

User specific privacy policies for collaborative BPaaS on the example of logistics

Björn Schwarzbach*, Michael Glöckner*, Arkadius Schier[†], Marcin Robak* and Bogdan Franczyk[‡]

*Leipzig University,

Grimmaische Strasse 12, 04109, Leipzig, Germany

Email: {schwarzbach, gloeckner, robak}@wifa.uni-leipzig.de

[†]Fraunhofer Institute for Material Flow and Logistics IML

Joseph-von-Fraunhofer-Str. 2-4, 44227, Dortmund, Germany

Email: arkadius.schier@iml.fraunhofer.de

[‡]Wroclaw University of Economics

Kommandorska 118/120, 53-345, Wroclaw, Poland

Email: bogdan.franczyk@ue.wroc.pl

Abstract—Today’s business is more and more organized in collaborative networks. Although decision makers know the benefits of collaboration, they are afraid of losing control of their data, which is one of the main impediments for Cloud Computing. We propose a novel cloud based approach for collaboration in business processes with guaranteed control of the privacy of the data. The platform ensures the compliance with the companies’ privacy policies and laws. The paper shows the definition of privacy policies and how they are converted into a well established access control language. An example helps to clarify the methods.

I. INTRODUCTION

ALMOST ten years ago, in 2008, Thomas J. Bittman, vice president of Gartner Research, published his view on future Cloud Computing development. Back in the beginning of Cloud Computing, cloud services were build on proprietary architectures of few dominant cloud service providers, e.g. Google, Amazon and Microsoft. The main problem in these times was the missing interoperability and compatibility of the cloud services of different cloud service providers. During the second phase the vertical supply chain distinguished itself as first ecosystems of smaller cloud companies emerged within the Cloud Computing market. New cloud service providers use the proprietary Cloud platforms of the dominant providers of the first phase to provide their own services. During the last phase these smaller cloud service providers unite to form horizontal federations. This union increased their earnings by expanding their capacities while reducing the costs at the same time through more efficient resource allocation. In parallel, open interoperability standards of service communication in intercloud-environment have been developed[1].

In the past years more and more companies adopted Cloud Computing by integrating cloud services into their supply chain. Especially in Germany the use of Cloud Computing in companies has increased from 2011 to 2014 by 16 percent,

The work presented in this paper was funded by the German Federal Ministry of Education and Research under the projects PREsTiGE (BMBF 16KIS0082K) and LSEM (BMBF 03IPT504X).

almost every second company is consuming at least one cloud service[2]. These cloud services reach from Infrastructure as a Service to Software as a Service. Especially the Software as a Service provides an tremendous number of services for every kind of task that can be achieved by software. Unfortunately these services are often provided by smaller cloud service providers while the cooperation between them, i.e. Bittman’s third phase, is not well established. Every service has it’s own interfaces with different message formats, even two services that provide the same functionality can differ in message format, behaviour, and constraints. According to an interview among approx. 120 german small and medium sized companies the target group of these services, i.e. these companies, does not have the knowledge how to tackle this problem.

In [3] we have proposed an architecture of a platform that enables those companies to consume cloud services of different clouds. The platform offers the features of a business process management system by orchestrating the individual cloud services in business processes that have been modelled by the consumers, i.e. the companies. Hence, we call this approach and the service provided by the platform Business Process as a Service.

In the interview mentioned before we discovered that most of the companies who do not consume cloud services are reluctant because of a fear of losing control of their data, which is even more important because of the hacks of global players (e.g. Sony) in the past months. But also those companies who consume at least one cloud service are concerned because of privacy issues. [4] comes to the same conclusion.

So one of the main challenges for such a platform is preserving the privacy of the data and the compliance with privacy laws while the business process is executed. This becomes even more important when multiple companies are involved in one business process and need to share data to each other. In [5] we have proposed an approach for secure service interaction, which has shown it’s feasibility in multiple tests. The architecture proposed in [3] also provides a component

for adding privacy policies to the business processes and the individual activities which are evaluated and enforced by the platform while the business processes are executed.

This paper discusses a new and flexible approach to define privacy policies for data that is transmitted while a business process is executed.

The remainder of this paper is structured as follows. After a brief presentation of the platform's architecture and the relevant components, the concepts behind the privacy policies for collaborative business processes is shown. The next section shows the translation of these policies into machine readable and evaluateable form. The algorithm for the translation is applied to an example based on a logistics use case. The paper is completed by a conclusion, which also reveals open tasks and questions.

II. ARCHITECTURE OF THE PLATFORM

This section gives a very brief overview on the architecture of the platform for privacy preserving collaborative business processes.

The platform first presented in [3] is shown in Fig. 1. The components are represented as rectangles, their interfaces are shown as the lines between the components.

The most important component of the platform is the business process management system, which is located in the center. The business processes are modeled by the user with the configurator, which also enables the user to define process and activity related privacy policies and assign them to the appropriate objects. Privacy policies that are not related to one particular process or activity are defined in the privacy management component, which also stores the privacy policies defined in the configurator. The privacy management passes all known privacy policies to the identity and access management system (IAMS).

When the BPMS executes a business process and reaches an activity that needs some data as input to call the cloud service related to this activity, the BPMS queries the IAMS whether the service is allowed to access this data in the current context. If the IAMS grants access to the data (with potential obligations) the BPMS instantiates a gateway that takes care of a secure service interaction as described in [3]. The gateway also takes care of the obligations for data access.

The components of the platform provide RESTful web services to communicate. The communication is secured by SSL and client certificates. User data, except for the companies' core data, is held within the BPMS, there is no interface which offers the data to other components or external users other than through gateways, which ensures that no unauthorized entity can access data.

III. PRIVACY POLICIES FOR COLLABORATIVE BUSINESS PROCESSES

This section describes in detail our approach for defining privacy policies in the context of collaborative business processes. One of our main requirements for the approach was to provide the companies with a tool that they could understand.

To define privacy policies that can be evaluated automatically and be used to decide whether a service is allowed to access some data or not we rely on use access control approaches. Basically there are four different types of access control. The mandatory access control and discretionary access control where applied in computer systems in the 70s of the last century. While mandatory access control describes security from the system itself by policies like "access is only granted from localhost", discretionary access control assigns each identity the appropriate access rights.[6] Mandatory access control is still used nowadays, e.g. SELinux is applying this approach [7].

In the late 80s and early 90s more and more users where using computer systems, hence assigning each individual user, i.e. identity, the correct access rights was not feasible any more. So in the beginning of the 90s role based access control emerged [8]. Role based access control assigns roles to identities and access rights are assigned to roles. This approach is used in Linux and Windows file systems and almost every modern software. Roles can be organized hierarically as shown in Fig. 2. [9], [10], [11]

Because of the well established application of role based access control our first approach for defining privacy policies was to apply role based access control.

During multiple workshops with local companies we discovered that the companies do not think about privacy identically. One common thing is that all companies separated the actors who want to access data into groups. But while some companies had have a very easy and strict approach for group setup, others could not clearly tell us which companies are member of which group. Instead they used phrases like "The driver of the truck while he is in the destination city is allowed to get the recipient's phone number to call the recipient to tell him his arrival time". This simple phrase contains the following information:

- The basic role of the person requesting access to the recipient's phone number is *driver*.
- The person requesting access to the recipient's phone number has to be located in the city where the shipment has to be delivered.
- Even if the first two conditions are met, the driver is only allowed to get the phone number if he want's to use it to call the recipient to announce his arrival.

This simple policy cannot be represented easily with roles because the location of the driver is changing over time. To tackle such requirements the research community followed two core approaches: extend role based access control with additional features, e.g. context or attributes, and creating a new access control model. [10]

Following the role based access control [12] has developed an access control model, which extends role based access control for virtual organizations. Unfortunately this model does not cover business processes, workflows, and cloud computing. Other approaches in this direction do cover business processes but leave out the cooperational aspect. Ref. [13], [14], [15] proposed and evaluated an extended role based access control

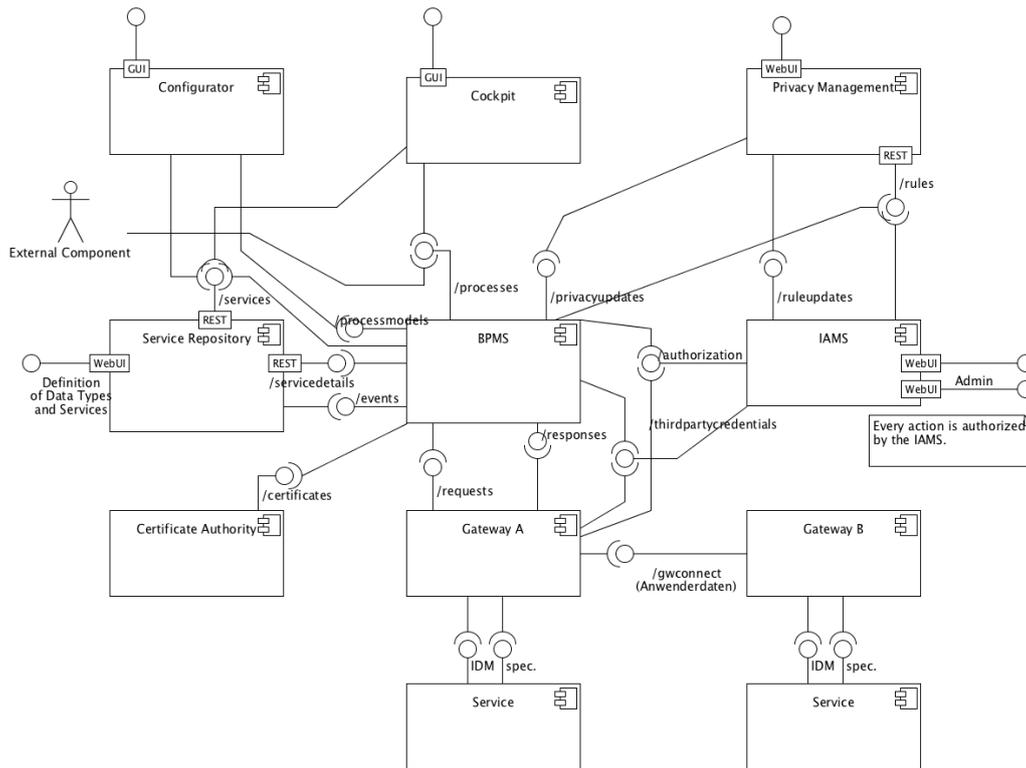


Fig. 1. Architecture of the platform for secure and privacy preserving collaborative business process as a service

model for team collaboration and workflows in the health sector.

All of the proposed models do not provide the flexibility in policy definition language that was needed by the participants of our workshops.

To achieve a maximum flexibility the research community developed a novel approach, the attribute based access control. In attribute based access control policies are based on attributes of subjects and objects. According to [16] attribute based access control is:

"An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions." [16]

The entity requesting access is called a subject. Typical attributes of subjects are their id, e.g. username, company name, and name of the department. The data that subjects want to access is called object or resources. A policy in attribute based access control is a triple of a subject, a resource, an operation, where the operation describes what access type the subject wants to have, and a result, e.g. grant or deny access. A policy can also comprise one or more conditions.

The policy "The driver of the truck while he is in the destination city is allowed to get the recipient's phone number to call the recipient to tell him his arrival time." consists of:

Subject The driver of the truck who is located in the destination city.

Resource Recipient's phone number

Operation Read

Condition Current activity in the workflow is "call recipient for dispatch notification"

Result Permit

The remainder of this section presents our approach on applying attribute based access control for privacy preservation to collaborative business process as a service. Privacy of data is always specified by the owner of the data, i.e. the entity who has created it.

First of all, in our platform business processes consist of activities that call external cloud based web services. Hence, there are two very basic roles in our platform. A process designer is an entity that models the business process, that is responsible for the correctness of the process itself, and that offers the resulting business process as a service to its customers. The second role is the service provider. A service provider is an entity that provides the external services that are being orchestrated in the business process by the process designer.

Our approach enables both roles to define their privacy policies independently from each other. It also includes privacy policies defined by law. Hence, the combined privacy policy consists of three columns as shown in Fig. 3 that can be

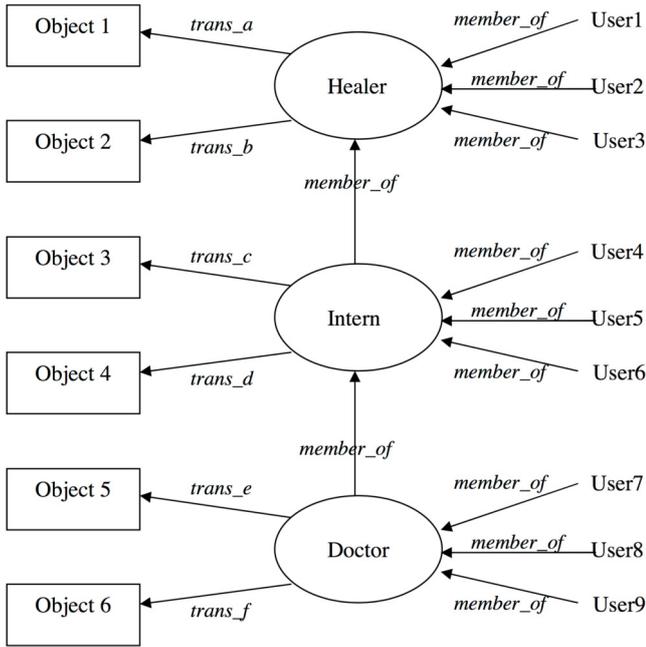


Fig. 2. Role based access control [11]

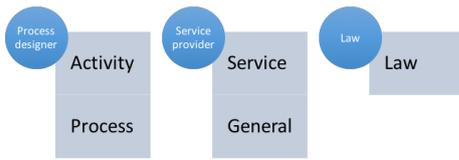


Fig. 3. Three columns of privacy policy process designer, service provider, and law

evaluated independently. The combined privacy policy results in permit if all three columns result in permit, else it results in deny.

To simplify the process of policy definition and to reduce redundancy in policies we provide each role with two levels of policies.

First a process designer can specify general privacy policies that are valid for the whole business process. E.g. a process designer may restrict access to all data to companies that are located in the Europe Union to ensure no data is transferred to other countries. Such privacy policies are visualized as tables where the objects are in the rows while the subjects are located in the columns. The cells contain either permit or deny depending on whether or not the subject is allowed to access the object. The subjects are defined by filters using attributes. So in this example the subject filter would be:

Companies meeting the condition: all locations have an attribute country with the a value that is in a list of the countries of the European Union.

The relevant section of the table for the privacy policy "Data can only be accessed by European companies." is shown in Fig. 4. Apart from the groups created by the process designer, every table does have an additional column *Default*. The

Object	Default	EU Companies
All data	deny	permit

Fig. 4. Privacy policy "Data can only be accessed by European companies." in table form

Object	Default	EU Companies
Parent	deny	permit
Child	permit	n/s

Fig. 5. Expandable objects in table to define a privacy policy for a child element different from the privacy policy for the parent element

algorithm to select the correct column when the evaluation of a policy, i.e. table, takes place is select the rightmost column whose filter does accept the subject, where *Default* accepts every subject.

The rows of the table represent the objects, i.e. the data the policy is about. The data is organized in an object hierarchy. Objects can be expanded to define policies for child elements as shown in Fig. 5. This table also states that EU Companies have unspecified access to the object *Child*. The evaluation algorithm handles *t/s* as if this column does not exist. The only column that is not allowed to have *t/s* is the *Default* column since else there would be no result for the evaluation of the policy.

The process level privacy policy applies to all data created by activities of the business process, i.e. it is assigned to all activities. If the process designer wants to define a different policy for a specific activity he defines a privacy policy on activity level. Privacy policies on activity level are evaluated before the process level privacy policies, i.e. activity level overrides process level. On activity level even the *Default* column can be set to *n/s*. If the evaluation of activity level policy results in *n/s* the policy on process level is evaluated.

The groups of subjects of a process designer's privacy policies can use both, companies and roles of the business process, as target. In case the process designer wants to use a business process role as the subject's filter, the systems shows up a list of the names of all swim lanes of the process. The process designer selects the appropriate entries and specifies the access rights as he does for company based filters.

The second role, i.e. the service providers, can define privacy roles that are applied to all data generated by their services in any business process. This is done on the level *General*. The definition of the policies follows the same concepts as for the process designer's policies. A service provider can override his general privacy policies by setting up a service specific privacy policy.

The third type of privacy policies are laws. Laws are provided by the platform provider as is and are not represented in a easy to read form as the process designer's and service provider's policy are.

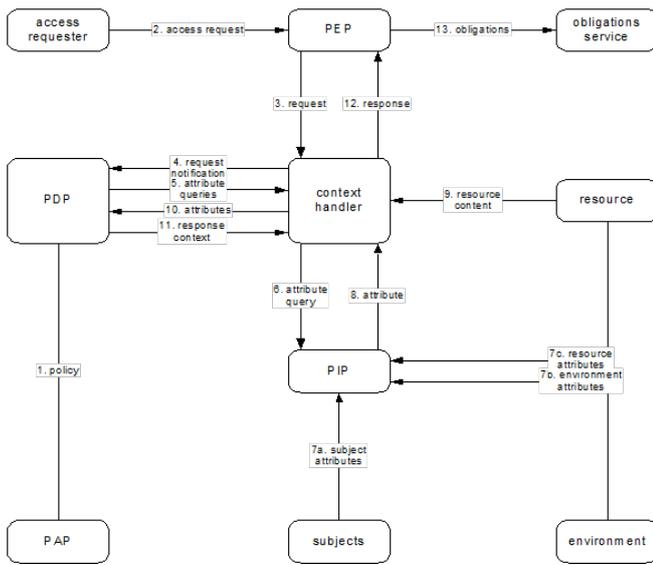


Fig. 6. Architecture and workflow of XACML 3.0 [18]

IV. TRANSLATION TO XACML

This section shows how the process of transformation from table form to machine readable form of privacy policies works. As standard policy language and architecture we have selected XACML which is a well established and accepted language and architecture to define attribute based access control policies. XACML is available in version 3.0 since the beginning of 2013 [17]. Unfortunately the support by tools is not very well at this time. Most of the tools are not available for the current XACML version, others are available but are not passing the conformance tests. Fortunately AT&T published the source code of a XACML 3.0 implementation that is almost complete. The project is in a frozen state and will become an Apache Incubator Project [18].

The core architecture of XACML 3.0 is depicted in Fig. 6. Applied to our platform's architecture the PDP is the IAMS, the PEP is the BPMS and the PAP is the Privacy Management.

The document specified by the XACML language specification is xml based and specifies three main elements: *PolicySet*, *Policy*, and *Rule*.

The root element of the XACML document is either *PolicySet* or *Policy*. A *PolicySet* can contain any number of *PolicySets* or *Policies*, while a *Policy* can only contain *Rules*. A *Rule* is a single expression with an effect and a condition. Each level of these elements can have a target. Targets are used as a very easy and fast selector for applicable section of the XACML-document. Hence, target provide limited functionality but speed up evaluation time of the XACML-document tremendously.

XACML adopts the attribute based access control structure of subjects, resources, conditions, and actions. This simplifies the algorithm for translating out table based privacy policies to XACML based ones. The algorithm consists of n main steps:

- 1) Creation of missing tables on activity / service level

Algorithm 1 Evaluate all cells of a activity table to either permit or deny

Require: table is not empty

Ensure: $\forall \text{ cell} \in \text{table} : \text{cell} = \text{permit or deny}$

expand all object

for all rows from top to bottom **do**

for all cells from right to left **do**

if cell is N/S **then**

if cell in default column is N/S **then**

if cell in corresponding column in process level table is N/S **then**

 cell \leftarrow value of the cell in default column of process level table

else

 cell \leftarrow value of the cell in corresponding column of process level table

end if

else

 cell \leftarrow value of the cell in default column

end if

end if

end for

end for

- 2) Evaluation of each cell of the activity / service level tables

- 3) Translation of each activity / service level table into XACML

- 4) Combination of the XACML fragments into the target XACML documents

The algorithm is executed separately for process designer and for service provider privacy policies. For process designer view the algorithm first identifies all activities of the business process that have to privacy policy table assigned and creates such a table with all cells set to *n/s*. This ensures that all activities can be handles the same way. The second step is the most important one. In this step all *n/s* values are evaluated to either permit or deny according to the algorithm 1.

After this step there are privacy policy tables for each activity of the process containing only permit or deny. These tables are now translated into XACML policies. Each table is transferred into a *PolicySet*. The target of this *PolicySet* is set to "resource:generator:activityid equals activity-id". The resource:generator:activityid is an attribute that is derived at runtime from the context of the BPMS. The *PolicySet* contains a *Policy* per attribute of the objects and a *Rule* per column of the table. The target of the *Policy* is set to "resource-id equals objectname:attributename", where objectname ist the name of the object and attributename is the name of the attribute of the current row. The rule's effect is set to the cell's value. In rules we do not use target's but we use condition's instead, since condition's are more powerful. The condition's of the rule is set according to the subject filter of the column.

When all tables are translated into XACML *PolicySets*, all *PolicySets* are put together in one *PolicySet*. This *PolicySet*

has "process-id equals id of the process" set as target. This *PolicySet* is the first of the three document types needed by our platform, the process related document. The platform holds one document of this type per business process.

The second document type handles the service provider's privacy policies. The platform creates one document that contains all privacy policies of all services of all service providers. The steps are very similar to the ones for the process designer's privacy policies. The following steps are executed for every service provider.

First it is ensured that there is a privacy policy table for every service. If one is missing, the system generates a table containing only *n/s* in all cells. After this step an algorithm similar to algorithm 1 is applied, with service provider level tables instead of process level tables. The result of this step is a set of privacy policy tables, one for each service, containing only permit or deny.

Each table is transformed into a *PolicySet* with the target set to "resource:generator:serviceid equals service-id", where service-id is replaced with the current service-id and resource:generator:serviceid is derived from the context of the BPMS at runtime. This *PolicySet* contains one *Policy* per attribute of the objects. The *Policies* contain one *Rule* per column as for the process based documents. All *PolicySets* of all Services are combined together into one *PolicySet* that is the second document type of the platform.

The third document type is produced by the platform provider and contains a *PolicySet* containing *Policies* for each law that is relevant to the platforms privacy preservation feature.

The process flow of this algorithm will be explained with an example in the next section.

V. EXAMPLE

The algorithm presented in the previous section will be processed in this section based on the following simple use case of the logistics sector. Due to the big similarity between the process designer's and the service provider's version of the algorithm this example is focussing on the service provider's privacy policies.

In this use case there is only one resource object called *address* containing the child elements *street*, *zipcode*, and *city*. Furthermore there is a service provider called ACME that is offering two services ACME-DE and ACME-WW to the platform. The company ACME has created two subject filters, one is named "GoodRelations" and contains companies that ACME likes to work with, and one is named "NeverAgain", this filter contains companies who ACME does not want to work with again any more.

ACME defines the default privacy policy as follows. Companies are allowed to access the zipcode and the city of an address but not the street. Companies matching the filter GoodRelations are allowed to access the street, companies matching the filter NeverAgain are not allowed to access the zipcode. This general privacy policy is represented in Fig. 7.

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	Permit	N/S	Deny
City	Permit	N/S	N/S

Fig. 7. Privacy policy for service providers on global level of ACME

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	N/S	Permit	Deny
City	Permit	Permit	N/S

Fig. 8. Privacy policy for service providers for service ACME-DE of ACME

In addition ACME defines a privacy policy for its service ACME-DE as follows. Companies should not be able to access a streetname but should be able to access the city of an address. Companies that ACME has good relation with are allowed to access the whole address generated by the service ACME-DE and companies that ACME has made bad experiences with are not allowed to access streetnames or zipcodes of addresses generated by the service ACME-DE. No statement is made for zipcode for default companies and city for companies matching the filter NeverAgain. The resulting table form of the privacy policy is shown in Fig. 8.

ACME does not define a special privacy policy for the service ACME-WW.

The first step is to create a privacy policy table for every service. There is already a privacy policy table for ACME-DE but none for ACME-WW. Hence, the system creates such a table with only *N/S* in the cells as shown in Fig. 9.

In the next step the *N/S* entries of the privacy policy tables of each service are evaluated to either permit or deny. This is shown in Fig. 10 and Fig. 11. The arrows show the source of the entry.

While the process for ACME-DE is easy, the process for ACME-WW needs some explanation. For example the cell Zipcode / GoodRelations contains *N/S* in the service specific table. During evaluation the algorithm first looks up the value

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	N/S	N/S	N/S
Zipcode	N/S	N/S	N/S
City	N/S	N/S	N/S

Fig. 9. Privacy policy for service providers for service ACME-WW of ACME

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	Permit	N/S	Deny
City	Permit	N/S	N/S

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	N/S	Permit	Deny
City	Permit	Permit	N/S

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	Permit	Permit	Deny
City	Permit	Permit	Permit

Fig. 10. Resulting privacy table for ACME-DE

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	Permit	N/S	Deny
City	Permit	N/S	N/S

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	N/S	N/S	N/S
Zipcode	N/S	N/S	N/S
City	N/S	N/S	N/S

Attribute	Default	GoodRelations	NeverAgain
Address			
Street	Deny	Permit	Deny
Zipcode	Permit	Permit	Deny
City	Permit	Permit	Permit

Fig. 11. Resulting privacy table for ACME-WW

in the default column for Zipcode in the service specific table. There it reads *N/S*, too. Hence, the algorithm looks up the value of the cell Zipcode / GoodRelations in the global level privacy policy table of ACME. There access control is set to *N/S* once again. So finally the algorithm falls back to the cell Zipcode / Default where it finds the resulting Permit.

For the transformation of the privacy policy tables

to XACML we assume that the filters are based on company names, although they good have any complexity. The companies' names are kept in the XACML attribute *urn:prestige:iams:ldap:Company:Name*. The resulting XACML portion for the service ACME-DE is shown below.

```

<PolicySet xmlns="
  urn:oasis:names:tc:xacml:3.0
  :core:schema:wd-17" PolicySetId="
  urn:com:xacml:policy:id:8cadbdca
  -1592-4ff9-bf49-a9ccccc064cf" Version="
  1" PolicyCombiningAlgId="
  urn:oasis:names:tc:xacml:1.0:policy-
  combining-algorithm:first-applicable">
  <Description />
  <Target>
  <AnyOf>
  <AllOf>
  <Match MatchId="
    urn:oasis:names:tc:xacml:1.0
    :function:string-equal">
    <AttributeValue DataType="http://www
    .w3.org/2001/XMLSchema#string">
    ACME-DE</AttributeValue>
    <AttributeDesignator Category="
    urn:oasis:names:tc:xacml:3.0
    :attribute-category:resource"
    AttributeId="
    urn:prestige:attribute:owner:service
    -id" DataType="http://www.w3.org
    /2001/XMLSchema#string"
    MustBePresent="true" />
  </Match>
  </AllOf>
  </AnyOf>
  </Target>
  <Policy PolicyId="
    urn:com:xacml:policy:id:62aa47ff-cbe5
    -4bfa-afef-737cb8e10ad4" Version="1"
    RuleCombiningAlgId="
    urn:oasis:names:tc:xacml:1.0:rule-
    combining-algorithm:first-applicable"
  >
  <Target>
  <AnyOf>
  <AllOf>
  <Match MatchId="
    urn:oasis:names:tc:xacml:1.0
    :function:string-equal">
  <AttributeValue DataType="http://
  www.w3.org/2001/XMLSchema#string"
  ">
    urn:prestige:data:address:street
  </AttributeValue>
  <AttributeDesignator Category="
    urn:oasis:names:tc:xacml:3.0
  
```

```

:attribute -category:resource "
AttributeId="
urn:oasis:names:tc:xacml:1.0
:resource:resource -id" DataType=
"http://www.w3.org/2001/
XMLSchema#string" MustBePresent=
"true"/>
</Match>
</AllOf>
</AnyOf>
</Target>
<Rule RuleId="
urn:com:xacml:rule:id:0fff9941 -8b89
-455c-a009-26e9107e0902" Effect="
Deny">
<Description>NeverAgain for Street</
Description>
<Target/>
<Condition>
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -is -in">
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -one -and -only">
<AttributeDesignator Category="
urn:oasis:names:tc:xacml:1.0
:subject -category:access -subject
" AttributeId="
urn:prestige:iams:ldap:Company:Name
" DataType="http://www.w3.org
/2001/XMLSchema#string"
MustBePresent="true"/>
</Apply>
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -bag">
<AttributeValue DataType="http://
www.w3.org/2001/XMLSchema#string
">NeverAgainCompanyName1</
AttributeValue>
<AttributeValue DataType="http://
www.w3.org/2001/XMLSchema#string
">NeverAgainCompanyName2</
AttributeValue>
</Apply>
</Apply>
</Condition>
</Rule>
<Rule RuleId="
urn:com:xacml:rule:id:db1e3bca -2cb3
-42f6-bcfe -299c40189b70" Effect="
Permit">
<Description>GoodRelations for Street<
/Description>
<Target/>
<Condition>
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -is -in">
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -one -and -only">
<AttributeDesignator Category="
urn:oasis:names:tc:xacml:1.0
:subject -category:access -subject
" AttributeId="
urn:prestige:iams:ldap:Company:Name
" DataType="http://www.w3.org
/2001/XMLSchema#string"
MustBePresent="true"/>
</Apply>
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -bag">
<AttributeValue DataType="http://
www.w3.org/2001/XMLSchema#string
">NeverAgainCompanyName1</
AttributeValue>
<AttributeValue DataType="http://
www.w3.org/2001/XMLSchema#string
">NeverAgainCompanyName2</
AttributeValue>
</Apply>
</Apply>
</Condition>
</Rule>
<Rule RuleId="
urn:com:xacml:rule:id:467ed7f7 -d7b6
-49e2 -970c -a32fb5b66a8a" Effect="
Deny">
<Description>Default for Street</
Description>
<Target/>
</Rule>
</Policy>
<!-- Skipping policies for zipcode and
city -->
</PolicySet>
<Condition>
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -is -in">
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -one -and -only">
<AttributeDesignator Category="
urn:oasis:names:tc:xacml:1.0
:subject -category:access -subject
" AttributeId="
urn:prestige:iams:ldap:Company:Name
" DataType="http://www.w3.org
/2001/XMLSchema#string"
MustBePresent="true"/>
</Apply>
<Apply FunctionId="
urn:oasis:names:tc:xacml:1.0
:function:string -bag">
<AttributeValue DataType="http://
www.w3.org/2001/XMLSchema#string
">GoodRelationsCompanyName1</
AttributeValue>
<AttributeValue DataType="http://
www.w3.org/2001/XMLSchema#string
">GoodRelationsCompanyName2</
AttributeValue>
</Apply>
</Apply>
</Condition>
</Rule>
<Rule RuleId="
urn:com:xacml:rule:id:467ed7f7 -d7b6
-49e2 -970c -a32fb5b66a8a" Effect="
Deny">
<Description>Default for Street</
Description>
<Target/>
</Rule>
</Policy>
<!-- Skipping policies for zipcode and
city -->
</PolicySet>

```

The resulting *PolicySets* for ACME-DE and ACME-WW will be combined and transferred by the Privacy Management to the IAMS. The same procedure applies to the business process related privacy policies.

VI. CONCLUSION

In this paper we have proposed a novel approach for defining privacy policies in business process and Cloud based scenarios. The definition is done by the end users with an easy to understand table based presentation and at the same time offers enough flexibility to fit the needs of the users. We have evaluated the approach with members of the target

group and found that it is feasible and easy to use. Especially the definition of global policies and local policies only where necessary was rated very good.

The technical implementation of the platform is working and fast enough even for a big number of policies. We have tested the platform with 500 services and a business process containing 40 activities. Every XACML request, i.e. request for privacy policy evaluation, was answered within a maximum of 27 ms over local network with no significant CPU load.

In the near future we will try to improve our platform especially in terms of visualization of privacy policies. Above all at the moment the definition of subject filters is either flexible or easy to use, depending on whether the user uses a code view or a list to select the companies from. We are planing to provide the user with a tool to define the filters using a set of attributes and a graphical editor to arrange those attributes.

Another task for the future is to perform system tests and experiments of the whole platform with companies in controlled laboratory and real world, as well.

REFERENCES

- [1] T. Bittman, "The evolution of the cloud computing market," *Gartner Blog Network*, <http://blogs.gartner.com/thomasbittman/2008/11/03/theevolution-of-the-cloud-computing-market>, 2008.
- [2] Statista, "Nutzung von cloud computing in unternehmen in deutschland in den jahren 2011 bis 2014," 2016. [Online]. Available: <http://de.statista.com/statistik/daten/studie/177484/umfrage/einsatz-von-cloud-computing-in-deutschen-unternehmen-2011/>
- [3] B. Schwarzbach, A. Pirogov, A. Schier, and B. Franczyk, "Inter-cloud architecture for privacy-preserving collaborative bpaas," *QUIS14*, 2015.
- [4] Statistisches Bundesamt, "12 % der unternehmen setzen auf cloud computing," 2014. [Online]. Available: <https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2014/12/PD14\textrunderscore467\textrunderscore52911.html>
- [5] B. Schwarzbach, M. Glöckner, A. Pirogov, M. M. Röhling, and B. Franczyk, "Secure service interaction for collaborative business processes in the inter-cloud," in *2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems. IEEE, 2015, pp. 1377–1386.
- [6] D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, N. Cuppens-Boulahia, F. Cuppens, and J. Garcia-Alfaro, Eds., *Data and Applications Security and Privacy XXVI*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [7] H. Lindqvist, "Mandatory access control," *Master's Thesis in Computing Science*, Umea University, Department of Computing Science, SE-901, vol. 87, 2006.
- [8] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (rbac): Features and motivations," in *Proceedings of 11th annual computer security application conference*, 1995, pp. 241–248.
- [9] I. Zahid and N. Josef, "Towards semantic-enhanced attribute-based access control for cloud services," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 2012, pp. 1223–1230. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6296118>
- [10] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering dac, mac and rbac," in *Data and Applications Security and Privacy XXVI*, ser. Lecture Notes in Computer Science, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, N. Cuppens-Boulahia, F. Cuppens, and J. Garcia-Alfaro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7371, pp. 41–55.
- [11] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," *arXiv preprint arXiv:0903.2171*, 2009.
- [12] A. Gouglidis and I. Mavridis, "domrbac: An access control model for modern collaborative systems," *computers & security*, vol. 31, no. 4, pp. 540–556, 2012.
- [13] X. H. Le, T. Doll, M. Barbosu, A. Luque, and D. Wang, "An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow," *Journal of biomedical informatics*, vol. 45, no. 6, pp. 1084–1107, 2012.
- [14] —, "Evaluation of an enhanced role-based access control model to manage information access in collaborative processes for a statewide clinical education program," *Journal of biomedical informatics*, vol. 50, pp. 184–195, 2014.
- [15] X. H. Le and D. Wang, "Development of a system framework for implementation of an enhanced role-based access control model to support collaborative processes," in *Proc 3rd USENIX Workshops on Health Security and Privacy*, 2012.
- [16] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*. National Institute of Standards and Technology, 2014.
- [17] OASIS, "extensible access control markup language (xacml) version 3.0," 2013. [Online]. Available: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>
- [18] AT&T, "At&t xacml 3.0 implementation," 2015. [Online]. Available: <https://github.com/att/XACML>