# Best of Two Worlds: Secure Cloud Federations meet eIDAS

Thomas Zefferer
A-SIT Plus GmbH
Seidlgasse 22
1030 Vienna, Austria
thomas.zefferer@a-sit.at

Dominik Ziegler
Know-Center GmbH
Inffeldgasse 13/6
8010 Graz, Austria
dominik.ziegler@tugraz.at

Andreas Reiter
IAIK
Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria
andreas.reiter@iaik.tugraz.at

*Abstract*—**The federation of information technology (IT) systems is a common approach to bundle capabilities and get the best results for all participants. Cloud computing and electronic identity (eID) are only two out of many domains, for which federated solutions have been a topic of scientific and corporate interest during the past years. Recently, the H2020 project SUNFISH has introduced a new cloud-federation approach called 'Federation as a Service' (FaaS). FaaS enables secure cloud federations, where data owners remain in full control of their data and workflows. In this paper, we identify shortcomings of the FaaS approach in terms of secure and reliable user authentication. In this sense, data security and protection mechanisms are only as good as the applied authentication measures. We solve this issue by proposing the integration of an existing pan-European federation of national eID systems into FaaS. This increases security guarantees of FaaS by using a trustworthy and liable eID solution that has a strong legal basis in the form of the EU eIDAS Regulation. A first successful implementation and deployment of the proposed solution demonstrates its feasibility and shows the great potential of combining federation solutions from the cloud domain and the eID domain.**

***Keywords-cloud computing, electronic identity, federation, security***

## I. INTRODUCTION

The federation of IT systems has become an important topic in both research and industry. In general, a federation consists of different entities agreeing on a certain standard of operation and thus achieving interoperability. The advantages are apparent: by joining a federation, each federation member can benefit from functionality of all other members of the federation.

One area with a high potential to benefit from federation concepts is cloud computing. If multiple clouds are interconnected, i.e. federated, users from one cloud can benefit from the combined functionality of the entire federation. For instance, two cooperating companies could agree to federate their private clouds in order to benefit from each other's infrastructure and data-processing capabilities. Despite its high potentials, the federation of clouds is a complex proposition, whose implementation raises several challenges, many of them related to data security. In the example given above, the two companies might certainly agree to exchange certain data with the other party, as part of using its infrastructure, but will naturally refrain from granting universal access to all data processed in their own private clouds. Maintaining control of own data is hence the key challenge in federated cloud environments.

Overcoming this challenge has been the goal of the European Union (EU) funded research project SUNFISH[1]. SUNFSIH has introduced the concept of Federation as a Service (FaaS). Its goal is to enable secure cloud federations that provide a higher degree of functionality by allowing the cross-cloud exchange of data, but are still able to meet security requirements of these data. The proposed FaaS concept and its implementation developed by SUNFISH achieve this.

The FaaS solution developed by SUNFISH bases on reliable policy-definition and policy-enforcement methods. These methods restrict data access to authorized users with the necessary assigned privileges. This approach requires users to be reliably identified and authenticated beforehand. Only if the identity of users is reliably verified by means of secure authentication schemes, policy-enforcement methods can work effectively. The FaaS solution developed by SUNFISH relies on the implicit assumption that end users are reliably authenticated within their home cloud-environment. Accordingly, all federation members build an implicit circle of trust and assume that all members act responsibly when authenticating its users. However, in certain scenarios, this assumption might be unrealistic. If a member cloud of the federation fails to authenticate reliably its users, restricting data access to users with certain roles or privileges is actually pointless.

In this paper, we propose a solution to this problem. Concretely, we propose to combine SUNFISH's FaaS solution with the federation of European national eID solutions defined by the European Union's eIDAS Regulation [11]. The eIDAS Regulation defines an interoperability solution to federate existing national eID solutions such as the Austrian Citizen Card [5], the Belgian eID card [2], or the Swedish eID [12]. This solution can be used to reliably identify and authenticate users across Europe. Backed by the EU eIDAS Regulation, this eID solution has a strong legal foundation. In addition, well-defined requirements and established governance processes guarantee an adequate level of security. The eIDAS-based federated eID solution can hence be regarded as ready-to-use

[1] http://www.sunfishproject.eu

building block that provides secure and reliable user identification and authentication. We propose to integrate this building block into SUNFISH's FaaS concept and implementation. We show how to combine the federation concepts behind FaaS and the eIDAS-based eID solution on conceptual and architectural level and demonstrate a working proof of concept.

Accordingly, the remainder of this paper is structured as follows. Section II provides relevant background information on SUNFISH's FaaS concept and the eIDAS-based eID federation. Section III then introduces in detail our proposal to combine eIDAS-based user authentication with SUNFISH's FaaS concept. Findings obtained during evaluation of the proposed solution are discussed in Section IV. Finally, conclusions are drawn in Section V.

## II. Background and Related Work

This section elaborates on the two baseline technologies combined in this work. While Section II.A focuses on the research project SUNFISH and its FaaS concept, Section II.B introduces the EU eIDAS Regulation and its definition of a secure eID federation across Europe.

### A. *SUNFISH: Secure Cloud Federations*

SUNFISH is a H2020 project funded by the European Union with the goal to enable secure data sharing and federation of clouds [10]. SUNFISH ensures that data owners maintain control of their data and can define flexibly who is allowed to access data for which purposes and to which extent. For this purpose, SUNFISH has developed FaaS, an extended and flexible data-security governance model, which supports a variety of scenarios. FaaS relies on classical eXtensible Access Control Markup Language (XACML) [7] based approaches, which are already widely used today to enforce data-access control [1] [4].

The core of the XACML enforcement model comprises several well-defined entities including the Policy Enforcement Point (PEP), the Policy Decision Point (PDP), the Policy Administration Point (PAP) and the Policy Information Point (PIP). This separation of components yields a clear assignment of responsibilities with regard to policy enforcement. The PEP is the contact point for applications and issues access-decision requests to the PDP. The responses respectively the decisions are then enforced by the PEP. The policy store itself is logically separated into the Policy Retrieval Point (PRP) and the Policy Administration Point (PAP). The flow for a usual policy-decision request starts at the PEP, where the request is generated, and is passed on to the PDP. The PDP gathers potentially missing attributes from connected Policy Information Points (PIPs) and retrieves a list of matching policies from the PRP. The evaluation result is returned to the PEP where the decision is finally enforced.

FaaS, as introduced by SUNFISH and described in detail by Suzic and Reiter [9], extends this generic XACML approach with additional components. Furthermore, it provides a concrete implementation of the XACML approach for federated cloud environments and closes identified gaps of the XACML specification. Figure 1 illustrates the FaaS concept developed by SUNFISH. Different cloud environments (each a member of the cloud federation) are modeled as so-called tenants. A special infrastructure tenant operates common services like the policy-decision service and the policy store. The Blockchain technology [13] [14] is used by the federation to guarantee integrity of stored policies.

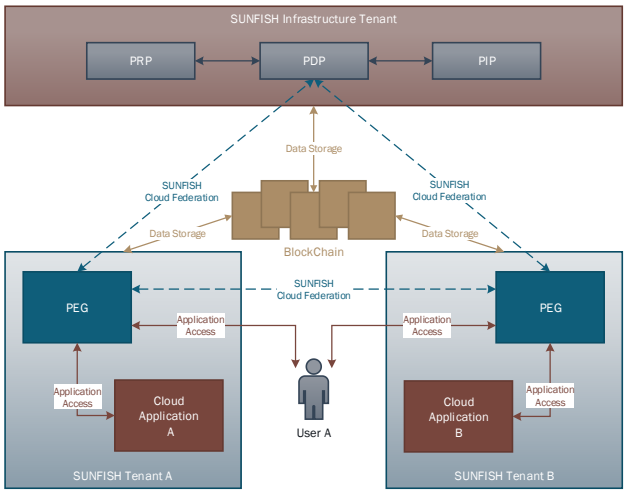


Figure 1. Basic building blocks of the FaaS concept.

All tenants dedicate computational resources to the federation, which are used to deploy and operate services and applications. The classical XACML model is extended by introducing the Policy Enforcement Gateways (PEG) entity. The PEG extends the responsibilities of the PEP by means of gateway functionality. PEGs are deployed at the edge of each tenant. Acting as a gateway, they protect the entire communication entering or leaving this tenant. The PEG analyzes the traffic and issues policy-decision requests to the PDP. Based on the policies defined by the data owner, a decision is derived, e.g., if an application from a certain tenant is allowed to access a service deployed in another tenant or if a user may access a particular service.

Summarizing, the FaaS concept developed by SUNFISH enables the federation of clouds while still leaving control of the data in the hands of the respective data owners. Apparently, the FaaS concept does not focus explicitly on the authentication of users. Instead, FaaS implicitly assumes that users are authenticated reliably in their respective home tenants and that user-specific policies can therefore be enforced reliably.

### B. *eIDAS: Secure Identity Federations*

The *Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market* (eIDAS Regulation) [11] is the legal basis for the provision of cross-border eID and trust services in the European Union. Although covering different kinds of trust services, the regulation has a strong focus on eID and on achieving interoperability between existing national eID solutions of EU Member States (MS). Essentially, the eIDAS Regulation defines an interoperability framework that enables European citizens to use an eID issued by MS X for

identification and authentication at a service provided by MS Y. Technical foundations of the eIDAS Regulation have been developed in several European large-scale pilots (LSPs) such as STORK[2] and STORK 2.0[3] [6].

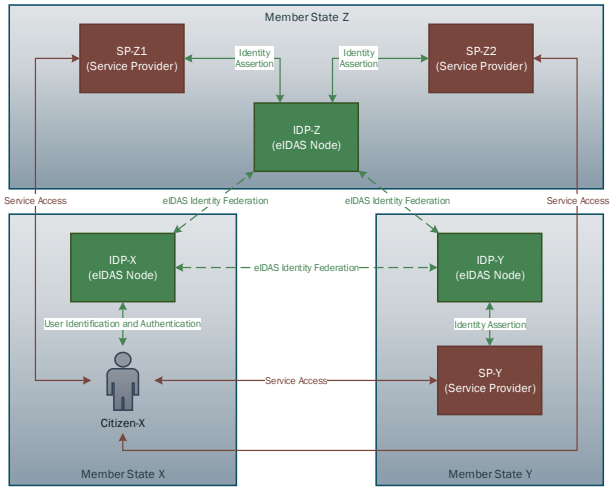


Figure 2. eIDAS-based eID federation.

The general architecture of the eIDAS interoperability framework is shown in Figure 2 by means of three exemplary EU MS. Each MS operates an own identity provider (IDP) denoted as eIDAS Node. Each national eIDAS Node is able to identify and authenticate citizens from the Node's own MS and accepts authentication requests from Service Providers located in the same MS. In order to allow citizens access to services in other MS as well, all eIDAS Nodes are federated, i.e. build a cross-border circle of trust.

The functionality provided by this eID federation is explained best by means of a concrete example. Assume that (according to Figure 2) a citizen from MS X accesses a service provided by MS Y, i.e. SP-Y. To identify and authenticate the citizen, SP-Y sends an authentication request to the eIDAS Node of MS Y, i.e. IDP-Y. As the citizen is from MS X, IDP-Y is unable to authenticate the citizen itself. Therefore, it sends an authentication request to the eIDAS Node of MS X, i.e. IDP-X. IDP-X identifies and authenticates the citizen, issues an identity assertion, and returns this assertion to IDP-Y. Due to the established eID federation, IDP-X and IDP-Y mutually accept issued identity assertions. This way, IDP-Y supplies SP-Y with a valid identity assertion, which is finally used by SP-Y to identify and authenticate the citizen from MS X.

In summary, the eIDAS interoperability framework assures that all European citizens can continue to use their existing national eIDs. At the same time, national IDPs only need to support their own national eID solution. Support for foreign eIDs is achieved by federating IDPs (eIDAS Nodes) of other MS. The technical interoperability framework defined by the eIDAS Regulation is currently being set up in EU MS.

[2] https://www.eid-stork.eu

[3] https://www.eid-stork2.eu/

## III. Proposed Solution

The main problem tackled in this paper is the potentially weak realization of user authentication in FaaS-based cloud federations. This section introduces a solution to this problem by extending the concept of FaaS with secure user authentication provided by the existing eIDAS-based eID federation.

### A. *Relevant Use Cases and Requirements*

Requirements to be met by the proposed solution have been derived beforehand from relevant use cases. Use cases and derived requirements are detailed in this section.

Identification of relevant use cases has been based on three general assumptions. First, we have assumed that each FaaS tenant features an IDP. Second, we have relied on the assumption that the IDP of a FaaS tenant is able to identify and authenticate users originating from this tenant. Third, we have assumed that applications deployed in a FaaS tenant can request the IDP of the same tenant to identify and authenticate users. All three assumptions can be regarded as realistic and comply with usual cloud deployments.

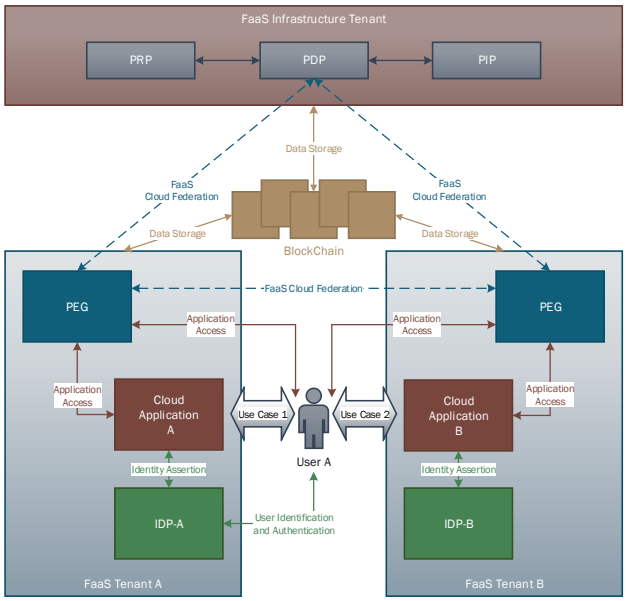


Figure 3. Relevant use cases

Considering the three assumptions made, yields the more detailed model of SUNFISH's FaaS concept shown in Figure 3. Note that Figure 3 still resembles Figure 1, i.e. shows the original FaaS concept. In addition, Figure 3 puts an additional focus on user authentication via IDPs. Figure 3 shows that two basic authentication scenarios can be distinguished, yielding the following two general use cases:

- **Use Case 1:** A user originating from Tenant A authenticates at a cloud application deployed in Tenant A. This means that user and application originate from the same FaaS tenant.
- **Use Case 2:** A user originating from Tenant A authenticates at a cloud application deployed in

Tenant B. In contrast to Use Case 1, user and application originate from different tenants in this use case.

The first use case resembles a classical setup of an identity management system consisting of a user, a service provider (application), and an IDP. The IDP authenticates the user on behalf of the service provider. In the context of SUNFISH's FaaS concept, the main challenge in this scenario is the FaaS-specific PEG component acting as gateway between the user and the application to be accessed. This component must be appropriately integrated into the authentication process.

The second use case is even more complex. In this use case, user and application originate from different tenants. Hence, there does not exist any IDP that is able to do both, receive authentication requests from the application and authenticate the user. Instead, two different IDPs are involved and need to interact in order to complete a successful user-authentication process. In addition, the same challenge as in Use Case 1 applies, i.e. the FaaS-specific PEG component needs to be integrated appropriately into the authentication process.

From the two sketched use cases, two general requirements can be derived for the proposed solution:

- **Requirement 1:** User identification and authentication functionality must be integrated into FaaS-based cloud federations such that FaaS-specific architecture components like the PEG are adequately addressed.
- **Requirement 2:** IDPs deployed in different tenants must be federated, in order to assure that users can identify and authenticate at applications deployed in other tenants.

Obviously, Requirement 2 suggests reliance on an established eID-federation solution. In Europe, such a solution is defined by the eIDAS Regulation and currently set up in EU MS. It is hence reasonable to employ the full potential of this approved solution and integrate it into the FaaS cloud federation model. The architecture that results from combining SUNFISH's FaaS concept with the eIDAS-based eID federation is introduced in the following section.

### *B. Architecture*

We have combined SUNFISH's FaaS concept with the eIDAS-based eID federation according to the architecture shown in Figure 4. Note that in addition to architectural components, Figure 4 also shows implementation-related entities to support a better understanding. These entities are elaborated in more detail in Section C. For the sake of simplicity, the architecture shown in Figure 4 sketches only two MS participating in the eID federation. In reality, the federation spans all EU MS that have successfully set up the eIDAS-based interoperability framework.

Figure 4 illustrates how the proposed solution meets Requirement 1 defined in Section A. In order to combine the two federation approaches, the solution transfers the conceptual role of the service provider from the cloud application to the PEG. Accordingly, the PEG interacts with the corresponding IDP and receives identity assertions issued by this IDP. This implies that the user authenticates at the PEG instead of the cloud application itself. Accordingly, the PEG can implement access-control mechanisms based on the user's confirmed identity. As a downside of this approach, the cloud application no longer receives identity assertions from the IDP. Consequently, the cloud application cannot use identity information contained in the assertion to provide identity-based features or to apply identity-based access-control schemes itself. To remove this drawback, the proposed solution foresees that the PEG supplies the cloud application with required identity and role attributes. This way, the cloud application benefits from the user's identity information without assuming the role of a service provider.

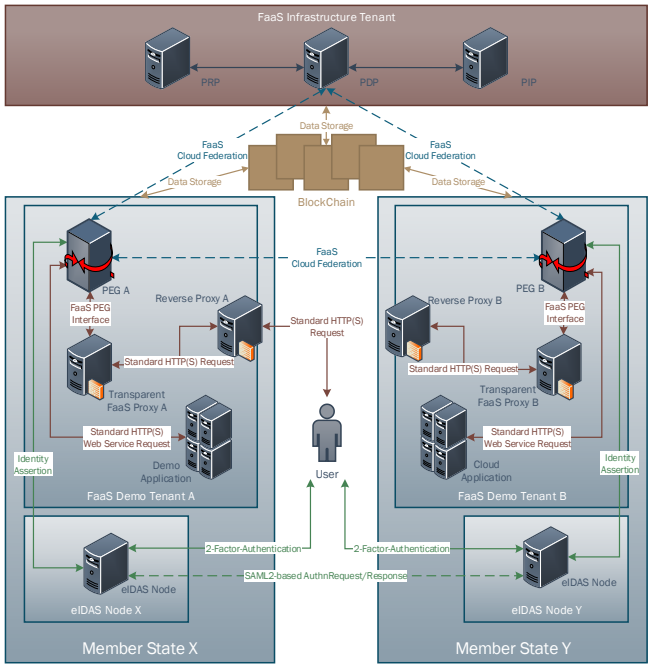


Figure 4. Architecture of the proposed solution

Requirement 2 defined in Section A is implicitly met by integrating the eIDAS-based eID-federation framework. Reliance on this framework and on eIDAS-compliant IDPs guarantees that users can authenticate at arbitrary cloud applications, independent from the tenant, in which the respective application is deployed. In theory, each PEG can access and use each IDP. However, in practice each PEG will most likely be able to communicate with one IDP only, i.e. with the eIDAS Node of the MS, in which the PEG itself is deployed. Note that this does not limit functionality, as all eIDAS-compliant IDPs are federated. Benefits of this federation also become apparent from the process-flow description provided in Section D.

### *C. Implementation*

Before elaborating on the proposed solution's basic process flow, we introduce a few implementation details to support a deeper understanding of the proposed solution. To evaluate its feasibility, we have implemented all relevant building blocks

of the proposed solution and deployed them in an evaluation environment. Details on evaluation results are provided in Section IV. Our implementation has been based on components developed by SUNFISH. The deployment spans across Europe by integrating the Austrian and Swedish eIDAS infrastructures with their already available eIDAS Nodes into the SUNFISH infrastructure. These countries have been chosen as their national eIDAS infrastructures are sufficiently set up. In addition to the eIDAS Nodes, each MS is assumed to host a computational FaaS tenant in this evaluation setup. An integral part of the FaaS model is the Infrastructure Tenant, thus also part of the evaluation setup. The Infrastructure Tenant hosts components related to policy evaluation and supports the PEGs of other tenants in enforcing defined policies.

In addition to the architecture, Figure 4 also shows some implementation-related components. When digging more into technical details, it becomes apparent that the PEG actually comprises the gateway components itself as well as two proxy components. First, the Reverse Proxy serves as a basic router to handle incoming HTTP(S) requests. This approach allows mapping multiple services to the same instance of the Transparent FaaS Proxy. Consequently, the Reverse Proxy can provide transport-layer security on a domain basis or load balancing for increased scalability and flexibility. Additionally, this entails that services can be migrated transparently to the FaaS model, without the need to change existing application configurations. Second, the Transparent FaaS Proxy serves as intermediate layer between legacy applications and the FaaS cloud-federation infrastructure. This enables an efficient integration of legacy applications into the FaaS infrastructure, without interfering with existing workflows. By aiming at a transparent deployment of services and by bridging the gap between legacy applications and managed cloud federations, a better acceptance of the overall system can be achieved.

### *D. Process Flow*

To bring the introduction of the proposed solution down to a round figure, a typical user-authentication process is sketched in this section. Focus is put on Use Case 2 as defined in Section A, since this use case is regarded more complex and thus more challenging.

Use Case 2 assumes that a user (User Y) from MS Y wants to access a cloud application located in FaaS Demo Tenant A. It is further assumed that the user can only be authenticated by eIDAS Node Y, while the accessed cloud application can communicate with eIDAS Node X only. This setup requires IDP X (eIDAS Node X) and IDP Y (eIDAS Node Y) to be federated, in order to complete a successful user-authentication process and to enable identity-based access-control mechanisms.

The overall process flow for this scenario is illustrated in Figure 5. In the beginning, the user triggers the process by requesting access to a Demo Application. User Y uses a client that is unaware of the underlying FaaS infrastructure. Hence, the user enters the public URL of the Demo Application. The FaaS infrastructure intercepts the user's request using the Reverse Proxy in FaaS Demo Tenant A (1).

The Reverse Proxy forwards the intercepted request to the Transparent FaaS Proxy located in the same tenant. The Transparent FaaS Proxy acts as adapter for legacy (FaaS-unaware) applications and integrates them with the FaaS policy-enforcement infrastructure (2). The Transparent FaaS Proxy transforms the original request into a format suitable for the FaaS infrastructure (3). This request is then forwarded to the PEG deployed in FaaS Demo Tenant A (4). The PEG issues an authorization request to the PDP (5). As User Y is not authenticated at that time, the PDP denies access to the requested Demo Application (6). Consequently, the PEG requests an identity assertion from eIDAS Node X (7). The eIDAS Node X initiates the user-authentication process by displaying a selection screen of all available and supported eIDAS-compliant IDPs (8). User Y selects the preferred eIDAS node (i.e. eIDAS Node Y) (9). Based on this selection, eIDAS Node X requests an identity assertion from eIDAS Node Y using the eIDAS-based interoperability protocol (10).

The eIDAS Node Y authenticates User Y by means of the user's national eID (11) (12). After successful verification of User Y's identity, eIDAS Node Y issues an eIDAS-compliant identity assertion (13) and forwards it to eIDAS Node X (14). The eIDAS Node X transforms the received assertion into its national format (15) and returns the transformed identity assertion to the PEG (16). The PEG extracts required identity attributes from the assertion (17), but still needs to verify whether the provided attributes are sufficient to grant access to the Demo Application. Thus, the PEG again issues an authorization request to the PDP, this time including available user information (18). If the data provided meets all defined access policies, the PDP grants access to the requested Demo Application (19). In the last step, the PEG forwards the initial user request to the Demo Application located in FaaS Demo Tenant A and includes available user information (20). This completes the user-authentication process.

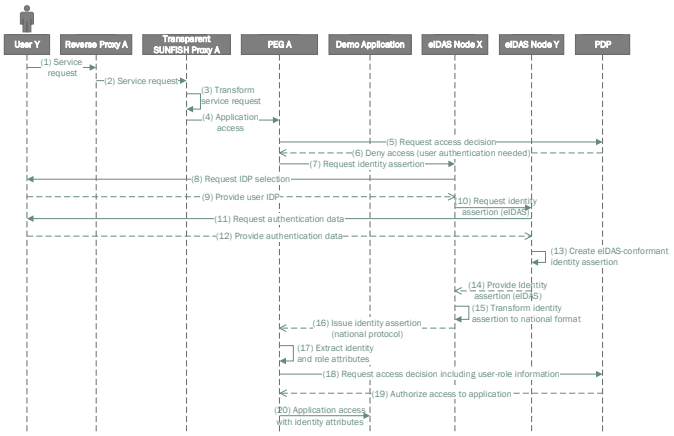


Figure 5. Process flow of Use Case 2

Note that the process flow for Use Case 1 is merely a subset of Use Case 2's process flow sketched above. The main difference is the lack of communication between IDPs, as in Use Case 1 the initially requested IDP is already able to authenticate the user. Thus, we do not elaborate on Use Case 1 in more detail at this point.

## IV. Evaluation

We have conducted a first evaluation of the proposed solution to verify its feasibility and to learn its benefits and drawbacks. Therefore, we have created a working prototype implementation. The prototype has been deployed and operated in an evaluation environment during an evaluation period. As expected, several lessons have been learned during this evaluation period. From these lessons learned, we were able to derive benefits and open issues of the proposed solution.

The conducted evaluation has shown that the proposed solution has indeed several advantages over SUNFISH's initial FaaS concept. Overall, the proposed solution provides a ***higher level of security***, as it takes away the responsibility of user authentication from members of cloud federations and assigns this task to a highly specialized and approved solution, i.e. the eIDAS-based eID federation. Since the eIDAS Regulation defines strict security requirements for eID solutions being part of this federation, the overall security of FaaS solutions is raised. Another advantage of the proposed solution is its ***strong legal foundation*** in the form of the eIDAS Regulation. This implies that responsibilities and liabilities are clearly defined on legal level, again leveraging the quality and reliability of the used eID federation. Its implementation and operation have shown that the proposed solution also provides the advantage of ***easy integration***. Due to its generic architecture, the eIDAS-based eID solution integrates smoothly into the FaaS concept. Information exchange between these two worlds, i.e. cloud federation and eID federation, take place via well-defined interfaces only, yielding a neat integrated solution. Finally, evaluation has shown that the proposed solution is in ***full conformance with SUNFISH's FaaS approach***. All concepts of FaaS and its underlying processing model can be left unmodified when integrating eIDAS-based user authentication. The eIDAS-based eID federation does not affect negatively functionality of FaaS-based cloud federations by any means.

In addition to these benefits, lessons learned during evaluation have also yielded some open issues. One challenge the proposed solution needs to face is the rather ***slow take-up of the eIDAS Regulation***. Although the regulation is already in effect, its implementation in the EU MS is still an ongoing process. Accordingly, it cannot be assumed that all EU citizens already have assigned a compliant eID issued by their MS. However, it can be expected that the situation will improve in future, as MS continue to implement the eIDAS Regulation. Another issue to be considered is the existence of ***decoupled eID systems***. In most corporate enterprises, proprietary eID systems are established and used. Users are not identified by means of their eIDAS-based eIDs (even if they have one), but by means of internal eIDs issued by the respective enterprise or institution. Integrating our proposed solution to such a scenario leads to two decoupled eID systems in place. In this case, means must be applied to map eIDAS-based eIDs to legacy eIDs used internally. Although such a mapping is expected to be technically feasible in most cases, it introduces additional effort.

Overall, the conducted evaluation has shown that the proposed solution is feasible and significantly improves the SUNFISH's original FaaS concept in several aspects.

## V. Conclusions

In this paper, we have proposed a solution that enhances SUNFISH's FaaS concept by integrating a highly secure and reliable pan-European federated eID solution, i.e. the eIDAS-based eID federation. We achieved an integration of eIDAS-based user authentication in full conformance with the original FaaS processing model. Our solution outsources the security-critical task of user identification and authentication to an approved solution that is backed by European law and applicable throughout Europe. This way, our solution eliminates a potential weakness of FaaS solutions, i.e. the weak implementation of user-authentication schemes, and enhances the security of SUNFISH's FaaS concept.

For future work, we plan to tackle remaining issues that we have derived from lessons learned during evaluation. In particular, we will work on solutions to problems that arise from the fact that the eIDAS-based eID solution might be decoupled from proprietary eID systems used in the respective members of the cloud federation. Despite these open issues to be addressed, we believe the proposed solution is mature enough to be integrated into FaaS-based cloud federations, in order to raise their level of security.

## References


[1] A. Bertolino, T. Y. Le, F. Lonetti, E. Marchetti, and T. Mouelhi. Validation of Access Control Systems. pages 210{233, 2014.

[2] A. Fairchild and B. de Vuyst. The Evolution of the e-ID card in Belgium: Data Privacy and Multi-Application Usage. In The Sixth International Conference on Digital Society, pages 13{16, Valencia, 2012.

[3] D. Hardt. The oauth 2.0 authorization framework. 2012.

[4] S. Kasem-Madani and M. Meier. Security and Privacy Policy Languages: A Survey, Categorization and Gap Identication. CoRR, page 13, 2015.

[5] H. Leitold, A. Hollosi, and R. Posch. Security Architecture of the Austrian Citizen Card Concept. In 18th Annual Computer Security Applications Conference, 2002. Proceedings, pages 391{400, 2002.

[6] H. Leitold, A. Lioy, and C. Ribeiro. STORK 2.0 : Breaking New Grounds on eID and Mandates. Proceedings of ID World International Congress, (Idm):1{8, 2014.

[7] B. Parducci and H. Lockhart. eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard, (January):1{154, 2013.

[8] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore. Openid connect core 1.0. The OpenID Foundation, 2014.

[9] B. Suzic and A. Reiter. Towards Secure Collaboration in Federated Cloud Environments. Workshop on Security, Privacy, and Identity Management in the Cloud, In Press, 2016.

[10] B. Suzic, A. Reiter, F. Reimair, D. Venturi, and B. Kubo. Secure Data Sharing and Processing in Heterogeneous Clouds. Procedia Computer Science, 68(316):116{126, 2015.

[11] The European Parliament and the Council of the European Union. REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identication and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, 2014.

[12] E. Wihlborg. Secure electronic identication (eID) in the intersection of politics and technology. International Journal of Electronic Governance, 6(2):143{151, 2013.

[13] D. G. Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Sante Publique, 28(3):391{397, 2016.

[14] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, 9. https://doi.org/10.1007/s10838-008-9062-0