Identity & Access Control Management Infrastructure Blueprint — Design Principles for True Informational Self-Determination

Daniel Pluta, Peter Weinert, Wolfgang Hommel
Leibniz Supercomputing Centre, Germany
pluta@lrz.de, weinert@lrz.de, hommel@lrz.de

LDAPCon 2011 — October 10th–11th, 2011 — Heidelberg, Germany

***Abstract*—We present the design criteria and the reference implementation of an identity & access control management (I&ACM) infrastructure, which does not only support *self-service* functionality to all of its users, but achieves true *informational self-determination* (ISD). Privileged proxy-users are completely forgone and each user stays in full control of which personal data is shared with which other users and services with the granularity of LDAP attributes. In analogy to Kim Cameron's well-known *seven laws of identity*, we present the *law of ISD-aware IT service design*. Its seven clauses are intended to be used as guidance and as checklist for the implementation of any IT service, including I&ACM.**

I. Introduction and motivation

Unless an IT service is explicitly intended to be used anonymously or has only a single user, it usually requires two important management components: First, an *identity management* (IDM) component focuses on mapping real-world entities to digital identities. These digital identities represent individuals' personal data (also known as personally identifiable information (PII)) and are referred to as *accounts*, *users*, or *identities*. This IDM component also serves as the basis for the second building block that is intended to handle *access management* (AM) issues by assigning permissions to digital identities, based on concepts such as *groups*, *roles*, or a combination thereof.

Literature and textbooks on identity and access management (I&AM) – including a book we contributed to a few years ago [1] – tell us that an organization that operates multiple IT services can reduce the redundant, time-consuming, costly, and error-prone management overhead of service-specific user and permission management, by implementing a single, central I&AM system. However, I&AM systems are typically complex systems that are designed and implemented very specific to a single organization. The design usually reflects a trade-off between the demands of IT services, which often evolve around data security, and individual user concerns, especially regarding privacy. Traditionally, the technical criteria dominate the design rationale, including custom adaptations based on a "business process" view. Unfortunately, a system-management-centric approach cannot map the individual personal rights.

In this article, we present a fundamentally different design of an identity & access control management (I&ACM) infrastructure. Our approach derives its design constraints from a person's perspective: *informational self-determination* (ISD).

ISD as the basis of our I&ACM design deserves some clarification. First of all, ISD is *not* the functionality that is provided by the web-based self-service portals found in many of today's I&AM implementations. ISD is neither the same as what the term *privacy* is currently typically used for. Instead, let us take a closer look at the explanation of ISD that is used in [7]:

> The term *informational self-determination* was first used in the context of a German constitutional ruling relating to personal information collected... (and) is often considered similar to the right to privacy but *has unique characteristics that distinguish it from the "Right to privacy"* in the United States tradition. Informational self-determination reflects Westin's [6] description of privacy: "The right of the individual to decide what information about himself should be communicated to others and under what circumstances."

It is important to note that ISD is not a free pass to do what you want to do in a completely anonymous manner. In contrary, our design is also based on the concept of *individual responsibility*:

> "Inseparable from the principle of self-determination... is the notion of individual responsibility." [3]

Thus, authenticity and non-repudiation of all user actions and I&ACM system transparency represent two key characteristics that are required for the broad acceptance and enforcement of individual responsibility.

Self-determination could eventually be achieved by *user-centric identity management* (UCIM); however, approaches in which the user's client stores and manages the personal data do – by design – not work very well for services, for which the personal data also needs to be available while the user is offline, e. g., email servers. This limitation of UCIM approaches lead to service-specific copies made of the PII, often without the user being aware of them, so basically the benefits of a centralized IDM are abandoned and there is no guarantee that ISD is not violated.

Thus, we focus on a centralized, LDAP-based solution that is characterized by fully honoring *the principles of ISD*. We will see that the resulting I&ACM infrastructure looks somewhat different compared to traditional I&AM systems. There are no unnecessarily privileged LDAP proxy-users, not even for graphical management front-ends or connected IT services, such as email or WWW. Furthermore our design offers several additional benefits to the LDAP server operators and the administrators of the involved IT services in comparison to a traditional I&AM system.

Our motivation can best be summarized as:

- *Never hesitate to put anything into question, including all currently well-known and generally accepted I&AM ideas and best practices.*
- *Strictly rely upon, trust in, and make use of the already existing skills, independence, and responsibility of each individual.*
- *Respect each individual's requirements, thus never judge others by one's own standards.*

Before we present our system design decisions, we will give a brief overview of the state of the art of currently deployed, organization-grade I&AM solutions.

II. The state of the art

The currently available – we refer to them as traditional – I&AM systems have their origin in system management. They represent highly customized solutions, which are often hard to analyze and understand in depth without knowing the organization-specific, evolutionary background. Therefore, an I&AM is designed by experienced system management architects. This results in a list of requirements that is limited to technical topics and features. The efforts on I&AM systems are purely technology-driven, individuals' personal motivations are often considered subordinate compared to the complex technical system management requirements. Taking *each individual's demands* into account seems to be completely out of scope or leading into chaos.

Let us have a look at two resulting principles of how currently deployed I&AM systems in many organizations are typically designed today:

1) **Leverage existing systems and data**: If identity and access information is already available in human resources management databases or in already established IT services, this data will be re-used when an I&AM system is built. Moreover, if one of the existing IT services already has powerful IDM and AM components, it may even be chosen to become the central I&AM system or at least its core component.
2) **Focus on current and foreseeable IT service requirements**: The I&AM system's features and characteristics are designed based on the current and foreseeable *technical requirements of the IT services*.

Sticking to these design principles has led to many I&AM implementations in the last decade that were considered successful because they met the considered requirements.

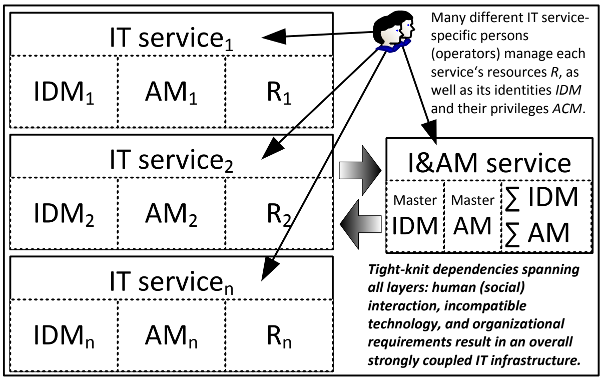

IT service$_1$
IDM$_1$
AM$_1$
R$_1$
IT service$_2$
IDM$_2$
AM$_2$
R$_2$
IT service$_n$
IDM$_n$
AM$_n$
R$_n$
Many different IT service-specific persons (operators) manage each service's resources *R*, as well as its identities *IDM* and their privileges *ACM*.
I&AM service
Master IDM
Master AM
∑ IDM
∑ AM
Tight-knit dependencies spanning all layers: human (social) interaction, incompatible technology, and organizational requirements result in an overall strongly coupled IT infrastructure.

Figure 1. Traditional I&AM infrastructure design leads to strong dependencies.

Figure 1 summarizes a common, broadly deployed, and thus well-known technology- and system-management-driven I&AM infrastructure in a generic manner. Its implementation approach and most important design ideas are the following: All the existing IT services' specific IDM and AM components are sourced out (aggregated) into an overall I&AM service, which internally uses its custom "Master IDM" and "Master AM" to manage the overall IDM and AM collections. Often one or more services are elected to represent the "leading data source". Depending on the amount of different data sources, the data is aggregated, correlated, and then provisioned into the services. Some data could also be requested online (using extra proxy-user accounts for example).

Most I&AM systems anticipate changes that are of purely technical nature, for example, an increasing number of stored digital identities, the integration of additional IT services, the evolvement of the data schema, and the placement of additional replica at further locations. We refer to this property of I&AM implementations as *system scalability*.

However, changes to business processes or individual user demands become almost impossible to implement, if the I&AM system was not designed to handle them a priori. Successfully dealing with changing requirements is a challenge for I&AM systems as it is for any other IT service. In practice, organizations then often give in and start to adapt their business processes and workflows to the existing IT infrastructure – in this case the limits of the deployed I&AM system. Also, many individual requirements will simply have to be ignored, leading anywhere from unhappy users to lost customers or departments that may even begin building their own I&AM infrastructure. We refer to the property to handle personal requirements from very small to arbitrarily large environments as *management scalability*.

In an ideal-world scenario, where no other IT service administration is needed and the overall dependencies are mastered successfully, this approach at best fulfills the need of I&AM administrators.

From the point of view of an experienced system manager, any managed IT system's resources either *deserve special protection* (e. g., due to business secrets, procedural advantages, . . .), *are dependent* (e. g., on file and VPN network resources, . . .), or *are dangerous* (e. g., given some users' criminal intent). Therefore, the existence of a trustworthy administrator or system manager represents a commonly accepted axiom of current system management operations. An administrator solely takes responsibility for secure and trustworthy management of the access privileges to resources, adding or removing accounts to groups or roles, and assigning privileges to them. Depending on a distinct system's management-related feature-set, access to the system's identity information is (partially) delegable or even completely shared among or accessible for all administrators. By design, system administrators are powerful – far more powerful than intended. As a consequence, they are forced to bear unnecessary and quite hard to handle responsibilities.

In sum, these challenges remain in traditional systems:

- *Focusing solely on technical aspects during I&AM system's design cannot result in an legitimate, universally deployable, trustworthy, and ISD-aware I&AM solution.*
- *Addressing personal concerns without a chance to respect ISD clearly misses the target.*
- *Burdening administrators with too much power and responsibility is not desirable.*

We present a fundamentally different approach to the design of centralized, general purpose IDM and AM solutions, which we refer to as Identity & Access Control Management (I&ACM). We base our design on the principle of ISD, which means that any individual – represented by a digital identity – has the right to decide what information about herself should be communicated to others and under which circumstances. That individual is responsible for her decisions. The primary goal is to not only fulfill technical and system-specific requirements, but to create a system that also fulfills each user's *individual* I&ACM requirements.

The remainder of this paper is structured as follows: In the next section, we present and explain the seven clauses of *the law of ISD-aware IT service design*. As several of this law's clauses are violated by traditional identity management implementations, we then present our ISD-aware reference implementation: Section IV explains design considerations, followed by Section V, which describes the OpenLDAP-based [2] backend including the flat directory information tree layout, schema design, and security properties. Important details about the semantics and the access control logic are then discussed in Section VI. An deployment, migration, and integration approach is then specified in Section VII. We discuss our work in Section VIII. A summary is given in Section IX.

III. The law of ISD-aware IT service design

Our law enforces encapsulation, transparency, care, and attention leading to ISD. Each IT service should undertake and obey this law:

§1 A service is an offer and has a mission.
A service competes for customers, i. e., individuals must not be forced to use it. A service resolves a distinct problem. An I&ACM system's mission is to host personal data and relationships. This is its only task.

§2 Each individual explicitly supplies her personal data.
The data is not supplied by a technical or administrative instance. Each person feeds her data herself into the I&ACM, therefore fulfilling legitimacy. The person has full control and individual responsibility over her data set.

§3 A central I&ACM is the single source of personal data.
This is the raison d'être of both I&AM and I&ACM. The advantage is that the data is in a single, private place, it is up-to-date and reliable, and a well-defined interface exists. Each person knows where her data is stored and whom she granted access, independent of the other infrastructure and services.

§4 Personal data is to be requested sparsely.
A service shall only request personal data it actually needs to provide its service and clearly state what the data is needed for. Data that only enhances a service must be optional. Any other data must not be requested.
This also includes a name or a generic login name. Instead, the I&ACM system supplies confidential, unique, and service-specific attributes for user identification separately for each service. The advantage is the achieved pseudonymity with regard to services and a generic support of miscellaneous authentication techniques. Pseudonymity can only be resolved at the administrative level of the I&ACM.

§5 Clearance relationships must be individual and explicit.
Each person explicitly declassifies her data herself in the I&ACM at the lowest granularity, that is single attributes, to each utilized service or demand, including logging or backup. The corresponding relationships belong to her data set.

§6 Clearance logic for existent data must be central, public, and statical.
The clearance logic must be encapsulated in the I&ACM. It must be at no higher level than the access control list (ACL). This relieves the graphical user interface (GUI) or application developer from the burden of access control logics. It enhances cohesion and looses coupling resulting in comprehensible correctness and reduced complexity. Being public, the ACL is at any time accessible anonymously in advance of the registration, to document and guarantee the privacy policy enforcement. Being statical, semantics of the existing ACL is at any time constant to prevent bad surprises and later disadvantages. ACL for new attributes can be added only if this is respected.

§7 **The service must be secured comprehensible.**
Clear, well-understood, and adequate security policies are part of the service. This includes, for example, rules and workflows for how pseudonymity can be resolved, how access to the I&ACM is restricted, how the server is physically accessed, and the separation of system and service administration. The policies must be accessible anonymously in advance of a new user's registration. They may not be softened, but only hardened to keep up with the time. The policies must be put into practice without any exception.

These clauses form the constraints for the architecture and the design of an IT service, including I&ACM. For ISD, not even one of them must be violated.

IV. Design considerations

We separated all the components nowadays involved in I&AM and reassembled the remaining and strictly ISD-subserving components. In particular, we split I&AM from system management, as the system management component's internal details remains mostly out of scope within this article, but will be addressed in our future work. We further demerged the I&AM component itself, which results into two standalone components: the IDM and the AM component. We also broke the AM component into an IDM-self-referencing AM and the remaining system-management-specific AM component, where the latter is not of detailed interest here, either. We are primary focusing on the ISD-specific requirements of the IDM and its specific AM component setup, that just for now represents the I&ACM's core feature.

We adhere to the principle of separation of concerns. Separation of operation and administration tasks has to be achieved. The term *operations* covers all system management related tasks like tuning of system parameters. The administration involves legal, organizational, and procedural aspects, for example, preventing the disclosure of any pseudonyms or ensuring legal certainty in the case that an exceptional lookup of an individual's data is requested by law enforcement.

Figure 2 illustrates the I&ACM's system core architecture. The traditional approach was illustrated in Figure 1. Our design decouples identity and access management from IT service operations. Decoupling is achieved via true social interaction using the I&ACM's PII storage to confidentially store and document the resulting relations. Each human being herself is the exclusive source (§2) regarding data input and the respective access control decisions (§5) within the I&ACM-Service. The I&ACM service itself is the only valid source of any kind of PII records for all other persons and any IT service a person is responsible for.

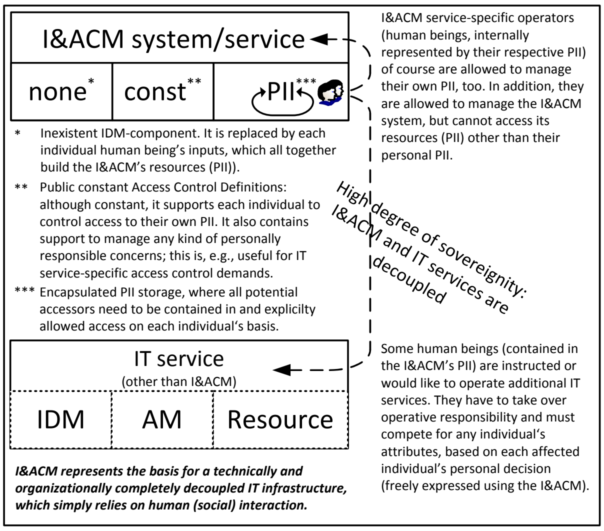

I&ACM system/service
none*
const**
PII***
* Inexistent IDM-component. It is replaced by each individual human being's inputs, which all together build the I&ACM's resources (PII)).
** Public constant Access Control Definitions: although constant, it supports each individual to control access to their own PII. It also contains support to manage any kind of personally responsible concerns; this is, e.g., useful for IT service-specific access control demands.
*** Encapsulated PII storage, where all potential accessors need to be contained in and explicilty allowed access on each individual's basis.
IT service
(other than I&ACM)
IDM
AM
Resource
I&ACM represents the basis for a technically and organizationally completely decoupled IT infrastructure, which simply relies on human (social) interaction.
I&ACM service-specific operators (human beings, internally represented by their respective PII) of course are allowed to manage their own PII, too. In addition, they are allowed to manage the I&ACM system, but cannot access its resources (PII) other than their personal PII.
High degree of sovereignity: I&ACM and IT services are decoupled
Some human beings (contained in the I&ACM's PII) are instructed or would like to operate additional IT services. They have to take over operative responsibility and must compete for any individual's attributes, based on each affected individual's personal decision (freely expressed using the I&ACM).

Figure 2. In accordance with our law's §1, the I&ACM represents just another independent IT service. Furthermore, the overall system architecture results in a completely decoupled IT service infrastructure.

Opening the registration interface opens a possibility for denial of service attacks by anonymously creating accounts until the storage capacity is reached. Therefore, our design can use a secured registration-interface. This interface distinguishes persons from robots. It empowers a dedicated I&ACM proxy-user for registration purposes with add-only privileges. This proxy-user is represented by a person or it is encapsulated within a dedicated registration-specific GUI. It depends on the organizational requirements. The I&ACM's system design offers support for both. Please note that this is the only GUI that is needed.

Instead, any existing non-privileged LDAP client can be used without limitations to interact with personal owned data. Offering a single customized GUI would represent an overall higher service quality level. Nevertheless, the development of such a GUI is not in the scope of this article.

V. Directory service backend design

In this section we present the directory backend for the I&ACM's prototype implementation. We describe the most important design decisions regarding our data structures: the I&ACM's core schema, a service's service-specific schema, and the I&ACM's directory information tree (DIT).

A. I&ACM's core schema

The two objectClasses *isdPerson* and *serviceOffer* and their respective attributes represent the I&ACM's core schema. isdPerson is the base class (structural) for a person's PII record. Although [5] states "The 'person' object class is the basis of an entry that represents a human being." we are not allowed to use this class as a basis, because it violates our law's §4. Objectclass *person* tries to represent a persons. As persons do not want to be forced to be represented by a surname (*sn*), this objectclass's mandatory *sn* attribute fails to represent a basis of an entry that represents persons. Similar arguments also argue against the other optional attributes.

Therefore our base class just contains two mandatory attributes: *cn* and *userIdentifier*. The LDAP standard does not include a dedicated unique identifier attribute definition. Using any already existing attribute is problematic under various aspects. It leads to a potential disclosure of login information and fails cohesion. For this reason, we specify a dedicated attribute called *userIdentifier* to be used as a personal generally confidential unique identifier. A person can change this identifier's value at any time, as long as it stays globally unique (slapd's unique-overlay ensures this for us).

```
dn: cn={nn}iacmService,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {nn}iacmService
olcAttributeTypes: {0}( 1.3.6.1.4.1.... NAME ’userIdentifier’ DESC ’iacm user
 identifier’ EQUALITY octetStringMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
olcAttributeTypes: {1}( 1.3.6.1.4.1.... NAME ’sqa’ DESC ’service quality
 assurance’ EQUALITY octetStringMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
 SINGLE-VALUE )
olcObjectClasses: {0}( 1.3.6.1.4.1.... NAME ’isdPerson’ DESC ’PII base class
 for iacm’ SUP top STRUCTURAL MUST ( cn $ userIdentifier ) )
olcObjectClasses: {1}( 1.3.6.1.4.1.... NAME ’serviceOffer’ DESC ’service offer’
 AUXILIARY MUST sqa )
```

We use the auxiliary class *serviceOffer*'s attribute *sqa* to enhance entries representing service-specific proxy-users. It stores a detailed "service quality assurance" (sqa), useful to compete for customers. Next to a description of a service's operation, it explains the list of required PII attributes, their intended purpose, the services' service level agreement, or its security measures. sqa's syntax is octetString, supporting storage of cryptographically signed documents. It is defined single-valued, so that this value, once written, cannot be changed (our ACLs ensure this).

Another service can connect to I&ACM as described next. This may result in a service-specific service registration and schema extension, illustrated in Figure 4. The service-specific schema extensions are not part of I&ACM's core data structure.

B. HPC service schema extension

The following schema definition exemplifies the schema extension for a common high performance computing (HPC) service that should be integrated into our I&ACM. We assume that the hpc service is running on any POSIX-conform UNIX server system.

All attributes are derived from the standardized posixAccount objectclass's attributes. We group the attributes into two objectclasses called *hpcServiceConstant* and *hpcServiceVolatile*. Together they build the superior classes of the *hpcService*. This splitting offers advantages regarding the I&ACM's internal ACL checking: the volatile attributes can be changed freely, the constant attributes can only be written once. The hpcService objectclass collects (schema inheritance) all service-specific attributes from the split objectclasses' attributes. The constant attributes are add-only, the volatile are writeable by their owner. The overall hpcService objectclass increases efficiency regarding a persons service-specific clearance.

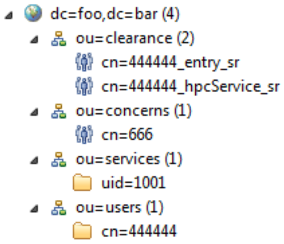
```
dn: cn={15}hpcService,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {15}hpcService
olcAttributeTypes: {0}( 1.3.6.1.4.1.... NAME ’hpcGidNumber’
 DESC ’hpc gidNumber’ SUP gidNumber )
        ... other attributes derived from posixAccount ...
olcObjectClasses: {0}( 1.3.6.1.4.1.... NAME ’hpcServiceVolatile’
DESC ’hpc service ACL: to self write’ AUXILIARY
 MAY ( hpcGecos $ hpcLoginShell $ hpcUserPassword ) )
olcObjectClasses: {1}( 1.3.6.1.4.1.... NAME ’hpcServiceConstant’
 DESC ’hpc service, single-value-ACL: to self add’ AUXILIARY
 MUST ( hpcGidNumber $ hpcUidNumber $ hpcHomeDirectory $ hpcUid ) )
olcObjectClasses: {2}( 1.3.6.1.4.1.... NAME ’hpcService’ DESC ’hpcService’
 AUXILIARY SUP ( hpcServiceVolatile $ hpcServiceConstant ) )
```

According to our previous assumption the above schema extension in principle is also applicable to any other UNIX host that should be integrated. Next to the HPC system, our compute cluster can be represented by a similar objectclass called clusterService for example.

C. Directory information tree

A DIT's layout depends on the usage scenarios and the services' requirements. The LDAP standard does not formalize any distinct layout. A well-known design approach is to keep the DIT as flat as possible. While this advice is generally helpful, it opens up a huge field of interpretations, resulting in complicated and often exhausting implementation specific discussions.

We decided to group all semantically identical entries of the same structural objectclass together into a single (flat) container. We do not further distinguish between adjacent entries.



```
◢ dc=foo,dc=bar (4)
  ◢ ou=clearance (2)
      cn=444444_entry_sr
      cn=444444_hpcService_sr
  ◢ ou=concerns (1)
      cn=666
  ◢ ou=services (1)
      uid=1001
  ◢ ou=users (1)
      cn=444444
```

Figure 3. 444444's personal DIT includes clearances and concerns she is individually responsible for as well as list of available services (here "uid=1001" represents the hpcService's proxy-user).

Which attribute should be used for an entry's relative distinguished name (RDN)? In general, any attribute can be used as a naming attribute. Thus, a follow-up issue is: In case the *cn* attribute has been chosen to build the RDN. How to avoid collisions, and what kind of content should be used as its value? Within I&ACM any DN that potentially needs disclosure, must not carry any other information except its addressing information.

We use a randomized RDN value for isdPerson entries. This offers collision detection and prevention. In particular, to our law's §3 and §2 the randomization is the best possibility to increase security (§7): it protects persons from an unintentional disclosure of private information. As the I&ACM does not support simple bind authentication for person entries, the randomization has no effect regarding usability. A person's DN and can be disclosed if needed.

For the reminder of our DIT the I&ACM's internal ACL logic enforces meaningless values for all potentially disclosed RDNs (see Figure 3).

VI. Data processing logic

According to our law's §2 to §7 we present our I&ACM's internal logic. This logic is stored centrally within slapd

(ACL engine). The ACL set we present here successfully achieves to

- protect personal PII attributes (privacy),
- offer a fine-grained clearance management for personal PII attributes, and
- completely publish I&ACM's internal logic and security enhancing settings (transparency).

The ACLs are statical for a given schema.

A. Users container

A directory service encapsulates each entry and guarantees, that it only operates on its own. Each entry is addressable using its DN.

Existing personal base class entries can be modified freely by its representing person (ACL {13}).

```
{13}to dn.one="ou=users,..."  by self write
```

We suggest to modify the userIdentifier and the userPassword attribute before a person adds additional PII records to her base entry,

B. Clearance container

Clearance entries use the objectclass groupOfNames to establish an indirection. A clearance entry is intended for disclosure of personal information to other persons. It is not intended for disclosure of informations for any services a person runs (see next subsection):

```
{8}to dn.regex="^cn=([0-9]+),ou=users,dc=foo,dc=bar$" attrs=givenName
 by self write by group.expand="cn=$1_givenName_sr,ou=clearance,..." =sr
{16}to dn.regex="^cn=([0-9]+)_([a-z]+)_(s|r|sr),ou=clearance,...$"
 attrs=entry,@groupOfNames by dn.exact,expand="cn=$1,ou=users,..." write
```

A person discloses information for a specific purpose. The receiver must take care and handle the information confidentially. This also applies to a person that is responsible for a service.

ACL {16} allows each person to express individual responsible clearances. ACL {8} represents the clearance for the givenName attribute. This also works fine for objectclasses instead of attributes, which is important for service-specific clearances.

C. Services container

A person responsible for a service acts on behalf of her role of this service's responsible administrator. For two reasons, an entry in the server container serves as an indirection:

- Separate personal clearances and service-specific clearances.
- Support to take over responsibilities (see subsection VI-G for details).

Any ACL that contains a “dnattr=. . .“ <who> expression (ACLs {21} - {23}) represent a major challenge with regard to non-repudiation and individual responsibilities.

```
{20}to dn.one="ou=services,..." attrs=userPassword
 by self none by dnattr=seeAlso write by anonymous auth
{21}to dn.one="ou=services,..." attrs=seeAlso
 by self none by dnattr=seeAlso write
{22}to dn.one="ou=services,..." attrs=description
 by dnattr=seeAlso write by users read
{23}to dn.regex="^cn=[0-9],ou=services,...$" attrs=entry,@account,
 @simpleSecurityObject,@serviceOffer by dnattr=seeAlso add by * none break
```

In subsection VI-G we suggest an enhancement regarding a possible solution. Clearance and service entries represent the I&ACM's internal core logic features exclusively useful for I&ACM's core identity- and personal access-control management. Thus they both only have an internal clearance effect: persons vis-á-vis persons or persons vis-á-vis service-proxies.

D. Concerns container

The confidential management of personal concerns offers a powerful opportunity to extend the I&ACM's internal clearance logic to also take effect for external services. We therefore benefit from slapd's memberOf-overlay as an indirection.

```
{17}to dn.base="ou=concerns,dc=foo,dc=bar" attrs=children by users write
{18}to dn.regex="^cn=[0-9]+,ou=concerns,dc=foo,dc=bar$" attrs=entry,
 @groupOfNames by dnattr=owner write
```

Similar to the clearance entries, concerns are represented by personally accessible groupOfNames objects. slapd-internally the memberOf-overlay adds the concern's DN to each member's PII attribute. Any person's memberOf-attribute represents a collection of tags (DN as a label). These individual labels can be used for flexible access control checks, so called label based security. As illustrated in Figure 5, the access control service is a separate service. The ACLs {17} and {18} grant each person the creation and deletion of her individual responsible concerns (§5). As stated above, please refer to subsection VI-G regarding the resulting challenge of non-repudiation.

E. Selected transparency measures

According to §6, the ACL set represents the I&ACM's only logic. There is no logic outside the ACLs. The ACLs are disclosed for anonymous access, including slapd's front-end, cn=config, and the PII storage itself. These are the most important ACLs dedicated to cn=config:

```
{0}to dn.subtree="cn=config" attrs=olcAuthzRegexp,olcRootDN
 by group/organizationalRole/roleOccupant="cn=iacmAdm,ou=roles,ou=iacm,..." write
{1}to dn.subtree="cn=config" attrs=olcAccess,olcRootPW
 by group/organizationalRole/roleOccupant="cn=iacmAdm,ou=roles,ou=iacm,..." write
 by * read
{2}to dn.subtree="cn=config" attrs=olcLogLevel,olcAuthzPolicy,olcConfigFile
 by * read
{3}to dn.subtree="cn=config"
 by group/organizationalRole/roleOccupant="cn=iacmOp,ou=roles,ou=iacm,..." write
 by * none break
{4} ...
```

As we are forced to specify an olcRootDN for slapd internal operations (e. g., memberOf), we explicitly grant global read privileges to any olcRootPW attribute (ACL {1}). This increases transparency and credibility, as it demonstrates that this value is empty and there is no "administrative super user backdoor" via LDAP. In a similar manner we grant read access to olcLogLevel, olcAuthzPolicy, and olcConfigFile (ACL {2}). These three attributes can be accessed by anyone but cannot be modified by anyone via cn=config. olcLoglevel and olcAuthzPolicy are ISD-relevant, while olcConfigFile should be always empty, like for example any olcRootPW attribute. Granting read access to these attributes contributes to transparency. Similar to published cryptographic algorithms, the complete I&ACM's internal logic is publicly reviewable.

Additionally the logic separates operational and administrative access (ACLs {0}, {1}, and {3}).

F. Thoughts about security

A detailed discussion of security related features is not part of this article. Nevertheless, we are aware of our overall responsibility with regard to ISD and to our role as the I&ACM service designers. We use I&ACM to demonstrate how to address security topics.

We keep following our approach: overall transparency and respect all persons' individual demands. Security features in general have to be in accordance with the rule of law and need to be first designed and accepted on the organizational layer. Afterwards, appropriate technical solutions can be implemented. We exemplify this process using our I&ACM:

The I&ACM's user registration and the offer to create unlimited personal service proxy-users and concerns represents a security threat. A denial-of-service (DOS) attack, where the I&ACM's overall capacity gets filled with garbage, may represent a risk for an organization, but not for the I&ACM itself. As a premise, the organization must define, follow, and enforce a hardened registration process to verify persons. Persons register themselves at the registration desk, where they are forced to identify themselves. Please note the vocabulary "forced": While this improves security, it confines the personal right of ISD. Thus this procedure needs to be published in advance, so that each interested person is aware of the resulting data processing. A person has to determine on her own whether to accept this kind of registration process or not.

The most efficient solution is to implement the registration into the I&ACM's logic and profit from its automatically disclosure of the internal processing: The user container is protected using ACLs. Being aware of our restriction of the ISD we suggest: according to our law's §4 a minimum privileged proxy-user holds add-only privileges that are restricted to isdPerson and simpleSecurity attributes. Thus the following ACL results in an anonymous person entry, addressable via its DN. The registration uses a random but unique value to initialize the userIdentifier and a password policy conform userPassword value.

```
{4}to dn.one="ou=users,..." attrs=entry,@isdPerson,@simpleSecurityObject
 by group/organizationalRole/roleOccupant.exact="cn=iacmri,ou=roles,..." =a
 by * none break
```

This proxy-user ACL successfully avoids anonymous persons. It is used to distinguish persons from machines and to verify their identity. Registered persons starting an attack can now be identified and take their personal responsibility. Only the registration knows which DN belongs to which person's identity card for example. As this lookup information is outside the I&ACM there is no chance for a successful internal lookup.

A registration desk is only allowed to create a base class without further personal attributes. ACL {4} protects new users from accidental misuse, because the registration proxy-user is unable to add any additional personal attributes.

G. Enhancement for dnattr=...ACL

As mentioned above the ACLs' <who> clause "dnattr=attribute...“ represents a challenge while maintaining individual responsibility. Its default behavior is to grant the DNs contained in the attribute's value access. We use this attribute's value (i. e. owner, seeAlso, ...) to document and ensure responsibility for an entry. However, granting write privileges introduces the chance for repudiation of individual responsibility. Write access implies that a person is also allowed to delete her own DN from this attribute, resulting in dangling entries without any responsible person. For demonstration purposes we have implemented a prototype patch for the ACL engine to avoid this behavior [4]. Currently our I&ACM testbed runs with this patch applied. Please also note that all the above ACL settings strictly rely on this patch.

We would like to suggest an extension of slapd's ACL engine similar to the "dnattr=member selfwrite" statement. For example, a "dnattr=member noSelfDelete" can grant *self* write access but denies delete access to *self*. This feature encapsulates responsibility and furthermore introduces support for what we call a "two-way handshake based responsibility transition". In our I&ACM scenario, this is useful for service proxy-users or concerns: In case a service's responsible person leaves the organization, she is allowed to add the DN of her successor. In case the successor is willing to take over responsibility, he deletes the previous DN. Next to the problem of repudiation of responsibilities, the successor needs to start over or the administration has to step in (updating the DN). In both ways the overall efficiency is decreased.

Next to the above enhancement of the ACL engine we would like to suggest to further increase the flexibility of common configuration options with regard to container-specific settings, e. g. for memberOf or to disallow bind_simple.

VII. Deployment, migration, and integration

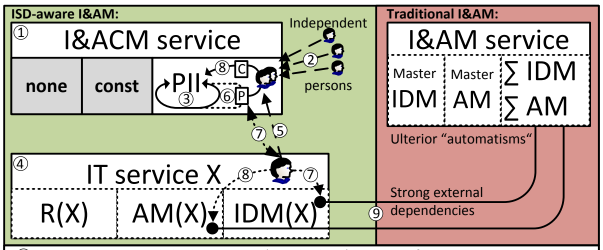
I&ACM can be deployed independently of any other services, even in parallel to an existing I&AM system, without conflicts or restrictions. According to §2, connectors, data migration, aggregation, transformation scripts are not allowed to migrate existing data into the I&ACM. Instead, each individual must be attracted by I&ACM's features. Users submit their data into the I&ACM. An I&ACM service has an immediate value for them. It can be used, for example, as an ISD-aware address book or even as a social network, where a person has a detailed control in place about which personal data to disclose to whom. From the beginning of the migration phase, the existing I&AM can be discontinued gracefully. According to §1 and our underlying system design (see Figure 2), the I&ACM system represents just another IT service. This, and the transitivity of ISD, is also the reason why the process of deployment, migration, and integration of ISD-aware services are all equal.

We use ordinary server hardware. We assume it offers enough capacity for successful initial deployment and enough performance to start operating the I&ACM. The I&ACM's backend utilizes LDAP's features, including its availability and performance features. As an obvious consequence, this system scales very well with arbitrarily growing requirements.

A deployed I&ACM will attract people. Also operators of other services, like email, will join and want to connect to the I&ACM service for their service.

An interested service operator submits the required and optional list of attributes (§4) including attributes for authentication (e. g., the hpcService objectclass presented in section V-B) (§7) and their description to the I&ACM's administration (§3). The administration decides whether any new attributes need to be introduced as a schema or ACL extension. The ACL extension is designed with care and must not alter existing semantics (§6).

The service operator then creates a dedicated service-specific proxy-user entry P within the I&ACM (see Figure 4, step 5). P contains a detailed and accessible service description including the necessary amount of personal data (§4), the security measures, the used infrastructure (§7), and the service level agreement (SLA) (§1). This description is stored in P's sqa attribute, which we have defined in section V. This process applies to deployment of I&ACM, as well. Now, persons will evaluate the offer make a decision. Signing in, a person will grant access to the listed attributes only for the service-specific proxy-user and become a customer.



① A person designs and ***deploys I&ACM*** (stand-a-lone) and waits for customers.

② Other persons (customers) may ***register and enrich*** their ***personal PII*** because they ...

③ ... want to confidentially collaborate using various IT services. Therfor, they ***share personal information*** and ***classify others regarding personal concerns***.

④ A person has another business idea: ***service X*** (benefits from existing ISD-awareness)

⑤ After this person registers herself, she creates a ***service-specific proxy entry*** P, she is ***individually responsible*** for (traceable by published ACLs). P represents an indirection and contains a public accessible service description attribute.

⑥ Persons convinced of **X**'s quality grant P access to personal attributes.

⑦ The person responsible for service **X** uses P to transfer the personal information.

⑧ Benefit from the concerns C as an indirection to derive AM information from a customer's attributes, using P (without the general loss of confidentiality – label-based security).

⑨ Cut overarching dependencies: stay in control of your service's internal automatisms, directly ***stay in touch with your users to understand and respect their demands***.

Figure 4. A person decides to offer an ISD-aware I&AM service. She deploys it, her users use it, services integrate into it and organizations migrate there I&AM towards ISD-awareness.

Figure 4 summarizes the deployment and service integration steps, which in the end lead to a successful migration, taking down any existing I&AM. Step 1 is the deployment of I&ACM. I&ACM's usage are represented by steps 2 and 3. Steps 4 to 8 illustrate the service integration process of any existing services (even in parallel connected to traditional I&AM services). After all services have been integrated step 9 finally concludes the migration phase.

Given that an I&ACM is deployed securely, the I&ACM is ISD-aware. That means, it represents a confidential decision documentation tool that relies on individual responsibility. Any decision only has an effect on the person herself (her PII attributes) or has effects within a connected I&ACM-integrated service (e. g., access to her individual responsible concerns). As a decision only involves a person's own affairs (IDM and AM), chances for a credible repudiation of individual responsibility like blaming a proxy-user, do not exist. Based on the high degree of encapsulation and its single interface (LDAP) the I&ACM does only offers one single peril point. Thus, it is very important to continuously monitor this point in real-time using alarms and notification mechanisms. For various reasons, logging is not acceptable for an I&ACM in general. It only offers a look into the past which does not prevent currently running attacks and thus cannot increase an operational service's security level. Logging also slows down system performance and last but not least clearly violates our law's §3 of ISD-aware system design: Any kind of logging of personal data weakens the high degree of encapsulation.

For the same reason (violation of §3), data backup service is in-active by default. This is obviously also true for all kind of other IT services, which log or backup personal data without explicit prior individual clarification and permission. This currently represents a major challenge for all IT service providers. Nevertheless, when using the I&ACM for the first time, the possibility exist to efficiently get each user's clearance for backup and logging of distinct values, even of the I&ACM's PII storage itself. Backup and logging are just another kind of IT services. Figure 5 exemplifies an I&ACM infrastructure that represents an ISD-aware IT service landscape containing various IT services.

VIII. Discussion of the Results

Traditional I&AM systems have been designed by administrators mainly targeting on IT providers' and their administrators' technical demands. The development thus has focused on decreasing the total cost of ownership: simplify all-day operations, e. g., by introducing various kind of automatisms. Therefore, many currently deployed I&AM systems represent extensively customized and often hardly portable solutions. The tight coupling between system management, platform-operational requirements, and efficient I&AM automatisms additionally increases the overall complexity. Trying to extend these systems into the direction of our goal of ISD is at worst impossible, and at best it is a very hard-to-implement feature that obviously does not seem to provide any immediate additional profit. Today's deployed traditional I&AM systems are effective in operation, but they have failed with regard to ISD, furthermore it seems to be futile to try to integrate ISD into these platforms afterwards because of the potentially generated security threats caused by many powerful automatisms that need to be protected.

In comparison to these highly customized and thus often confusing state-of-the-art I&AM infrastructures, to our knowledge the presented I&ACM is in fact the first ISD-aware I&AM solution.

In difference to all the existing traditional I&AM solutions, our I&ACM approach represents a completely decoupled universal core IT service. As a result of the I&ACM's high level of independence, it is indeed as easy to be deployed standalone as it is to integrate into any existing IT service infrastructure. Continuously honoring the principle of ISD and the resulting in individual responsibility the achievable flexibility during operation offers support for any kind of migration scenarios. An I&ACM is able to effectively and efficiently redeem any traditional I&AM infrastructure. The degree of fulfillment regarding the efficiency (whether integration, migration or operation) directly scales with the efficiency the embedding organization is able to achieve: As the I&ACM consequently follows the approach to decrease the initially needed count of operational proxy-users, no extra personal is needed to administrate all these proxy-users. Each user is free to self-responsible create non-privileged proxy-users to optimize her daily operational tasks. Blaming the organizational I&AM for being complex or hard-to-handle thus can always directly be referred back either to the originating person or the organizational structures and requirements she is a part of. An I&ACM that conforms to our law can hold a mirror up to a protester.

The amount of employees needed to successfully operate an I&ACM is constant: In difference to traditional I&AM systems, I&ACM only needs to scale with itself, but not with the overall organization's growth. In the past, the I&AM operation represented the bottleneck for overall organizational growth and flexibility. Now, the deployment of an IT service represents the bottleneck, because they take more time to be deployed compared to the time a schema and ACL extensions takes. Both extensions can be done without any downtime of the I&ACM.

One of the most important and first unique feature of the I&ACM is its straight-forwardness and the resulting compactness that offers the best possible internal and external (overall) comprehensibility even for non-experts: Internally, from an organization's personal responsible (e. g., executive officer) as well as externally from any potentially interested future user's perspective. As a direct result of the achieved simplicity, the probably most important effect of an operational I&ACM is the chance to question or even de-mystify most of the currently cryptic IT-security related discussions, the resulting organizational decisions and procedural directives: The I&ACM not just solves, for example, the default backup of personal data issue and also the overall logging, it also provides practical alternative ways to achieve the same benefits of identity management at the core of the IT service infrastructure without the need to restrict the principle of ISD.

Another very important and also unique feature of our system design, which is a direct result of the high degree of encapsulation of personal data records within the I&ACM is its independence from any kind of management GUI. All communications use the plain LDAP protocol together with individual, personally bound credentials. As a consequence, no separate and typically hard-to-secure abstraction layer, e. g., an I&AM management API, is required to be maintained.

An I&ACM stands out against other I&AM services due to its explicit compliance with ISD. One could guess that I&ACM shifts the responsibility for ISD-aware processing of personal data to the providers of other services. This is a misjudgment. Everyone who processes personal data always has had to carry out this duty already in the past. However, not until now was it possible for a service provider to implement true ISD completely. The personal data was ordinarily made available technically obviating the person. Now the person explicitly clears her required data to a service. The resulting personal relationships and the individual responsibilities are documented in the I&ACM. A service provider can benefit from our law and should employ it, as it is done in I&ACM. She will have to use I&ACM instead of traditional systems to transitively become ISD-aware. Only then a service provider not only offers a technical service, but a service that does not lose sight of the rights of its users and customers.

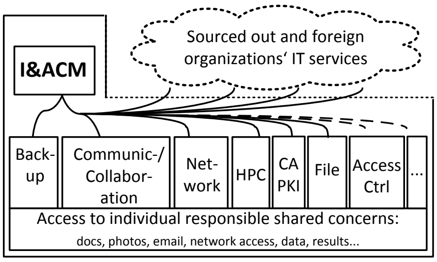

I&ACM
Sourced out and foreign organizations' IT services
Back-up
Communic-/Collabor-ation
Net-work
HPC
CA PKI
File
Access Ctrl
...
Access to individual responsible shared concerns:
docs, photos, email, network access, data, results...

Figure 5. A fictitious organization runs an ISD-aware IT landscape: As an I&ACM is inexpensive to deploy and cheap to operate any organization can afford it, even if to just express its respect vis-à-vis their employees, customers, patients, students, guests, and any human user in general.

Finally, the fictitious organization presented in Figure 5 introduces important organizational questions: Where should *the* I&ACM be deployed and who should get administrative access, how? Only the chief executive managers, any kind of trustworthy board, an external security officer, the CEO's best friend, or only the sovereign . . . The bad news is, it depends, a generally always correct answer does not exist. However, the very good news is, I&ACM is the first also even ISD-aware I&AM that legally supports all scopes of deployment, as long as our law's §3 is considered.

Depending on a distinct deployment's scope, capacity, and monetary aspects need to be considered, as well. Our I&ACM and any other future ISD-aware I&AM service, internally does not offer support to analyze personal data or any kind of resulting social relations.

IX. Conclusion

In this paper, we presented the *law of ISD-aware IT service design*. To demonstrate that the law and its resulting guidelines can be met, we implemented a prototype and described the utilization for ISD-aware I&AM, which is referred to as I&ACM.

Our I&ACM represents the first ISD-aware I&AM service. It can be deployed independently. Its system architecture is completely decoupled and thus enables the easy integration of additional services within an overall ISD-aware IT service landscape.

I&ACM also provides a flexible management tool to map relations of (real-life) social networking. We also demonstrated that utilizing the principle of individual responsibility results in increased service management efficiency.

Altogether, I&ACM represents the first true *identity & access management as a service*.

References

[1] Arndt Bode and Borgeest Rolf (Hrsg.). *Informationsmanagement in Hochschulen, ISBN-13 978-364-2947-190*. Springer-Verlag, 2010.

[2] The OpenLDAP Foundation. The OpenLDAP Project, http://www.openldap.org/. [Online; accessed September 28th 2011].

[3] Eidgenössisches Departement für auswärtige Angelegenheiten EDA. Self-Determination in Switzerland, http://www.image-schweiz.ch/fileadmin/user_upload/pdf/e/Formulare_und_Dokumente/Self-determination_in_Switzerland.pdf. [Online; accessed September 28th 2011].

[4] Daniel Pluta. dnattr acl statement: users can produce dangling entries, http://www.openldap.org/its/index.cgi/incoming?id=6900;selectid=6900, April 2011. [Online, accessed September 28th 2011].

[5] A. Sciberras. Lightweight Directory Access Protocol (LDAP): Schema for User Applications. RFC 4519 (Proposed Standard), June 2006.

[6] Alan F. Westin. *Privacy and freedom*. Atheneum, New York, 1970.

[7] Wikipedia. Informational self-determination – Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Informational_self-determination&oldid=386747585, 2010. [Online; accessed July 5th 2011].