Heinrich C. Mayr, Martin Pinzger (Hrsg.): INFORMATIK 2016,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016 275

Consideration of Values in the Design of Access Control Systems

Till Neudecker[1], Arsen Hayrapetyan[2], Alexander Degitz[1], Philipp Andelfinger[1]

Abstract: Since access control systems codify many of the power structures that govern everyday life, the design of such systems has direct ramifications w.r.t. moral values held by the system's designers, users, or operators. As an alternative to a design process based solely on functional or economical requirements, "value-sensitive design" has been proposed as a structured approach to produce systems that are congruent with given sets of moral values. However, the literature has pointed out the lack of methods for handling tradeoffs between values that may limit the practical utility of the approach. In this position paper, we explore the value-sensitive design of an access control system in a data sharing scenario. To this end, we step through the analysis and evaluation of design alternatives from a purely qualitative consideration to a simple formalization that enables discussion and comparison of designs with respect to tradeoffs between values. While any final design decision depends on value judgments by the stakeholders, we believe that by making value judgments explicit, the formalization can substantiate design discussions and lead to more satisfying designs.

Keywords: value-sensitive design, access control, data sharing

1 Introduction

Many of the interpersonal and institutional hierarchies and power structures present in everyday life can be understood in terms of being granted or denied access to certain resources. On a technical level, the specification and enforcement of access policies is performed using access control systems. Due to this central role, access control systems must be designed carefully so that – in addition to satisfying functional requirements – their operation does not violate moral values.

Value-sensitive design [Fr96] is "a theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner throughout the design process" [Fr13]. Value-sensitive design is descriptive as opposed to normative [MH11], i.e., the approach supports the analysis of systems, but does not provide means for resolving conflicts between values or handling tradeoffs when considering multiple system design alternatives [MH11].

In this paper, we explore a systematic approach of designing IT systems that adhere to given sets of moral values. On the example of access control in two variants of a data sharing scenario, we step through a design process from a qualitative level to a consideration of

[1] Karlsruhe Institute of Technology, Steinbuch Centre of Computing and Institute of Telematics, Engesserstraße 2, 76131 Karlsruhe, {till.neudecker, alexander.degitz, philipp andelfinger}@kit.edu
[2] Karlsruhe Institute of Technology, Steinbuch Centre of Computing, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, arsen.hayrapetyan@kit.edu

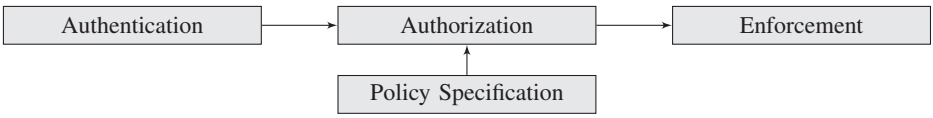

Fig. 1: Access Control Process.

moral values and finally towards a simple formalization of the consideration of values. The aim of the formalization is to make tradeoffs between values explicit and to substantiate discussions of design alternatives among stakeholders.

Our main contributions are as follows: we propose a mapping between the "Objective, Model, Architecture, Mechanism" (OM-AM) framework from the access control literature [Sa00] and the "Values, Norms, Design Requirements" framework from value-sensitive design [VdP13]. The mapping clarifies the role of different "design requirements". Further, we analyze key implications of design alternatives for access control on the example of a data sharing scenario with respect to trust relationships, norms, and values. Finally, we propose a simple formalization for evaluation of system design alternatives with respect to moral values. The formalization is applied to the data sharing scenario to illustrate how the users' prioritization of values lead to different access control system designs.

2 Related Work

2.1 Access Control

Access control is a security element which "is concerned with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system" [VCH06]. An access control system usually consists of an entry point, authentication, authorization and access control policy enforcement services. The entry point provides the user with the means to present their security credentials for authentication. The authentication service is responsible for establishing the identity of the user based on the credentials presented. The authorization service is responsible for verifying if the authenticated user has the necessary rights to access the resource. Finally, the policy enforcement service is responsible for applying the authorization decision, either allowing or denying the access to the protected resource accordingly. The access control process consists of the following major steps (Fig. 1):

– Policy specification: The resource owner specifies with whom to share the resource, usually by assigning access rights (e.g., read, write) to identities (e.g., user names, email addresses). However, access rights can also be assigned to groups or based on attributes.

– Authentication & Attribute assertion: Before the user can access the resource, the user's identity must be established using an authentication mechanism (e.g., password based, using certificates, smart-card). Optionally, the identity is then linked to attributes.

– Authorization: Based on the formalized policy specification and the identity or attribute information a binary (yes/no) authorization decision has to be made that encodes whether the user should be allowed to access a resource.

– Enforcement: Depending on the authorization decision, the access is allowed or not. This is performed by the resource provider as the user attempts an access.

The "Objective, Model, Architecture-Mechanism" (OM-AM) framework was proposed by Sandhu [Sa00] as a tool for engineering authority and trust relationships across organizations and individuals. The framework consists of four layers: the Objective and Model layers specify the security objectives, requirements and tradeoffs, while the Architecture and Mechanism layers describe the means to address those requirements. In [Sa00], OM-AM is applied to articulate the corresponding aspects of the role-based access control models. In [PS01], for example, the framework is applied to the Usage control system.

There are a number of authentication, authorization and access control policy specification frameworks, e.g., Security Assertion Markup Language (SAML) for exchanging authentication and authorization data between security domain; X.509 for the Public Key Infrastructure; OAuth for authorization and secure delegated access, and eXtensible Access Control Markup Language (XACML) for access control policy specification.

2.2 Frameworks for Considering Values in System Design

When considering "values", we focus on "human values", i.e., "what a person or group of people consider important in life" [Fr13]. Here, we disregard frameworks focusing on economical consideration such as "business value" or "stakeholder value" [Co05, Bo06].

Socio-technical system design (STSD) is a set of approaches for a "joint optimization of the social and technical systems" [Mu06]. Baxtor et al. [BS11] provided a survey of STSD methods and noted the lack of methods to guide system synthesis. VENUS [GH14] is a research project building on STSD to design socially aware systems in the field of ubiquitous computing: normative propositions are refined to more specific normative criteria a design can be judged against. Now, technological requirements are formulated that adhere to the normative criteria and finally, design proposals are made.

Value-sensitive design (VSD) [Fr96] is "a theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner throughout the design process" [Fr13]. Due to the significant research activities in the field of VSD in the past years, we base our analysis of access control systems on VSD concepts. In [Fr13], an iterative tripartite methodology is described that comprises *conceptual* (identification of stakeholders, clarification of terms), *empirical* (assessment of the human context by observation and measurement) and *technical* (analysis and design of systems in light of the stakeholders' values) investigations. Our work focuses on conceptual and technical investigations by analyzing the trust relationships and resulting considerations of values in the context of an access control system design. Manders-Huits notes that VSD should be complemented with an ethical theory to enable systematic resolution of tradeoffs between values [MH11]. To structure the consideration of values in VSD, Van de Poel [VdP13] proposes a layered hierarchy of *values*, *norms*, and *design requirements*. Values are "what a person or group of people consider important in life", e.g., "autonomy". Norms are "prescriptions or restrictions on action" [VdP13], e.g., "only administrators may grant access rights". Design requirements are technical properties of the envisioned system. Typically, there are many options in determining elements of lower layers of the

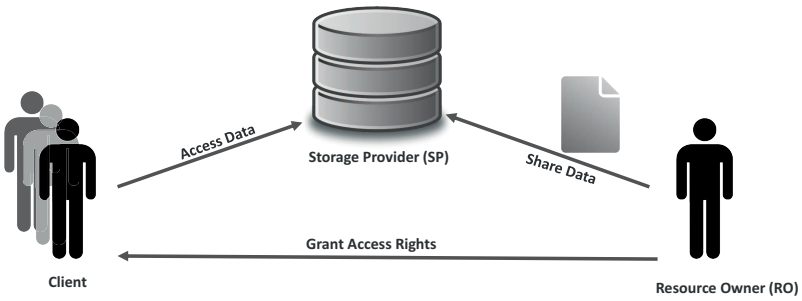

Fig. 2: Sharing Outsourced Data Scenario.

hierarchy that conform with higher layers. However, elements on lower layers can hinder or promote elements on higher layers. In a recent work in VSD, Puylaert has investigated the values of different stakeholders in the design of automated vehicles [Pu16].

We base our analysis on concrete values taken from Schwartz' refinement [Sc12b] of his theory of basic individual values [Sc12a], one result of which is a set of 19 values that can to some degree be considered as interculturally recognized, e.g., "self-direction – action", i.e., "the freedom to determine one's own actions" and "security – personal", i.e., "safety in one's immediate environment". Depending on context, it might be appropriate to determine relevant values according to further sources, e.g., human rights [OB16].

3 Sharing Outsourced Data Scenario and Design Process

We will now first describe a **scenario** of sharing outsourced data that we will consider in the following. This description characterizes the core functionality of the data sharing system and also makes assumptions that limit the possible design space: A user (*Resource Owner, RO*) wants to share files with one or more other known users (*Client*, CL). We assume that the RO stores the files at a dedicated *Storage Provider* (SP), such as Dropbox. Although there are numerous ways of sharing files over the Internet (peer-to-peer networks, email, own web servers, etc.), we acknowledge the fact that using a dedicated storage provider has become the predominant method of personal file sharing and, therefore, exclude listed alternatives from further analysis. Regardless of the concrete realization of this scenario, the access control system performs all tasks described in Section 2.1. It is important that the system design supports the users' values, e.g., if the sharing system only allows sharing of data with the public or not at all, undesired accesses by third parties may occur.

In order to obtain the relationships between values, norms and design requirements in a given scenario, we propose a **design process** that is depicted in Fig. 3 consists of three steps. **Step 1** is the exploration of the technically feasible design space based on the scenario description (see above) and the state of the art. This step is exemplarily executed in Section 5 and, contrary to the subsequent steps, can be regarded as being based on techni-

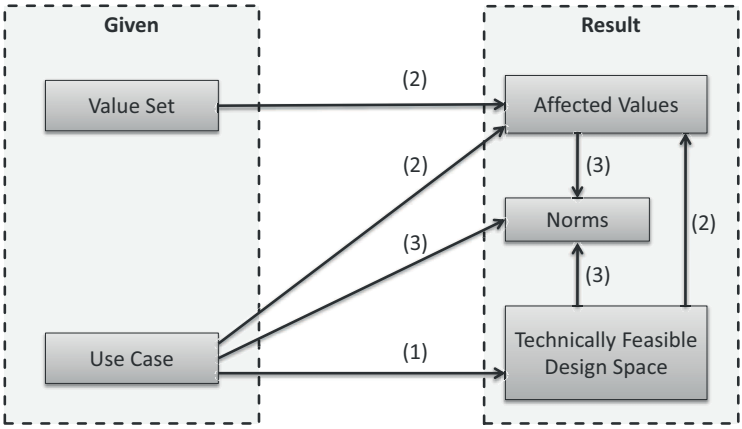

Fig. 3: Proposed Design Process.

cal aspects alone, i.e., independent of subjective judgements. In **step 2**, the set of possibly *affected values* is derived from a given set of values (in our case from [Sc12a]) by an intuition-led process, in which effects of design choices on values are gathered. Above's example of a system design that only allows sharing of data with the public or not at all intuitively affects certain values of a user. Additionally, the affected values depend on the use case, e.g., it makes a difference whether the data to be shared contains weather information or clinical data.

Finally, **step 3** leads to the relevant norms, which link concrete design requirements to abstract values (cf. Fig. 5) by providing an answer to the question *why* a certain value is affected by a specific design requirement and how strong the impact is. This step depends strongly on subjective judgments by the stakeholders and is also dependent on the use case. Although we do not provide a more objective method for this step, we argue that our process makes divergent subjective judgments explicit and hence enables discussion. In the remainder of the paper, we will sketch the described process for our example scenario. In order to proceed with the first step, we need to refine the notion of *design requirements* with respect to access control systems. To this end, we will now propose a mapping that enables us to use fine-grained descriptions of access control systems within the framework of value sensitive design.

4 Value-Sensitive Design and the OM-AM Framework

This section presents a mapping beween the values hierarchy from [VdP13] and the OM-AM framework. The former focuses on non-technical notions of values and norms, while the latter underlines the technical aspects of the design requirements. The combination gives a more detailed and technical picture of the system design based on the values. The OM-AM layers articulate the following aspects of an access control system:

- **Objective:** The security objective of the system expressed as a set of policies to be achieved. An example of a security objective is ”no unauthorized access to the resources” (e.g., access is denied to any entity accessing the files owned by the RO, for which she has not granted the access rights) in the scenario presented in Section 3.
- **Model:** The mathematical formalization of the security objectives. For the above scenario, the ABAC model is an example of a model.
- **Architecture:** The description of the functionality of the system's logical components and their interdependencies. The architecture typically includes components responsible for authentication, authorization and policy enforcement. For the above scenario, the architecture would include authentication and authorization servers responsible for the authentication of the RO and CL, and making authorization decisions for file access requests, as well as a policy enforcement component at the storage provider.
- **Mechanism:** The protocols and software implementations of the functionality of architecture components. For the above scenario, the mechanism could be the OAuth protocol to exchange authentication and authorization information.

We propose the following mapping between OM-AM and the VSD hierarchy (cf. Fig 4):

- **Values**: no equivalent in OM-AM. Since OM-AM is a technical framework whose goal is to articulate the aspects of an access control system with a specific security objective (or set of security objectives), OM-AM has no layer corresponding to the high-level abstract Value notion in the values hierarchy.
- **Norms**: correspond to *Objectives* in OM-AM. Both Norms and Objectives specify the domain-specific goals to be achieved. In the scenario presented in Section 3, the general norm is “no unauthorized access is allowed”
- **Design Requirements**: subsumes Model, Architecture and Mechanism:
  - *Model*: The Model layer specifies formally the *goals* to be strived for to achieve the objective. For the scenario from Section 3, the model is ABAC, and it specifies what it means to allow “no unauthorized access”: to deny access to a user who does not possess the attributes required for an access.
  - *Architecture*: This layer specifies the scope of the norm, defining what is meant by the statement “the authentication and authorization are enforced in a trustful way.” For the data sharing scenario, it describes the components enforcing authentication and authorization as well as the trust relationships between those components.
  - *Mechanism*: The Mechanism layer specifies the actions to achieve the aims formalized by the Model. For the data sharing scenario, the OAuth framework defines the enforcement protocols and the format of authentication and authorization data.

Value Sensitive Design	OM-AM
Values	
Norms	Objectives
Design Requirements	Model
	Architecture
	Mechanism

Fig. 4: The proposed mapping between the VSD hierarchy and the OM-AM framework.

We will see in Section 5 that an analysis of design requirements in terms of model, architecture and mechanism enables a more fine-grained analysis of design decisions.

5 Qualitative Analysis of Design Alternatives

The data outsourcing scenario presented in Section 3 describes the core functionality of the system to be designed. As the access control process (cf. Section 2.1) neither specifies which tasks are performed by whom, nor how interactions between the tasks should be implemented, we will now explore these degrees of freedom in order to establish the set of possible *designs*. Each design has several *characteristics*, such as the used protocol or the system architecture. It must be possible to evaluate each characteristic against *design requirements*, i.e., answer the question whether a given characteristic satisfies a design requirement or not, or whether it is independent of the design requirement. As shown in Section 4, design requirements can be on the abstraction level of models, architectures or mechanisms. Our exploration follows the OM-AM model by walking through the layers and identifying characteristics for each layer.

The access control **model** defines how access policies can be specified. Hence, it also affects the authorization, as the formalization of the policy must be evaluable by the authorization point. It can also impose constraints on the authentication, e.g., the authentication must provide the clients' attributes in order to use an attribute based access control model. Finally, the protocols within the access control process must provide the required expressiveness in order to transfer the policy specification or attributes. Characteristics that affect the model layer include, therefore, the expressiveness of the used protocol and the capabilities of the authentication and authorization point.

The **architecture** of the AC system is the most influential layer, as the architecture specifies each entity's tasks. It is clear that the policy enforcement must be performed by the storage provider and that the policy specification must be performed by the resource owner.[4] Authentication and Authorization, however, can be done by either one, or by another party, such as an *identity provider* (IdP), which has a large effect on the overall system.

Table 1 gives an overview of the characteristics of different architecture designs. Whenever a task is outsourced by the RO to a third party, the RO trusts the third party in some regard.

[4] The resource owner could encrypt the files to be shared and, therefore, cryptographically enforce access control. However, as this would require additional key distribution, we focus only on logical access control here.

Tab. 1: Characteristics of architecture level design alternatives. ✓: fulfilled, X: non-fulfilled

		Authentication			Authorization		
		by IdP	by SP	by RO	by IdP	by SP	by RO
Trust	File access possible by...	IdP, SP	IdP, SP	SP	IdP, SP	IdP, SP	SP
	Access monitored by...	IdP, SP	IdP, SP	SP	IdP, SP	IdP, SP	SP
	Policies known to...	N/A	N/A	N/A	IdP(, SP)	SP	✓
Functionality	Free choice of AC model	N/A	N/A	N/A	X	X	✓
	No IDM Overhead	✓	✓/X	✓/X	N/A	N/A	N/A
Non-functional	Availability	✓	✓	X	✓	✓	X
	Cost	✓	✓	X	✓	✓	X
	Performance	✓	✓	X	✓	✓	X

For example, if authorization is done by a third-party IdP, this IdP could impersonate clients and access outsourced data. Third parties can also learn something about the Client or RO, e.g., by monitoring the access or analyzing the policies. In all these cases, the RO trusts the third party, that it does not perform these malicious actions. Depending on the architecture design, the required trust in third parties varies.

In terms of functionality, the architecture layer overlaps with the model layer, because the choice of the access control model is limited to the access control model offered by the authorization point. Hence, a RO requiring full control of the access control model should not outsource the authorization step. Another functional characteristic is the possibility to rely on an already established identity management system. In most cases of personal data sharing, outsourcing the authentication to a provider with a large user base (e.g., Google or Facebook) decreases the overhead for identity management (IDM). In cases of academic or commercial data sharing, all clients might already have an account at the RO's organization. Non-functional characteristics include performance, cost and availability, which can be satisfied to a higher degree by specialized providers. The **mechanisms** used in an AC system also primarily impact non-functional characteristics. For instance, authentication mechanisms could apply symmetric or asymmetric cryptography, resulting in different processing delays due to different complexity of calculations.

The performed qualitative assessment of the core aspects of the system design is the first step (cf. Sec. 3) of our design process. It also demonstrates the challenge of a comprehensive assessment. Hence, we described the design in terms of the "Values, Norms, Design Requirements" hierarchy (cf. Fig. 5). In the figure, we focus on the design decision whether the Authentication and Authorization (A&A) process is outsourced to a third party. We identify three norms that are clearly affected by the design decision: First, **minimizing the monetary cost** for the resource owner: we assume that it is less expensive to outsource A&A than it is to host the necessary infrastructure oneself. Second, **specification of arbitrary access control policies**: in case A&A is outsourced, the resource owner may need to confine herself to the types of access control policies allowed by a third party. Third, **monitoring of enforcement**: in case the RO hosts the A&A infrastructure herself, she is able to monitor all steps of the A&A process, which may be infeasible in case of outsourcing.

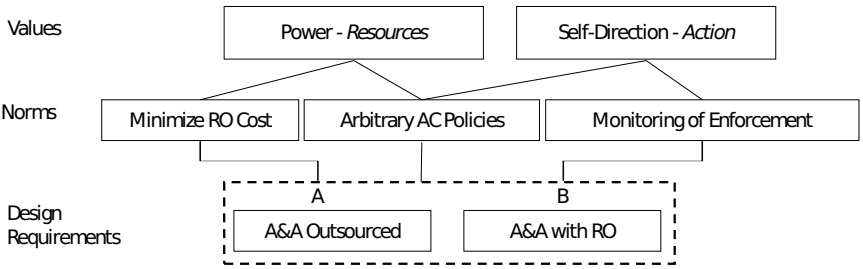

Fig. 5: Values, norms and design requirements in the private data sharing scenario.

We consider the following values (from [Sc12b]): First, **Power - Resources**, i.e., "Power through control of material and social resources". Here, Power - Resources is affected by the monetary cost of a design. Second, **Self-Direction - Action**, i.e., "Freedom to determine one's own actions". Here, Self-Direction - Action is affected by the possibility to constrain or react to the behavior of users accessing the shared data. The consideration of values and norms gives an overview of the effects of different design decisions, but does not allow for design decisions in case multiple different designs are possible. To support design decisions, in the following, we move towards a quantitative assessment.

6 An Attempt at Formalization

Van de Poel studied conditions for aggregating value scores to determine an overall most desirable design [VdP15]. Such a procedure depends on the ability to compare value scores across design alternatives on a common scale ("value commensurability"). Further, Van de Poel analyzed concrete procedures to evaluate designs: in *cost-benefit analysis*, considerations of values are expressed in terms of monetary units. Van de Poel notes a number of issues with cost-benefit analysis, e.g, the non-linearity in the utility of money, which is shared by the approach of *direct trade-offs*, where designs are scored on an individual scale for each value. To compare alternatives, there must be a way to compare scores across values. Like cost-benefit analysis, direct trade-offs assumes that "a loss in one value can always be compensated by a gain in another value". Counter-examples to this have been called *taboo trade-offs* [Te03]. In the *maximin* approach, the design with the highest score is selected according to each design's lowest-scoring value. Using *satisficing*, a design is evaluated only w.r.t. whether a certain minimum score with respect to each relevant value is exceeded, which does not suggest a "best" design. Van de Poel proposes an overall design process in which satisficing is performed first, and which, if morally acceptable designs can be identified, proceeds to selecting the "best" design using cost-benefit analysis, direct tradeoffs, or maximin. In the following, we assume that satisficing has been applied and morally inacceptable designs discarded. The comparison is based on direct trade-offs.

We formalize the analysis of a set of design alternatives as follows: a *design analysis* is characterized by a tuple (R,N,V,I_R,I_N,W_V), where R is a set of design requirements, N is a set of norms, and V is a set of values. $I_R : R \times N \to \mathbb{R}$ reflects the impact of design requirements on norms, if fulfilled. $I_N : N \times V \to \mathbb{R}$ reflects the impact of norms on values,

if fulfilled. For a design requirement r and a norm n, $I_R(r,n) > 0$ iff r adheres to or supports the norm n. $I_R(r,n) < 0$ iff r hinders or violates n. $I_R(r,n) = 0$, if r has no bearing on n. $I_N(n,v)$ is defined analogously for each pair of a norm n and a value v. $W_V : V \to \mathbb{R}_+$ reflects the importance of each value in the context of the considered system.

To evaluate a (partial) design w.r.t. the considered values, we first select a conflict-free set of design requirements $D \subseteq R$. To determine the value score of a design, it may be sufficient to consider those design requirements with large impact on the value score. We assume that in a real-world setting, stakeholders will iteratively extend the *design analysis* with values, norms and design requirements they consider to be relevant. Now we can evaluate the design's adherence to the considered values by calculating the design's score:

$$S(D) := \sum_{d \in D,\ n \in N,\ v \in V} I_R(d;n) I_N(n;v) W_V(v).$$

By comparing the scores of all feasible designs, the design that best fulfills the considered values can be determined. Meaningful comparisons require that the weights on each level of the "Values, Norms, Design Requirements" hierarchy are chosen so that they represent the relative impacts consistently. Certainly, some situations may require a more expressive language to comfortably express intricacies of the consideration of values:

- Adherence to one design requirement r_1 may be sufficient to fully adhere to a norm n, so that additional adherence to a design requirement r_2 has no further effect. A solution would be to specify maximum scores that each norm can attain.
- Values may be more or less important w.r.t. different aspects of a design. In such cases, values should be specified further to differentiate the relevant aspects.
- Our formalization scores values and norms linearly, e.g., diminishing returns w.r.t. a certain value or norm cannot be specified. As solution would be to allow for arbitrary functions that determine the score w.r.t. a given value based on its linear score.

Although the proposed extensions may results in an analysis process that more accurately reflects the underlying considerations of values, the goal of making explicit the reasoning that guides design decisions still suggests settling for the simplest suitable formalization.

7 Making Design Decisions

Now, using our formalization that enables scoring of different designs according to the stakeholders' value judgments, we determine value scores with respect to three assumed users. In addition to the values *Power - Resources* and *Self-Direction - Action* introduced in Section 5, we consider the value *Face*, i.e., "Security and power through maintaining one's public image and avoiding humiliation" [Sc12b]. In our scenario, we consider *Face* to be affected by the possibility to monitor who accesses the shared pieces of data, and by the availability of the resources to the users. Figure 6 shows our consideration of values,

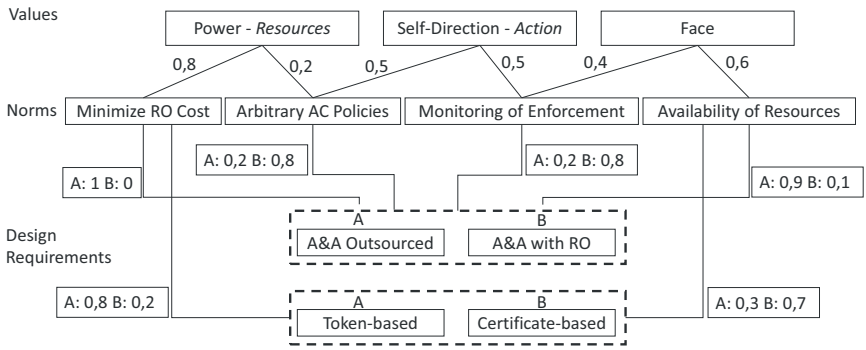

Fig. 6: Scoring the congruence of a design with values considered in the private data sharing scenario.

norms and design requirements. It represents an example for a value-sensitive discussion about design requirements. The considered norms and impacts can be derived in step 3 of the design process (cf. Section 3) and were exemplarily chosen here. Since the value ”Power-Resources” has more influence on the norm ”Minimize RO Cost” than on the norm ”Arbitrary AC Policies”, the edge weights differ significantly (0.8 vs. 0.2). The design requirements in question are the following:

A. Whether to outsource the Authentication and Authorization (A&A) process to a third party: a discussion of the implications of this aspect was given in Section 5.

B. Whether to use a token-based or a certificate-based A&A mechanism: in a token-based A&A approach like OAuth 2.0, the IdP and the authorization server do not manage client connections, but only issue tokens, which the clients send to the next server according to the used protocol. Token-based mechanisms tend to be less resource-intensive, since the need to store connection information is avoided. For this reason, the edge weight between ”Minimize RO Cost” and ”Token-based” (A) is 0.8, compared to the weight for ”Certificate-based” (B) 0.2. In contrast, Certificate-based A&A approaches not only need to maintain a public key infrastructure, but usually also need to manage client connections, which are redirected from the IdP to the corresponding authorization server. However, certificate-based approaches tend to be less error-prone, since connections are managed by the server at all times. Therefore the edge weight from ”Availability of Resources” to ”Certificate-based” (B) is 0.7, whereas the weight for ”Token-based” is only 0.3.

Figure 5 depicts a quantitative analysis of the impacts of the discussed design requirements according to the proposed formalization. Edges are assigned weights representing how well a value is represented by a norm or how well a norm is represented by a design requirement, enabling quantitative discussions about the degree to which a design requirement fulfills a set of values. As an example, we consider three assumed users:

User 1 is only interested in the monetary cost of sharing her data and thus only considers the value *Power- resources* (importance 1.0). The score in Figure 6 suggests that user 1 should outsource A&A to a third party provider and that she should use token-based A&A.

Tab. 2: Three users with differing scores with respect to their values for the data sharing scenario.

	Score (1)	Score (2)	Design Decision
User 1: Money (1;0;0)	A: 0.84 B: 0.16	A: 0.64 B: 0.16	A - A
User 2: Availability (0.1;0.1;0.8)	A: 0.6 B: 0.4	A: 0.21 B: 0.35	A - B
User 3: Self-direction (0.2;0.7;0.1)	A: 0.37 B: 0.63	A: 0.15 B: 0.07	B - A

User 2 is mostly interested in availability, so he sets the highest importance to the value *Face* (importance 0.8). He expects to "lose face" and possibly business partners if his shared data is not available. User 2 also considers the cost and his ability to set up and monitor customized rules (both importance 0.1). With these weights set, User 2 should also outsource A&A, but use a certificate-based A&A approach, which are usually more reliable.

User 3 is highly privacy-aware. She wants to customize and monitor the enforcement of access rights herself through the value of *Self-direction* (importance 0.7). She is also interested in a cost-efficient solution (importance 0.2) while still considering the availability of her data to a small amount (importance 0.1). With these weights set, User 3 should not choose to outsource the A&A process. Like User 1, she should choose a Token-based approach, because she valued the cost of the solution more than the availability of her data.

Table 2 summarizes the value scores and selected designs for the three assumed users.

8 Summary and Discussion

In this paper, we considered the value-sensitive design of access control systems. As a first step, we performed a purely qualitative and informal analysis of an assumed access control scenario. In the analysis, extending the "Values, Norms, Design Requirements" hierarchy from value-sensitive design with the Object-Model-Architecture-Mechanism framework from the access control literature enabled a more fine-grained consideration of design requirements. Based on existing proposed design procedures, we introduced a simple formalization that enables a numerical scoring of the congruence of system designs with the stakeholders' values. The formalization considers the effects of design decisions on the considered values, making the value judgments in the design process more explicit. We calculated scores reflecting the congruence with the values held by three assumed types of users of the envisioned access control system with distinct prioritizations of values. The per-user scoring demonstrated that the design of even a relatively simple access control system can be affected strongly by value judgments of the stakeholders.

The presented analysis highlights some of the key challenges on the path towards a value-sensitive analysis and design of real-world systems:

– While technically feasible system designs can be derived from the requirements defined by a given use case based on technical properties of the envisioned system, determining the largely subjective impacts of designs on norms and values poses larger difficulties. Similarly to approaches for determining security metrics w.r.t. IT environments, quantification and comparison is a challenging issue.

– In the value-sensitive design process sketched in Section 3, a major challenge lies in determining norms on a suitable level of abstraction, based on previously identified design requirements and values. An abstraction and generalization of the design requirements, e.g., formally using domain-specific languages, and a specification of the relevant values in the given context, e.g., in the form of concrete laws or policies, could aid in bridging the remaining gap between technical and ethical considerations. A similar problem is given in policy-based management, where generic policies must be translated to concrete actions.

– In real-world IT environments, the consideration of values may not necessarily be a priority in the design process. Instead, administrators and users are frequently confronted with existing systems that reflect the implicit value judgments made at the time the system was designed. Further, in a previous design process, the compatibility with existing IT environments may be the predominant decision factor. In such cases, we argue that a systematic evaluation of the existing system in light of the values held by the stakeholders' may enable a clearer analysis and critique of the system and could suggest steps to align the system more closely with the stakeholders' values.

– A possible focus of future work could be the question of scoring designs with respect to *privacy*-related values. For certain types of values, a prioritization can be performed in terms of well-defined numerical weights, e.g., using probabilities or established currencies. For other values, the selection of suitable weights seems less clear. Particularly in the case of access control systems, values related to privacy can come into focus. It may not be possible to enumerate or evaluate the potential consequences of unintended data flows, although in many scenarios such data flows are clearly undesirable. This is related to the general problem of justification of privacy concerns [So07]. Possibly, numerical weights associated with privacy-related values may therefore only be understood as expressing subjective prioritizations in relation to other values.

– We think that the benefits of more explicit consideration of values in system design can only be proven in a real-world setting. User studies such as [Pu16] might give pointers to improve the process of determining value-congruent system designs systematically.

– Finally, some of the steps in resolving conflicts and tradeoffs between values in the system design process seem to have natural counterparts in mathematical logic and optimization: *cost-benefit analysis* and *direct trade-offs* can be considered as mathematical optimization problems. Further, it seems natural to formalize problems of *satisficing* as satisfiability problems from mathematical logic. Hence, similarly to approaches in IT security [Kö15], given a formal description of the impacts of different design requirements on the relevant values for a desired system functionality, it may be possible to generate or parametrize designs in an automated fashion to accommodate varying scenarios or users.

References

[Bo06] Boehm, Barry W: Value-Based Software Engineering: Overview and Agenda. In: Value-based software engineering, pp. 3–14. Springer, 2006.

[BS11] Baxter, Gordon; Sommerville, Ian: Socio-Technical Systems: From Design Methods to Systems Engineering. Interacting with computers, 23(1):4–17, 2011.

[Co05] Cockton, Gilbert: A Development Framework for Value-Centred Design. In: CHI'05 extended abstracts on Human factors in computing systems. ACM, pp. 1292–1295, 2005.

[Fr96] Friedman, Batya: Value-Sensitive Design. Interactions, 3(6):16–23, December 1996.

[Fr13] Friedman, Batya; Kahn, Peter H.; Borning, Alan; Huldtgren, Alina: Early Engagement and New Technologies: Opening up the Laboratory. In (Doorn, Neelke; Schuurbiers, Daan; van de Poel, Ibo; Gorman, E. Michael, eds): The Handbook of Information and Computer Ethics, chapter Value Sensitive Design and Information Systems, pp. 55–95. Springer Netherlands, Dordrech, 2013.

[GH14] Geihs, Kurt; Hoffmann, Holger: Socio-Technical Design of Ubiquitous Computing Systems. chapter A Research Agenda for the Socio-Technical Design of Ubiquitous Computing Systems, pp. 3–18. Springer International Publishing, Cham, 2014.

[Kö15] Köhler, Jens: Tunable Security for Deployable Data Outsourcing. KIT Scientific Publishing, 2015. Dissertation, Department of Computer Science.

[MH11] Manders-Huits, Noëmi: What Values in Design? The Challenge of Incorporating Moral Values into Design. Science and engineering ethics, 17(2):271–287, 2011.

[Mu06] Mumford, Enid: The Story of Socio-Technical Design: Reflections on its Successes, Failures and Potential. Information Systems Journal, 16(4):317–342, 2006.

[OB16] Orwat, Carsten; Bless, Roland: Values and Networks – Steps Toward Exploring the Relationships. Computer Communication Review, 2016. Editorial Note.

[PS01] Park, Jaehong; Sandhu, Ravi: Towards an Engineering Framework for Usage Control and Digital Rights Management. 2001.

[Pu16] Puylaert, S.A.A.: Social Desirability and Mobility Impacts of Early Forms of Automated Vehicles. Master's thesis, Delft University of Technology, 2016.

[Sa00] Sandhu, Ravi: Engineering Authority and Trust in Cyberspace: The OM-AM and RBAC Way. In: Proceedings of the fifth ACM workshop on Role-based access control. ACM, pp. 111–119, 2000.

[Sc12a] Schwartz, Shalom H: An Overview of the Schwartz Theory of Basic Values. Online Readings in Psychology and Culture, 2(1):11, 2012.

[Sc12b] Schwartz, Shalom H; Cieciuch, Jan; Vecchione, Michele; Davidov, Eldad; Fischer, Ronald; Beierlein, Constanze; Ramos, Alice; Verkasalo, Markku; Lönnqvist, Jan-Erik; Demirutku, Kursad et al.: Refining the Theory of Basic Individual Values. Journal of personality and social psychology, 103(4):663, 2012.

[So07] Solove, Daniel J: 'I've Got Nothing to Hide' and Other Misunderstandings of Privacy. San Diego law review, 44:745, 2007.

[Te03] Tetlock, Philip E: Thinking the Unthinkable: Sacred Values and Taboo Cognitions. Trends in cognitive sciences, 7(7):320–324, 2003.

[VCH06] Vincent C. Hu, David F. Ferraiolo, D. Rick Kuhn: Assessment of Access Control Systems. 2006.

[VdP13] Van de Poel, Ibo: Translating Values into Design Requirements. In: Philosophy and engineering: Reflections on practice, principles and process, pp. 253–266. Springer, 2013.

[VdP15] Van de Poel, Ibo: Conflicting Values in Design for Values. Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains, pp. 89–116, 2015.