Enhancing Keycloak: Implementing OpenID Connect for Identity Assurance

Brendon Vicente R. Silva[1], **Frederico Schardong**[1,2], **Ricardo F. Custódio**[1]

[1]Laboratório de Segurança em Computação (LabSEC)
Universidade Federal de Santa Catarina (UFSC)

[2]Instituto Federal do Rio Grande do Sul (IFRS)

Abstract. *Electronic identities are pivotal in a world where digital interactions are evolving fast. In this context, OpenID Connect, a cornerstone of user identification and authentication, plays an important role in the operation of electronic identification services. This paper describes a novel implementation of the OpenID Connect for Identity Assurance 1.0 (OIDC4IDA), an extension to OpenID Connect, on Keycloak, the leading open-source identity provider.*

1. Introduction

Preventing fraud, ensuring information integrity, and promoting data confidentiality have always been inherent concerns in computer security. From this perspective, the concept of electronic identities emerged, proposing not only the management of information stored in digital environments but also serving as a mechanism to enable secure communication between Service Providers (SP) and their users [Ferdous 2015, Miyata et al. 2006]. In this scenario, electronic identity protocols like OAuth 2.0 and OpenID Connect 1.0, used in large systems and renowned applications worldwide, were established to standardize and maintain best practices in the processes involved [Hardt 2012, Sakimura et al. 2014].

The OAuth 2.0 protocol establishes formal guidelines for virtual data sharing based on user authorization. It serves as the foundation for OpenID Connect, which introduces an additional identity layer. This layer defines new behaviors, flows, terminology, and standards crucial for electronic identity management. In order to maintain high-security standards, even the most well-established protocols must be continually updated to address new perspectives and use cases. Consequently, these specifications allow for documentation extensions to be proposed, enabling the integration of new functions.

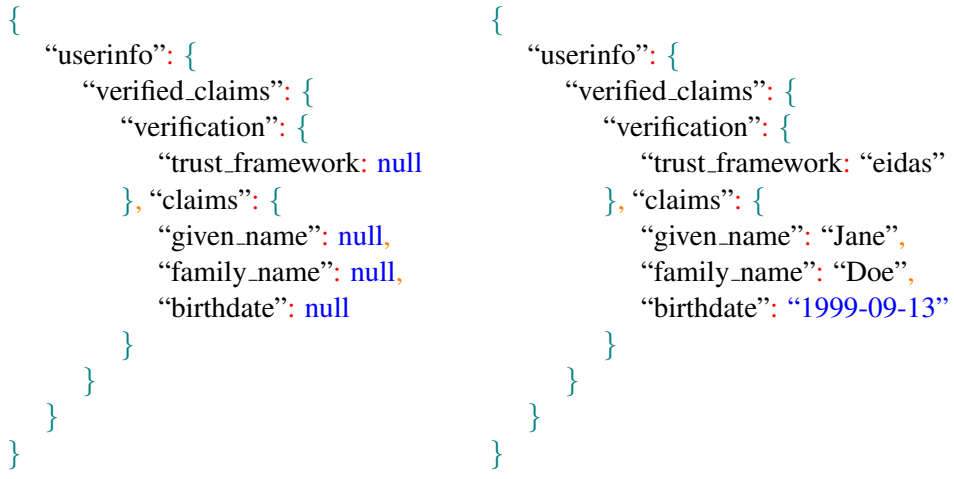
This paper describes the process of integrating OpenID Connect for Identity Assurance 1.0 (OIDC4IDA), a proposed extension to OpenID Connect, into Keycloak, one of the market's leading open-source identity provider systems [Lodderstedt et al. 2022, Keycloak 2024]. Alongside this, prominent aspects of the proposed solution are discussed, highlighting its peculiarities compared to other existing proposals. In this project, the objective is to build an open-source solution ensuring versatility and compatibility. Moreover, the implementation needs to be adapted for different use cases and ensure that the protocol's functionalities are fulfilled with integrity.

In this paper, Section 2 contextualizes the OIDC4IDA protocol, briefly describing its purpose and operation. Section 3 presents existing market solutions claiming to support the studied protocol, highlighting their strengths and aspects for improvement. Section 4 discusses the design decisions taken for the development of the proposed implementation,

considering the defined scope. Finally, Section 5 summarizes the topics covered, exposing the immediate results of the work developed and the expected future outcomes.

2. OpenID Connect for Identity Assurance 1.0

Officially proposed in March 2019, OpenID Connect for Identity Assurance 1.0 is a protocol that extends the functioning of OpenID Connect in an additive way. It brings mechanisms that enable the representation of supporting data complementary to the information in an identity [Lodderstedt et al. 2022]. In other words, the objective of this specification is to standardize the transmission of information that verifies the attributes of an electronic identity. This includes the regulation under which the Identity Provider (IdP) operates and under which the data was validated, as well as evidence regarding the collection, integrity, and verification of this information. Figure 1 shows an example of an OIDC4IDA simple full name and birthdate request, while Figure 2 exemplifies a possible response to it.



(a) Requesting verified data.

(b) Verified data response.

Fig. 1. OIDC4IDA request and response examples.

3. Related Work

The search for work related to this paper focused on libraries, tools, and IdP creation systems like Keycloak, specifically those claiming compatibility with OIDC4IDA. The research reveals two proprietary services that are monetized and thus do not fit directly into the proposed solution scope: the *Authlete* [Authlete 2024a] and *Connect2id Server* [Connect2id 2024] platforms. These are IdP creation services that, in addition to providing support for the protocol, offer free programming libraries that concentrate tools for implementing their functionalities. It is noteworthy that the provided libraries, although free, do not function as isolated systems and, therefore, require development effort to be effectively used. The analysis of these products was effective even outside the scope, as the cited libraries were used later in the proposed implementation.

Although it claims OIDC4IDA integration on Keycloak's system, the *eKYC Hub* [Identity First Tech 2021] faces notable challenges in terms of maintenance and reliability; the biggest of them is that its operation exclusively depends on an external service called *Passbase*. However, attempts to communicate with it constantly fail, and the provided documentation contains inoperative and outdated links, indicating possible discontinuation. These reasons make eKYC Hub not a viable alternative for real applications.

Furthermore, an unnamed solution discussed in Keycloak's development repository forums describes a native OIDC4IDA integration accomplished by modifying Keycloak's source code. Doing so distances the implementation from the desired scope, as it brings several disadvantages regarding compatibility between different system versions and its adoption in real-world scenarios. Moreover, using external data storage services is imperative for the functioning of this solution, harming its potential for versatility and adaptability. Finally, the prototype presents inconsistent behavior compared to what is described by OIDC4IDA, making its implementation unrealistic.

4. Proposed Implementation

Building on top of the related work's results, the solution proposed in this paper is designed with OIDC4IDA compliance as its main premise. To ensure this, the `authlete-java-common` package [Authlete 2024b], a programming library detailed in Section 3, is used to compute the specification's requests. By processing these requests using a tool with official OIDC4IDA support and developed by an organization actively involved in the specification's construction, strict compliance with the behaviors outlined in the protocol is guaranteed [Lodderstedt et al. 2023, Lodderstedt et al. 2022].

Furthermore, another key point of the proposed implementation is the validation of the protocol's requests and responses through JavaScript Object Notation (JSON) Schemas [Wright et al. 2022] provided by the OIDC4IDA development team. This behavior ensures the specification compliance. The solution also presents a modular approach. Thus, it can be easily integrated with multiple Keycloak instances without extra development effort. This adaptability is a key feature of the proposed work. Finally, the system allows the choice between internal or external services to store IdP users' verified claims, an extra functionality not seen in other market solutions.

To further validate the solution, several tests were conducted using sample files provided by the OIDC4IDA development team. These files contain examples of verified claims, requests, and expected responses from the protocol. A simulation environment was also set up by encapsulating a Keycloak instance in Docker containers. This environment, along with more technical details of the solution, is available for public access, allowing anyone to replicate the tests and validate the implementation by themselves.

5. Final Remarks

The related work analysis reveals a gap between existing solutions and the OIDC4IDA protocol. Consequently, the work conducted is expected to have a relevant impact on adopting OIDC4IDA, which should contribute to a more reliable digital environment.

The presented solution meets the initial objectives and proposes additional functionalities. The Keycloak integration ensures that the OIDC4IDA can be adopted in a widely recognized open-source tool. Its modular approach, which emphasizes adaptability, brings advantages such as ease of integration with ongoing projects and interoperability between different Keycloak versions. Furthermore, the full compliance between the proposed solution and the specification stands out as a decisive point for their adoption.

The proposed implementation's source code is available at `https://github.com/Bredstone/Keycloak-Extension-OIDC4IDA`.

For future works, the authors will integrate this implementation with a nationwide electronic identification system in Brazil: the Autenticação Eletrônica do Registro Civil do Brasil (IdRC) [Silva et al. 2023]. Additionally, efforts are underway to list it in OIDC4IDA development repositories [Lodderstedt et al. 2023] and Keycloak's communication channels, popularizing the solution and inviting community contributions.

References

Authlete (2024a). Authlete - homepage. Available at `https://web.archive.org/web/20240515152813/https://www.authlete.com/`, accessed on 19/06/2024.

Authlete (2024b). Authlete common library for java. Available at `https://github.com/authlete/authlete-java-common/tree/master`, accessed on 19/06/2024.

Connect2id (2024). Connect2id server. Available at `https://web.archive.org/web/20240520231313/https://connect2id.com/products/server`, accessed on 20/06/2024.

Ferdous, M. S. (2015). *User-controlled identity management systems using mobile devices*. PhD thesis, University of Glasgow.

Hardt, D. (2012). The OAuth 2.0 Authorization Framework. RFC 6749.

Identity First Tech (2021). ekyc hub. Available at `https://web.archive.org/web/20220623224237/https://github.com/identityfirst/eKYC-Hub`, accessed on 20/06/2024.

Keycloak (2024). Open source identity and access management. Available at `https://web.archive.org/web/20240619221535/https://www.keycloak.org/`, accessed on 18/06/2024.

Lodderstedt, T., Fett, D., Haine, M., Pulido, A., Lehmann, K., and Koiwai, K. (2022). Openid connect for identity assurance 1.0.

Lodderstedt, T., Sanz, M., and Haine, M. (2023). Openid connect for identity assurance implementations. Available at `https://web.archive.org/web/20240224103438/https://bitbucket.org/openid/ekyc-ida/wiki/Implementations`, accessed on 20/06/2024.

Miyata, T., Koga, Y., Madsen, P., Adachi, S.-I., Tsuchiya, Y., Sakamoto, Y., and Takahashi, K. (2006). A survey on identity management protocols and standards. *IEICE TRANSACTIONS on Information and Systems*, 89(1):112–123.

Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., and Mortimore, C. (2014). Openid connect core 1.0. *The OpenID Foundation*, page S3.

Silva, B., Schardong, F., Custódio, R., and Vendramin, L. (2023). Identificação eletrônica do registro civil do brasil. In *Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. SBC.

Wright, A., Andrews, H., Hutton, B., and Dennis, G. (2022). Json schema: A media type for describing json documents.