Graz University of Technology
Institute of Applied Information Processing and Communications

Bojan Suzic

Towards Secure Integration and Interoperability in Heterogeneous Environments

Ph.D. Proposal

22nd August 2016

Abstract

The emergence of technologies and business models that extensively rely on sharing of resources raised new challenges concerning efficient, interoperable and secure management of data sharing in complex environments. Due to an increasing degree of cross-system dependence and diversity of operating environments, the overall security governance of resources hosted at various third parties becomes progressively complex.

In our work, we approach the problem of authorization management of multi-entity service-based interactions. We advance the confidentiality and privacy of cloud-scale resource sharing by decoupling security management from proprietary platforms and implementing collaborative, model-driven and context-aware definition and enforcement of security policies. By relying on semantic technologies, we enable expressive and transformative policies applicable in diverse environments and multilateral flows.

Our contribution includes a data sharing and processing framework that consists of the architectural and interaction model, semantic vocabularies and enabling software components. This work is the result of activities performed in the scope of SUNFISH and several A-SIT projects.

The outcome of our work has been scrutinized by the scientific community through the peer-review and publication of six papers, with an ongoing review of one publication. In this proposal, we present these results and outline the future work that leads towards the consolidation of the contributions and completion of the Ph.D. thesis.

Contents

1 Introduction

Many organizations have already adopted or consider the integration of cloud services into their business workflows. Among the increasing number of entities, the adoption levels start to shift from the complement to the full replacement of existing systems with their cloud-based counterparts. A similar transition can be observed on the side of vendors, many of which focus their strategies on cloud-based product offerings. This is confirmed by the findings of various industry research groups. Among them, IDC expects Software-as-a-Service (SaaS) cloud delivery model [56] to significantly outpace traditional software product delivery model, with a nearly fivefold growth rate up to 2019 [54]. Similarly, Cisco and Forrester predict SaaS to be the most highly deployed global cloud service model at the end of the current decade [58, 4].

Besides the cloud, the other concept that is expected to significantly impact global patterns of data and service usage is Internet of Everything (IoE) [7]. Coined by Cisco, this term extends the reach of Internet of Things (IoT) by considering people and processes as additional internetworked entities [10]. Coupled with cloud and mobile technologies, IoE is predicted to increase the volume of data and heterogeneity of its distribution among devices and locations, resulting in more data stored on diverse ranges of smartphones, tablets, and machine-to-machine (M2M) devices [58].

The industry widely recognizes that the application of cloud technologies accelerates the intensity and capacity of inter-organizational collaborations [5]. The emergence of novel business models affects the complexity of these processes even further [103]. In an environment where the rising amount of data is stored on diverse systems and resides in domains of multiple subjects, the effective and efficient governance of data sharing processes becomes a growing concern.

In this proposal, we approach the problem of authorization management of multi-party cloud-based integrations. Our proposal focuses on transactions performed in the scope of Service-Oriented-Architecture (SOA) and Web APIs, which represent a prevailing building block of inter-organizational collaborations [53, 13, 103]. We introduce multilateral and semantic perspective to integrative processes, with the aim

to enable expressive and collaborative management of security controls in the cloud. By applying a model-based approach, we decouple security controls from proprietary platforms, enabling integrated, dynamic and resource-aware definition and enforcement of security policies. Our goal is to support confidentiality and privacy of cloud-scale interactions across heterogeneous entities by establishing a unified framework for externalized, granular and context-aware authorization.

In the rest of this chapter we present motivational use-cases that further elaborate the problem and we establish the research objectives. In the subsequent chapter, we position our proposal in the context of related work. Following that, we present preliminary results derived from the current work and draw an outline of the future work towards the completion of the Ph.D. thesis. Finally, we conclude this proposal with the overview of other relevant aspects.

1.1 Motivational Use-Cases

Cloud Integration platforms

Integration platforms serve as integration and automation environments that unify, bridge and orchestrate various backend and frontend services for their customers[1]. Figure 1.1 illustrates the execution flow of a system deployed at third party cloud and consumed as *Integration Platform as a Service* (IPaaS) [64].

Based on a predefined workflow, IPaaS accesses organizational and third-party resources co-located at various cloud service providers (CSP) or organizational premises (including external organizations). The connections with these entities can be established using various techniques, whereas OAuth 2.0 [29] serves as a dominantly applied authorization management framework for integrations based on Web-APIs [13, 101].

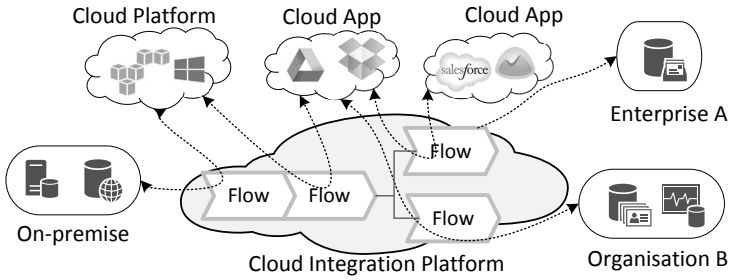

Figure 1.1. Integration flows executed by cloud integration platform.

Challenges arising from this scenario are manifold. In terms of confidentiality and privacy, coarse-grained, hardwired and context-insensitive OAuth 2.0 [29] authorizations support the conformance to the *least privilege principle* [79] at suboptimal level. Due to the static access scopes, integration platforms can retrieve more information

[1] The literature review in Section 2.1 provides additional details

than necessary to accomplish their tasks. Similarly, they may be able to alter client's data on other systems. The coupling and hardwired authorization extents[2] render the overall management of authorizations in multilateral interactions as costly and complex endeavor [90, 91].

Furthermore, as cloud services often rely on a multi-tenant delivery model, CSPs may get access to an unprecedented amount of resources belonging to different clients, which produces a significant global risk and makes these systems presumable targets of a range of sophisticated attacks. We have provided more details on these challenges in [89, 90, 91].

Being a representative example due to their complex setup and integration with different entities, IPaaS should not be considered as the only setup that introduces these challenges. Instead, even standard peer-to-peer interactions based on OAuth 2.0 scope confinement suffer from similar issues. The case of IPaaS brings these issues only to a new level, as these services aggregate data of many users and consequently represent a threat on a global scale.

Automation tools and services

The services such as Zapier [34], IFTTT [32] or Elastic [26] provide automation environments for end users, enabling them to connect devices and sources such as home automation systems, vehicles, mobile devices, and cloud services.

To illustrate automation scenarios we provide a couple of recipes from IFTTT [33]:

(i) *Add a reminder to Google Calendar when you miss a call on your Android*
(ii) *Post your uploaded YouTube videos to your WordPress blog*
(iii) *When you arrive home in your BMW signal GarageIO to open your garage door*
(iv) *See notifications for upcoming Google Calendar events on your Tesla dashboard*
(v) *Activate SmartThings device when you arrive home*

By relying on similar architecture as integration platforms, automation tools establish connections between various sources under user's control. While some control activities get performed at user's devices, in a typical case they are executed by automation platform, with the most of the processes occurring in a cloud, under a control of an external system.

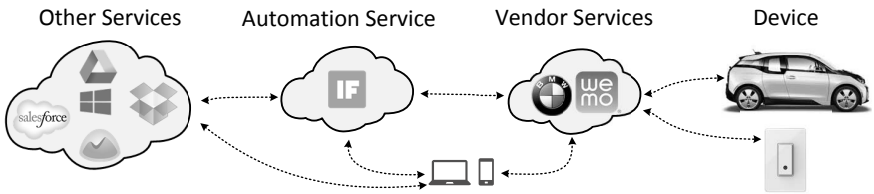

Figure 1.2. Management flows and entities in cloud automation scenario.

[2] We will interchangeably reuse this term to refer to an access scope and its generic application beyond OAuth 2.0 and UMA frameworks

The integration trend can be observed by inspecting BMW's ConnectedDrive, which employs cloud-deployed APIs for management of onboard vehicular resources [20]. The user has to connect to BMW cloud in order to be able to control or read the parameters of its vehicle. A similar model is found in home appliances examined by Notra et al. [60].

The management flows that represent these scenarios are shown in Figure 1.2. As related services typically rely on Web APIs and OAuth 2.0 authorization framework [29], they share similar issues as cloud integration platforms, with additional security concerns stemming from the presence of diverse devices and potential impact on the physical world.

Consolidated access management

The proliferation of various business models and service functionalities resulted in a broad heterogeneity, even between the cloud services of the same type. In a typical case, these services provide two main entry points for administrative tasks: web-based interfaces and web-based APIs[3].

While web interfaces provide a practical way to manage security controls at one provider, they are impractical for actions that need to be repeatedly performed and automated, especially in the case of many different providers. Hence, web APIs are deemed as more suitable interfaces for automated execution and integration of administrative controls.

In practice, however, providers tend to expose management APIs whose implementations differ not only in structure and organization but also in applied REST maturity levels [73] and exposed resource models [71]. Instead to allow flexible, adjustable and reusable security management controls, these interfaces typically force users to rely on integrated, predefined and static view derived from provider's restricted application scenario and business goals.

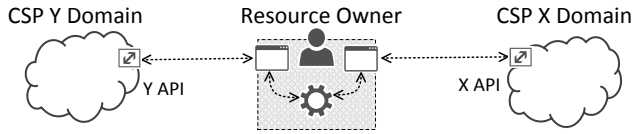

Figure 1.3. Management of user resources at different CSPs.

The simplified scenario from Figure 1.3 shows management flows between one user and two different CSPs. In practice, due to a larger number of CSPs utilized in each organization, the management of organizational accounts is established on a one-to-many basis. In the figure, these integrations are represented as the service-specific API and its corresponding local component at resource owner's premise. Translated to the context of many different users, this setup leads to many-to-many integrations

[3] We have provided more details on this setting in [90]

that need to be globally implemented to utilize APIs of each provider. This results in hardwired and tightly coupled implementations that are error-prone, costly to develop and complex to maintain.

As each provider exposes its proprietary interface, orchestrating security controls among different services and resource types becomes the complex and challenging task.

1.2 Research Objectives and Approach

The primary aim of this research is to advance the security of complex web interactions that drive resource sharing and consumption across diverse subjects and systems.

For this purpose, we have identified two main objectives. In our first objective, we address the challenge of service-coupled, implementation-specific and unilaterally established security management of users' resources at various service providers. We detach the security management from particular resources and interfaces, allowing this process to be performed using flexible and multilaterally adjustable controls.
We realize this separation in two planes. Firstly, we introduce the modeling of resources, services, policies and processes using abstract constructs that are reusable across the platforms in an integrative way. Secondly, we introduce the gateway that is responsible for security management functions, including policy management and context-sensitive, resource-aware and transformative policy enforcement. By reusing the constructs from the abstract plane, the gateway further allows the establishment of interoperability of security controls and consolidated security management across diverse parties.

In our second objective, we approach the challenge of static, coarse-grained and inflexible authorization extents that are currently broadly applied for inter-entity resource sharing. By relying on graph-based representations, we introduce structured and self-descriptive authorization extents that extend existing protocols with expressive, granular and dynamic capabilities. We apply these constructs to support confidentiality and privacy requirements and allow the collaborative and refined definition of data sharing restrictions. Our further aim is to support the application of these extents at multiple levels of granularity and complex authorization delegation chains.

In our work, we aim to consider both objectives and resulting work in a holistic and integrative view. Thus, we aim to establish a connection between security policies and authorization extents, supporting reuse of common building blocks in both areas that allow complementary and synergistic action. By organizing our research results as data sharing and processing framework, our goal is to provide consolidated contributions and validate their practical application by performing implementation and evaluation in a range of plausible case scenarios.

In the following chapter, we present the related work in a broader scope and respectively position our work. In the subsequent chapter, we then present our preliminary results that support presented objectives. In the next chapter, we outline the further work, presenting a consolidated roadmap that leads towards the complete realization of these objectives and conclusion of the work.

2
Related Work

In this chapter, we review the work related to our proposal, concerning both the current and future publications. Considering that the main building blocks of the proposal relate to the security policy management and cloud and web services, the sections in this chapter are organized in subtopics to reflect both theoretical and practical motivation and provide a broader context of the proposed work.

2.1 Enterprise Integration

As information systems and processes have been dominantly used within intra-organizational context, the research and development activities in the previous decades have been focused on answering the challenges of intra-organizational integration. The primary motivator behind this scenario was the need of organizations to interconnect heterogeneous systems and technologies and implement a higher degree of automation. By recognizing the need to increase data quality, reuse, and its availability, organizations increasingly benefited from integration processes by achieving cost savings and improving productivity, as well [42].

By considering both horizontal and vertical perspectives, Giachetti [25] identified four integration degrees of an enterprise system, referring to *coordination*, *interoperability*, *data sharing* and *connectivity*. Based on organizational affiliation of these systems, integration approaches are frequently characterised as *intra-enterprise* and *inter-enterprise* integration scenarios [30, 16].

With the advent and availability of internet connectivity, the interest of organizations shifted towards inter-enterprise integration, engaging in cross-entity transactions categorized as *business-to-business* (B2B) and *business-to-customer* (B2C) collaborations [30]. Being based on processes that span across the boundaries of different entities, inter-service integration raised additional challenges in the terms of dependability, reliability, interoperability and security of its transactions. Kurz et al. identified *data*, *applications* and *business processes* as layers that designate these integrations [43].

Categorized as point-to-point connections, the challenges of B2B scenarios were frequently addressed with the systems that connected adjacent entities under a clear separation of concerns. However, the emergence of cloud-based deployment and service models extended operational environments of enterprises beyond their organizational or physical boundaries, enabling a new range of application and integration scenarios. In that context, data and applications can be hosted on a variety of disparate entities, where the responsibilities may interweave.

Two concepts establish cloud-based integration. *Cloud Service Integration* at technical level embraces various techniques for integration of cloud services. When implemented and deployed as *Cloud Integration Service*, they represent an integration technology provided as a cloud service [42]. This model is recognized in the industry as Integration-as-a-Service (IaaS) or Integration-Platform-as-a-Service (iPaaS) [64]. Pezzini and Lheureux identified the following common integration scenarios [64]:

(i) cloud to on-premises
(ii) cloud to cloud
(iii) on-premises to on-premises integration.

Considered from the perspective of the single organization, in both of these scenarios data and services can be hosted at organizational premises or outsourced to various third-party providers. This view hence extends the reach of traditional intra-enterprise integrations to the cloud services offered by third parties. Even when relying on third-party infrastructure, the organization still exhibits a significant degree of control over its resources, as they still reside in its tenant domain.

The same cloud-based scenarios can be observed from the perspective of inter-enterprise integrations, as well. In this case, the integration is performed between systems residing in different organizational domains, potentially hosted in a tenant environment at a third-party cloud provider. The complexity of this scenario can be further extended by transforming the role of a cloud provider from a passive entity to the integration service that acts as a central integration point, processing tenant data and managing interconnections in *many-to-many* manner.

One of the significant improvements that the cloud integration platforms bring is the ability to abstract and transparently handle the broad range of chained backend services and tasks, supporting their further reuse and specialization [64]. Integration platforms hence relieve their customers from complex tasks that include administration of various APIs, platforms, systems development and lifecycle management and enable them to stay more focused on their core business concepts and competencies [42, 65].

In the scope of our work, we take cloud integration platforms as a demonstrative use case for cross-entity integrations. By relying on existing building blocks, integration platforms reuse and extend a range of technologies, applying additional complexity and multilateral perspective that challenge the security of their workflows. In our work, we aim to revise these challenges and derive a framework that addresses underlying security issues by introducing controls and processes that enable multi-organizational, dynamic and process-aware management of security in integration workflows.

2.2 Cloud and Web services

Service-oriented computing, established around WS-* family of technologies based on the SOAP protocol and WSDL service description language have been subject to intensive standardization in the previous decade. As most of the efforts were directed to establish syntactic interoperability between different services, the research community recognized the need to introduce more advanced means of interoperability that allow automated service composition and interactions based on complex scenarios [55].

One of the initial contributions in this direction was proposed by Sycara et al. In their work Sycara et al. [96] identified and approached three categories of challenges that need to be addressed. These categories include 1) the representation of capabilities of web services and their matching with requested functionalities, 2) the specification of information that web service requires and provides, and 3) the description of interaction protocol and service-invoking mechanisms on different levels [96].
The vision presented by Sycara et al. suggested the realization of *semantic web services* (SWS), which should result from the integration of semantic metadata, ontologies, formal tools and web services infrastructure. McIlraith and Martin additionally envisaged SWS as a way to enable a broad range of automation tasks that include interoperation, execution monitoring, and recovery [55]. Some of the noticeable outcomes from these endeavors resulted in semantic web services (SWS) frameworks, including OWL-S, WSMO and WSDL-S [72].

While web services attained a broader adoption, mostly in the enterprise context, semantic web services did not gain a significant traction. Verborgh argues that the reliance on remote procedure calling (RPC) with resulting treatment of web as a simple black-box, followed by the lack of actually implemented use cases, contributed to the low SWS adoption [100]. Lanthaler and Gütl consider perceived complexity and status of disruptive technology to contribute to the lower acceptance of semantic web technologies among developers [44]. In our work we rely on vision of Verborgh, which suggests a *bottom-up* and *self-descriptive* approaches in building the descriptions of web APIs [100]. We furthermore aim to reduce the complexity of solution and dependability on particular tools by applying Linked Open Data practices and employing JSON-LD with its reusable cross-platform tool stack [85, 46].

Unlike SOAP and WSDL based web services, RESTful style gained significant adoption in recent years. This trend is observable by looking at ProgrammableWeb API directory[1], which in June 2016 hosted more than 9,375 REST and 2,470 SOAP declared APIs. The same trend can be confirmed retroactively by looking at the analysis of Bülthoff and Maleshkova from 2014 [13].

Although the RESTful architectural style introduced by Fielding [23] defined a strict model with resources serving as a key abstraction of information in Web APIs [22], many implementations do not straightly follow this model in practice. They rather

[1] http://www.programmableweb.com/category/all/apis

employ hybrid systems, which additionally expose functions as RPC-like constructs[2] [71]. This results in a great diversity among implementation interfaces, leading to the need to reestablish API *maturity models* [73].

We aim to apply our proposal on RESTful-based services, which today represent a de-facto primary mechanism of SOA interactions and resource exchange for automated agents on the web. Instead of establishing a new approach to service descriptions, we rely on existing architectures and protocols, focusing on novel capabilities that are critical for security in cross-entity authorizations and policy management.

In the domain of descriptions of RESTful APIs, we noticed that many approaches, such as Hydra by Lanthaler [45], WSMO-Lite or hRESTS by Roman et al. [72] do not consider security requirements at all. The work of Alarcon and Erik [1] and its further refinement in the form of ReLL-S by Sepulveda et al. [80] do tackle security, but in a way that applies high-level notations with low practical relevance.

In our proposal we aim to analyze and enhance these approaches, enabling their reuse and practical application in a broader context of web-scale authorization and security policy management.

Recently emerged initiatives such as OpenAPI [57], RAML[3] or Restlet[4], focus on delivering API management and generation functionalities [66]. We aim to rely on these specifications in automatic derivation and provision of service models that are integrated with security policies. Similarly, we aim to reuse and enhance the concepts of open vocabularies, such as Schema.org[5] and Core Vocabularies[6]

2.3 Access Control

Traditional access control models that gained broad adoption include Mandatory Access Control (MAC), Discretionary-based Access Control (DAC) and Role-based Access Control (RBAC) [76]. While MAC and DAC each consider access capabilities from the perspectives of organization or the user which owns the data, RBAC builds on centralized MAC view, introducing the concept of roles derived from the notion of organizational duties.

The standardized Core RBAC defines the concepts of *users*, *roles*, *objects*, *operations*, *permissions* and *sessions* [21, 35]. OrBAC model by El Kalam et al. [40] extends RBAC with an additional abstraction layer, mapping the concepts of *subject*, *action* and *object* into the notions of *role*, *activity* and *view* in the context-specific setting. It also extends the permissions in RBAC with *prohibitions* and *obligations*. The vast majority of systems today, even the ones present in distributed and cloud environments, conceptually depend either on these models or their derivations.

The evolution of distributed and federated computing imposed the need for access

[2] This is common for traditional web services
[3] http://raml.org/
[4] https://github.com/restlet/restlet-framework-java
[5] http://schema.org
[6] http://ec.europa.eu/isa/ready-to-use-solutions/core-vocabularies_en.htm

control models that enable the definition and evaluation of access control from the perspective of connected systems in multi-entity context. The models that rely on user identities, such as DAC or MAC, are not completely suitable for decentralized and distributed systems. In the general scenario, the user or its identity has to be known at the time of access definition and enforcement, which is not trivial to accomplish in dynamic, multi-domain environments. Furthermore, the user's identity itself can encapsulate more information than it is needed or allowed to accomplish the transaction, raising the issue of privacy and legislative conformance.

Attribute-based access control (ABAC) model shifted focus from user identities and applied an intensional approach that relies on the properties of principals. These properties are provided as attributes that can include beliefs about principals or serve as the basis for trusting these beliefs. Additionally, the attributes can be used to characterize the contextual conditions and requirements [77, 38]. It is generally assumed that ABAC can provide benefits from other models, such as DAC, RBAC or MAC, while overcoming some of their limitations. With their ABAC-α model, Jin et al. [38] provided contributions in that direction. UCON ABC family of models introduced the concepts of *obligations*, *conditions*, *continuous enforcement* and *mutability of attributes* [62] in usage control. The summary and recommendations over various contributions related to ABAC are provided in the guide published by NIST [31].

In our work we go beyond theoretical constructs and provide a framework that allows the application of these models in a multi-organizational environment. For this purpose, we refine RBAC and ABAC models and apply them in a range of inter-entity workflows for the purpose of integrated policy-based security management.

2.4 Security Policy Management

Policy-based management (PBM) is a paradigm that enables the separation of the rules that govern the behavior of a system from its functionality. Application of policies promises the reduction of maintenance costs in ICT systems while improving their flexibility and adaptivity in a dynamic manner [8, 84]. Security policies define security requirements for a given system [27]. Traditionally, they deal with *access control*, *information* and *availability* [78, 84].

On a more specialized level, *authorization policies* allow users to define a structured and reusable set of rules and requirements that are applied in the process of policy enforcement. Authorization policies are considered as enablers of effective data protection and access control [24, 74]. The separation of definition and enforcement of authorization policies allows for a greater degree of flexibility, traceability, and manageability, especially in complex and distributed environments.

In the domain of cloud-based services, Singhal et al. proposed a framework that allows dynamic, on-the-fly collaborations and resource sharing among different organizations [83]. Their framework envisages a range of proxy-based architectures for multi-clouds collaboration.

As Singhal et al. provide a high-level overview and analysis of potential security issues in multi-cloud collaborations, their work serves as a motivational basis for our work on collaborative policy management.

Jung et al. proposed a policy decision and enforcement framework for enabling usage control in cloud scenario [39]. Integrated into VMware virtualization environment at infrastructure service level [56], the proposed framework enforces context-aware policies by relying on the environmental information. As it currently focuses on a *reactive* behavior, Jung et al. plan to investigate preventive enforcement in the future [39]. Pustchi et al. applied multi-organizational perspective, describing the concept of authorization federation in IaaS cloud environments [67]. They focused on the formalization of the trust model for homogeneous peer-to-peer federations, providing the implementation for the OpenStack[7].

Multi-tenant access control for Intercloud proposed in recent work of Ngo et al. [59] relies on infrastructure description models to generate policies for dynamic objects from predefined policy templates.

The both of these contributions consider the management on an infrastructural level, with the work of Jung et al. and Pustchi et al. focusing on a setting of particular software platforms. Furthermore, these contributions do not assume collaborative and distributed policy management. Our work distinguishes by aiming to provide a more general approach that applies to different cloud service models, not being tied to the particular platform.

KAoS policy management framework proposed by Bradshaw et al. consists of a set of components and services for policy and domain management integrated through the three layers [12]. The first layer uses hypertext-like graphical interface, enabling policy specification in constrained English sentences. *Policy management layer* employs OWL [28] to encode and manage policy-related ontologies. This information is then used by Distributed Directory Service (DDS) to analyze and test policies. The third layer consists of policy monitoring and enforcement components that employ policy representations derived from OWL, optimized for efficient execution. KAoS supports *authorization* and *obligation* policies, integrating positive and negative actions for both sets. However, KAoS is mainly intended to network configuration and operation. It does not consider cloud requirement and SOA domain [68].

Ponder is a policy management framework developed at Imperial College. It consists of a Ponder policy specification language, general architecture, and policy deployment model. The framework categorizes security policies into *authorization*, *filtering*, *refrain*, *delegation* and *obligation* policies. The latter are used to support event-condition-action (ECA) paradigm [3] by performing management actions. In a policy enforcement model of Ponder, policies are compiled by the Ponder compiler into Java classes and deployed in virtual machines at enforcement points. Following that, each enforcement point needs to implement a *policy enforcement interface* to enable the loading, management, and enforcement of policies. This setup, as well as

[7] Open-source software for cloud management at infrastructural level, `https://www.openstack.org`

the reliance on experts to define the policies, restricts the practical applicability of Ponder to enterprise-centered environments.

The work of Modica and Tomarchio [18] applies semantic technologies to annotate existing security policies with ad-hoc content. They designed a general ontology that defines main security concepts. Those concepts are applied in the process of matching between customer's and provider's security requirements and capabilities. In this work, they implemented the prototype relying on WS-Policy specification and tested it in two case scenarios. In contrast to this approach, in our work, we aim to provide a range of modular and granular ontologies for different domains. We furthermore aim to enable policy definition and cross-entity annotation that considers RESTful architectures, which represent a prevailing service paradigm in the web.

XACML is an XML-based declarative language standardized by OASIS, providing the means to specify access control policies based on an extensive set of built-in data types, functions, combining algorithms and supported profiles [69]. Primary elements of XACML are *rule*, *policy* and *policy set*. Its data flow model supports the separation of functions, encompassing a distributed architecture consisting of entities with clearly separated roles [104]. In this architectural model, the access to resources is protected by policy enforcement point (PEP), which implements access decisions provided by policy decision point (PDP), in cooperation with policy information point (PIP).

Compared to other models, XACML distinguishes by being standardized and broadly adopted. As its main use-case relates to intra-enterprise environments, the application in cross-domain context and related interoperability issues require non-trivial effort to be invested for broader adoption. Our work can be considered as orthogonal to XACML, as it aims to provide a policy translation approach that extends the application of XACML and its implementations beyond single environments.

Context-aware access control framework [97] and its adaptive policy model [98] proposed by Toninelli et al. use a combination of description logic and logic programming Their model treats context as a first-class principle, supporting the specification of authorization and obligation policies. It is however not clear how proposed approach implements policy enforcement and refinement in practice and what are its application domains. Obligation model does not foresee transformative measures. In our work, we aim to provide contributions in that direction by extending the understanding of the context with resource properties and applying them in the scope of enforceable transformative security controls.

In a recent survey of Kassem-Madani and Meier [41] the lack of languages to support privacy-utility tradeoff negotiations and agreements is observed. This potentially leads to policies resulting in binary decisions and limited flexibility. Tonti et al. found that existing policy specification solutions tend to diverge, being best suited for particular ranges of applications. From this point they identified a number of advantages of semantic web languages for policy representing and reasoning, identifying *expressiveness*, *analyzability*, *ease-of-use* and *enforceability* as critical aspects for their further development [99]. This view has been later confirmed by Bradshaw and Montanari [11].

In our work, we follow semantic-based approach to reach the goals of interoperability and rich expressivity in diversified environments. We rely on semantic technologies to allow machine-based understanding and process awareness that span across different resource and data layers, supporting the interactions and security management beyond the traditional syntactic boundaries.

2.5 Web Authorization

Web authorization management can be considered as application-specific subset of general security policy management domain, with additional refinements that apply to web-based interactions and particular scenarios that involve different entities and agents. In this proposal we describe it separately due to relevance to the current work.

OAuth 2.0 [29] represents a broadly adopted authorization framework aimed at enabling resource sharing in the web environment. In its typical scenario, OAuth 2.0 enables *clients* to access protected resources on a behalf of a *resource owner*. A client is a device agnostic term that refers to an application that accesses the protected resource hosted by *resource server*. A fourth entity in this setting is an *authorization server*, which issues access tokens to the client, providing that the authorization consent has been previously obtained by the resource owner. This protocol is dominantly adopted for web API protection [13].

User-managed access (UMA) is an emerging profile of OAuth 2.0, designed to provide individuals with a unified control point for authorizing the access to their personal data, content, and services [50]. The profile is based on previous work by Machulak et al. [48, 49], aimed at enabling an user-centered access management for resources hosted at various web applications. UMA introduces new actors and resource owner policies, as well as centralized authorization model and trust elevation based on claims gathering flow. It furthermore establishes two additional APIs that separately govern the access to the authorization server for clients and resource servers. Core protocol specifications include User-Managed Access Profile of OAuth 2.0 [52] and OAuth 2.0 Resource Set Registration [51].

Similarly, as OAuth 2.0, UMA depends on *access scopes* to constrain the extent of information that can be provided to clients. However, these access scopes are defined statically and tightly coupled with the provider service on a syntactic level. Aside from the primary security analysis of OAuth 2.0 that considers unintentional granting of too wide scopes [47], or a work of Shebab and Marouf that revises requested scopes using collaborative recommender system [81], no attention has been paid to the inherent nature of scopes, which are designated by unilaterally defined extents in various dimensions.

In our work, we aim to tackle this issue and propose a model that establishes access scopes that can be defined by different parties in a dynamic, reusable and semantically transparent manner. With this proposal, we aim to provide the means for access scopes that allow fine-grained, resource-specific constraints, supporting the *principle of the least privlege* [79] in cross-entity authorizations.

Birgisson et al. [6] recently proposed a mechanism for restricting delegations using *caveats*, the predicates that determine the context in which delegated credential may be used. Issued in the scope of *Macaroons* bearer credentials, the goal of this structure is to support decentralized delegation between principals. Macaroons are conceptualized to encompass *first-party* and *third-party* caveats, which are attached to the credential in a tamper-proof manner that enables *attenuation* and *contextual confinment* of authorization scope.
While first-party caveats allow target services to check request conformance to related predicates, third-party caveats enable additional flexibility by specifying any number of *holder-of-key proofs* that need to be satisfied for the request to be authorized. Based on that, additional requirements may be embedded in the macaroon, such as revocation-checking, extended authentication steps, anti-virus scanning or other activities. In its current instatiation Macaroons are syntactically tied to a service and out-of-the-band processes, drawing similar issues as OAuth 2.0 access tokens when it comes to semantic interoperability.

In our proposal, we apply mechanisms similar to attenuation and confinement. We, however, implement these mechanisms in an extent of a collaborative authorization process, where the resource owner inspects and refines requested authorizations. Based on resource owner policies, these authorizations can be established in an automatic and independent process that does not require owner's presence nor the knowledge of the accessing client. Furthermore, in our proposal we aim to establish confinement process that transparently ensures its properties across diverse systems.

In the following chapter we first present our contributions resulting from our previous work. Then, in the subsequent chapter we review the future work aimed towards the completion of the thesis.

3
Preliminary Results

This chapter introduces preliminary results of the research activities performed so far, presented in the topical order.

In the first section, we review our initial work which was focused on an assesment on existing literature, mechanisms and related challenges. This work establishes a motivational basis and building block for the work described in other section.

The second section introduces our concept on service integration framework. This work revisits security issues related to web-based authorization flows and proposes an initial framework consisting of a semantic vocabulary for general and selected domains, architectural and interaction model and an initial prototype. These components enable multi-party collaborative model-based service integration with the focus on data confidentiality.

In the third section, we present our concept of interoperable security policies that are defined and enforced in a separate layer, supporting policy management and definition across diverse entities.

The fourth section presents the refinement of the previously introduced model. It partially relies on ongoing work, which aims to establish deeper integration of policies and provider-specific data models and further consolidate the components.

Finally, we provide a summarized overview of relevant contributions.

3.1 Cloud Data Sharing Challenges

In our initial research activities, we examined the security of service and system integrations in distributed environments. This work has been motivated by the need of public administrations to establish secure private cloud federations, enabling secure data exchange and inter-entity service consumption.

The first contribution [95] focused on investigating the aspects of access control, data and security policy languages, and cryptographic approaches that enable fine-grained security and data processing in semi-trusted and interconnected environments. We

examined a range of matching techniques and models in each category and identified gaps in their application for establishing secure private cloud federations in multi-organizational context.

In our second contribution [89] we examined the security of service and system integrations in cloud-based collaborative environments. For this purpose we refined the RMIAS framework [14], establishing a range of security requirements that consider the context of cross-organizational interactions. Based on these requirements and using the guidance provided by the Cloud Security Alliance [2] we identified supporting security controls and derived their features. Following these criteria, we evaluated the capabilities of OAuth 2.0 [29], UMA [52] and XACML [70] for integration in cross-entity, heterogeneous, interoperable and delegated context.

The findings are summarized in Table 3.1. The analysis pointed at limited features of OAuth 2.0 and complementary capabilities of UMA and XACML. Although initially they were not designed for the same purpose, these approaches were selected based on their adoption level, capabilities and relevance for web and distributed context.

Being broadly adopted and the primary choice for cross-domain authorizations on the web, in our analysis OAuth 2.0 demonstrated a range of drawbacks that hinder optimal manageability of security. The first of them is the access scope, a construct that enables consent-based resource sharing. As we further pointed in [90, 91], this construct is, arbitrary and solely defined by the service provider using out-of-the-band processes, without considering the perspectives of other actors, such as a resource owner (service subscriber) or an accessing client.

Other properties of an access scope include static, non-standardized approaches to its definition, coarse-grained permissions, and detached semantics, as we elaborated in [89, 90, 91]. Together, they introduce some issues in security management. First, by coupling the scope structure and its semantics to a particular environment, its authorization extent must be agreed and implemented on a per-case and out-of-the-band basis, requiring additional integration and maintenance overhead. This is especially notable in one-to-many scenarios, as each client has to integrate and maintain different implementations for each service provider. Due to non-derivable semantics, the automated security management in such environments is hindered as well. Finally, the confidentiality of data can be only partially ensured. This can be observed through the suboptimal conformance to the *principle of least privilege* [79], which can be only partially supported by unstructured and static scopes.

Although it has been designed for intra-enterprise scenarios, with security policies as a primary concern, other capabilities of XACML demonstrate its relevance for the domain of inter-organizational authorization management. This can be observed from the development of UMA [49, 52], which partially adopted distributed architecture of XACML and concepts introduced in ISO/IEC 10181-3:1996 [36]. Based on this, UMA integrates security policies and enables their enforcement and evaluation on distributed infrastructure consisting of resource and authorization servers.

Despite its support for access models beyond the DAC-resembling [76, 75] consent

Table 3.1. Protocol support for security controls [89].

Control \ Protocol	OAuth 2.0	UMA	XACML
Access Control Models	∼	✓	✓
Access Granularity	∼	✓	✓
Accountability	✗	✗	✗
Auditability	✗	✗	✗
Authentication of Actors	✗	∼	✗
Contextual Awareness	✗	∼	∼
Cross-Domain Flows	∼	✓	✗
Data Transformation	✗	✗	∼
Delegation	∼	∼	∼
Legal Awareness	✗	✗	✗
Data/Process Integrity	✗	✗	✗
Resource Management	✗	∼	✗
Security Policies	✗	∼	✓
Transaction Integrity	✗	✗	✓

Not supported: ✗ Partially supported: ∼ Supported: ✓

and capability-based authorization in OAuth 2.0, UMA still lacks the structured and reusable data properties for inter-organizational service compositions.

As we have observed in [89], by adopting and refining out-of-the-band processes from OAuth 2.0, UMA inherits limitations in the terms of access scopes and resource and policy management. As the format and semantics of security policies are omitted, their extent and actual capabilities are left to particular implementations. This further facilitates the diversity among solutions, hindering the cross-domain security management and process awareness for autonomous agents. Instead of being able to reuse uniform security policies and access scopes across different entities (or their basic constructs), users and clients still have to manage one-to-many definitions and implementations of such constructs across the whole integration chains and maintenance lifecycles.

The findings presented in these contributions partially serve as motivations for the further work and contributions presented in the following sections of this proposal.

Relevant publications:

[95] Bojan Suzic et al. 'Secure Data Sharing and Processing in Heterogeneous Clouds'. In: Procedia Computer Science 68 (2015). 1st International Conference on Cloud Forward: From Distributed to Complete Computing.

[89] Bojan Suzic. Integration of Cross-Domain Distributed Systems: Approaches and Security Challenges. Accepted as a short paper at 24th Euromicro Intl. Conference on Parallel, Distributed, and Network-Based Processing (2016).

[37] Keith Jeferry et al. 'Challenges Emerging from Future Cloud Application Scenarios'. In: Procedia Computer Science 68 (2015). 1st International Conference on Cloud Forward: From Distributed to Complete Computing.

3.2 Service Integration Framework

In [93] we considered collaborative environments that host repetitive interactions between different entities operating in adjacent domains and jurisdictions. A typical scenario identifies three types of entities. *Service provider* delivers online services to its customers, typically realized in a cloud and corresponding to a particular cloud service layer [105]. By consuming these services, a *resource owner* partially outsources its business processes, data, processing or infrastructure to third parties. By involving in transactions with other subjects, resource owners may allow their *clients* to access or consume resources outsourced at service providers.

In order to enable resource-aware, granular and interoperable policy management, in our proposal we first identify the need to provide machine-readable descriptions of services taking part in collaborative transactions. The purpose of these descriptions is to provide a model of a service, its capabilities and supported interactions, which can be further reused for service and policy management in a multilateral context.

In the first instance, accessing clients can derive capabilities of service providers and available resources by inspecting exposed descriptions. These capabilities can be correlated to particular use-case needs and reused by the clients to voluntarily express their access requirements and acceptable restrictions on data sharing. In the second instance, users (resource owners) can rely on service and capability descriptions to define security policies applicable in the resource sharing process. In this work we focused on interactions that take place in the scope of RESTful architecture with cross-entity authorizations performed using OAuth 2.0 flows.

In the semantic framework introduced in [93] we modeled limited general vocabulary and two domain-specific vocabularies specialized for cloud email and storage services. In addition to service description vocabulary, our contribution includes the policy description vocabulary, which partially reuses the terminology defined in [104] and the concepts from XACML language [69]. Proposed sets support hierarchical resource representations and introduce the concept of *operations*, which serve both to express supported resource transformations and instruct their dynamic execution prior to resource delivery.

By relying on the provided semantic framework, collaborating parties use its common and domain specific vocabularies to describe their own resources and processes. The general interaction model encompassing these entities is presented in Figure 3.1.

In the step (1) involved parties use common and domain specific vocabularies to describe their resources and processes. Following that, a service provider models and exposes descriptions of its resources using these vocabularies, while client and resource owner rely on vocabularies to correlate representations and available concepts with their internal processes and data models. In the step (2) the resource owner and the client fetch the service model from the service provider. In the consenquent steps, the resource owner establishes security policies over resources exposed in the service model (3), while clients structure their requests and access the resources in the course of further interactions (4).

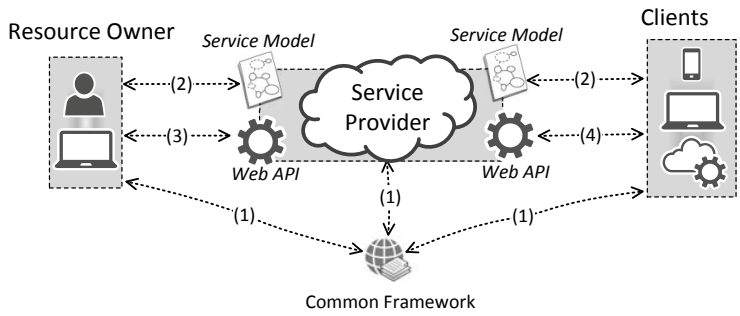

Figure 3.1. Interaction model and flows for resource-aware integrations.

Hence, the presented architectural framework assumes the existence of three structured entities: *service model*, *policy model* and *access request*. These entities are instantiated, fetched, extended or updated, depending on the subject and position in interaction flow. By reusing the concepts established in the common framework and following *any-to-one decentralized interoperability model* [102], they allow expressive and transparent integration of services and processes on the semantic layer.

In Figure 3.2 the sample model of an email service is depicted. The items in the picture are unlabeled for the purpose of simplification. The model reuses terminology established in the core and domain-specific vocabularies, instantiating descriptions of service capabilities as graph nodes with edges representing relationships between them. In the terms of Brachman et al. [9], these nodes can be considered as an ABox component of knowledge representation model. Using RDF's built-in *type* property [15] they are connected with base terminology classes, which correspond to a TBox component of the model.

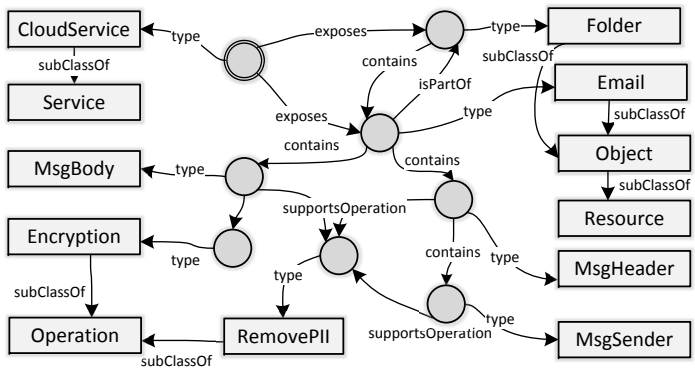

Figure 3.2. Service description for email model.

Being conceived as graph-based representations, the models that we introduced in [93] can be instantiated and exchanged between machines in various formats. While

typical instantiation might assume the use of RDF/XML serialization format, in our proposal we have applied JSON-LD representations [85]. The primary motivation behind that is to simplify the integration with existing systems that may be semantically-unaware or unwilling to participate due to perceived complexity or integration requirements [44]. Since JSON-LD maintains full compatibility with the JSON, the developers can continue to rely on preferred software and libraries with the benefit of added semantic layer that supports advanced expressivity, extensibility, and evolvability of a system [46]. On the other hand, the systems that utilize semantic technologies in a broader extent can directly read, instantiate and reason over models serialized using JSON-LD.

Figure 3.3 shows a sample access request created by a client agent, corresponding to the interaction step (4) on Figure 3.1, which is performed between a third-party client and a service provider. In this request, the client asks for an access to a particular resource type (email resource), stating the extent of an acceptable subresource (message header) and operation (dynamic removal of PII[1]), whose application still allows the client to fullfill its intended task. The resource types, their hieararhical representations and supported operations stem from the service model description, provided in the step (2) and visualized in Figure 3.2.

By expressing their access requirements and acceptable constraints, clients allow for a more fine-grained, collaborative and automated restriction of access authorizations. This supports the fulfillment of the *principle of least privilege* [79] in cross-entity resource sharing, allowing for a more fine-grained and confidentiality-aware transactions. Semantically annotated interactions additionally enable deeper inspection and reasoning over inter-organizational resource sharing. In this sense, we envisage the extension of existing or the potential emergence of new, automated solutions that, based on derived knowledge, perform automated security management across diverse platforms and systems.

```
{ "@context": {
  "dasp-email": "http://www.daspsec.org/on/dasp-email#",
  "owl": "http://www.w3.org/2002/07/owl#",
  "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
  "xsd": "http://www.w3.org/2001/XMLSchema#",
  "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
  "dasp-general": "http://www.daspsec.org/on/dasp-general#"
}, "@graph": [
  { "@id": "http://www.daspsec.org/on/dasp-general",
    "@type": "owl:Ontology", "owl:imports": [
      { "@id": "http://www.daspsec.org/on/dasp-general" },
      { "@id": "http://www.daspsec.org/on/dasp-email" } ] },
  { "@id": "dasp-general:AccessRequest",
    "@type": [ "dasp-general:Request", "owl:NamedIndividual" ],
    "dasp-general:acceptsOperation": { "@id": "dasp-general:RemovePII" },
    "dasp-general:acceptsSubtype": { "@id": "dasp-email:MsgHeader" },
    "dasp-general:requestsAccess": { "@id": "dasp-email:Email" } } ] }
```

Figure 3.3. Resource sharing request.

[1] Personally identifiable information

In our proposal we, therefore, aim to enrich existing REST APIs with lightweight, domain-specific descriptions that introduce additional security management layer in existing systems without imposing significant implementation or maintenance related overheads.

Although our model relies on semantic technologies to establish cross-entity interoperability and allow integration in workflows of autonomous agents, the conceived architecture and its integration in existing environments do not require advanced knowledge engineering skills or related infrastructure on the side of service providers or clients. This contrasts a typical top-down and full-fledged application of semantic technologies, whose resulting complexity and processing requirements hinder the practical application, as observed by Verborgh et al. [100]. From this standpoint, our work goes along with the recommendations of Verborgh et al., which suggests the building of Web APIs out of reusable blocks using self-descriptive, bottom-up approach.

In contrast to Resource Linking Language (ReLL) [1] or Hydra [45], our model is not primarily intended to support automated service traversal and extraction based on hypermedia constraints, but its aim is to provide structural and non-exhaustive description of exposed services that facilitates data, process and entity awareness across automated agents for the purpose of security management. Unlike ReLL-S [80], our proposal does not aim to describe existing security goals and mechanisms using high-level concepts, but to provide the means for cross-entity integration and implementation of different security mechanisms, including the novel ones, as we have presented in our work [93].

Relevant publications:

[93] Bojan Suzic. 'User-centered Security Management of API-based Data Integration Workflows'. In: 2016 IFIP/IEEE Network Operations and Management Symposium (NOMS). 2016.

[91] Bojan Suzic. 'Securing Integration of Cloud Services in Cross-domain Distributed Environments'. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. SAC '16. Pisa, Italy: ACM, 2016.

3.3 Interoperable Security Policies

In our initial contribution, we focused on the role of security policies in the definition and enforcement of enhanced access control in Web API-based transactions. Existing and broadly adopted web authorization frameworks exhibit different approaches to access control management. For instance, OAuth 2.0 does not assume the existence of security policies but relies on user's explicit consent communicated using access scope for access control enforcement. Its related UMA profile, however, introduces the notion of a *policy*, defining it as a set of configuration parameters at the authorization server that effect the access management of resources [52]. UMA, however, does not specify the format, model, and application of security policies, leaving most aspects to be provided by particular implementations. These missing aspects are covered in XACML

framework, which establishes a language for expressing security policies. Due to its orientation towards single enterprise, the applicability of security policies in XACML is typically restricted to intra-enterprise context. For the same reason, XACML does not tackle other processes relevant for inter-domain security management, as we have shown in [89]

In our proposal, we go a significant step further by providing the novel framework for specification and exchange of security policies that aims to be system and platform agnostic. Our goal is to enable systems residing in heterogeneous environments to expose their policy models, allowing the realization of policy management and reasoning functions using a separate layer, beyond the scope of a single platform. For this purpose, we define an architectural and interaction model, establish a semantic interoperability framework and provide a range of integration components. The results of this work were published in [93].

The proposed model of data sharing policies partially relies on the terminology defined by Westerinen et al. [104] and the concepts from XACML language [69]. It considers the *rules* as a basic building block, which are organized in sets of *policies*. Each rule states its *target* (object), action, subject and alternatively context and obligation. The decision over a policy set consisting of multiple rules is done by applying predefined *combinatorial algorithm*.

A high-level interaction model and architectural components, as conceived in [93, 87], are shown on Figure 3.4. This model assumes the policy management and enforcement to be performed using a gateway, which intercepts and evaluates the interactions between clients and service providers. This allows a transparent integration with existing systems, incurring a minimal implementation and deployment overhead on the service provider. The proposed management approach shares *any-to-one decentralized interoperability model* [102] and relies on components introduced in Section 3.2 for the purpose of service integration.

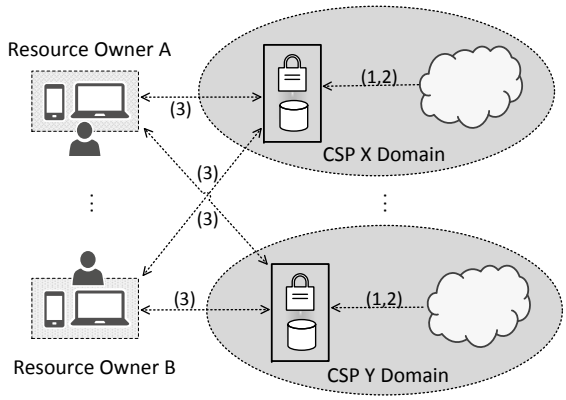

Figure 3.4. Policy specification.

Considering that many existing environments already rely on authorization management protocols, the proposed model supports the transparent integration of the broadly adopted OAuth framework. Serving as a second security enforcement layer, the gateway augments security capabilities of the underlying platform with fine-grained, dynamic and interoperable enforcement of policies, allowing the platform-independent policy management as well as context-based and client-bound online data transformation. Data transformation may be performed using functions internally implemented in the software component, or by relying on an external data transformation service, as we have presented in [94].

Security management workflow in the proposed framework is performed through the following steps:

(1) Defining and exposing the policy model
(2) Policy model discovery
(3) Policy generation
(4) Policy update

In the first step, the service provider establishes the policy model using an available semantic framework. A simplified sample model for the cloud storage provider is depicted on Figure 3.5. This model uses core and storage domain-specific vocabularies to instantiate a graph with nodes representing the supported subset of available classes and relationships. In the scope of policy model discovery (2), the instantiated model can be provided to the resource owner along with the resources, or by using a separate endpoint. In both cases, the granularity of representation is flexible and depends on the resource abstraction level. Furthermore, each of abstracted resources (or their ranges) is described and managed separately.

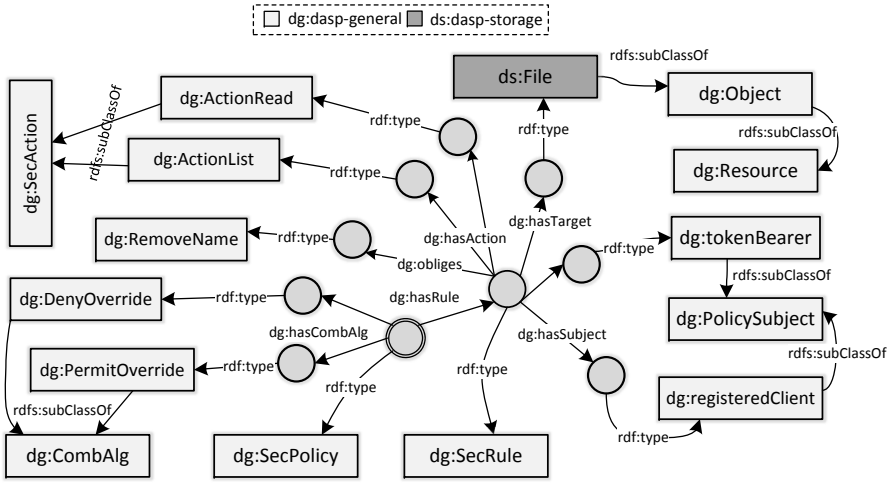

Figure 3.5. Policy model for cloud storage domain.

Policy generation (3) in the proposed model is performed on the user's side, using

clients such as a web browser or a mobile application. The policies are created by instantiating a subset of resources supported by the exposed policy model (1) and assigning them user-selected values. In the example model depicted on Figure 3.5, this definition would allow the specification of OAuth access tokens and granted actions for the requestor, as well as resource transformation prior to a delivery. In a more complex example, the policy can refer additional context (such as client's IP address, time or other constraints) or impose the execution of more complex obligations.

The final step, policy update (4), is performed as an HTTP PUT request to a relevant endpoint, using reserved request headers. From that point on, the gateway evaluates interactions between the service provider and the clients and applies the policies over server-generated responses. Hence, layered policy execution may restrict broad OAuth scopes [90, 91] or impose additional data processing independently of service provider capabilities [94].

Relevant publications:

[93] Bojan Suzic. 'User-centered Security Management of API-based Data Integration Workflows'. In: 2016 IFIP/IEEE Network Operations and Management Symposium (NOMS). 2016.

[94] Bojan Suzic and Reiter Andreas. 'Towards Secure Collaboration in Federated Cloud Environments'. In: Availability, Reliability and Security (ARES), 2016 11th International Conference on. (To appear.) IEEE. 2016.

3.4 Refining Policy Model

In the ongoing work [87, 92] we aim to extend our previous contributions [93, 94] by (1) applying refined and extended policy model, (2) introducing dynamic, instance-level based granularity and (3) enhancing policy enforcement with bidirectional support.

The revised conceptual model of policy vocabulary is presented in Figure 3.7. The refined policy model includes the possibility to dynamically reuse and reference data elements present in a target resource. Compared with our initial proposal [93], this allows the inclusion of additional, resource-specific parameters in security policies and their run-time evaluation according to the properties of target objects or their elements. Besides security rules and context representations, the target-driven parametrisation is enabled for specification of obligations that are executed along with policy decision.

The policy evaluation in the course of presented framework is shown in Figure 3.6. By extending our initial work [93], which supported the evaluation of service provider responses against security policies, in [87] we apply bidirectional evaluation, allowing the evaluation and transformation of client requests that may have a mutable effect on resources. These include requests such as HTTP PUT or POST, and additional declaration of GET requests that may have a mutable effect.

Typical policy evaluation in proposed framework is therefore performed in two phases, corresponding to steps (2) and (3) in Figure 3.6.

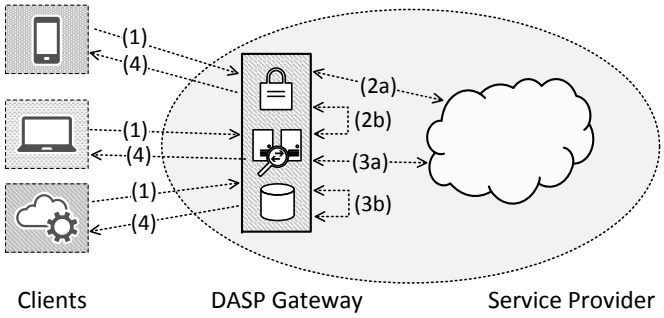

Figure 3.6. Policy execution.

In the first phase, the gateway checks if the client request triggers mutable action. If so, it is evaluated against security policies and necessarily transformed in the form of obligation. The policy-based evaluation allows restriction of requests that would be otherwise executed under too broad scopes, like OAuth-based requests. By integrating elements from terminology shown in 3.7, the policies may include complex contextual requirements that depend on diverse *intrinstic* or *extrinstic* properties[2].

After the client request is forwarded to the target service, and its response sent back to the client, the second evaluation phase is triggered, which corresponds to steps (3a) and (3b) on Figure 3.6. In this phase, the service provider response is evaluated against security policies. Depending on contextual requirements expressed in the policies and evaluation result, the service provider response may be again transformed to conform to user security and privacy preferences, including the *principle of least privilege* [79]. The transformation, executed as a *restriction*, ensures that the accessor receives the data for particular access context. In comparison with work introduced in [93], the refinement provided in [87] enables the parametrization of restrictions, allowing the execution of separate restrictions for each accessor or based on other property retrievable at a run-time.

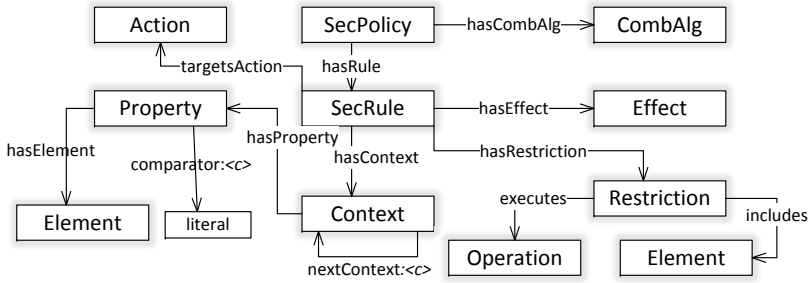

Figure 3.7. Terminology for policy definition.

[2] While intrinsic properties may reuse values of target service, resource, or its element, extrinsic properties may refer to the environment or accessor attributes.

Being the part of the semantic interoperability framework, the policies expressed using the security vocabulary (Figure 3.7) may rely on other parts of the framework, enabling cross-referencing and integration of security infrastructure with web services and resources on the semantic level. The example for such dependence is interaction in the step (2a), shown on Figure 3.6. Thus, if the policy depends on an intrinsic property of a target resource, such as the value of data field, then the gateway has to fetch the resource prior to the policy evaluation. This is performed automatically by relying on instances of *Selector* class from extended framework provided in [87]. These instances provide information on how the resource can be fetched, following hypermedia approach similar to one envisaged by Fielding [23] and later addressed by Sheth [82], Pedrinaci et al. [63] and Lanthaler and Gütl [45].

Relevant publications:

[93] Bojan Suzic. 'User-centered Security Management of API-based Data Integration Workflows'. In: 2016 IFIP/IEEE Network Operations and Management Symposium (NOMS). 2016.

[87] Bojan Suzic. 'Collaborative Policy Management and Enforcement for Cross-Domain Web Services'. (In preparation).

[92] Bojan Suzic. 'Structuring the Scope: Towards Integrated Multiorganizational Authorization Management'. (In preparation).

3.5 Summary of Contributions

This section presents a summarized overview of contributions introduced in this chapter. We group presented work to reflect the structure of this chapter. For details on each contribution, we refer to sections 3.1 to 3.4.

In our initial research activities, we examined the security of service and system integrations in distributed environments. Our first contribution [95] investigated the aspects of access control, security policy languages, and cryptographic approaches that enable fine-grained security and data processing in semi-trusted and interconnected environments. We examined potential techniques and identified gaps in their application for establishing secure private cloud federations in multi-organizational context.

In our second contribution [89], we defined security requirements and derived supporting security controls for cross-organizational interactions in cloud-based environments. We evaluated OAuth 2.0, UMA and XACML frameworks for adherence to these requirements and analyzed their capability to support derived security controls. This work identified a range of drawbacks of analyzed frameworks, which motivated our further research.

Based on the findings from our previous contributions [89, 95], in the further phase, we focused on service interactions established around RESTful interfaces that rely on web authorization frameworks. We identified cloud integration platforms (iPaaS) as a representative use case of complex service compositions that involve multiple entities.

In the first contribution [91] in this direction, we extended Apache Camel integration

framework to support interactions secured with UMA. Subsequently, we analyzed and discussed the security of integration flows of UMA and OAuth 2.0 frameworks in an iPaaS environment. While the both frameworks exhibited drawbacks in most of the analyzed categories, we demonstrated the better alignment of UMA with the security requirements. Following the outcome of this work, we identified the need to reconsider the inter-entity data and service integrations from the holistic point of view that allows resource, process and context awareness in distributed authorization management.

The following contribution [93] introduced the approach for modeling of services, policies, resources, and capabilities. These resources are exposed as the concept instantiations from the common interoperability framework, on an additional descriptive layer. The framework is reused and utilized among different entities to support service and policy management. Our work presented in [93] and [90] provides an architectural and interaction model, semantic vocabularies and a software prototype that implements and supports the proposed framework. The initial version of the prototype integrates with existing OAuth 2.0 deployments and performs unidirectional enforcement of security policies, supporting dynamic, policy-based and context-sensitive constraining of resources exposed using RESTful APIs.

Considering the integrations of services in the scope of private cloud federations, in [94] we presented the architecture and processes that establish cross-organizational management and enforcement of data security policies. This architecture enables policy-driven restriction and dynamic transformation of data flows in the federated environment, allowing both proactive enforcement and its post-executional conformance verification. Our contribution in this work includes the service and data description framework that enables fine-grained policy definition and enforcement in the federated environment. In the subsequent work [86] we refined existing policy model, enabling a more granular and expressive characterization of involved entities and interactions. This allows for a more expressive, context-sensitive, process and entity-aware definition and enforcement of policies. In this work we introduced the implementation prototype and validated the scalability of the architecture and policy evaluation components.

In the scope of the current activities, we work in two tracks. First, in [87] we extend our approach presented in [93] by refining policy model and advancing policy enforcement with the bidirectional support that relies on dynamic resource features. In the second work [92], we enhance the interaction model proposed in [93] and extend existing OAuth 2.0 and UMA protocol flows by allowing the resource owner to redefine client-requested authorizations and apply sharing constraints independently of cloud service provider. We furthermore apply this process to allow the users to perform consolidated security and authorization management of their resources at different providers.

4 Outlook

In this chapter, we present an overview of the work planned towards the completion of the Ph.D. program. We present the activities in the three terms and detail level related to their distance. The temporal reference point for planning is the completed proposal defense. This would account for a total time of approximately 30 months since the acceptance of the initial Ph.D. proposal[1].

4.1 Short Term (0 - 4 months)

Following our current activities, the first objective is to complete the ongoing work aimed at establishing an access scope definition that relies on and applies the concepts introduced in the prior work [93]. While in our previous work we investigated enhanced policy-driven information filtering using unidirectional proxy-gateway on OAuth-based flows, currently we work on establishing and applying novel token scope [92]. The aim is to support the flexibility of this structure, enabling it to be used both with existing OAuth or UMA flows and independently.

By integrating the existing service and data models, the new structure is meant to enhance existing authorizations with additional granularity and expressiveness, allowing service specific confinements that are horizontally interoperable and reusable across the services. In this contribution, we aim to focus on the consolidation and formal definition of the structure. We also plan to evaluate its application in different environments, protocols[2], and use-scenarios.

In our second ongoing contribution [87], we aim to complete the work that refines policy management flows described in Section 3.4. In the first instance, we plan to introduce bidirectional evaluations of mutable actions and integrate additional contextual requirements in the form of both intrinsic and extrinsic properties. The support for enhanced intrinsic property evaluation would enable referencing and

[1] May 2015

[2] Including integration with protocols such as OpenID Connect

reusing of resource-level data segments or features in security policy evaluation and enforcement, including transformative actions based on obligations [87, 94].

The expected outcome of these contributions are two papers and the extended components and libraries that are going be applied in the further work.

4.2 Mid Term (4 - 10 months)

In the course of mid-term planning, we intend to work on three additional contributions. In the first instance, we aim to extend existing policy decision engine. In the current configuration, the policy evaluation is performed by relying on Apache Jena[3] semantic framework for graph-based queries. We intend to investigate the applicability of OWL-based axioms [28] and SWRL/N3 [61] rules to achieve automated consistency check of policies and a higher degree of expressibility and policy automation. We furthermore plan to investigate the application of different reasoners, including EYE [17], and to evaluate their practical applicability for various scenarios and scalability requirements.

In the second instance, we aim to enable integration with existing XACML-based infrastructures [69]. Considering the acceptance of XACML for intra-enterprise policy definition and enforcement, we intend to enable the translation of policies specified using our framework and relying on JSON-LD [85] for translation to XACML-based policies in existing environments. This would enable service providers to enhance their infrastructures by transparently integrating with the proposed multilateral policy management approach.

In the third contribution, we aim to automate service modeling by allowing the integration of existing API generation and modeling frameworks, such as RAML or OpenAPI [57]. This would enable reuse and easy integration of existing API models. Furthermore, we aim to enhance the framework with additional crawling and reasoning capabilities, allowing semi-automated model generation based on current knowledge.

The expected outcome of these contributions are two-to-three publications and an extended set of existing and new software components that would bring the overall framework on a more mature level. Furthermore, during this phase we intend to provide an outline of the Ph.D. thesis, including particular chapters in advanced draft.

4.3 Long Term (10 - 14 months)

In a long term, we plan to apply up to two additional contributions, depending on the results that are going to be achieved during the execution of tasks scheduled for short and mid-term work.

One intended contribution is the extension of the components to provide intelligent user interface and browser component that allow easy, flexible and collaborative management of policies and tokens derived in the proposed short-term work. This

[3] https://jena.apache.org

contribution would furthermore encompass supporting components for clients and service providers to transparently and easily model their requests and services and integrate into existing environments.

A second potential contribution would be investigation of applicability of proposed solution in the scope of other platforms, such as mobile or IoT, as these platforms already support some of the aspects related to the current work[4].

Finally, during this phase of the work we intend to further consolidate and advance the overall framework for the practical application. Parallelly, we intend to intensify work on the final thesis, leading to its completion during the 14th month.

[4] Such as web authorization protocols

5 Other Relevant Aspects

This Ph.D. topic has been accepted and announced in May 2015.
The rest of this section provides additional information relevant to this proposal.

5.1 Courses

I have completed the following courses related to the Ph.D. programme:

1. 705.065 Angewandte Kryptografie 2
2. 700.011 Wissenschaftliches Arbeiten
3. 930.001 Fundamental and Applied Research: Third-Party Funding, Grant Proposals, Collaboration, Resources and Impact
4. 930.002 Inventions, Patents, and Technology Exploitation

5.2 Teaching

I have been involved in the supervision of three completed bachelor theses.
Currently I am involved in the supervision of one ongoing master's thesis.

5.3 Projects

I have contributed to the following national and international research projects:

- A-SIT
- SUNFISH
- Cloud for Europe
- eSENS
- STORK 2.0

In the scope of these projects I have contributed to the following deliverables:

Sunfish:

1. D2.1 State of the Art and Legal Aspects
2. D2.2 Requirement Definition and Threat Model
3. D4.1 Data Security Policy and SLA Definition Language
4. D4.4 Information Sharing Governance Model

Cloud for Europe:

5. D3.1 Standards, Normalization and Certifications Associated
6. D3.3 Public Administration Requirements and Market Vendor Offering

eSENS:

7. D6.2 Enterprise Interoperability Architecture n^o 1
8. D6.3 European Interoperability Reference Architecture

STORK 2.0:

9. D4.8 Final Version of Process Flows
10. D4.9 Final Version of Functional Design
11. D4.10 Final Version of Technical Design
12. D4.11 Final Version of Technical Specifications for the Cross Border Interface
13. D4.13 Final Version of Common Building Blocks
14. D5.3.1 Technical & Business Objectives and Specifications
15. D5.3.2 eGov4Business Go Live Planning
16. D5.3.3 eGov4Business Pilot Running Phase Planning
17. D5.3.5 eGov4Business Pilot Final Report

5.4 Relevant Publications

The following publications are relevant for this thesis proposal:

[95] Bojan Suzic et al. 'Secure Data Sharing and Processing in Heterogeneous Clouds'. In: Procedia Computer Science 68 (2015). 1st International Conference on Cloud Forward: From Distributed to Complete Computing.

[89] Bojan Suzic. Integration of Cross-Domain Distributed Systems: Approaches and Security Challenges. Accepted as a short paper at 24th Euromicro Int. Conference on Parallel, Distributed, and Network-Based Processing (2016).

[37] Keith Jeferry et al. 'Challenges Emerging from Future Cloud Application Scenarios'. In: Procedia Computer Science 68 (2015). 1st International Conference on Cloud Forward: From Distributed to Complete Computing.

[91] Bojan Suzic. 'Securing Integration of Cloud Services in Cross-domain Distributed Environments'. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. SAC '16. Pisa, Italy: ACM, 2016.

[93] Bojan Suzic. 'User-centered Security Management of API-based Data Integration Workflows'. In: 2016 IFIP/IEEE Network Operations and Management Symposium (NOMS). 2016.

[94] Bojan Suzic and Reiter Andreas. 'Towards Secure Collaboration in Federated

Cloud Environments'. In: Availability, Reliability and Security (ARES), 2016 11th International Conference on. (To appear.) IEEE. 2016.

[86] Bojan Suzic et al. 'Balancing Utility and Security: Securing Cloud Federations of Public Entities'. In: On the Move to Meaningful Internet Systems: OTM 2016 Conferences. Springer International Publishing, (2016). (In review)

[87] Bojan Suzic. 'Collaborative Policy Management and Enforcement for Cross-Domain Web Services'. (In preparation).

[92] Bojan Suzic. 'Structuring the Scope: Towards Integrated Multiorganizational Authorization Management'. (In preparation).

5.5 Other Publications and Reports

[19] DPSP Cluster WG. 'Challenges for trustworthy (multi-)Cloud-based services in the Digital Single Market'. 2016.

[90] Bojan Suzic. 'Multidimensional Security Policies'. Tech. rep. 2016.

[88] Bojan Suzic. 'e-ID in the Cloud with SCIM'. Tech. rep. 2015.

[106] B. Zwattendorfer et al. 'PaaSPort - A unified PaaS-Cloud Management Application avoiding Vendor Lock-In'. In: Proceedings of the 13th International Conference e-Society 2015. IADIS Press, 2015.

[107] B. Zwattendorfer et al. 'Secure Hardware-Based Public Cloud Storage'. In: Open Identity Summit 2013. Springer, 2013.

[108] B. Zwattendorfer et al. 'Sicheres Speichern in der Public Cloud mittels Smart Cards'. In: D-A-CH Security 2013.

Bibliography

[1] Rosa Alarcon and Erik Wilde. 'Linking data from restful services'. In: *Third Workshop on Linked Data on the Web, Raleigh, North Carolina (April 2010)*. 2010.

[2] C Alliance. 'Security guidance for critical areas of focus in cloud computing v3.0'. In: *Cloud Security Alliance* (2011).

[3] E Emanuel Almeida, Jonathan E Luntz and Dawn M Tilbury. 'Event-condition-action systems for reconfigurable logic control'. In: *IEEE Transactions on Automation Science and Engineering* 4.2 (2007), pp. 167–181. DOI: 10.1109/TASE.2006.880857.

[4] Andrew Bartels, John R. Rymer and James Staten. 'The Public Cloud Market is Now in Hypergrowth'. In: *Forrester Research* 24.4 (2014), pp. 1–27. URL: https://www.forrester.com/report/The+Public+Cloud+Market+Is+Now+In+Hypergrowth/-/E-RES113365.

[5] Saul J Berman et al. 'How cloud computing enables process and business model innovation'. In: *Strategy & Leadership* 40.4 (2012), pp. 27–35. DOI: 10.1108/10878571211242920.

[6] Arnar Birgisson et al. 'Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud'. In: *Network and Distributed System Security Symposium*. 2014. DOI: 10.14722/ndss.2014.23212.

[7] Irena Bojanova, George Hurlburt and Jeffrey Voas. 'Imagineering an Internet of Anything'. In: *Computer* 47.6 (June 2014), pp. 72–77. ISSN: 0018-9162. DOI: 10.1109/MC.2014.150.

[8] Raouf Boutaba and Issam Aib. 'Policy-based Management: A Historical Perspective'. In: *Journal of Network and Systems Management* 15.4 (2007), pp. 447–480. DOI: 10.1007/s10922-007-9083-8.

[9] Ronald J. Brachman, Richard E Fikes and Hector J. Levesque. 'Krypton: A functional approach to knowledge representation'. In: *Computer;(United States)* 10 (1983).

[10] Joseph Bradley et al. *Internet of Everything: A 4.6 trillion Public-sector Opportunity*. 2013. URL: http://internetofeverything.cisco.com/sites/default/files/docs/en/ioe_public_sector_vas_whitepaper_121913final.pdf.

[11] Jeffrey M Bradshaw and Rebecca Montanari. 'Policy-based governance of complex distributed systems: What past trends can teach us about future requirements'. In: *Engineering Adaptive, Dynamic, and Resilient Systems*. CRC Press - Taylor & Francis, 2014, pp. 259–284.

[12] Jeffrey M Bradshaw et al. 'The KAoS policy services framework'. In: *Proc. 8th Cyber Security and Information Intelligence Research Workshop*. 2013. URL: http://jeffreymbradshaw.com/publications/CSIIRW%20KAoS%20papers.pdf.

[13] Frederik Bülthoff and Maria Maleshkova. 'RESTful or RESTless–Current State of Today's Top Web APIs'. In: *European Semantic Web Conference*. Springer. 2014, pp. 64–74. DOI: 10.1007/978-3-319-11955-7_6.

[14] Yulia Cherdantseva and Jeremy Hilton. 'A reference model of information assurance & security'. In: *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. IEEE. 2013, pp. 546–555.

[15] Richard Cyganiak, David Wood and Markus Lanthaler. 'RDF 1.1 concepts and abstract syntax'. In: *W3C Recommendation. Feb* (2014).

[16] Li Da Xu. 'Enterprise systems: state-of-the-art and future trends'. In: *IEEE Transactions on Industrial Informatics* 7.4 (2011), pp. 630–640.

[17] Ben De Meester et al. 'Event-Driven Rule-Based Reasoning using EYE'. In: *Joint Proceedings of the 1st Joint International Workshop on Semantic Sensor Networks and Terra Cognita and the 4th International Workshop on Ordering and Reasoning*. Vol. 1488. CEUR Workshop Proceedings. Oct. 2015. URL: http://ceur-ws.org/Vol-1488/paper-08.pdf.

[18] Giuseppe Di Modica and Orazio Tomarchio. 'Matchmaking semantic security policies in heterogeneous clouds'. In: *Future Generation Computer Systems* 55 (2016), pp. 176–185.

[19] DPSP Cluster WG. *Challenges for trustworthy (multi-)Cloud-based services in the Digital Single Market*. Jan. 2016.

[20] Terence Eden. *BMW i Remote API*. 2016. URL: https://github.com/edent/BMW-i-Remote/ (visited on 01/07/2016).

[21] David F Ferraiolo et al. 'Proposed NIST standard for role-based access control'. In: *ACM Transactions on Information and System Security (TISSEC)* 4.3 (2001), pp. 224–274.

[22] Roy T Fielding and Richard N Taylor. 'Principled design of the modern Web architecture'. In: *ACM Transactions on Internet Technology (TOIT)* 2.2 (2002), pp. 115–150.

[23] Roy Thomas Fielding. 'Architectural styles and the design of network-based software architectures'. PhD thesis. University of California, Irvine, 2000.

[24] Cédric Fournet, Andrew D Gordon and Sergio Maffeis. 'A type discipline for authorization policies'. In: *Programming Languages and Systems: 14th European Symposium on Programming, ESOP 2005*. Springer Berlin Heidelberg, 2006, pp. 141–156. DOI: 10.1007/978-3-540-31987-0_11.

[25] Ronald E Giachetti. 'A framework to review the information integration of the enterprise'. In: *International Journal of Production Research* 42.6 (2004), pp. 1147–1166.

[26] Elastic.IO GmbH. *Hybrid Integration Platform*. 2016. URL: http://www.elastic.io (visited on 01/07/2016).

[27] Joseph A Goguen and José Meseguer. 'Security policies and security models. Security and Privacy'. In: *IEEE Symposium on Security and Privacy*. Vol. 11. 1982, p. 77.

[28] W3C OWL Working Group et al. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. 2012. URL: https://www.w3.org/TR/owl2-overview.

[29] Dick Hardt. 'The OAuth 2.0 authorization framework'. In: (2012).

[30] Wu He and Li Da Xu. 'Integration of distributed enterprise applications: a survey'. In: *IEEE Transactions on Industrial Informatics* 10.1 (2014), pp. 35–42.

[31] Vincent C Hu et al. 'Guide to attribute based access control (ABAC) definition and considerations (draft)'. In: *NIST Special Publication* 800.162 (2013).

[32] IFTTT Inc. *Connect the apps and devices you love with "if this, then that" statements*. 2016. URL: http://www.ifttt.com (visited on 01/07/2016).

[33] IFTTT Inc. *Expore and add IFTTT recipes*. 2016. URL: http://www.ifttt.com/recipes (visited on 01/07/2016).

[34] Zapier Inc. *Connect Your Apps and Automate Workflows*. 2016. URL: http://www.zapier.com (visited on 01/07/2016).

[35] ANSI INCITS. *INCITS 359-2004. Role-based Access Control*. 2004.

[36] ISO. *ISO/IEC 10181-3:1996 - Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework*. Standard. International Organization for Standardization, 1996.

[37] Keith Jeferry et al. 'Challenges Emerging from Future Cloud Application Scenarios'. In: *Procedia Computer Science* 68 (2015). 1st International Conference on Cloud Forward: From Distributed to Complete Computing, pp. 227 –237. ISSN: 1877-0509. DOI: http://dx.doi.org/10.1016/j.procs.2015.09.238.

[38] Xin Jin, Ram Krishnan and Ravi Sandhu. 'A unified attribute-based access control model covering DAC, MAC and RBAC'. In: *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer. 2012, pp. 41–55.

[39] Christian Jung, Andreas Eitel and Reinhard Schwarz. 'Enhancing Cloud Security with Context-aware Usage Control Policies'. In: *CloudCycle 2014 Workshop on Provisioning and Management of Portable and Secure Cloud-Services*. 2014. URL: http://cs.emis.de/LNI/Proceedings/Proceedings232/P-232.pdf.

[40] Anas Abou El Kalam et al. 'Organization based access control'. In: *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*. IEEE. 2003, pp. 120–131.

[41] Saffija Kasem-Madani and Michael Meier. 'Security and Privacy Policy Languages: A Survey, Categorization and Gap Identification'. In: *arXiv preprint arXiv:1512.00201* (2015).

[42] Michael Kleeberg, Christian Zirpins and Holger Kirchner. 'Information systems integration in the cloud: scenarios, challenges and technology trends'. In: *Future Business Software*. 00006. Springer, 2014, pp. 39–54. DOI: 10.1007/978-3-319-04144-5_4.

[43] Christian Kurz, Ewald Hotop and Günter Haring. 'Evaluation and characterization of business-to-business integration systems'. In: *4th International Conference on Electronic Commerce Research*. Citeseer, 2001, pp. 424–438.

[44] Markus Lanthaler and Christian Gütl. 'A semantic description language for RESTful data services to combat Semaphobia'. In: *Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on*. IEEE. 2011, pp. 47–53.

[45] Markus Lanthaler and Christian Gütl. 'Hydra: A Vocabulary for Hypermedia-Driven Web APIs'. In: *LDOW* 996 (2013).

[46] Markus Lanthaler and Christian Gütl. 'On using JSON-LD to create evolvable RESTful services'. In: *Proceedings of the Third International Workshop on RESTful Design*. ACM. 2012, pp. 25–32.

[47] Torsten Lodderstedt, Mark McGloin and Phil Hunt. *OAuth 2.0 threat model and security considerations*. Tech. rep. 2013.

[48] Maciej Machulak, Moren Lukasz and Aad van Moorsel. 'Design and Implementation of User-managed Access Framework for Web 2.0 Applications'. In: *Proceedings of the 5th International Workshop on Middleware for Service Oriented Computing*. MW4SOC '10. Bangalore, India: ACM, 2010, pp. 1–6. DOI: 10.1145/1890912.1890913.

[49] Maciej P Machulak et al. 'User-managed access to web resources'. In: *Proceedings of the 6th ACM workshop on Digital identity management*. ACM. 2010, pp. 35–44.

[50] E. Maler. 'Extending the Power of Consent with User-Managed Access: A Standard Architecture for Asynchronous, Centralizable, Internet-Scalable Consent'. In: *Security and Privacy Workshops (SPW), 2015 IEEE*. 2015, pp. 175–179. DOI: 10.1109/SPW.2015.34.

[51] Eve Maler et al. ‘OAuth 2.0 Resource Set Registration’. In: (2015).

[52] Eve Maler et al. ‘User-Managed Access (UMA) Profile of OAuth 2.0’. In: (2015).

[53] Maria Maleshkova, Carlos Pedrinaci and John Domingue. ‘Investigating Web APIs on the World Wide Web’. In: *Web Services (ECOWS), 2010 IEEE 8th European Conference on*. IEEE. 2010, pp. 107–114. DOI: 10.1109/ECOWS.2010.9.

[54] Benjamin McGrath and Robert P. Mahowald. *Worldwide SaaS Enterprise Applications 2015–2019 Forecast and 2014 Vendor Shares*. 2015. URL: http://www.idc.com/getdoc.jsp?containerId=252568.

[55] Sheila A McIlraith and David L Martin. ‘Bringing semantics to web services’. In: *IEEE Intelligent systems* 18.1 (2003), pp. 90–93.

[56] Peter M Mell and Timothy Grance. *SP 800-145. The NIST Definition of Cloud Computing*. National Institute of Standards & Technology, 2011. DOI: 10.6028/NIST.SP.800-145.

[57] Fathoni A Musyaffa et al. ‘Minimally Invasive Semantification of Lightweight Service Descriptions’. In: *IEEE International Conference on Web Services*. IEEE. 2016.

[58] Cisco Visual Networking. *Cisco Global Cloud Index: Forecast and Methodology, 2013-2018*. 2014. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf.

[59] Canh Ngo, Yuri Demchenko and Cees de Laat. ‘Multi-tenant attribute-based access control for cloud infrastructure services’. In: *Journal of Information Security and Applications* 27 (2016), pp. 65–84.

[60] Sukhvir Notra et al. ‘An experimental study of security and privacy risks with emerging household appliances’. In: *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE. 2014, pp. 79–84.

[61] Martin O’connor et al. ‘Supporting rule system interoperability on the semantic web with SWRL’. In: *International Semantic Web Conference*. Springer. 2005, pp. 974–986.

[62] Jaehong Park and Ravi Sandhu. ‘The UCON ABC usage control model’. In: *ACM Transactions on Information and System Security (TISSEC)* 7.1 (2004), pp. 128–174.

[63] Carlos Pedrinaci et al. ‘iServe: a linked services publishing platform’. In: *CEUR workshop proceedings*. Vol. 596. 2010.

[64] Massimo Pezzini and B. J. Lheureux. ‘Integration platform as a service: moving integration to the cloud’. In: *Gartner* (2011).

[65] Martin Potočnik and Matjaz B Juric. ‘Integration of SaaS using IPaaS’. In: *The 1st International Conference on CLoud Assisted ServiceS*. 2012, p. 35.

[66] Tomás Procházka. *Model-Driven Development of REST APIs*. 2015. URL: https://brage.bibsys.no/xmlui//bitstream/handle/11250/2353592/13585_FULLTEXT.pdf.

[67] Navid Pustchi, Ram Krishnan and Ravi Sandhu. 'Authorization federation in IaaS multi cloud'. In: *Proceedings of the 3rd International Workshop on Security in Cloud Computing*. ACM. 2015, pp. 63–71.

[68] A. C. Riekstin et al. 'A Survey of Policy Refinement Methods as a Support for Sustainable Networks'. In: *IEEE Communications Surveys Tutorials* 18.1 (2016), pp. 222–235. DOI: 10.1109/COMST.2015.2463811.

[69] Erik Rissanen. *eXtensible access control markup language (XACML) version 3.0 OASIS standard*. 2012.

[70] Erik Rissanen et al. *Extensible access control markup language (XACML) version 3.0*. 2013.

[71] Carlos Rodríguez et al. 'REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices'. In: *International Conference on Web Engineering*. Springer. 2016, pp. 21–39.

[72] Dumitru Roman et al. 'WSMO-Lite and hRESTS: Lightweight semantic annotations for Web services and RESTful APIs'. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 31 (2015), pp. 39–58.

[73] Ivan Salvadori and Frank Siqueira. 'A Maturity Model for Semantic RESTful Web APIs'. In: *Web Services (ICWS), 2015 IEEE International Conference on*. IEEE. 2015, pp. 703–710.

[74] Pierangela Samarati and Sabrina De Capitani Di Vimercati. 'Access control: Policies, models, and mechanisms'. In: *Lecture notes in computer science* (2001), pp. 137–196. DOI: 10.1007/3-540-45608-2_3.

[75] Pierangela Samarati and Sabrina Capitani de Vimercati. 'Foundations of Security Analysis and Design: Tutorial Lectures'. In: *Foundations of Security Analysis and Design: Tutorial Lectures*. Springer Berlin Heidelberg, 2001, pp. 137–196. DOI: 10.1007/3-540-45608-2_3.

[76] Ravi S Sandhu and Pierangela Samarati. 'Access control: principle and practice'. In: *Communications Magazine, IEEE* 32.9 (1994), pp. 40–48.

[77] Fred B Schneider. 'Chapter 9: Credentials-based authorization'. In: *Untitled Textbook on Cybersecurity*. 2015. URL: https://www.cs.cornell.edu/fbs/publications/chptr.CredsBased.pdf.

[78] Fred B Schneider. 'Enforceable security policies'. In: *ACM Transactions on Information and System Security (TISSEC)* 3.1 (2000), pp. 30–50. DOI: 10.1145/353323.353382.

[79] Fred B Schneider. 'Least privilege and more'. In: *Computer Systems*. Springer, 2004, pp. 253–258.

[80] Cristian Sepulveda, Rosa Alarcon and Jesus Bellido. 'QoS aware descriptions for RESTful service composition: security domain'. In: *World Wide Web* 18.4 (2015), pp. 767–794.

[81] Mohamed Shehab and Said Marouf. 'Recommendation models for open authorization'. In: *IEEE Transactions on Dependable and Secure Computing* 9.4 (2012), pp. 583–596. DOI: 10.1109/TDSC.2012.34.

[82] Amit P Sheth, Karthik Gomadam and Jon Lathem. 'SA-REST: Semantically interoperable and easier-to-use services and mashups'. In: *IEEE Internet Computing* 11.6 (2007), p. 91.

[83] Mukesh Singhal et al. 'Collaboration in Multicloud Computing Environments: Framework and Security Issues.' In: *IEEE Computer* 46.2 (2013), pp. 76–84.

[84] M. Sloman and E. Lupu. 'Security and management policy specification'. In: *IEEE Network* 16.2 (2002), pp. 10–19. DOI: 10.1109/65.993218.

[85] Manu Sporny et al. *JSON-LD 1.0*. 2014. URL: https://www.w3.org/TR/json-ld/.

[86] B. Suzic et al. 'Balancing Utility and Security: Securing Cloud Federations of Public Entities'. In: *On the Move to Meaningful Internet Systems: OTM 2016 Conferences: Confederated International Conferences: CoopIS, ODBASE, and C&TC 2015. Proceedings*. Springer International Publishing, 2016.

[87] Bojan Suzic. *Collaborative Policy Management and Enforcement for Cross-Domain Web Services*. (in preparation). 2016.

[88] Bojan Suzic. *e-ID in the Cloud with SCIM*. Tech. rep. 2015. URL: https://pure.tugraz.at/portal/files/3621911/Studie_eID_SCIM_Cloud_1.0.pdf.

[89] Bojan Suzic. *Integration of Cross-Domain Distributed Systems: Approaches and Security Challenges*. Accepted as short paper at 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (2016). URL: http://demo.a-sit.at/.

[90] Bojan Suzic. *Multidimensional Security Policies*. Tech. rep. 2016. URL: https://pure.tugraz.at/portal/files/3621889/Projektbericht_MDSecPol_fin.pdf.

[91] Bojan Suzic. 'Securing Integration of Cloud Services in Cross-domain Distributed Environments'. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. SAC '16. Pisa, Italy: ACM, 2016, pp. 398–405. DOI: 10.1145/2851613.2851622.

[92] Bojan Suzic. *Structuring the Scope: Towards Integrated Multiorganizational Authorization Management*. (in preparation). 2016.

[93] Bojan Suzic. 'User-centered Security Management of API-based Data Integration Workflows'. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. 2016, pp. 1233–1238. DOI: 10.1109/NOMS.2016.7502993.

[94] Bojan Suzic and Reiter Andreas. 'Towards Secure Collaboration in Federated Cloud Environments'. In: *Availability, Reliability and Security (ARES), 2016 11th International Conference on*. (To appear.) IEEE. 2016.

[95] Bojan Suzic et al. 'Secure Data Sharing and Processing in Heterogeneous Clouds'. In: *Procedia Computer Science* 68 (2015). 1st International Conference on Cloud Forward: From Distributed to Complete Computing, pp. 116 –126. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.09.228.

[96] Katia Sycara et al. 'Automated discovery, interaction and composition of semantic web services'. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1.1 (2003), pp. 27–46.

[97] Alessandra Toninelli et al. 'A semantic context-aware access control framework for secure collaborations in pervasive computing environments'. In: *International semantic web conference*. Springer. 2006, pp. 473–486.

[98] Alessandra Toninelli et al. 'Proteus: A semantic context-aware adaptive policy model'. In: *Eighth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07)*. IEEE. 2007, pp. 129–140.

[99] Gianluca Tonti et al. 'Semantic Web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder'. In: *International Semantic Web Conference*. Springer. 2003, pp. 419–437. DOI: 10.1007/978-3-540-39718-2_27.

[100] Ruben Verborgh, Erik Mannens and Rik Van de Walle. 'Bottom-up Web APIs with self-descriptive responses'. In: *Proceedings of the First Karlsruhe Service Summit Research Workshop*. Feb. 2015. URL: https://biblio.ugent.be/publication/5939097/file/5939101.

[101] Ruben Verborgh et al. 'Semantic Description of REST APIs'. In: *REST: Advanced Research Topics and Practical Applications*. Springer, 2014, pp. 69–89.

[102] Guido Vetere and Maurizio Lenzerini. 'Models for semantic interoperability in service-oriented architectures'. In: *IBM Systems Journal* 44.4 (2005), pp. 887–903.

[103] Maja Vukovic et al. 'Riding and thriving on the API hype cycle'. In: *Communications of the ACM* 59.3 (2016), pp. 35–37. DOI: 10.1145/2816812.

[104] Andrea Westerinen et al. *Terminology for policy-based management*. Tech. rep. 2001. DOI: 10.17487/RFC3198.

[105] Qi Zhang, Lu Cheng and Raouf Boutaba. 'Cloud computing: state-of-the-art and research challenges'. In: *Journal of internet services and applications* 1.1 (2010), pp. 7–18.

[106] Bernd Zwattendorfer, Bojan Suzic and Gabriel Schanner. 'PaaSPort – A unified PaaS-Cloud Management Application avoiding Vendor Lock-In'. In: *Proceedings of the 13th International Conference e-Society 2015*. IADIS Press, 2015, pp. 223–230. ISBN: 978-989-8533-32-6.

[107] Bernd Zwattendorfer et al. 'Secure Hardware-Based Public Cloud Storage'. In: *Open Identity Summit 2013*. Springer, 2013, pp. 43–54.

[108] Bernd Zwattendorfer et al. 'Sicheres Speichern in der Public Cloud mittels Smart Cards'. In: *D-A-CH Security 2013*. 2013, pp. 120–132.