# Reconciling Privacy and Data Sharing in a Smart and Connected Surrounding

Paul Tran-Van, Nicolas Anciaux, Philippe Pucheral

# Reconciling Privacy and Data Sharing in a Smart and Connected Surrounding

Paul Tran-Van[1,2,3]
[1] Cozy Cloud, France
paul@cozycloud.cc

Nicolas Anciaux[2,3]
[2] Inria, France
nicolas.anciaux@inria.fr

Philippe Pucheral [2,3]
[3] U. Versailles St-Q., France
philippe.pucheral@uvsq.fr

## 1 INTRODUCTION

When Alice goes trekking in the French Alps with friends, she is equipped with a pedometer to measure her efforts, takes pictures using her smartphone and uses a mobile coach app to monitor her trip and GPS trail. But how could Alice have a transversal view of the personal data she generates and how could she share -part of- her data with her friends?

The Personal Cloud paradigm emerges [1] (e.g., Cozy Cloud, ownCloud, Databox to cite a few) and holds the promise of a Privacy-by-Design storage and computing platform where each individual could gather her complete digital environment in one place and share it under control. Conjointly, smart disclosure initiatives pushed by legislators (e.g., EU General Data Protection Regulation) and industry-led consortiums (e.g., Blue Button for medical records in the US, Midata in the UK, MesInfos in France) give shape to this paradigm by letting individuals getting their personal data back from the applications that collected them. Hence, Alice could link her personal devices to the personal cloud platform of her choice and then manage the personal data she generates when trekking and regulate data sharing at will.

However, the personal cloud paradigm causes a gravity shift of data management and data security from organizations to individuals, who are usually not database administrators nor security experts. Unfortunately, the main existing access control models (e.g., RBAC, ABAC or TBAC [2]) are geared towards central authorities and require a deep expertise to define users, roles and privileges. Some decentralized models have been proposed to let individuals manually define their own sharing preferences, often based on Web of Trust-like approaches [3] or on the owner's social graph [4], but offer limited expressive power and poorly cope with the versatile nature of the Personal Cloud. To tackle this issue, several works aim to ease the sharing administration. For example, [5, 6] give the possibility for the owner to share any kind of personal data through the use of attribute-based sharing rules, while [7, 8] explore machine learning techniques to automatically infer the best sharing policies. However, they provide little means for individuals to control the actual effects of their policies and could actually result in unexpected data leakage. This contradicts a founding principle of the Personal Cloud paradigm, namely enabling individuals making sovereign decisions about the sharing of their data [1]. The problem is exacerbated in a ubiquitous and smart surrounding producing continuous flow of daily activity events.

We derive from these statements a new sharing paradigm dedicated to the personal cloud context, called SWYSWYK (*Share What You See with Who You Know*). SWYSWYK relies on two founding principles: (1) provide intuitive means to derive sharing rules directly from the personal cloud content and help the personal cloud owner administer the resulting sharing policy by visualizing and sanitizing its net effects and (2) provide a secure personal cloud architecture giving tangible guarantees that the sharing policy will be properly enforced, whatever the security expertise of the owner.

In [10], we investigated point (2) and proposed a secure architecture combining an untrusted, an isolated and secure execution environments. [11] presented a practical instantiation of this architecture where the reference monitor runs into a secure hardware device. In [9], we focus on point (1) and introduce the semantics of the SWYSWYK sharing paradigm and discuss the specificities of its administration.

This demonstration focuses on point (1) with the goal to assess the practical interest of the SWYSWYK paradigm. To this end, we have integrated SWYSWYK in a real personal cloud platform, namely Cozy, and apply it to a smart surrounding scenario inspired by Alice's one. A video of the demonstration is available online.

In this paper, Section 2 presents the SWYSWYK baseline, Section 3 gives the scenario and Section 4 concludes.

## 2 SWYSWYK PARADIGM

### 2.1 Baseline

SWYSWYK is not yet-another access control model. It is rather a new sharing paradigm, helping the derivation of expressive access control rules directly from the Personal Cloud content and providing convenient tools to administrate the resulting policies. The originality of SWYSWYK relies on two core principles helping circumventing the aforementioned difficulties of data sharing in the Personal Cloud context:

*Documents are rules*. The personal cloud content on its own conveys intuitive sharing rules, e.g., share pictures and related events of a trek with people who appear on these pictures and as such are identified as participants in that trek. Such rules should be straightforward to express, as the related permissions could be derived from the documents' content. The subjects targeted by the document, called *identifiees* [12], should be extracted from the document content and enter in the rule definition. We call *reflexive sharing rules* the rules based on this principle.

*Subjects and objects are documents.* The content of a personal cloud also intrinsically describes the individual's acquaintances under different forms (e.g., contact files, identity pictures.) and

conversely, acquaintances are associated with pieces of information in the owner's space (e.g., agenda entries, photos on which a friend appears). A corollary is that for each permission granted to a subject *s* on an object *o*, viewable documents should represent *s* and *o*. More generally, the result of a sharing policy (sets of sharing rules) must be viewable by the personal cloud owner, who can thus precisely understand what is the net effect of this policy. For example, a stream of GPS tracks may be represented as trajectories on a map and time series of activities logs could be represented in graphs.

The combination these principles gives substance to the *Share What You See with Who You Know* (SWYSWYK) paradigm.

## 2.2 Sharing Paradigm Semantics

SWYSWYK aims at providing simple expressions for sharing rules and make the sharing policy self-administrated when the personal cloud content evolves. We show here how this can be captured within simple semantics, combined with a set of simplifying assumptions. First, our sharing paradigm relies on a *closed policy*, i.e. every action not explicitly granted is denied. Actions are *CRUD* operations on documents in the personal cloud. The paradigm supports only authorizations (positive rules) but allows the owner to post-filter the produced Access Control List (ACL) when exceptions need to be declared. Consequently, there is a direct translation between sharing rules and sets of ACLs: an action *a* is granted to subject *s* on document *d* iff $(s, d, a) \in$ ACL and is denied otherwise. The sharing is by construction *consistent* (the decision is unique), *complete* (the decision always exists) and can be evaluated in logarithmic time.

For the sake of conciseness, we do not formally define here all the notations and operators of SWYSWYK. Rather, we illustrate the paradigm through a single type of sharing rules, namely the *reflexive sharing rules*, and refer to [9] for a complete description.

**Reflexive sharing rules.** These rules express the sharing of documents with subjects appearing on it. They implement the *documents are rules* principle and are thus considered as first-class citizen rules:

$$\text{ACL} \leftarrow \{(s,d,a) \in S \times D \times A \,/\, Filter(d,Q) \wedge MatchS(DI(d), SI(s))\ \}$$

*Filter* and *DI* are user-defined, platform dependent, functions. *Filter* returns *true* if a document *d* of the Personal Cloud satisfies the qualification *Q*, that can be expressed on the metadata or on the content of *d*. *DI* extracts identification traits of individuals, denoted next by *IT*, from *d*. *IT* must uniquely represent a subject in the Personal Cloud and can combine simple attributes (e.g., email, phone number) or complex representation (e.g., facial features, fingerprint). *SI* and *MatchS* are internal SWYSWYK operators. *SI* returns the *IT* from a registered subject *s* and *MatchS* returns true if the compared *IT*s are equivalent. Below are various illustrations of reflexive sharing rules.

*Example 1*. Share the pictures taken during my trekking sessions with the people appearing on it:

*Q: docType='photo' ∧ tagGallery='trek'*

*DI: face detection algorithm*

*MatchS*: here, compares the facial features extracted from DI with the ones returned by SI from known subjects.

*Example 2*. Share the minutes of meetings with the attendees:

*Q: docName like 'minutes-%.doc'*

*DI: extract attendee names from a minute document*

These two examples, extracted from two different application contexts, show the generality of the sharing paradigm.

## 2.3 Sharing Administration

To make the sharing paradigm practical, subject declaration and maintenance should be (quasi) automatic while respecting the owner's privacy. In SWYSWYK, the notion of *rule consistency* concretizes the fact that the effects of all rules can be visualized (and then easily controlled by the owner), the notion of *exceptions* permits customization of these effects according to the owner's preferences, and *subject administration* can be automatically performed such that the set of subjects grows along document insertions and rule declarations with minimal interactions.

**Rules Consistency.** A SWYSWYK sharing rule is said well-formed iff it produces only ACLs involving viewable documents shared with recognizable subjects: $\forall sr \in SR,\ \forall acl \in ACL,\ acl.d \in DV \wedge acl.s \in DS$, where *SR* is the set of sharing rules, *DV* the set of viewable documents and *DS* the subset of viewable documents characterizing a unique subject. Any *acl* which does not satisfy this condition is filtered out.

**Rules Exceptions.** Instead of introducing interdiction rules to capture exceptions, which makes the net effects of the resulting policy complex to apprehend, we simply give the owner the ability to filter out the permissions which hurt her privacy (considered as *suspicious ACLs*). We introduce three types of *watchdog triggers* to highlight suspicious permissions:

$$What(Q_S, A) \rightarrow \{(s,\{(d,a)\}) \,/\, (s,d,a) \in ACL^* \wedge s \in Q_S(S) \wedge a=A\}$$
$$Who(Q_D, A) \rightarrow \{(d,\{(s,a)\}) \,/\, (s,d,a) \in ACL^* \wedge d \in Q_D(D) \wedge a=A\}$$
$$Which(Q_S, Q_D, A) \rightarrow \{(s,d,a) \,/\, (s,d,a) \in ACL^* \wedge s \in Q_S(S) \wedge d \in Q_D(D) \wedge a=A\}$$

$ACL^*$ corresponds to the set of newly created/updated ACLs. *What* identifies, for each sensitive subject, the new set of (document, action) she is granted to (e.g., *which new documents can be seen by my boss?*). *Who* identifies, for each sensitive document, the new set of subjects s with granted action a on them (e.g., *which new subjects have a read access to my medical records?*). Finally, *Which* identifies new ACLs combining a selection of (sensitive) subjects and documents (e.g., *which new authorizations my colleagues have on my family photos?*).

**Subjects Administration.** New subjects can automatically be created while inserting new contact files or address book entries. The *IsS* SWYSWYK operator is invoked each time (1) documents are created or updated in the personal cloud and (2) a new rule invoking *IsS* is defined, thus enriching the set of subject S along document insertions and rule declarations as side-effects of the function. Each $s \in S$ is made of the extracted identification traits and at least one generated credential for the authentication.

## 2.4 Sharing Enforcement

**General principle**. The creation, maintenance and evaluation of a set of SWYSWYK permissions are as follows: (1) the owner

creates sharing rules and watchdog triggers to be applied on her personal cloud; (2) a *rule translator* translates the selected rules into candidate ($ACL^{*}$) and suspicious ($ACL^{?}$) ACLs and materialize them; (3) the owner checks the suspicious ACLs at will and accepts ($ACL^{+}$) or rejects ($ACL^{-}$) them using the *administration GUI*; (4) the *reference monitor* authenticates subjects and evaluates *Allowed* (i.e., *Allowed (s,d,a) = true iff (s, d, a)* $\in ACL^{+}$) and delivers the requested documents accordingly.

**ACL production and maintenance.** Five operators are required to translate any sharing rule into ACLs, namely *Filter*, *DI*, *SI*, *IsS* and *MatchS*. The data flow between the operators to translate a reflexive sharing rule into ACLs is shown in Fig. 1. At declaration time, the rule tree is evaluated over all documents of the personal cloud. First, *Filter* operators are evaluated at the leaf of each branch to select the targeted subjects documents (right branch) and the targeted objects documents containing *identifiees* (left branch). Then *DI* operators extract the list of *ITs* from the targeted subject documents (left branch) and from the objects documents (*ITs* of the *identifees*). In the right branch, *IsS* tries to match the extracted *ITs* with the subjects already registered in *S*, then *SI* appends the identified *ITs* to each subject. Finally, *MatchS* joins the left and right branches on subject *ITs* and produces the (candidate) ACLs. At insertion of a new document *d* in the personal cloud, the *Filters* of all rules are evaluated against *d* to check whether new candidate ACLs can be produced.

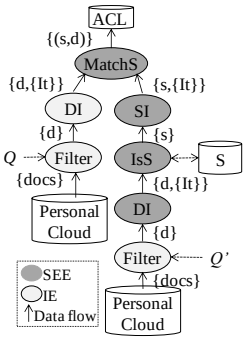


**Fig.1: ACL production**

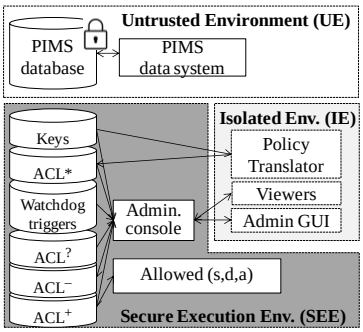


**Fig.2: Reference architecture**

**Secure enforcement of sharing policies.** The enforcement issue is exacerbated when the Personal Cloud platform runs on the owner's side, the security of which can be questioned. In [10], we proposed a reference architecture tackling this issue, displayed in Fig. 2. It consists of three environments: (i) an *untrusted environment (UE)* on which no security assumption is made for the code nor for the data, (ii) an *isolated environment (IE)* on which general purpose code can be run with the guarantee that it cannot leak any information but with no guarantee about the soundness and honesty of its output and (iii) a *Secure Execution Environment (SEE)* which runs only certified core programs and protects data and code against snooping and tampering. In Fig. 1, the light grey operators (*DI* and *Filter*) are run in the *IE*, as they are made of untrusted third parties code, while the ones in dark grey are executed in the *SEE*. Administrative tools helping the owner to control and sanitize the ACLs are partly hosted in SEE (e.g., watchdog triggers) and in IE (e.g., document viewers).

# 3 DEMONSTRATION

The objective of this demonstration is to show that the SWYSWYK paradigm makes sense in concrete environments. Hence, it considers a ubiquitous surrounding scenario with connected smart devices producing continuous data streams. These streams are stored in the Cozy personal cloud platform. SWYSWYK has been integrated in the Cozy stack, a simplified version being part of the next release (sharing in Coz y3).

## 3.1 Demonstration platform

The platform consists of an Android smartphone, a Withings smartwatch and a local Cozy instance running on a laptop with Ubuntu 16.04. Additionally, several Cozy instances running on a remote server are used to simulate other subjects' personal clouds. Pictures and GPS tracks are synced with the local Cozy instance thanks to the Android Cozy app. Pedometer data from the smartwatch is retrieved through the Withings' API. The Cozy stack running on the laptop is implemented in Go and stores documents in a CouchDB database. The Cozy apps are developed with the JavaScript React framework.

## 3.2 Demonstration scenario

The demonstration concentrates on the usage of the SWYSWYK paradigm. The scenario is composed of four steps, as summarized in Fig. 3 and described in a video accessible online:

*Step 1 - Data collection:* this step illustrates how surrounding data produced by smart devices can be collected by the Cozy platform to be further exploited. A Cozy instance is populated with a set of predefined documents and timely integrates data produced by Paul's smartphone and pedometer. A *Sharowalky* application developed on Cozy (for illustration purpose) manages Paul's trekking data, namely his photos, GPS trails and physical activity. The attendees are invited to connect to Cozy as Paul (the personal cloud owner and incidentally co-author of this paper), open the *Sharowalky* app and browse days of trekking.

*Step 2 - Sharing definition:* the *Sharowalky* app proposes the attendee to share the photos of a trekking day with Paul's friends appearing on them (among which Riad). The GUI presents the semantics of the underlying sharing rule, that is a typical SWYSWYK reflexive rule represented as logic-based predicates on Cozy metadata. The GUI allows the attendee to identify that Riad has been granted access to certain pictures of the circle -confirmed when connecting to Riad's personal cloud-. The attendee can also share the GPS and activity trails of the circle with Riad very easily.

*Step 3 - Sharing administration:* the access control console allows the attendee (playing Paul's role) to visualize and control the net effect of the current access control policy (set of all existing sharing rules). All resulting permissions are shown as viewable ACLs, i.e., triples <*subject*, *object*, *permission*> where each *subject* and *object* are personal cloud documents which can in turn be visualized. The GUI brings to light a suspicious permission that the attendee is invited to remove (or confirm according to her will).

*Step 4 - Dynamicity:* the demo operator finally selects from the Cozy the set of pictures taken during the conference, showing groups of people. Then, he takes a *selfie* with a demo attendee and creates her contact based on the photo, triggering her registration as a subject. This automatically grants her a read access on all the conference pictures on which she appears, including the *selfie* and forthcoming ones, thanks to a pre-trained face recognition model.

## 3.2 Demonstration results

The demonstrations shows an implementation of the SWYSWYK model in the Cozy platform. The semantics of the model and its administration principles, based on the combination of the *documents are rules* and *subjects and objects are documents* mottos, opens to a set of benefits:

**Ease-of-use.** The content of a personal cloud, which describes Alice's acquaintances, and conversely, acquaintances which are associated with pieces of information of Alice, are used to express sharing rules. Most interesting rules could also be easily shared via the Cozy marketplace, and reused among interested users.

**Self-administration.** The sharing rules are self-administrated while the personal cloud content evolves. Typically, new subjects (attendees) are automatically created while inserting new contact files or address book entries and a search of correspondences with potential content to share with them is automatically triggered.

**Visualization.** Subjects and objects are all viewable documents of the personal cloud. Hence the net effect of any sharing policy can be visualized and precisely apprehended by Alice (e.g., the GPS tracks pictured in a map that she is ready to share with a subject represented by her identity picture).

**Control.** Administration tools are provided to ease the detection of suspicious permissions and sanitize the access control policy. Pursuing this objective, Watchdog triggers highlight newly generated ACLs involving sensitive subjects, documents and the associations of presumed incompatible subject/object pairs.

# 4 CONCLUSION

Finding new ways for the individual to intuitively share personal data and apprehend the real effects of their sharing policies is paramount. This is particularly true in a ubiquitous context where highly sensitive personal data (e.g., well-being data, daily activity logs) are produced at an increasing rate by smart appliances. Gathering these data in a personal cloud allows the definition of new transversal services of great value for the individual and holds the promise of a better privacy than storing them in a central cloud. However, appropriate sharing tools are needed to regulate data sharing and prevent individuals from exposing their digital life because of too permissive sharing policies. This shows in this demonstration how the SWYSWYK model tackles this challenge. We hope that this work contributes to a new step in the privacy preservation of personal data.

# REFERENCES


[1] Abiteboul, S., André, B., & Kaplan, D. (2015). Managing your digital life. Communications of the ACM, 58(5), 32-35.
[2] Bertino, E., Ghinita, G., and Kamra, A. (2011). Access control for databases: Concepts and systems. In Foundations and Trends in Databases, 3(1-2).
[3] Tootoonchian, A., Saroiu, S., Ganjali, Y., Wolman A. Lockr: better privacy for social networks. In CoNEXT, 2009.
[4] Carminati, B., Ferrari, E., and Perego, A. Rule-Based Access Control for Social Networks. In On the Move to Meaningful Internet Systems, 2006.
[5] Mazurek, M.L., Liang, Y., et al. Toward strong, usable access control for shared distributed data. In USENIX conf. on File and Storage Tech., 2014.
[6] Wang L., Wijesekera D., and Jajodia S. A Logic-based Framework for Attribute based Access Control. In ACM FMSE, 2004.
[7] Squicciarini, A.C., Sundareswaran, et al. A3P: adaptive policy prediction for shared images over popular content sharing sites. In ACM HT, 2011.
[8] Fang, L., and LeFevre, K. Privacy wizards for social networking sites. In ACM conference on World Wide Web (WWW), 2010.
[9] Tran-Van, P., Anciaux, N., and Pucheral, P. A New Sharing Paradigm for the Personal Cloud. In Trust, Privacy & Security in D. Business (TrustBus), 2017.
[10] Tran-Van, P., Anciaux, N., & Pucheral, P. SWYSWYK: A Privacy-by-Design Paradigm for Personal Information Management Systems. In ISD, 2017.
[11] Tran-Van, P., Anciaux, N., & Pucheral, P. SWYSWYK: A new sharing Paradigm for the Personal cloud. In ADMA, demo paper, 2017.
[12] Park, J., and Sandhu, R. (2004). The UCON ABC usage control model. In ACM TISSEC, 7(1).


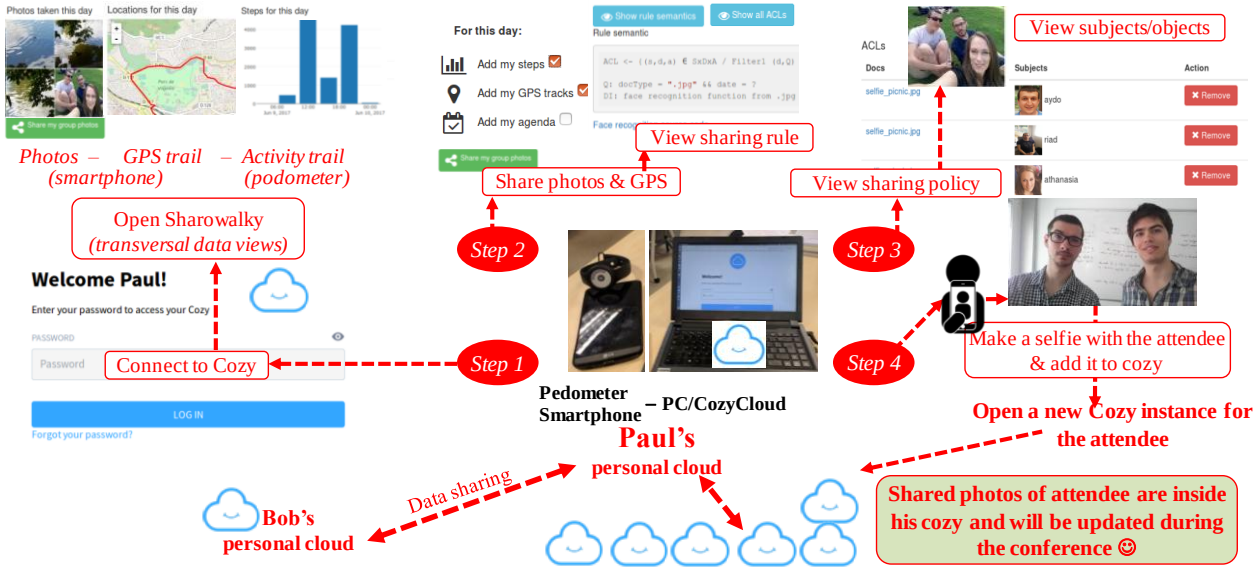


**Fig. 3. Demonstration scenario. Video available at http://wanda.inria.fr/demos/videos/swyswyk_model.avi**