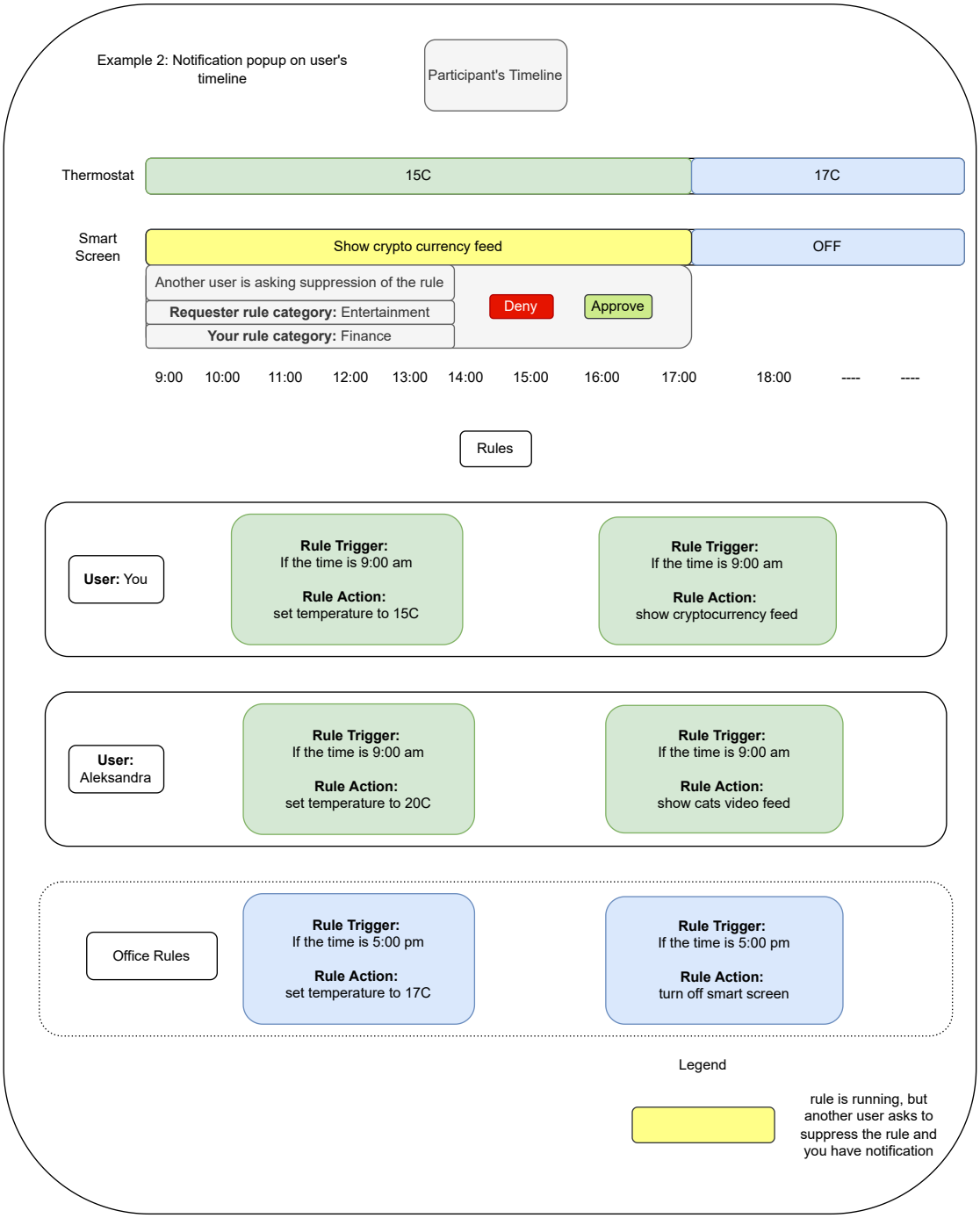


Figure 4.12: Question 7 - Visualisation Example 2

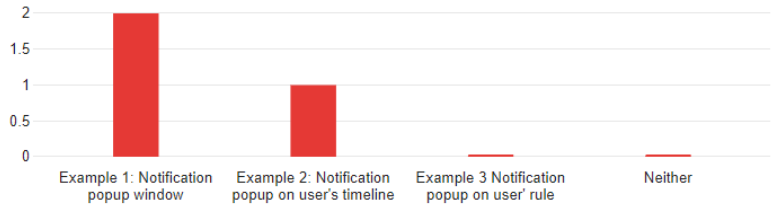


Figure 4.13: Question 7 Responses

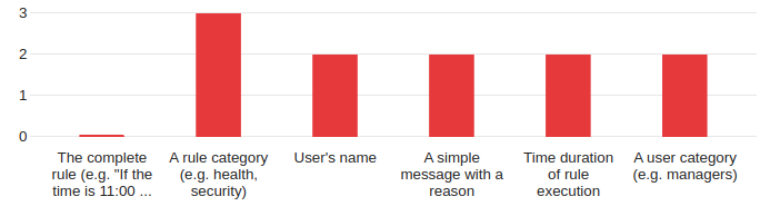


Figure 4.14: Question 8 Responses

Based on Figure 4.16 we see that most popular response is *rule category* factor followed by *User Proximity* which was chosen by 4 out of 14 participants.

Figure 4.15: Question 9 Responses

Finding 4: *"Users prefer 'Rule Category' as decision factor for automatic conflict resolution"*

Based on responses from question Q11 we see that participants desire to have information regarding why their rules may have been suppressed. The majority of participants (11 out of 14) would like to know why their rule was suppressed. In question Q12 we asked participants how we should visualise this information. We provided two visualisation examples. The first example where the suppression information is shown on the user's timeline is depicted

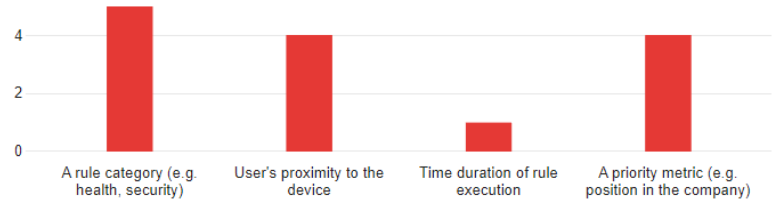


Figure 4.16: Question 10 Responses

on Figure 4.17 with the *mechanical icon* that depicts that the rule has been suppressed. The second example which shows the suppression information on the user's rules is depicted on Figure 4.18 with the *mechanical icon* that depicts that the rule has been suppressed. Figure 4.20 demonstrates results where we notice that more than half (7 out of 13) of the participants would prefer to show suppression information on the user timeline.

Finding 5: *"Most participants would like to receive feedback when the system suppresses their rules"*

In question Q13, also based on the work of Barkhuus et. al. [21] we asked participants whether administrator-level users should be able to decide which rules should be suppressed. Figure 4.21 demonstrates that the majority (12 out of 16) of participants positively responded to this question. We asked participants (Q14) also based on the work of Thyagaraju et. al. [5], which factors administrators should use to determine which rules should be suppressed. *Simple message* and *Priority Metric* were the most chosen options. This is depicted in Figure 4.22.

Figure 4.25, demonstrates that all participants want to receive feedback of their rule is suppressed by an administrator.We provided two example visualisations in order to depict this feedback and we asked participants to select which visualisation they prefer. Figure 4.23 depicts this feedback on the user's timeline with the use of *person icon* that depicts that the rule has been suppressed by an administrator. In contrast, figure 4.24 depicts the feedback on the user's rules with the use of *person icon* that depicts that the rule has been suppressed by an administrator. The results shown in Figure 4.26 show that more than half participants have a preference for having the feedback shown on the users' rules depicted on Figure 4.24.

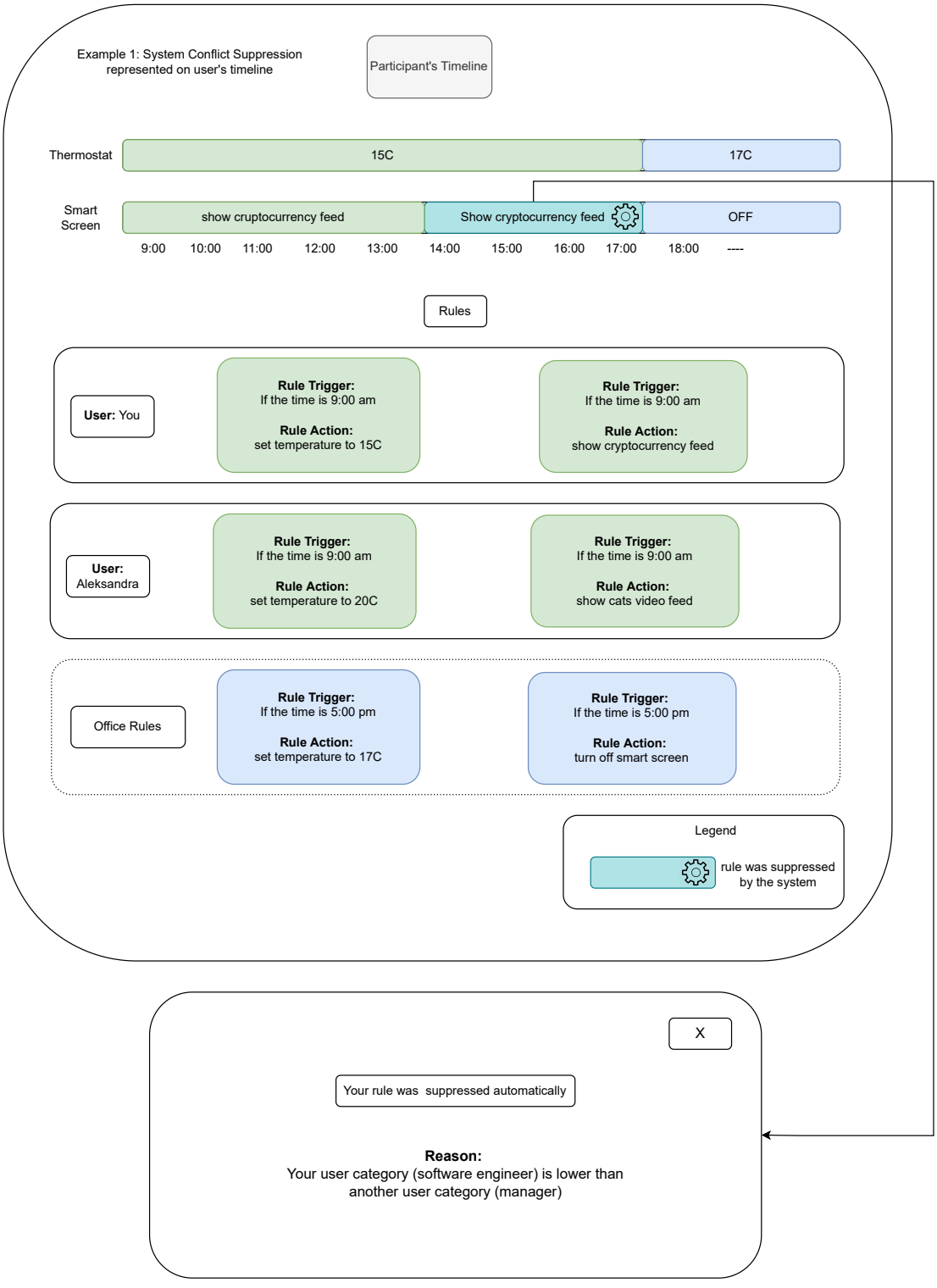


Figure 4.17: Question 12 - Visualisation Example 1

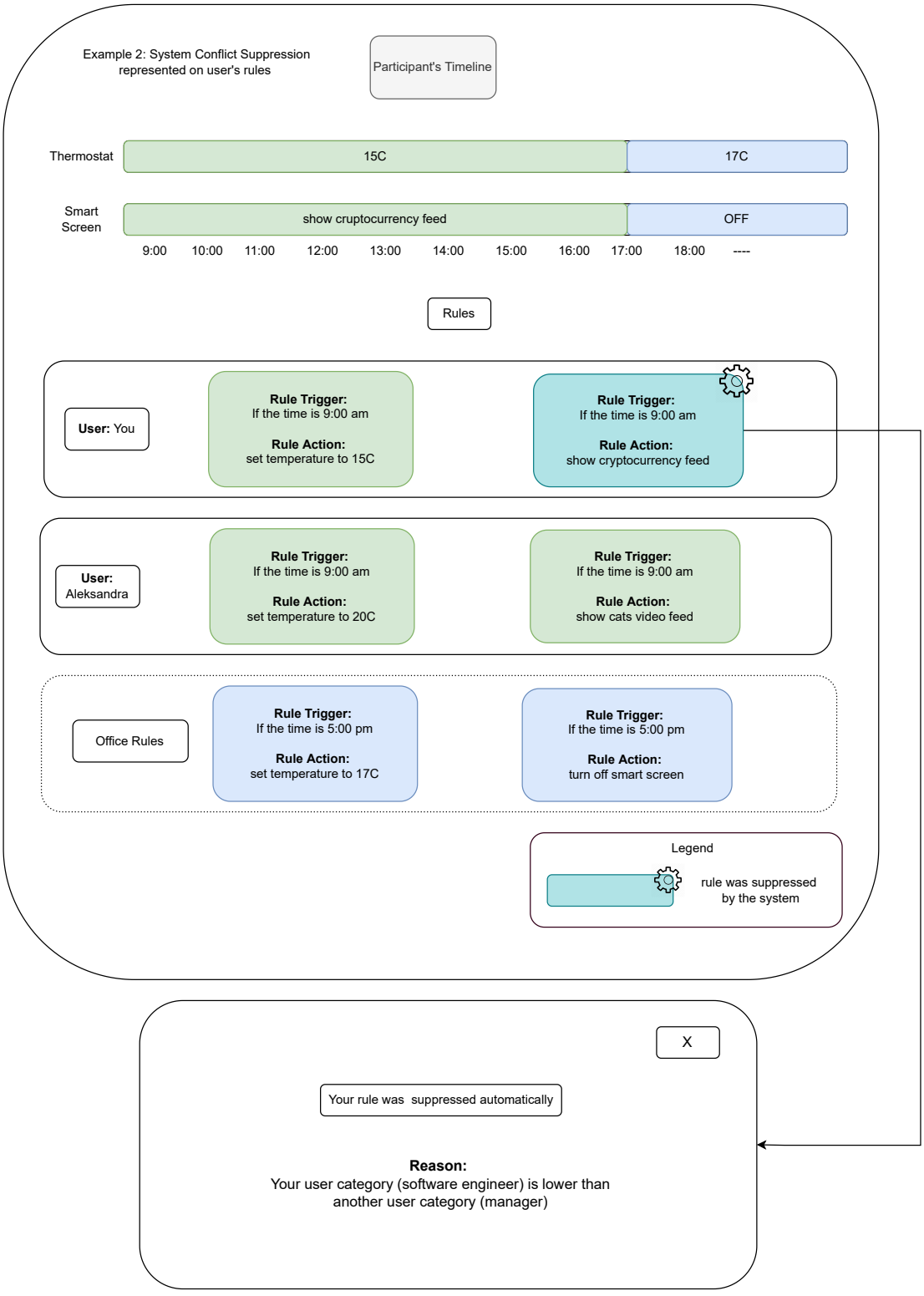


Figure 4.18: Question 12 - Visualisation Example 2

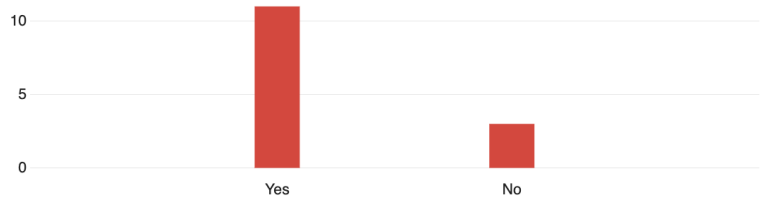


Figure 4.19: Question 11 Responses

Figure 4.20: Question 12 Responses

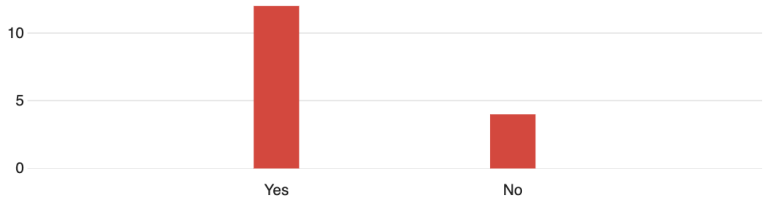


Figure 4.21: Question 13 Responses

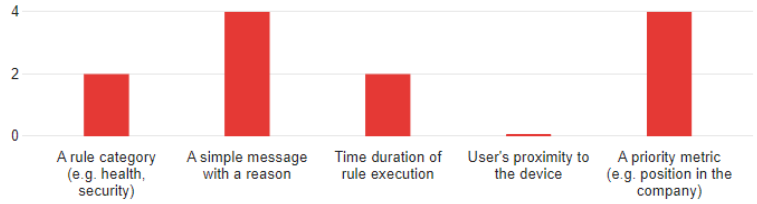


Figure 4.22: Question 14 Responses

Finding 6: *"Most participants are also in favour of having admin-based rule suppression"*

Additionally based on the related work [19], we asked participants (Q17) if they are in favour of having a new rule which is intended to be a compromise between conflicting rules. For example, in a situation where two rules set ambient temperature in the room to different temperatures, an alternative temperature can be suggested which may be satisfactory to both users. This mechanism can be applied for conflicting rules which have the same rule category. Figure 4.27 demonstrates that the majority of participants (12 out of 16) are in favour of such a feature.

Finding 7: *"Most participants are in favour of having a new rule which is a compromise between conflicting rules"*

Providing users feedback that such a compromised has been reached can be done in a similar way to how we provide feedback for suppressed rules.

4.4.1 Users' Remarks

We have collected several remarks from the participants. One participant stated that they would like to see more devices and rules that are conflicting with each other. The examples that we have are quite limited. Another participant stated a remark regarding rule compromise examples:

> Intermediate rules can be applied to both scenarios mentioned in this survey (e.g. split-screen for TV to demonstrate both tasks). The survey needs more cases describing the situation where the app/admin should decide whether the intermediate rule is applicable, or a task with higher priority must be selected.

Another participant stated similar remarks:

> It will be better to include more scenarios with conflicting interests and involve more smart devices.

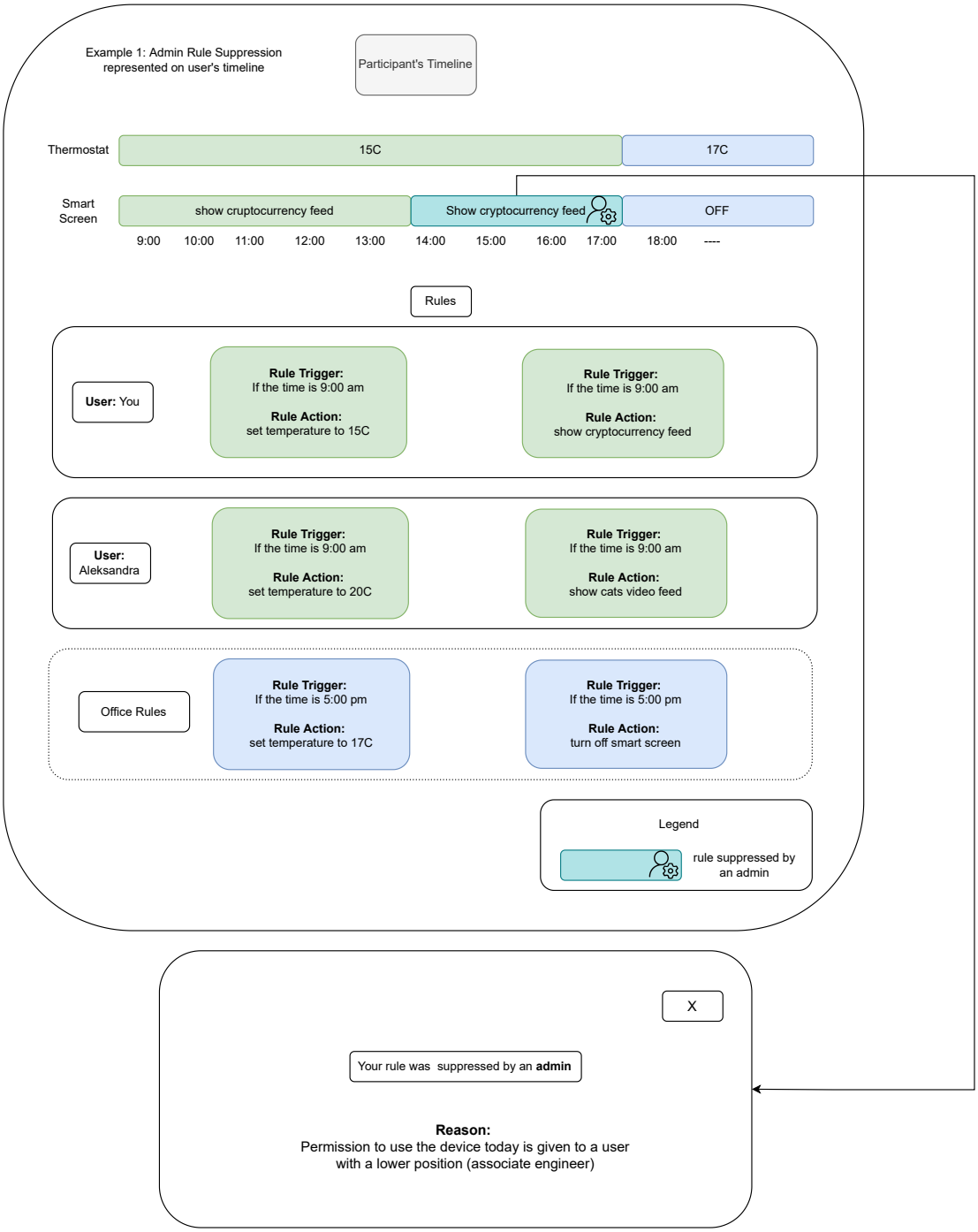


Figure 4.23: Question 16 - Visualisation Example 1

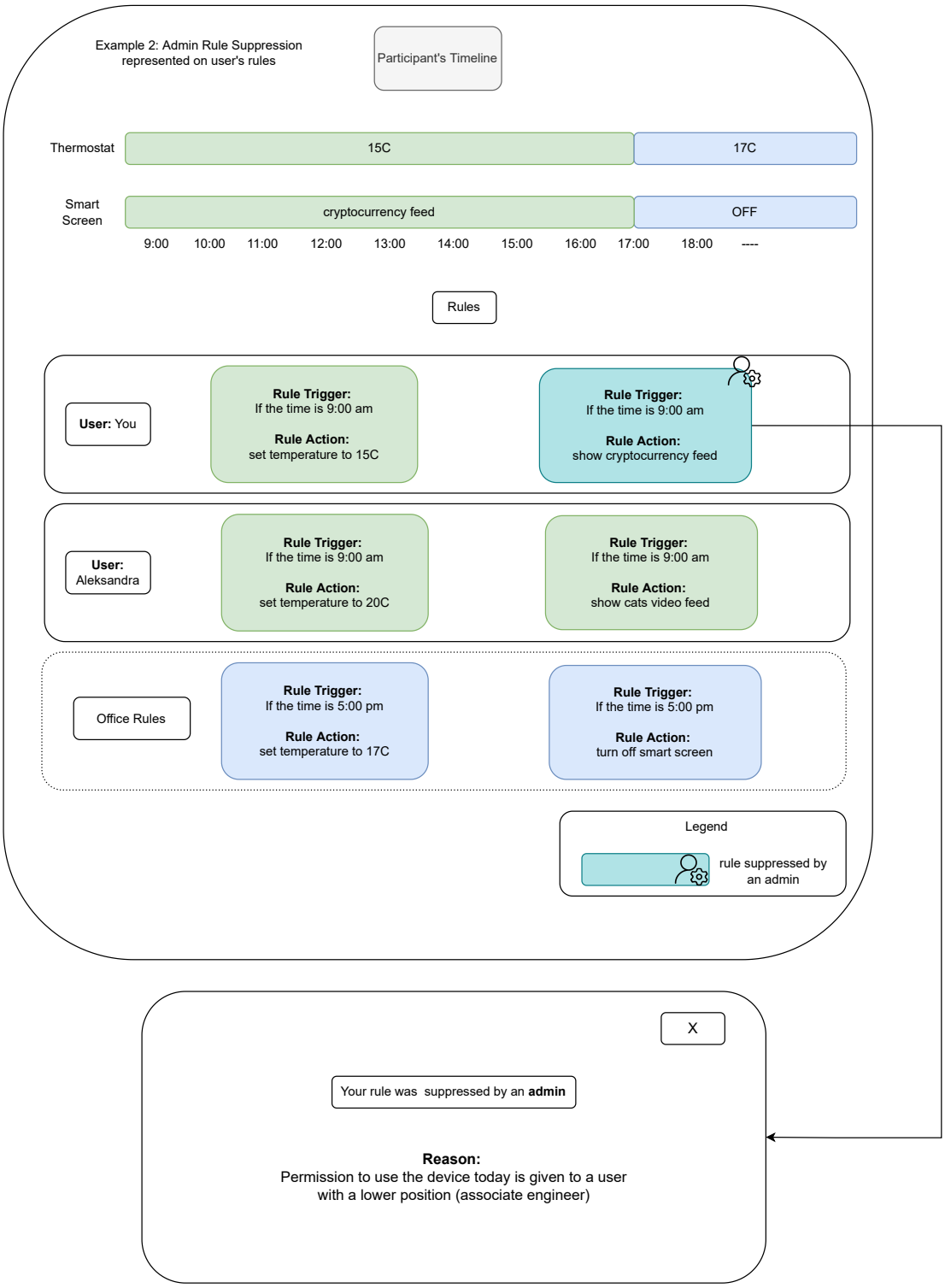


Figure 4.24: Question 16 - Visualisation Example 2

Figure 4.25: Question 15 Responses

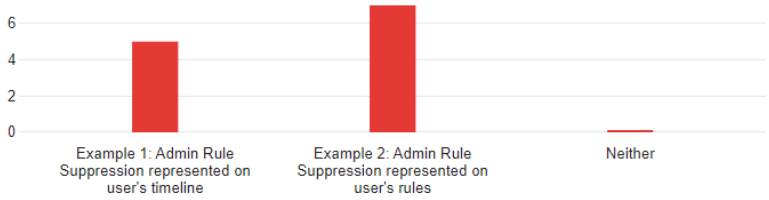


Figure 4.26: Question 16 Responses

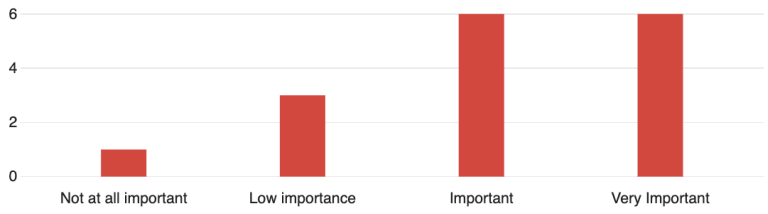


Figure 4.27: Question 17 Responses

4.4.2 Survey Conclusion

In a conclusion, we were able to determine 6 findings related to intelligibility in multi-user IoT environments. Participants have shown interest in automatic conflict resolution and corresponding notification mechanisms. They are interested in a system that can resolve conflicts automatically. At the same time participants want to know which rules are suppressed with a reason. Additionally, participants have shown interest in having intermediate rules in situations for conflicting rules where possible.

Chapter 5

Design Guidelines

In this section we describe design guidelines based on prior work and the results of our user study. We suggest that these guidelines are important for supporting intelligibility in systems that are used in multi-user IoT environments.

5.1 Data Ownership

Attoh and Signer determined the problem of data ownership where automation systems store all user's data in a centralised data store. This leads to the problem where users cannot control the data, since they simply do not own it. To solve this issue Attoh and Signer proposed to use *SOLID* as decentralised data store [29]. Task automation systems can use *SOLID* as decentralised data store that solves data ownership problem.

Solid is a decentralised platform for the social Web. Solid is based on Semantic Web technologies [30]. In *SOLID* each user stores data in an online independent storage space called a *pod*. REST calls are used to communicate between pods. Application data is organised as documents and identified by Uniform Resource Identifiers (URIs). *OAuth* and *OpenID* protocols are not suitable for *RDF-based* profile data identification. Therefore *WebID* is used for implementation of global identity management. Each pod has an access control list. Users have control over the data access. Users can revoke access. At the same time, other users or applications can request access.

5.2 Users Rules and Conflicting Rules Visualisation

In our survey we asked participants whether they would like to have visualisation that shows rules of other users as well as any conflicting rules. Survey responses reveal positive feedback for having such a visualisation feature. Based on our survey we propose task automation systems have visualisation component that can demonstrate users' rules and possible conflicting rules to help identify conflicts and their reasons.

5.3 Limited Sharing Of Rule Information

The results of our user survey show that users do not want to share complete information regarding their rules with other users. They would rather prefer to limit the information shown to other users in order to identify possible rule conflicts. Respondents are open to sharing information regarding conflicting devices and rule triggers. We propose that task automation systems should limit the rule information that is shared among users for privacy reasons.

5.4 System-based and Admin-based Conflict Resolution

The survey results have shown that majority of participants prefer to have a system-based and admin-user conflict resolution. The survey has shown that *rule category* should be used as a factor for conflict resolution. Conflicting rule that has higher *rule category* should be selected and conflicting rule with lower *rule category* should be suppressed.

Furthermore participants want to have intelligibility mechanisms that can explain suppressed rules. They have shown preference to have notification means as intelligibility mechanisms. Task automation user interface should have visualisation indicators that can depict whether rule is in conflict. Besides, task automation should have views that explain why certain rules are suppressed.

Chapter 6

Solution

6.1 Multi-User IoT Dashboard

In this section, we demonstrate a proof of concept dashboard application which mimics the application presented to users in our survey and implements the design guidelines that were proposed in the previous chapter. We assume that users have authored their rules using their preferred authoring tool. This application consists of a front-end part and a back-end part. The first part is a web user interface application where users first need to authorise and then be able to see devices and automation rules they have. The application helps users to identify which rules they have, which rules are running and which rules were suppressed by the automation algorithm. We call this application *Multi-User (MU-IoT) IoT Dashboard*.

The MU-IoT dashboard application is written with the following technology stack:

- Javascript
- Node.js
- Express.js
- SOLID
- MongoDB
- Mongoose
- Bootstrap 5
- Vis.js

6.1.1 MU-IoT Dashboard Architecture

In this subsection, we describe the architecture of the proof of concept application. Figure 6.1 demonstrates high-level architecture diagram.

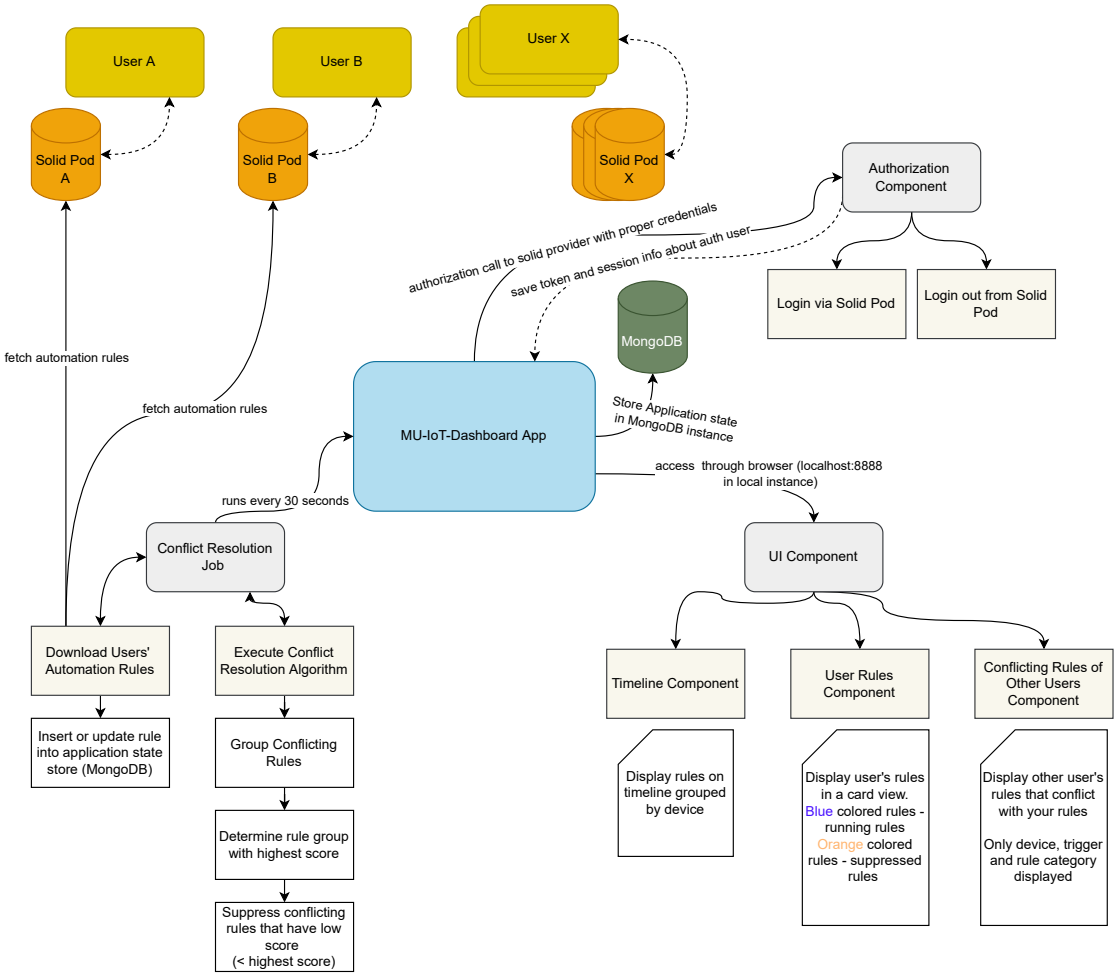


Figure 6.1: MU-IoT Architecture

As can be seen on Figure 6.1 the application consists of three main components:

- Authorisation
- Conflict Resolution
- User Interface

Authorisation Component

This component is responsible for user authorisation to their solid pod to prove user identity. The dashboard allows authorising and logout from the dashboard. This component stores user data in browser sessions.

Application State Validator Component

This component has several responsibilities, such as recurring fetch of users' rules from solid pods, update of state store and execution of conflict resolution algorithm. The component fetches users' rules from their pods every 30 seconds. This value can be tweaked by dashboard administrators. Every time the application successfully fetches users' rules it updates the state store. The application has a *MongoDB* database instance as a *state store*. The *state store* stores the users' rules, *rule category* selection and rules suppression logs. For state store we decided to use *MongoDB* database. This database is a *NoSQL* database type that has no schema requirements. This is a perfect choice since the implemented POC application architecture can change in the future work. The purpose of the state store is to have some context that can be then used for visualisation purposes.

The next step after the update of the state store is done is the execution of conflict resolution. One of the design guidelines requirements is to have system-based suppression mechanism where *rule category* factor is considered as deciding one. In related work Ospan et al. proposed a rule priority that differs based on user age category [27]. However, in our survey no participants indicated need to consider *age* as decision factor for conflict resolution. Therefore we decided that office system administrators should provide rule categories priority. In our solution administrators need to specify rule category priority by providing numerical value (e.g. 0 to 10) where higher category score rule should be prioritised. Therefore this score is internal representation of *rule category* to decide which rule has higher priority. As an example, if *rule A* has category health which has a score *8* and *rule B* has category entertainment with a score *3* then the latter rule is going to be suppressed if a conflict between the two rules arises. In our application, rules are in conflict when they have the same trigger condition but intend to perform different actions on the same device. For example one user's rule sets the smart thermostat to 20 degrees if the time is 9:00 AM while another user's rule sets the smart thermostat to 25 degrees if the time is 9:00 AM. Moreover, in a situations when conflicting rules have the same *rule category* the system cannot decide which rule to prioritise. Design guideline states that *admin-based* conflict resolution that can be used instead of *system-based* conflict resolution. In a situations when system cannot

determine which rule should run the admin user can resolve conflict manually. For proof-of-concept purposes, the admin user has to modify the status of the rule in the state store (MongoDB instance).

User Interface Component

In this subsection, we demonstrate the User Interface implemented for the proof of concept. The component consists of following sub-components:

- Authorisation Page Component
- Timeline Component
- Rules View Component

Users can use the dashboard only after successful authorisation via solid pod authentication. The dashboard uses the vis.js library that helps us to visualise items on the timeline. Bootstrap 5 CSS framework is used for the rapid development of the user interface. The dashboard application is implemented as a server-side rendered application which means that reload of the page is required to fetch the latest information.

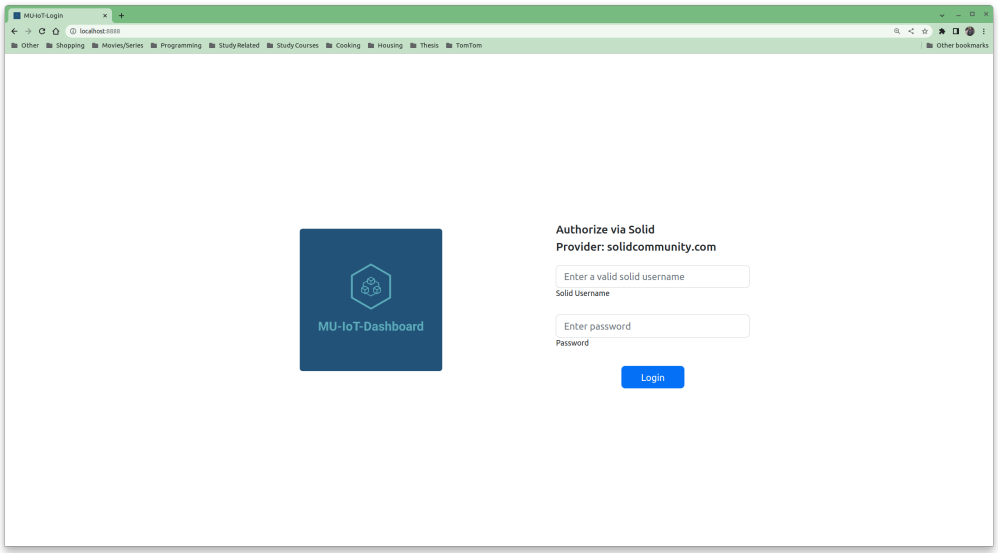


Figure 6.2: MU-IoT Dashboard Login Page

Figure 6.2 demonstrates login page. Users are required to fill in the correct login and password solid credentials to further move into the dashboard main page. In case incorrect credentials are entered the user is notified by a notification message at the top of the page.

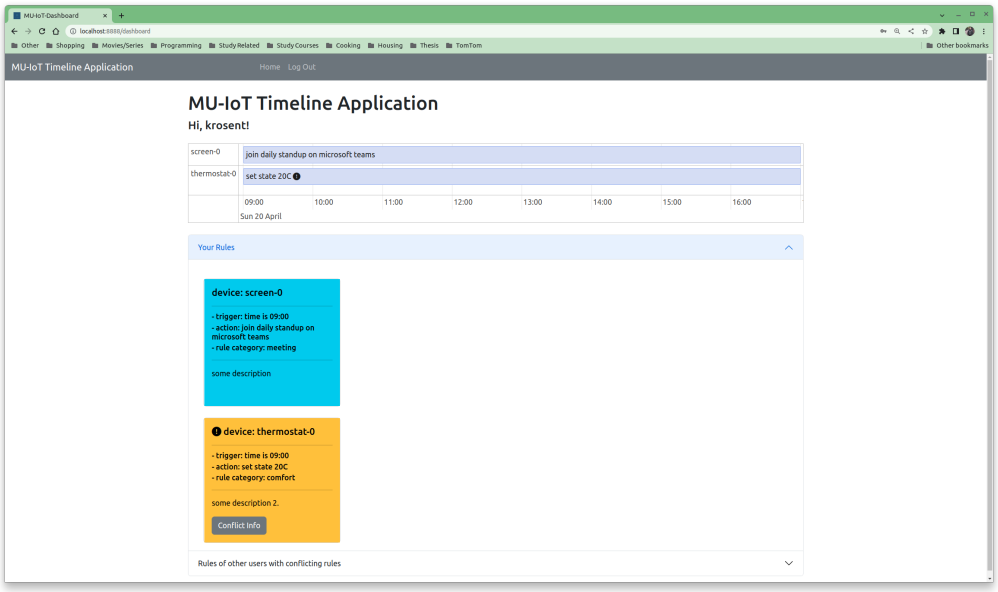


Figure 6.3: MU-IoT Dashboard Main Page showing timeline and user rules

The proposed design guidelines require automation application to have a visualisation view that can depict rules of other users. The information regarding these rules can be limited. Design guidelines require to show conflicting device and rule triggers as rule information. Furthermore the application should have intelligibility mechanisms that can show users which suppressed rules they have and why these rules are suppressed.

Figure 6.3 demonstrates main dashboard page. This page has three main components from which two of them are depicted. The first component is a timeline which depicts users' rules for specific devices and from which time they are running. Rules with exclamation marks are the rules that are suppressed by the system. Users can click on the rule on the timeline to get more information regarding the reason for suppression. The bottom section of the page demonstrates the rules that users have. Each rule has a user-defined description, the device that the rule is using, rule trigger and rule action. Rules that have a 'blue' colour are the rules that are successfully running and have no conflicts. In contrast, rules that have an 'orange' colour are the rules that have conflicts and were suppressed by the system. Users can click the 'Conflict Info' button to get the reasoning behind why the rule is being suppressed.

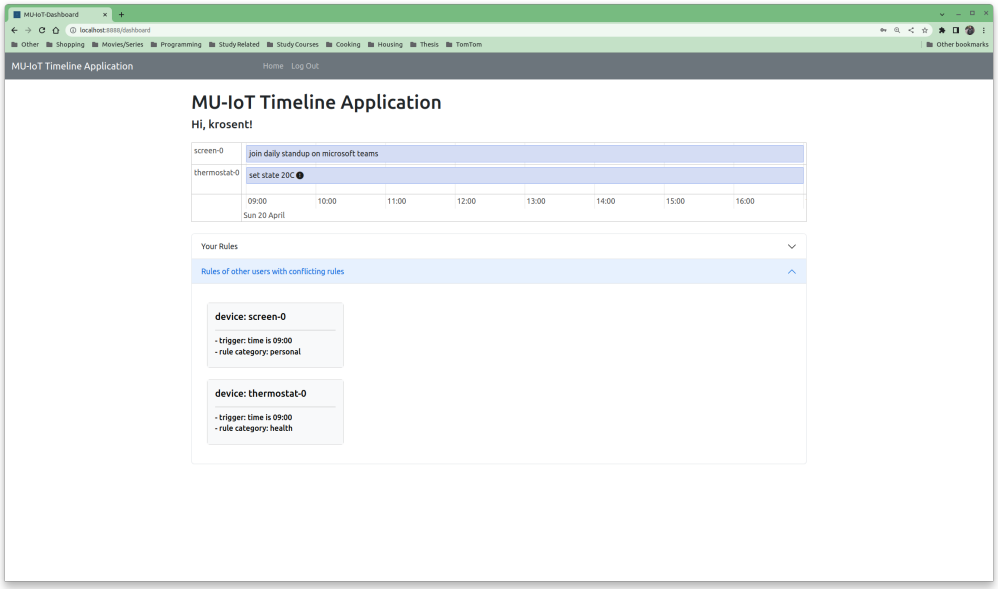


Figure 6.4: MU-IoT Dashboard Main Page showing other users rules that conflict with yours

Figure 6.4 shows another tab that was hidden in the previous Figure which represents the rules of other users that conflict with your rules. In contrast to the first view (which depicts 'your rules'), this view shows limited information regarding the rules of other users. This view shows only the device, rule trigger and rule category. The decision to show only this information is based on proposed design guidelines. This view does not show rules of other users that do not have conflicts with you hence guaranteeing user privacy.

Figure 6.5 demonstrates a popup window that provides information regarding rule suppression. This window shows the reason for suppression.

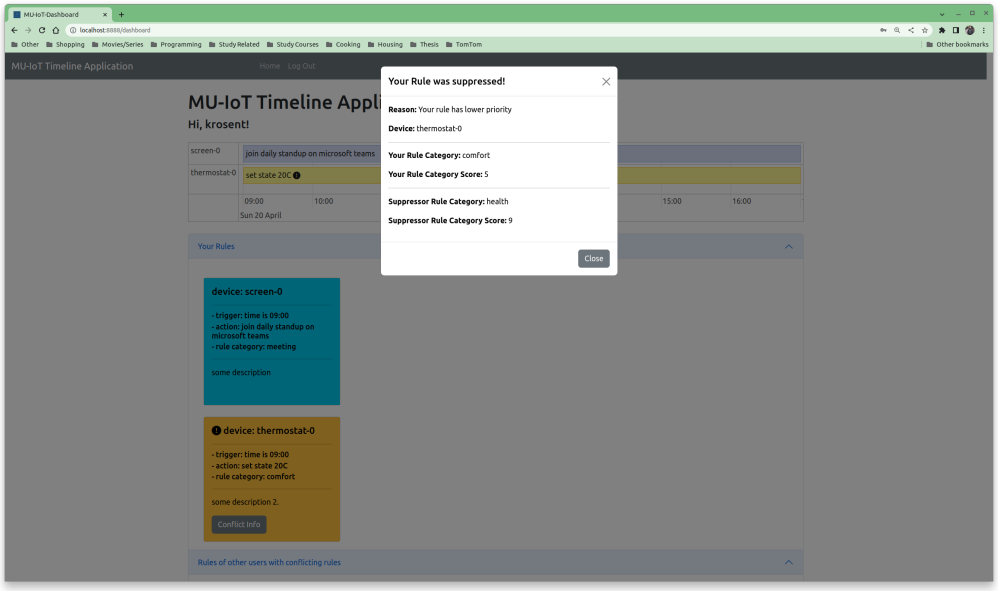


Figure 6.5: MU-IoT Dashboard show suppressed rule information

Additionally, users can see the conflicting device, *suppressed rule category* and its score. To understand why exactly the rule was suppressed the *suppressor rule category* and the score is displayed as well.

Moreover, based on the design guidelines, we added support for admin-based suppression. To suppress a rule administrator need to manually modify *Rule* record in *MongoDB* store. Figure 6.6 demonstrates how the rule suppressed by an administrator is depicted on the timeline. As can be seen, the rule has *person icon* that indicated the type of rule suppression.

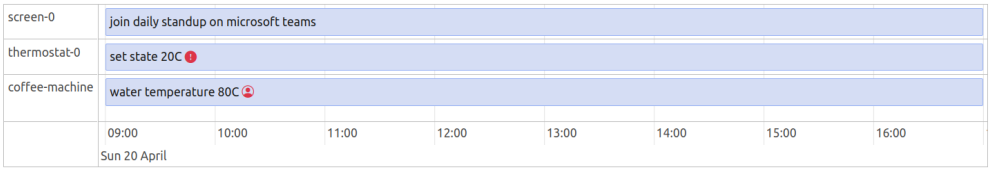


Figure 6.6: MU-IoT Timeline Rule Suppressed by Admin Icon

Figure 6.7 demonstrates how suppressed rule by administrator user is depicted in the list of rules in the dashboard. The visual difference between system-based and admin-based suppression is the use of the different icon. In system-base suppression *exclamation mark* icon is used, while in admin-based suppression *person* icon is used.

6.1.2 Design Space Support

Vermeulen et al. [17] proposed classification of IoT systems with correspondence to intelligibility. In this section we discuss how our proof-of-concept application fits in their design space classification. Out of the 6 factors listed, our proof-of-concept application respects *Generality* because it uses metaphors common to most software application (e.g alert icons, pop-up windows) to provide intelligibility rather than domain-specific metaphors, *Initiative* because it supports showing intelligibility based both on user request or system decision and *Modality* because intelligibility is provided both via text and graphical visualisations. Currently, the *Degree of co-location* for our application is external since it is not embedded into the task automation system itself. *Level of Control* and *Timing* are not supported in our application.

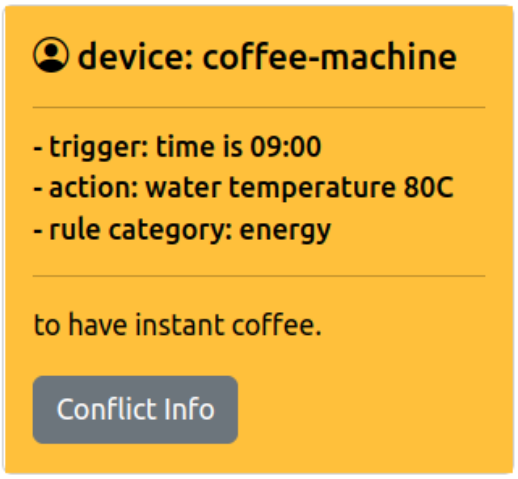


Figure 6.7: MU-IoT Depiction Rule Admin Suppression

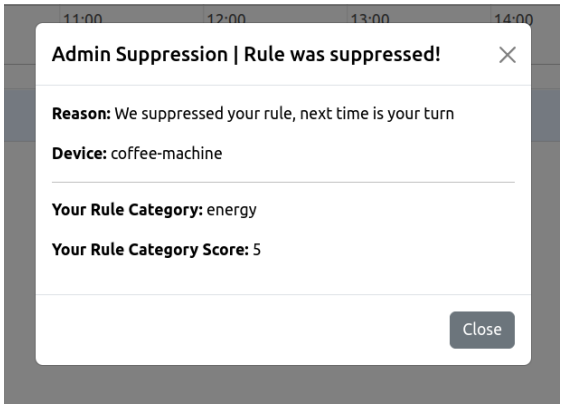


Figure 6.8: MU-IoT Admin Rule Suppression Reasoning

Finally, Figure 6.8 demonstrates a popup window with suppression reason which is suppressed by the administrator user. The window displays users a explicit explanation of rule suppression. A message with the reason, the device in question and the user's rule are mentioned as details that should help users in understanding why their rule was suppressed.

Chapter 7

Conclusion and Future Work

In this chapter, we discuss results obtained during our literature review, conducted a study and implemented a proof-of-concept application. Furthermore, we present some future work that can fix and improve the application.

Prior works have shown the need for proper intelligibility in handling multi-user IoT environments. Based on the findings we were able to implement a proof of concept application that implements the intelligibility needs of users in multi-user IoT environments. First finding identified that users want to be aware of the rules of other users and conflicts with their rules. At the same time, the second finding demonstrates that users want to share limited information about their rules. To address these findings we implemented a visual component that displays to the user the rules of other users and also those rules that conflict with theirs. The information shown to the user is limited and covers only a subset of information that was selected as was chosen by the participants in our user survey. Other findings indicate that users prefer to have system-based and admin-based suppression. The survey also showed that system-based suppression should be based on *rule category*. In our application, we implemented a basic algorithm for both system-based and admin-based suppression. Finally, findings reveal that users want to have feedback mechanisms to see why their rules are suppressed. In our implementation, we have indication icons (exclamation mark icon, person icon) on a timeline that shows users which of their rules are suppressed. Additionally in the rules visualisation section users see the conflicting rules depicted in 'orange' colour. They can click *conflict info* that opens a popup window which provides reason why the rule is suppressed. The MU-IoT application that we developed has proof-of-concept status which means not everything is implemented completely. However, even at the current stage, the POC can be used to analyse users' feedback and then can be further improved. As a part of future

work, we propose to integrate the dashboard into the existing task automation solutions (e. g. HomeAssistant or openHAB) so users can have a single application to meet all their needs. This integration will however also need to be validated with users to understand if it improves their user experience. Such an integrated solution will also will change the modality of the intelligibility mechanism discussed in Section 2.1 from *external* to *embedded*. During research, we conducted a few experiments with the integration of a timeline application from [4] into HomeAssistant but this needs more work. We also believe that the next iteration of the application should use real automation rules from existing task automation systems. An example of such a system can be HomeAssistant automation.
For future work, we suggest conducting a user survey based on the implemented POC and proposed design guidelines. This survey can validate the initial design guidelines and the implemented application. Hence the survey can show whether the design guidelines need to be further refined. Additionally, the survey can show how useful the proposed mechanisms are. Finally, the survey can ask questions regarding visualisation, whether participants like how it is implemented or have different preferences.
The next improvement can be an introduction of an onboarding mechanism. In the current implementation, we have to manually allow certain users to use the system. It is done by hard-coding certain usernames in the application code. This approach is acceptable for a proof-of-concept but can be further improved by the introduction of a proper onboarding mechanism.

Another improvement can be security considerations. In the current implementation, we do not assume security flaws that the system may have. As an example, we store some user authorisation information in browser sessions which can be improved by the use of more secure mechanisms.

We support four out of six of the general dimensions in the proposed Design Space by Vermeulen et. al. [17]. Support of *Level of Control* and *Timing* dimensions should also be considered as part of future work.

Finally the introduction of user control over intelligibility; that is, giving users the ability to control the amount of information they receive from their system, should be considered. In a user survey conducted by Zheng et. al. users have shown interest to have such a feature since some participants were overwhelmed by the number of notifications provided by the system [1]. This feature can enable *level of control* dimension of the Design Space in [17].

Appendix A

Survey Questions

Q102

Do you currently use or have you ever used IoT automation solutions to control smart devices such as smart lighting systems, smart TVs?

- Yes
- No

Q103

For how long have you used IoT or task automation solutions?

- Less than 1 year
- More than 1 year

Q104

How do you evaluate your expertise in IoT automations?

- No experience
- Newbie
- Average
- Experienced
- Professional

Q242

Have you ever used an IoT System in a shared environment such as an office where you are not completely familiar with all other users?

- Yes
- No

Q107 | Environment OverviewIn our user study, we consider the office as a shared environment. Let us a

Consider you and a user called Aleksandra are colleagues at VerySerious Company. You both work from home Monday to Thursday and come to the office on Friday in order to collaborate and have debriefing meetings with other colleagues before the weekend. You are also both avid IoT users and have recently bought and installed new smart devices and configured rules to manage your homes.

Both you and Aleksandra store your rules in **private Solid Pods.** Solid is a storage architecture that enables decentralized storage of data. It enables you and Aleksandra to always maintain ownership of your rules (data) and to give access to applications that wish to make use of those rules, as opposed to applications owning the rules (data). Therefore, you and Aleksandra have granted the IoT systems in your respective homes **READ** access to your IoT rules.

Let us assume that you are able to view your rules and the state of your devices on a timeline application interface just like the ones shown in the pictures below.

You have recently bought a smart thermostat to control the temperature in your home as well as a smart TV.
You recently configured the following rules for use in your home:

- *If the time is 9:00 am, then set the temperature to 15 degree Celsius*
- *If the time is 9:00 am, then show the current cryptocurrency trading prices and my cryptocurrency portfolio on the smart TV screen*

Aleksandra has also recently bought a smart thermostat to control the temperature in her home, as well a smart TV and camera and has configured the following rules for use at her house:

- *If the time is 9:00 am, then set the temperature to 20 degree Celsius*
- *If the time is 9:00 am, then show a video feed of my cats on the smart TV screen*

VerySerious company has informed its employees that it has recently installed a smart thermostat and smart TV in the brainstorming room on every floor and has granted its employees free access to these devices. Employees can give the IoT system in VerySerious company **READ** access to the rules in their SOLID pods and have the rules be executed by the system. **All employees have equal rights in the office and equal access to smart devices but the use of the smart devices currently works on a first-come-first-served basis. The workday starts at 9:00 am and ends at 5:00 pm. The company has implemented the rules:**

- ***If the time is 5:00 pm, then set the thermostat to 17°C***
- ***If the time is 5:00 pm, then set the smart screen to OFF***

As can be seen from the given scenario, both your rules and Aleksandra's rules have conflicting actions at 9 am. Your preferred temperature setting at 9:00 am is 15 degree Celsius, while Aleksandra's is 20 degree Celsius. You also expect to see the current cryptocurrency trading prices and your cryptocurrency portfolio on the screen in the brainstorming room at 9:00 am, while Aleksandra expects to see a video feed of her cats.

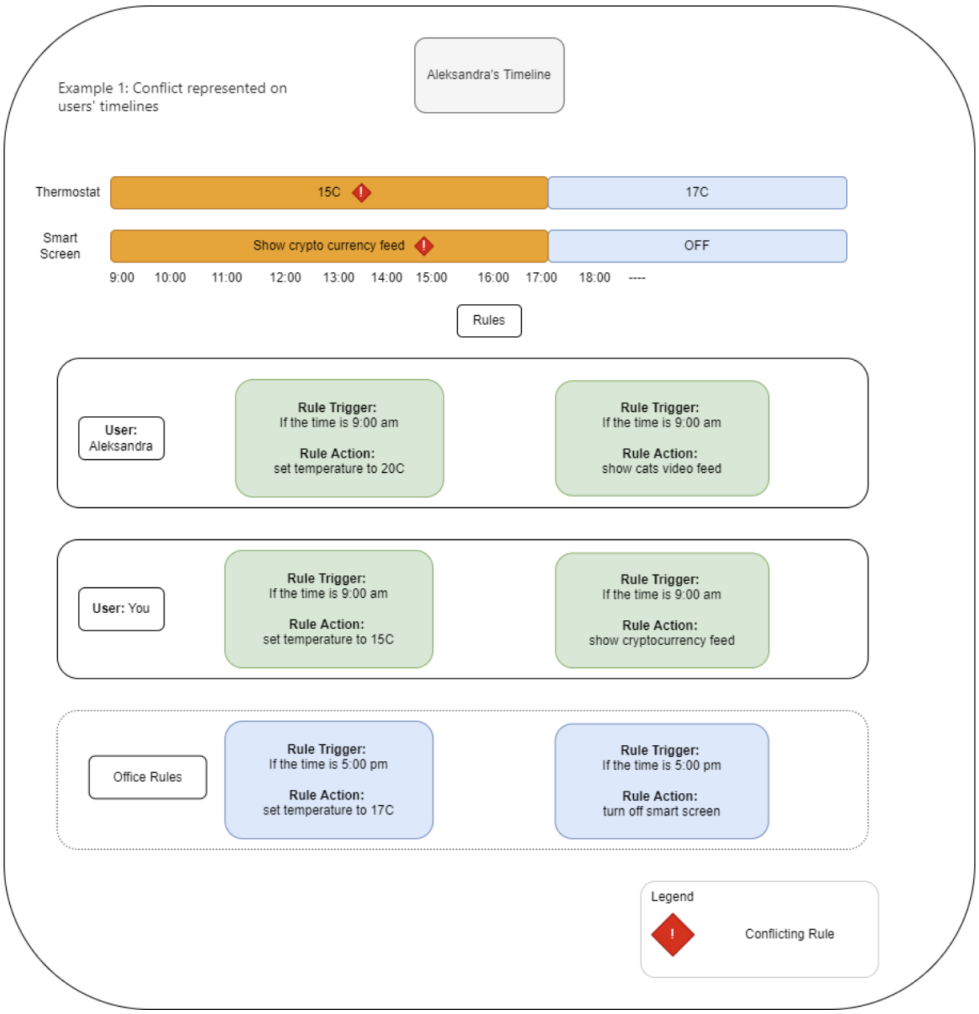
You and Aleksandra come to the office on a busy Friday with other colleagues and both expect your configured home rules to keep working in the office. You are both unaware of each other's rules. You arrive at the office before everyone else, therefore the current temperature is set to 15 degree Celsius and the smart screen is currently displaying the current cryptocurrency trading prices and your cryptocurrency portfolio. Aleksandra does not understand why her rules are not being executed by the system at the office when she arrives and is worried about the welfare of her cats.

Q171

Based on the described scenario

	Not at all important	Low importance	Important	Very important	N/A
How would you rate a feature that allows you and Aleksandra to view each other's rules on your respective timeline apps including possible conflicts?	○	○	○	○	○

Examples of timeline conflict visualisation




Example 2: Conflict represented on users' rules
Aleksandra's Timeline
Thermostat
15C
17C
Smart Screen
Show crypto currency feed
OFF
9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 ----
Rules
User: Aleksandra
Rule Trigger: If the time is 9:00 am
Rule Action: set temperature to 20C
Rule Trigger: If the time is 9:00 am
Rule Action: show cats video feed
User: You
Rule Trigger: If the time is 9:00 am
Rule Action: set temperature to 15C
Rule Trigger: If the time is 9:00 am
Rule Action: show cryptocurrency feed
Office Rules
Rule Trigger: If the time is 5:00 pm
Rule Action: set temperature to 17C
Rule Trigger: If the time is 5:00 pm
Rule Action: turn off smart screen
Legend
Your Conflicting Rule
Conflicting Rule of another user

Q222

Display this question

If Based on the described scenario How would you rate a feature that allows you and Aleksandra to view each other's rules on your respective timeline apps including possible conflicts? - Important Is Selected

Or Based on the described scenario How would you rate a feature that allows you and Aleksandra to view each other's rules on your respective timeline apps including possible conflicts? - Very important Is Selected

How would you like these possible conflicts to be depicted on the timeline application?

- Example 1: Conflict represented on users' timelines
- Example 2: Conflict represented on users' rules
- Neither

Q223

Display this question

If Based on the described scenario How would you rate a feature that allows you and Aleksandra to view each other's rules on your respective timeline apps including possible conflicts? - Important Is Selected

Or Based on the described scenario How would you rate a feature that allows you and Aleksandra to view each other's rules on your respective timeline apps including possible conflicts? - Very important Is Selected

Do you have any suggestions on how to improve the visualisation you prefer or on how best to visualise conflicts on the timeline?

Q201

Considering that both you and Aleksandra have rules which trigger actions that reveal personal information about yourselves

	Not at all important	Low importance	Important	Very important	N/A
How would you rate a feature that allows you to only reveal information which helps Aleksandra to see if her rules will be in conflict with yours?	○	○	○	○	○

Q202

Display this question

If Considering that both you and Aleksandra have rules which trigger actions that reveal personal in... How would you rate a feature that allows you to only reveal information which helps Aleksandra to see if her rules will be in conflict with yours? - Important Is Selected

Or Considering that both you and Aleksandra have rules which trigger actions that reveal personal in... How would you rate a feature that allows you to only reveal information which helps Aleksandra to see if her rules will be in conflict with yours? - Very important Is Selected

Which of your information do you consider is necessary to display on the timeline app to only help Aleksandra to see if her rules will be in conflict with yours?

- ☐ Your Username
- ☐ Only device in conflict (e.g. smart TV screen)
- ☐ Only Rule Trigger (e.g. "If the time is 9:00 am")
- ☐ Complete Rule (e.g. "If the time is 9:00, then set the temperature to 15 degrees")

Q203

Display this question

If Considering that both you and Aleksandra have rules which trigger actions that reveal personal in... How would you rate a feature that allows you to only reveal information which helps Aleksandra to see if her rules will be in conflict with yours? - Important Is Selected

Or Considering that both you and Aleksandra have rules which trigger actions that reveal personal in... How would you rate a feature that allows you to only reveal information which helps Aleksandra to see if her rules will be in conflict with yours? - Very important Is Selected

Do you have any other suggestions of relevant information to display on the timeline app to only help Aleksandra to see if her rules will be in conflict with yours?

Q204

The timeline application has been upgraded with a feature that allows a user to suppress the execution of the rules of another user. Aleksandra really wants to be sure that her cats are doing well so she therefore suppresses your rule for the smart TV screen and forces the system to execute her own rule. She can now thus see her cats on the screen.

Q205

Do you consider this rule suppression feature to be appropriate?

- ○ Yes
- ○ No

Q206

Display this question

If Do you consider this rule suppression feature to be appropriate? No Is Selected

What would you propose instead?

Q207

Display this question

If Do you consider this rule suppression feature to be appropriate? No Is Selected

Since you do not consider the rule suppression system to be appropriate

	Not at all important	Low importance	Important	Very important	N/A
How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours?	○	○	○	○	○

Q208

Display this question

If Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Important Is Selected

Or Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Very important Is Selected

What kind of information do you require from Aleksandra in order to be able to make a decision on whether or not to let her rule take priority over yours? Keep in mind that this choice will also be applicable to your rules.

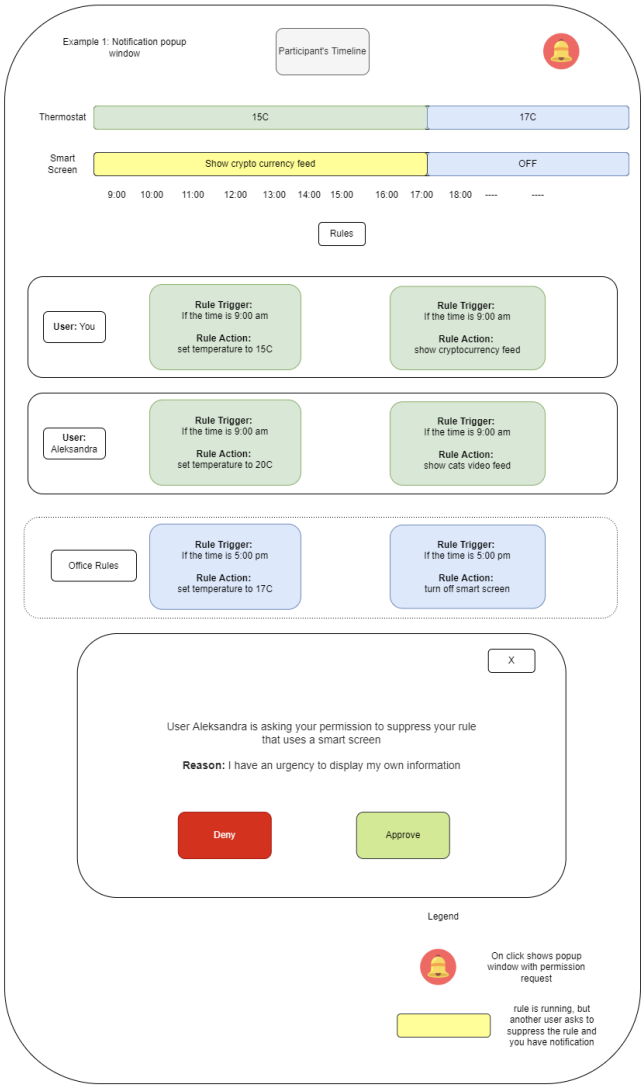
- ☐ The complete rule (e.g. "If the time is 11:00 am, then show a video feed of my cats on the smart TV screen")
- ☐ A rule category (e.g. health, security)
- ☐ User's name
- ☐ A simple message with a reason
- ☐ Time duration of rule execution
- ☐ A user category (e.g. managers)

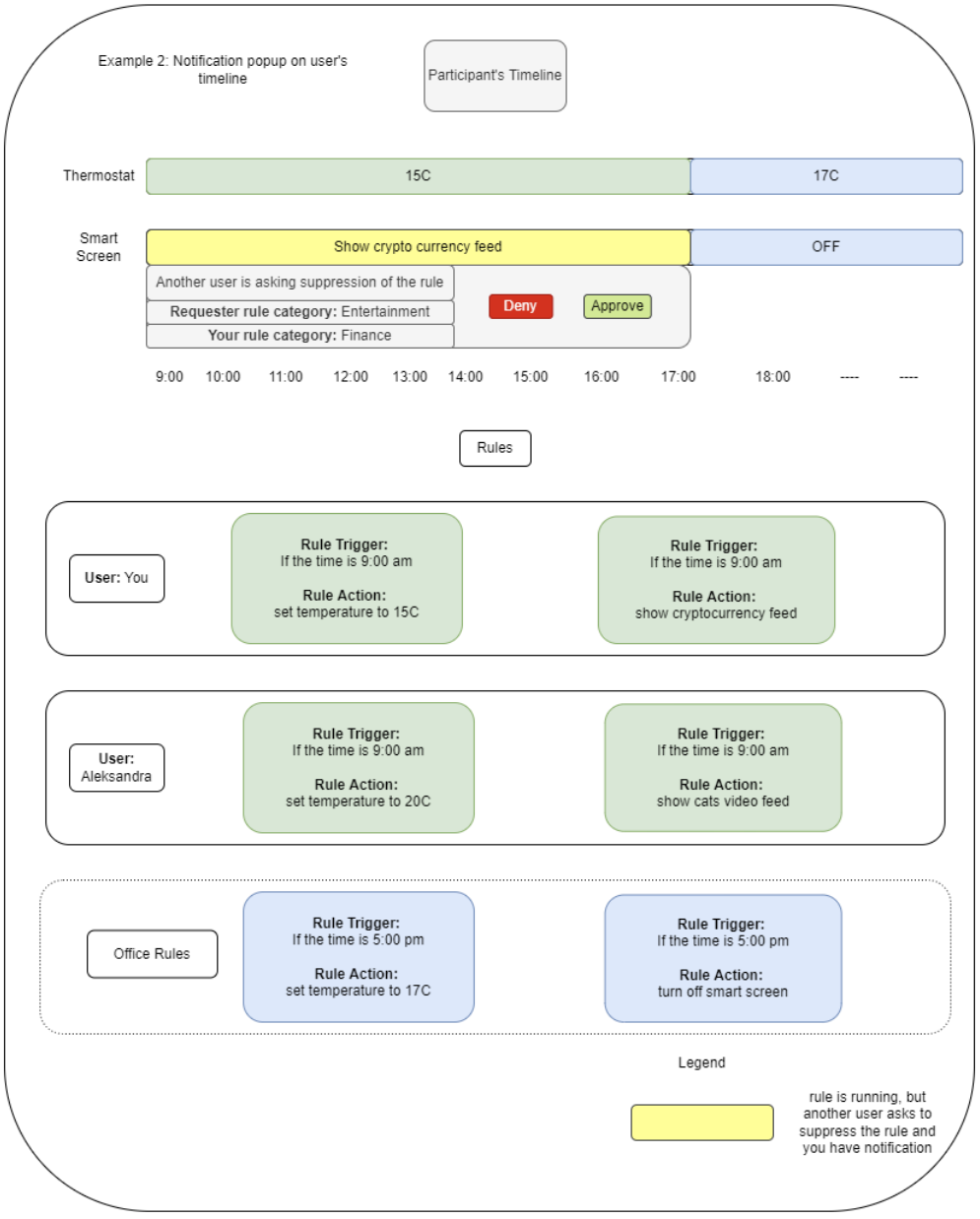
Display this question

If Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Important Is Selected

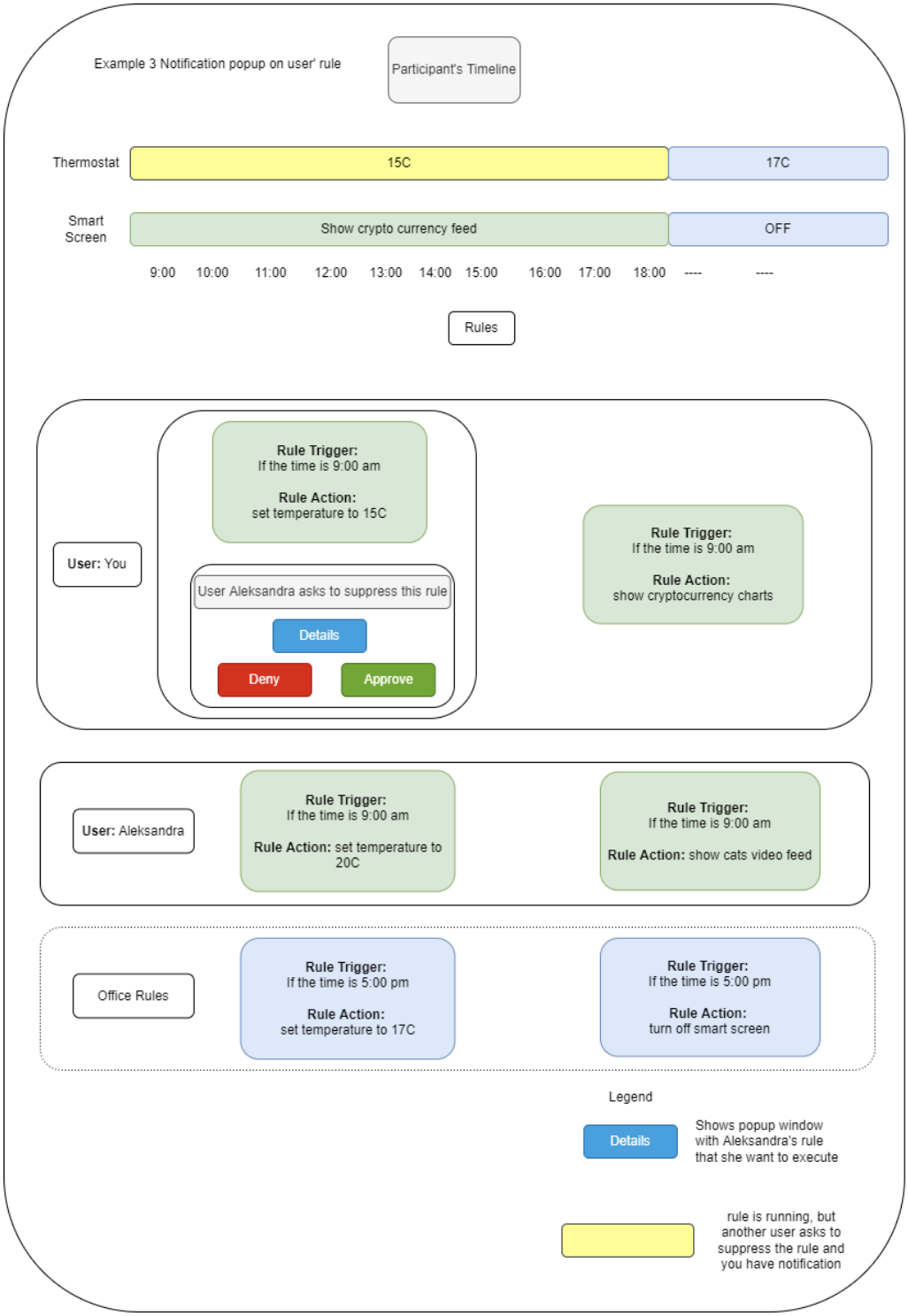
Or Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Very important Is Selected

Examples of a request notification from Alexandra to suppress your rules




Example 2: Notification popup on user's timeline
Participant's Timeline
Thermostat
15C
17C
Smart Screen
Show crypto currency feed
OFF
Another user is asking suppression of the rule
Requester rule category: Entertainment
Your rule category: Finance
Deny
Approve
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00

Rules
User: You
Rule Trigger:
If the time is 9:00 am
Rule Action:
set temperature to 15C
Rule Trigger:
If the time is 9:00 am
Rule Action:
show cryptocurrency feed
User:
Aleksandra
Rule Trigger:
If the time is 9:00 am
Rule Action:
set temperature to 20C
Rule Trigger:
If the time is 9:00 am
Rule Action:
show cats video feed
Office Rules
Rule Trigger:
If the time is 5:00 pm
Rule Action:
set temperature to 17C
Rule Trigger:
If the time is 5:00 pm
Rule Action:
turn off smart screen
Legend
rule is running, but another user asks to suppress the rule and you have notification


Example 3 Notification popup on user' rule
Participant's Timeline
Thermostat
15C
17C
Smart Screen
Show crypto currency feed
OFF
9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 ---- ----
Rules
User: You
Rule Trigger:
If the time is 9:00 am
Rule Action:
set temperature to 15C
User Aleksandra asks to suppress this rule
Details
Deny
Approve
Rule Trigger:
If the time is 9:00 am
Rule Action:
show cryptocurrency charts
User: Aleksandra
Rule Trigger:
If the time is 9:00 am
Rule Action: set temperature to 20C
Rule Trigger:
If the time is 9:00 am
Rule Action: show cats video feed
Office Rules
Rule Trigger:
If the time is 5:00 pm
Rule Action:
set temperature to 17C
Rule Trigger:
If the time is 5:00 pm
Rule Action:
turn off smart screen
Legend
Details
Shows popup window with Aleksandra's rule that she want to execute
rule is running, but another user asks to suppress the rule and you have notification

Q216

Display this question

If Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Important Is Selected

Or Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Very important Is Selected

How would you prefer this information be presented to you on the timeline application?

- Example 1: Notification popup window
- Example 2: Notification popup on user's timeline
- Example 3 Notification popup on user' rule
- Neither

Q225

Display this question

If Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Important Is Selected

Or Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Very important Is Selected

Do you have any suggestions on how to improve the visualisation you prefer or on how best to visualise the suppression request?

Q209

Display this question

If Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Important Is Selected

Or Since you do not consider the rule suppression system to be appropriate How would you rate a feature which allows Aleksandra to ask for your permission to have her rule be executed instead of yours? - Very important Is Selected

Do you have any other suggestions of relevant information you require from Aleksandra in order to be able to make a decision on whether or not to let her rule take priority over yours?

Q228

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

What kind of information do you think this decision should be based on?

- A rule category (e.g. health, security)
- User's proximity to the device
- Time duration of rule execution
- A priority metric (e.g. position in the company)

Q233

Display this question

If What kind of information do you think this decision should be based on? Time duration of rule execution Is Selected

For how long should a user's rule be allowed to make use of a device?

Q211

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

Do you have any other suggestions of relevant information that the system may require in order to be able to make a decision on whether or not to let a rule take priority over another?

Q217

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

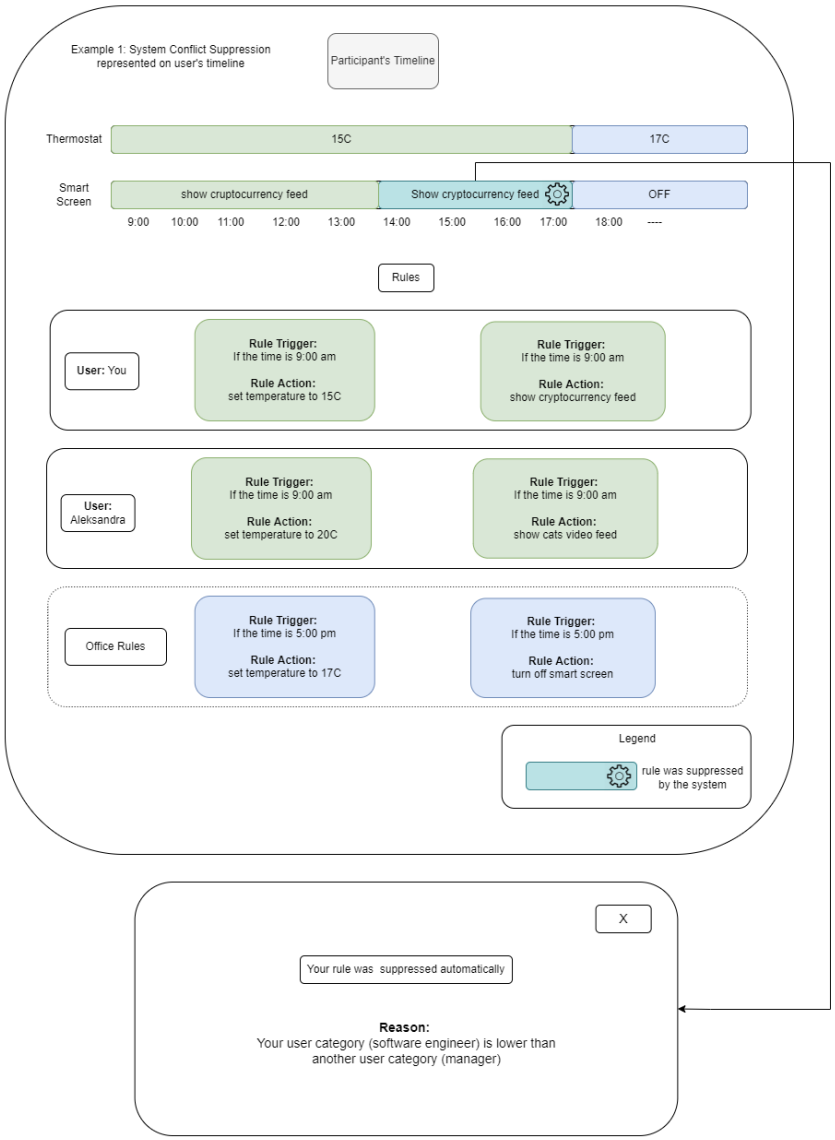
Would you like to receive information from the system about why your rule was not executed if the system determines it should be suppressed?

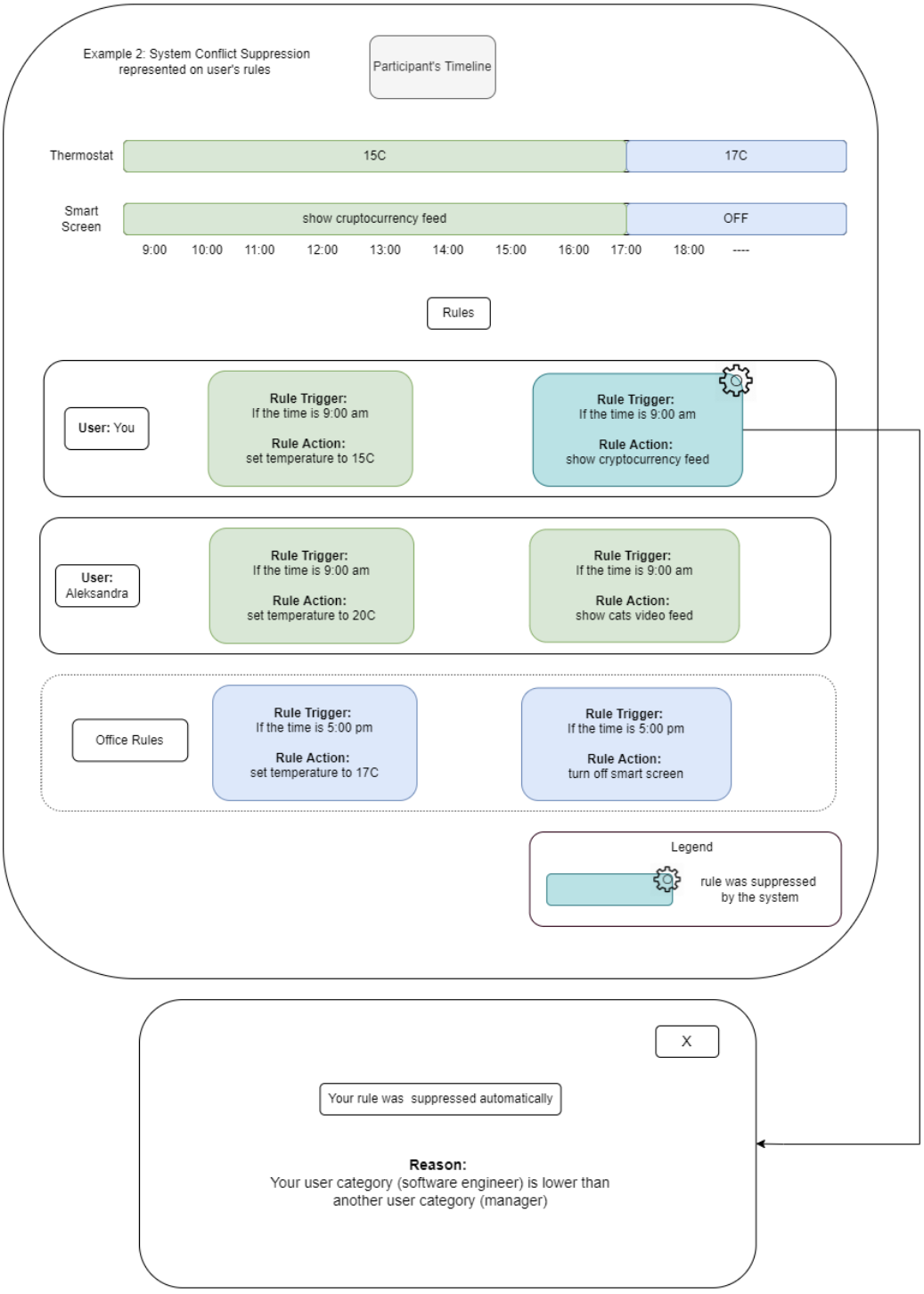
- Yes
- No

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

Examples of Conflict Suppression by the system




Example 2: System Conflict Suppression represented on user's rules
Participant's Timeline
Thermostat
15C
17C
Smart Screen
show cruptocurrency feed
OFF
9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 ----
Rules
User: You
Rule Trigger: If the time is 9:00 am
Rule Action: set temperature to 15C
Rule Trigger: If the time is 9:00 am
Rule Action: show cryptocurrency feed
User: Aleksandra
Rule Trigger: If the time is 9:00 am
Rule Action: set temperature to 20C
Rule Trigger: If the time is 9:00 am
Rule Action: show cats video feed
Office Rules
Rule Trigger: If the time is 5:00 pm
Rule Action: set temperature to 17C
Rule Trigger: If the time is 5:00 pm
Rule Action: turn off smart screen
Legend
rule was suppressed by the system
X
Your rule was suppressed automatically
Reason:
Your user category (software engineer) is lower than another user category (manager)

Q218

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

How would you like the reason for why the system determined your rule should be suppressed to be displayed on the timeline application?

- Example 1: System Conflict Suppression represented on user's timeline
- Example 2: System Conflict Suppression represented on user's rules
- Neither

Q240

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

Do you have any suggestions on how to improve the visualisation you prefer or on how best to visualise conflicts on the timeline?

Q212

Would you prefer that the timeline lets an administrator determine which rules should be suppressed or executed?

- Yes
- No

Q214

Display this question

If Would you prefer that the timeline lets an administrator determine which rules should be suppress... Yes Is Selected

What kind of information do you think this decision should be based on?

- A rule category (e.g. health, security)
- User's proximity to the device
- A simple message with a reason
- Time duration of rule execution
- A priority metric (e.g. position in the company)

Q215

Display this question

If Would you prefer that the timeline lets an administrator determine which rules should be suppress... Yes Is Selected

Do you have any other suggestions of relevant information that the administrator may require in order to be able to make a decision on whether or not to let a rule take priority over another?

Q219

Display this question

If Would you prefer that the timeline lets an administrator determine which rules should be suppress... Yes Is Selected

Would you like to receive information from the administrator about why they suppressed your rule?

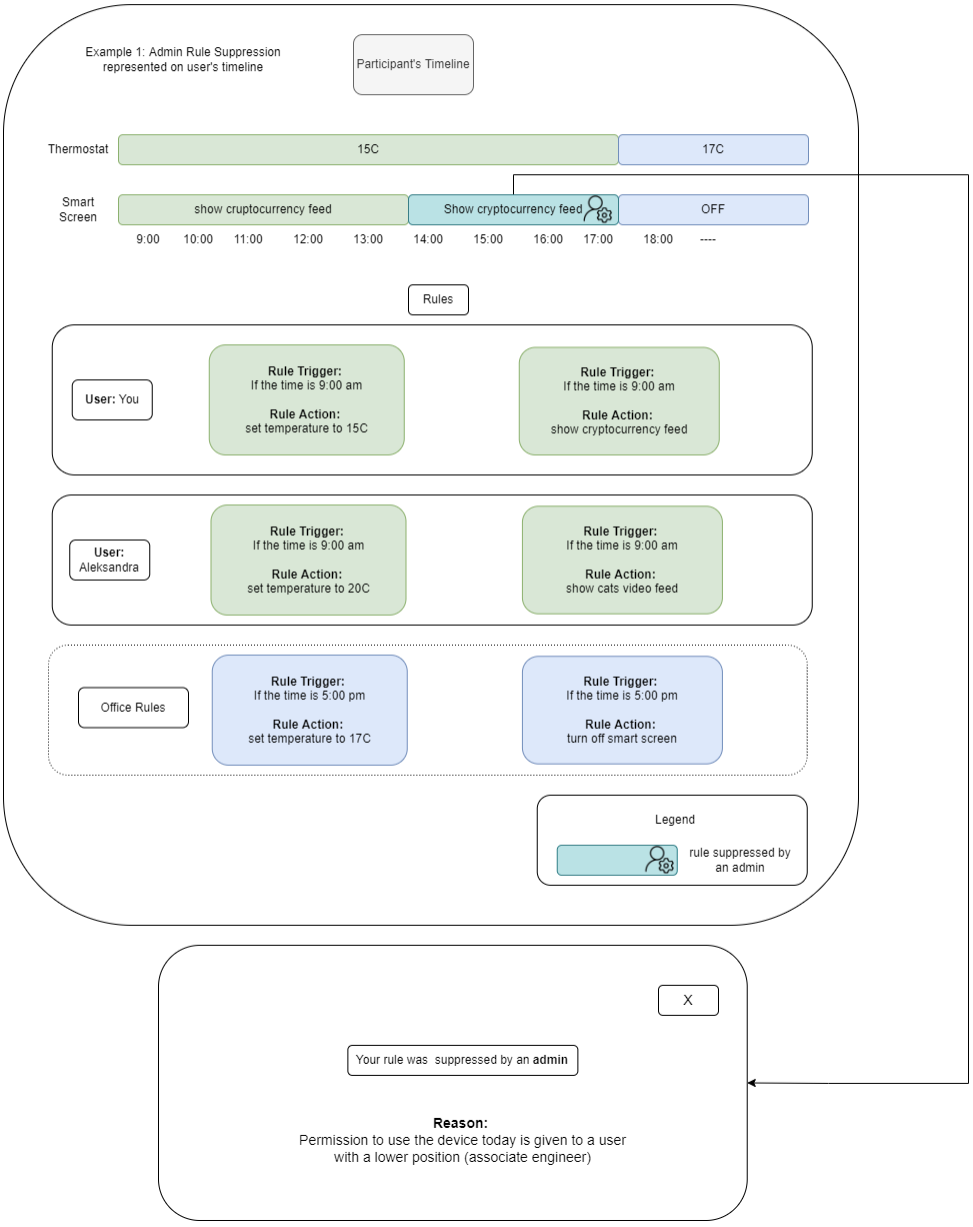
- Yes
- No

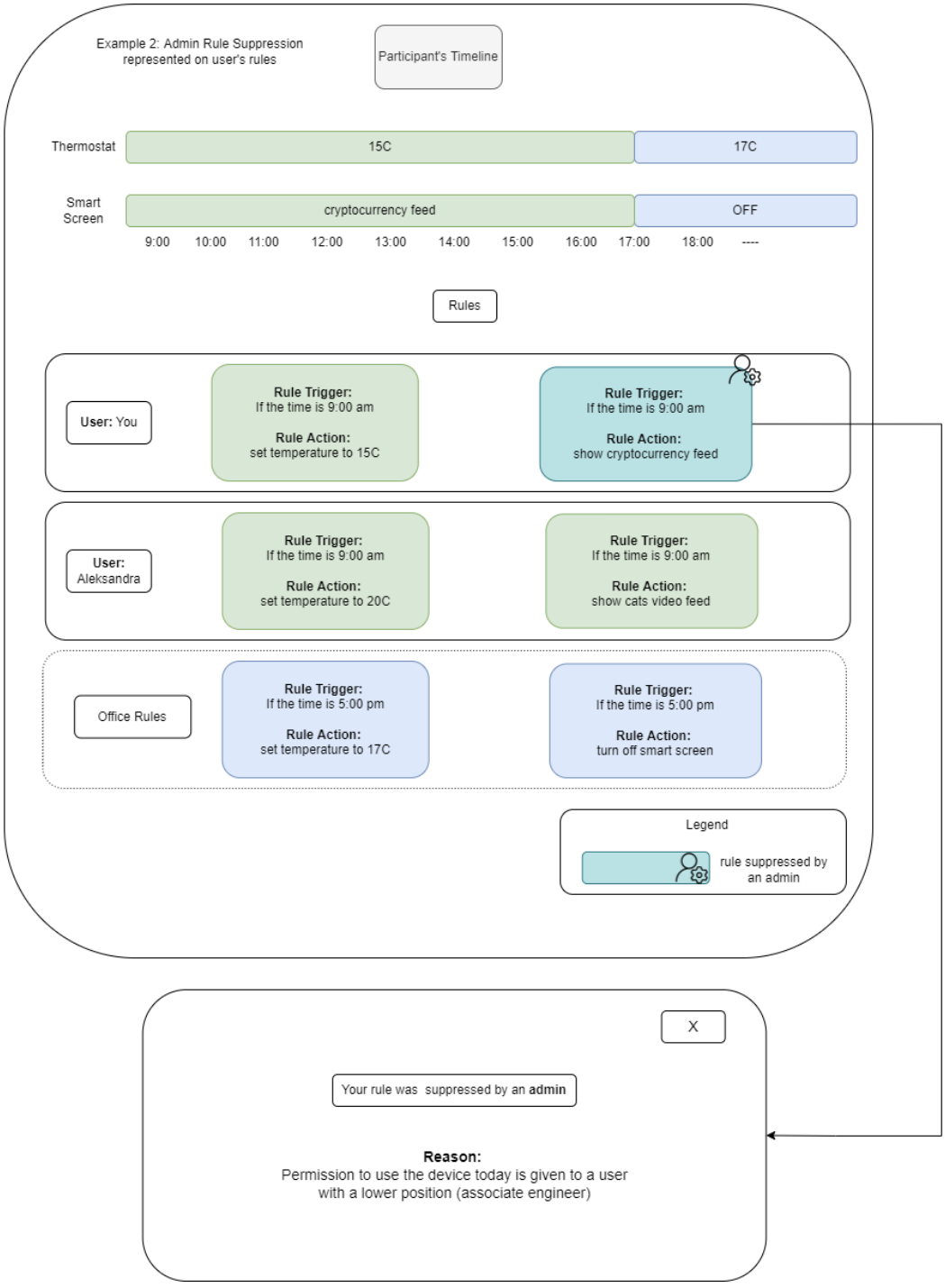
Q227

Display this question

If Would you prefer that the timeline lets an administrator determine which rules should be suppress... Yes Is Selected

Examples of admin conflict suppression visualisations




Example 2: Admin Rule Suppression represented on user's rules
Participant's Timeline
Thermostat
15C
17C
Smart Screen
cryptocurrency feed
OFF
9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 ----
Rules
User: You
Rule Trigger: If the time is 9:00 am
Rule Action: set temperature to 15C
Rule Trigger: If the time is 9:00 am
Rule Action: show cryptocurrency feed
User: Aleksandra
Rule Trigger: If the time is 9:00 am
Rule Action: set temperature to 20C
Rule Trigger: If the time is 9:00 am
Rule Action: show cats video feed
Office Rules
Rule Trigger: If the time is 5:00 pm
Rule Action: set temperature to 17C
Rule Trigger: If the time is 5:00 pm
Rule Action: turn off smart screen
Legend
rule suppressed by an admin
X
Your rule was suppressed by an admin
Reason:
Permission to use the device today is given to a user with a lower position (associate engineer)

Q220

Display this question

If Would you prefer that the timeline lets an administrator determine which rules should be suppress... Yes Is Selected

How would you like the reason for why the administrator determined your rule should be suppressed to be displayed on the timeline application?

- Example 1: Admin Rule Suppression represented on user's timeline
- Example 2: Admin Rule Suppression represented on user's rules
- Neither

Q239

Display this question

If Would you like to receive information from the administrator about why they suppressed your rule? Yes Is Selected

Do you have any suggestions on how to improve the visualisation you prefer or on how best to visualise conflicts on the timeline?

Q229

Display this question

If Would you prefer that the timeline application automatically determines which rules should be sup... Yes Is Selected

Or Would you prefer that the timeline application automatically determines which rules should be sup... No Is Selected

Or Would you prefer that the timeline lets an administrator determine which rules should be suppress... Yes Is Selected

Or Would you prefer that the timeline lets an administrator determine which rules should be suppress... No Is Selected

Please answer on the following question

	Not at all important	Low importance	Important	Very Important	N/A
How would you rate a feature that suggests a compromise rule that is based on two conflicting rules? As an example, it could suggest 18C temperature for both users	○	○	○	○	○

Q128

Do you have any further suggestions for this survey?

Q129

Do you have any suggestions on how to improve the user experience and understandability of IoT systems in environments such as the office we described which were covered in our survey?

Q130

What is your gender?

- Male
- Female
- Other

Q131

What is your age?

- Younger than 20
- 20-39
- 40-59
- 60 or older

Q132

What is the highest level of eductaion or highest degree you have received?

- Less than high school degree
- High school degree or equivalent
- Bachelor's degree
- Master's degree
- PhD degree

Q133

Display this question

If What is the highest level of eductaion or highest degree you have received? Bachelor's degree Is Selected
Or What is the highest level of eductaion or highest degree you have received? Master's degree Is Selected
Or What is the highest level of eductaion or highest degree you have received? PhD degree Is Selected

During your studies, did you get in contact with programming (computer science-related courses)?

- Yes
- No

Q134

Display this question

If During your studies, did you get in contact with programming (computer science-related courses)? Yes Is Selected

How many computer science-related activities did you follow?

- Not much (1-2 courses)
- A few (3-4 courses)
- Many (I am a computer scientist)

Q135 List of Countries

In which country were you born?

Afghanistan

Show Discussion (0)

Q136

Would you like to be contacted about future studies?

- Yes, leave email below
- No

Appendix B

MU-IoT Dashboard Installation Guide

- First of all you need to install a local instance of MongoDB[1]
- After *MongoDB* is installed you need to update *app.js file (:49)* where *mongoose.connect* function is located with your own credentials
- The second step is to have *NPM*[2] on your machine
- In the root folder of the project you need to execute command `npm install`
- You are ready to run our application, you need to execute following NPM command: `npm start`
- Give a system 30 seconds to download first automation rules and after that you can go into dashboard through `localhost:8888`

[1] https://www.mongodb.com/docs/manual/installation/
[2] https://www.npmjs.com/

Bibliography

[1] Eric Zeng and Franziska Roesner. "Understanding and Improving Security and Privacy in Multi-User Smart Homes: A Design Exploration and in-Home User Study". In: *Proceedings of the 28th USENIX Conference on Security Symposium*. SEC'19. Santa Clara, CA, USA: USENIX Association, 2019, pp. 159–176. ISBN: 9781939133069.

[2] Christine Geeng and Franziska Roesner. "Who's In Control? Interactions In Multi-User Smart Homes". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–13. ISBN: 9781450359702. DOI: 10.1145/3290605.3300498. URL: https://doi.org/10.1145/3290605.3300498.

[3] Blase Ur, Jaeyeon Jung, and Stuart Schechter. "Intruders versus Intrusiveness: Teens' and Parents' Perspectives on Home-Entryway Surveillance". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '14. Seattle, Washington: Association for Computing Machinery, 2014, pp. 129–139. ISBN: 9781450329682. DOI: 10.1145/2632048.2632107. URL: https://doi.org/10.1145/2632048.2632107.

[4] Sven Coppers, Davy Vanacken, and Kris Luyten. "FORTNIoT: Intelligible Predictions to Improve User Understanding of Smart Home Behavior". In: *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies* 4 (Dec. 2020). DOI: 10.1145/3432225.

[5] G. S. Thyagaraju et al. "Conflict Resolving Algorithms to Resolve Conflict in Multi-user Context-Aware Environments". In: *2009 IEEE International Advance Computing Conference*. 2009, pp. 202–208. DOI: 10.1109/IADCC.2009.4809007.

[6] Naser Hossein Motlagh et al. "Internet of Things (IoT) and the Energy Sector". In: *Energies* 13.2 (2020). ISSN: 1996-1073. DOI: 10.3390/en13020494. URL: https://www.mdpi.com/1996-1073/13/2/494.

[7] Kinza Shafique et al. “Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios”. In: *IEEE Access* 8 (2020), pp. 23022–23040. DOI: 10.1109/ACCESS.2020.2970118.

[8] Andrea Zanella et al. “Internet of Things for Smart Cities”. In: *Internet of Things Journal, IEEE* 1 (Jan. 2012). DOI: 10.1109/JIOT.2014.2306328.

[9] Omer Sezer, Erdogan Dogdu, and Murat Ozbayoglu. “Context Aware Computing, Learning and Big Data in Internet of Things: A Survey”. In: *IEEE Internet of Things Journal* Volume: 5 (Nov. 2017), pp. 1–27. DOI: 10.1109/JIOT.2017.2773600.

[10] B.N. Schilit and M.M. Theimer. “Disseminating active map information to mobile hosts”. In: *IEEE Network* 8.5 (1994), pp. 22–32. DOI: 10.1109/65.313011.

[11] Hossein Chegini et al. “Process automation in an IoT–fog–cloud ecosystem: A survey and taxonomy”. In: *IoT* 2.1 (2021), pp. 92–118.

[12] Charith Perera et al. “Context Aware Computing for The Internet of Things: A Survey”. In: *IEEE Communications Surveys and Tutorials* (May 2013). DOI: 10.1109/SURV.2013.042313.00197.

[13] Victoria Bellotti and Keith Edwards. “Intelligibility and Accountability: Human Considerations in Context Aware Systems”. In: *Human-Computer Interaction* 16 (Dec. 2001). DOI: 10.1207/S15327051HCI16234_05.

[14] Timo Jakobi et al. “Evolving Needs in IoT Control and Accountability: A Longitudinal Study on Smart Home Intelligibility”. In: *Proceedings in ACM Interactive Mobile Wearable Ubiquitous Technologies* 2.4 (Dec. 2018). DOI: 10.1145/3287049. URL: https://doi.org/10.1145/3287049.

[15] Eric Zeng, Shrirang Mare, and Franziska Roesner. “End User Security and Privacy Concerns with Smart Homes”. In: *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security*. SOUPS ’17. Santa Clara, CA, USA: USENIX Association, 2017, pp. 65–80. ISBN: 9781931971393.

[16] Yasir Arafat Malkani Mahmoud Aljawarneh Lachhman Das Dhomeja. “Resolving User Conflicts in Multi-user Context-aware Home Environment”. In: *PREPRINT (Version 1) available at Research Square* Volume: 5 (Dec. 2021). DOI: https://doi.org/10.21203/rs.3.rs-1045928/v1.

[17] Jo Vermeulen, Kris Luyten, and Karin Coninx. “Intelligibility Required: How to make Us Look Smart Again”. In: (Nov. 2013).

[18] Danilo Caivano et al. “We@Home: A Gamified Application for Collaboratively Managing a Smart Home”. In: June 2017, pp. 79–86. ISBN: 978-3-319-61117-4. DOI: 10.1007/978-3-319-61118-1_11.

[19] Amit Kumar Sikder et al. “Kratos: Multi-User Multi-Device-Aware Access Control System for the Smart Home”. In: *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec ’20. Linz, Austria: Association for Computing Machinery, 2020, pp. 1–12. ISBN: 9781450380065. DOI: 10.1145/3395351.3399358. URL: https://doi.org/10.1145/3395351.3399358.

[20] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The Internet of Things: A Survey”. In: *Comput. Netw.* 54.15 (Oct. 2010), pp. 2787–2805. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010. URL: https://doi.org/10.1016/j.comnet.2010.05.010.

[21] L Barkhuus and Anind Dey. “Is context-aware computing taking control away from the user? Three levels of interactivity examined”. In: Jan. 2003, pp. 149–156. ISBN: 3-540-20301-X.

[22] Fulvio Corno, Luigi De Russis, and Alberto Roffarello. “My IoT Puzzle: Debugging IF-THEN Rules Through the Jigsaw Metaphor”. In: July 2019, pp. 18–33. ISBN: 978-3-030-24780-5. DOI: 10.1007/978-3-030-24781-2_2.

[23] Jose Danado and Fabio Paternò. “Puzzle: A mobile application development environment using a jigsaw metaphor, journal = Journal of Visual Languages and Computing”. In: 25.4 (2014), pp. 297–315. ISSN: 1045-926X. DOI: https://doi.org/10.1016/j.jvlc.2014.03.005. URL: https://www.sciencedirect.com/science/article/pii/S1045926X14000445.

[24] Jo Vermeulen et al. “PervasiveCrystal: Asking and Answering Why and Why Not Questions about Pervasive Computing Applications”. In: Aug. 2010, pp. 271–276. DOI: 10.1109/IE.2010.56.

[25] Jo Vermeulen, Kris Luyten, and Karin Coninx. “Understanding Complex Environments with the Feedforward Torch”. In: Nov. 2012. ISBN: 978-3-642-34897-6. DOI: 10.1007/978-3-642-34898-3_22.

[26] Mirzel Avdic and Jo Vermeulen. *Studying Breakdowns in Interactions with Smart Speakers*. June 2019.

[27] Bauyrzhan Ospan et al. “Context Aware Virtual Assistant with Case-Based Conflict Resolution in Multi-User Smart Home Environment”. In: *2018 International Conference on Computing and Network Communications (CoCoNet)*. 2018, pp. 36–44. DOI: 10.1109/CoCoNet.2018.8476898.

[28] Weijia He et al. “Rethinking Access Control and Authentication for the Home Internet of Things”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 255–272. ISBN: 978-1-939133-04-5. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/he.

[29] Ekene Attoh and Beat Signer. “A Middleware for Implicit Human-Computer Interaction Across IoT Platforms”. In: Sept. 2021. DOI: 10.1145/3460418.3479311.

[30] Andrei Vlad Sambra et al. “Solid : A Platform for Decentralized Social Applications Based on Linked Data”. In: 2016.