Another approach, which has been discussed in the scenario, is where a user can make an access control policy that is subject to approval by a custodian. In such setting, two different approaches can be considered. A custodian can either only restrict the policy further (i.e. the resulting access control policy can be composed of a subset of rules as proposed by an owner of a resource). In the second approach, a custodian can define access control policies at his own discretion. This can mean that a custodian can restrict policies proposed by an owner of a resource by deleting certain rules, expand those policies by introducing new rules or change those policies completely. In any case, how ownership of a resource is preserved needs to be considered.

2.6 Access Control for Internal and External Web Applications

2.6.1 Introduction

According to a recent survey published by McKinsey [27], majority of organisations have been using “Web 2.0” applications such as Web services, peer-to-peer networking, blogs, podcasts, and online social networks. Moreover, a significant majority plans to increase their investments in “Web 2.0” technologies in coming years. Among the main incentives is the ability to communicate to customers and business partners, to increase collaboration inside the company and to manage knowledge and share ideas of employees. “Web 2.0” applications are either developed and deployed internally within organisations or those applications are used in form of Software-as-a-Service (SaaS).

Applications such as Confluence Wiki Software [2], Wordpress blogging software [23] and platform [22], and Google Apps [5] are all used to help businesses be more competitive and agile on the market and to decrease the cost of their operation. Those applications have advanced features specifically tailored for intra-organisational use. The Confluence Wiki provides enterprise-level access control which allows to define fine-grained access control to Wiki pages. It also allows to plug in existing Identity Management systems. Wordpress supports plugins that can extend its core security functionality. Google Apps, which is provided in form of a service accessible over the Web, allows organisations to reuse their Identity Management systems as well (e.g. products such as OpenSSO [21] can be configured with minimum effort to work with Google Apps).

Even though specialised “Web 2.0” applications are being developed to support security features as required by organisations, such approach is far from being perfect. Building applications from scratch for the purpose of adjusting it to particular requirements of organisations is not desirable. An organisation may already use its own Access Control Management system and may wish to reuse its functionality for the SaaS-based applications. Such organisations may additionally wish to give their employees more flexibility in how access control rules are defined in order not to decrease the collaborative value of “Web 2.0” applications.
In the next section we present how a User-Centric Access Control can be used to allow organisations plug in their access control management systems into SaaS-based applications. We show how employees can be empowered to define access control rules for resources that they wish to share and how those rules can be constrained by organisational access control policies.

2.6.2 Scenario

Company XYZ is making excellent products and is gaining advantage over its competitors. With the increased production it is able to cut down its prices. Additionally, with a high quality of its products, its customer base grows rapidly. To further increase their sales, decrease the cost of production and extend their portfolio, the company decides to enhance collaboration with both customers and foreign designers. It wants to combine professional help from designers with highly valuable opinions of its customers. By doing that the company wishes to create a huge knowledge base upon which future ideas can emerge within the company in a collaborative manner.

In order to build a knowledge base, the company opts to use a Wiki. It chooses to install the Confluence Wiki to benefit from its enterprise-level functionality and ease of maintenance. Such knowledge base will be accessible by business partners, customers and employees of the company. To cut the costs of deployment and the total cost of ownership, the company decides to use such Wiki as a service and not install it internally.

In order to further support collaboration and collective intelligence of its employees and other parties, the company decides to allow sharing of short video clips, presentations and documents. The company wants such multimedia content to be shared via a fast and reliable service. Therefore, the company decides to use a dedicated video sharing and online storage services (YouTube and Office Live Workspace respectively).

In order to be able to define access control rules that would be partly based on legal and financial considerations, the company chooses to offload access control from all of the above mentioned applications to a specialised component - Security Provider. In such setting, the company is able to compose access control policies internally and apply them to a distributed set of Web resources. It is able to reuse a magnitude of possible digital identities which may require access to those resources as well. Moreover, a different entity than the owner of resources can be responsible for access control for those resources.

In order to allow its employees (users) to compose their access control policies and still constrain those policies with organisational rules, the company decides that it will configure each of its Web applications to offload authorisation to both of the following Security Providers:

1. Employee’s Security Provider - such SP component will allow an employee, who acts as a user and an owner of resources, to define access control policies for resources. An employee is believed to have a proper knowledge about collaboration requirements and can share a resource quicker and in
many cases more effectively. Access control rules, as defined by employees, will be subject to restrictions that will be imposed by the Company’s Security Provider.

2. Company’s Security Provider - such security provider will be configured to ensure that resources are shared taking legal and financial considerations into account. Moreover, it will impose constraints on access control rules specified by employees to prevent data leaks.

The company decides that it will not use external SaaS-based Security Providers as none of them meets their requirements. It decides to build and deploy both components in-house. A different Security Provider is used by company’s IT specialists and a different one is used by employees.

Both Security Providers that are built by the company allow to control a magnitude of resources hosted by a distributed set of Web applications. IT Security Specialists are able to define policies based on legal and financial requirements. Such policies are subject to compliance checks. Moreover, IT Security Specialists are responsible for composing access control policies that will impose constraints on other security policies. Employees, who act as users of Web applications, on the other hand, are presented with an intuitive graphical interface which allows them to define access control policies with minimum effort. Both security experts and users are able to retrieve information from multiple distributed Identity Providers when defining access control rules.

All the “Web 2.0” applications that the company decides to use are then configured to offload its access control functionality to both company’s Security Providers. Access control decisions are then derived from applicable policies by those components and are enforced by each Web application. As Security Providers were developed internally, they are integrated with the company’s OpenSSO IAM solution. As such, a centrally located set of access control policies is applied to a distributed set of resources hosted by internal and external Web applications. Additionally, the company plugs in its internal Identity Provider to the entire system.

The collaboration between the company and its business partners and customers turns out to be very beneficial. The company is able to manage various projects easily. Its employees are able to define access control rules by themselves which results in a very efficient collaboration. Resources in form of video clips, presentations and documents can be shared very quickly so that ideas can be exchanged between distributed parties with minimum effort. Constraints imposed by the company’s Security Provider prevent any sensitive information from being leaked from the company. Moreover, they allow any access control rules that could possibly violate any company’s requirements to be spotted and prevented very quickly.

The company additionally establishes its own network on Facebook to further emphasise its willingness to collaborate with its customers on better designs. Customers are be able to join the network and comment on designs that are presented to them. Once customers become members of such networks, they are additionally given access to other resources hosted by other company’s
Web applications. No account synchronisation is needed as user attributes are retrieved directly from Facebook. In such setting, the Facebook acts as an Identity Provider and not as an application as it is only viewed as a source of user attributes that are later used by other Web applications.

Because both company’s Security Providers are able to retrieve user attributes from various Identity Providers, it is possible to allow business partners and customers to participate in creating the knowledge base very effectively. No registration is required from none of the participating parties as already used identities are sufficient to access company’s set of Web applications.

2.6.3 Architecture

The architecture for a User-Centric Access Control for the provided scenario is depicted in Figure 13.

![Figure 13: User-Centric Access Control in Organisational Access Control Restrictions - Architecture.](image)

Web applications can be configured to use multiple Security Providers to protect the same set of Web resources. When an access request is issued for a resource then an application contacts those SP components to obtain an authorisation decision. Based on the application, its configuration or other factors, the application may decide which decision is enforced in case decisions from multiple Security Providers are not consistent or even contradicting. The application may also decide to enforce a decision from one Security Provider if it does not violate any restrictions imposed by a different Security Provider.

2.6.4 Discussion

This scenario presents how a User-Centric Access Control can be used for authorisation of multiple applications hosted within and beyond a single administrative domain. Web applications offload authorisation to multiple Security
Providers which are controlled by different entities. Web applications must be able to resolve conflicts in access control decisions (similarly to an already presented scenario in Section 2.4). Users with different security knowledge (i.e. IT Security Specialists vs. Employees) have access to different Security Providers depending on their skills (see Section 2.3).

View of the actors presented in this scenario with regards to the generic architecture of a User-Centric Access Control (see Figure 2) is depicted in Figure 14. Presented diagram shows Security Providers (1a,1b), Users (2a,2b), sets of Web applications (3a,3b), Consumers (4) and Identity Providers (5a,5b).

Figure 14: User-Centric Access Control in Organisational Access Control Restrictions Scenario - View of the Actors.

Resources of a single user (e.g. employee) are scattered across multiple Web applications and security for those resources is controlled by multiple entities. A user who creates a resource is mostly concerned with how to share this resource. Such user has the best knowledge about the collaboration requirements (e.g. what are the other users who should have access to their resources) and is able to easily define access control rules according to those requirements. Security for those resources may be subject to additional constraints as imposed by other entities involved in the security management process (e.g. IT Security Specialists). As such, Web applications may use multiple Security Providers to obtain access control decisions which must be enforced.

The scenario presented in this section combines characteristics of some of the already discussed use case scenarios. Similarly to the use case presented in Section 2.4, multiple Security Providers are used for the same set of resources and the same set of Web applications. Moreover, as already presented in scenario in Section 2.3, those Security Providers can differ in their functionality and in the level of flexibility that they provide. Users who create and disseminate information may use Security Providers with easy-to-use graphical interfaces and recommenders that support them with policy composition process [30]. Users who are responsible for security within organisations may use more powerful
Security Providers.

Apart from combining already presented scenarios, this use case also demonstrates how a User-Centric Access Control can allow plugging in Identity Providers from distinct administrative domains and use attributes that come from those IdP components. As shown in Figure 13, different users of the same Web application are allowed to plug in their Security Providers that make use of Identity Providers from different domains. Attributes obtained from those Identity Providers are then used when composing access control policies.

An important aspect of the presented approach is the use of social networking applications not as containers for applications or resources but as Identity Providers. A user is able to establish links (connections) with other users of a social networking application. Then such user can define access control rules for various Web applications reusing such connections. A user can either reuse attributes of single users or connections with those users (Relationship-based Access Control [32]). An example would be when a user creates a network of his friends on Facebook and then creates an access control policy on his chosen Security Provider that allows his friends to view pictures stored by Picasa Web Albums. Once a friend is added or removed from the list on Facebook, the set of users able to view pictures stored by Picasa is updated automatically. No changes are required in the already composed access control policy that is stored and evaluated by a user’s Security Provider.

2.7 Access Control Using Opaque Tokens and Obligations

2.7.1 Introduction

Existing “Web 2.0” applications are as varied and complex as desktop applications and meet requirements of even most demanding Internet users. In many cases, those applications substitute their desktop equivalents and are used by professionals on a daily basis. For example, Web applications such as Picasa Web Albums [16] allow photographers to store pictures while applications such as Picnik [18] allow those pictures to be transformed online. Photographers do not have to store their pictures on their personal computers but can put them online and make them easily accessible from various devices. Moreover, they can make such pictures easily available to a wide audience. By having software available as a service, they can also work with those pictures without the need of any software maintenance.

It is often the case, that professionals may want to make profit on the work that they produce. By hosting it using Web applications they often wish to achieve two things. The first is that they want to make their work easily accessible and manageable. The second is that they can treat the Web as a potential channel through which their work could be acquired by interested parties.

Currently, however, it is virtually impossible to integrate popular services that aid professionals with their work with payment services that would allow those professionals to make profit. For example, integrating Picasa Web Albums and PayPal [14] or Google Checkout [6] services cannot be easily achieved. As