Computers & Security 168 (2026) 104934

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

Best current practices for privacy-preserving OpenID Connect: A study of their adoption in the wild

Gianluca Sassetti [a,b,*], Amir Sharif [b,**], Giada Sciarretta [b], Roberto Carbone [b], Silvio Ranise [b,c]

[a] Department of Computer Science, Bioengineering, Robotics and Systems Engineering, University of Genova, Genova, Italy
[b] Center for Cybersecurity, Fondazione Bruno Kessler, Trento, Italy
[c] Department of Mathematics, University of Trento, Trento, Italy

ARTICLE INFO

Keywords:
Digital identity
Privacy
General Data Protection Regulation
OpenID Connect
eIDAS

ABSTRACT

The transition from centralized identity architecture to a decentralized one introduces profound shifts in the privacy protection of users' data. Yet, as decentralized identity continues to mature, today's online services still overwhelmingly depend on centralized and federated identity management solutions built on top of OpenID Connect (OIDC) as the most widespread solution. Ensuring privacy-preserving OIDC deployments is therefore critical for safeguarding users' personal data and maintaining compliance with regulatory frameworks such as the General Data Protection Regulation (GDPR) and trust frameworks such as the Electronic Identification, Authentication and Trust Services (eIDAS). However, the current OIDC ecosystem lacks a coherent set of privacy Best Current Practices (BCPs) and a study of how widely these privacy-enhancing features are adopted in real-world deployments. To this end, this work addresses the aforementioned gaps on two fronts. First, we propose a structured set of privacy BCPs derived from official OIDC specifications and current implementation trends, identifying easy-to-deploy privacy-enhancing features that strengthen the OIDC deployments' baseline privacy without altering the protocol or compromising interoperability. Furthermore, the BCPs also help achieve the GDPR privacy principles, such as data minimization, confidentiality, and unlinkability. Second, this work provides a comprehensive survey of OpenID Providers (OPs) in the wild to identify gaps in privacy-preserving configurations in both private and public (i.e., national) sectors OPs. The study employs a dual methodology: first, a manual review performed in 2022; subsequently, an automated compliance analysis performed in 2025 surveying a dataset of 10000 OPs worldwide. The results reveal a concerning lack of privacy-enhancing features among private OPs and a wide gap between private and national OPs, with the latter group providing, on average, much higher baseline privacy. We have also found a prevalence of OPs not complying with the OIDC specifications, resulting in misconfigured OPs hampering interoperability and, in some cases, security. The paper emphasizes the importance of adopting actionable BCPs to improve baseline privacy and demonstrates the need for an automated framework for ongoing privacy compliance assessments in OIDC ecosystems.

1. Introduction

The current web infrastructure is based on centralized and federated identity architectures, where a single Identity Provider (IdP) authenticates multiple users for different Service Providers (SPs). This is the basis for Single Sign-On (SSO): users log into online SPs by authenticating at an IdP, which then transmits the identity data to the SPs. The rise of decentralized three-party model (issuer-wallet-verifier) architectures (a.k.a. wallet-based architecture) marks a sharp shift with respect to the traditional centralized and federated ones. In wallet-based architectures, *issuers* provide users with their digital identity, much alike IdPs, that is then stored on an application running on the user's device known as wallet application. To authenticate, users simply *present* the identity data in their wallet to verifier, equivalent to SPs, similarly to what we do with identity cards in everyday life. The new identity paradigm also disrupts the traditional responsibility models. In the centralized and federated architectures, the responsibility of user

* Corresponding author at: Department of Computer Science, Bioengineering, Robotics and Systems Engineering, University of Genova, Genova, Italy.
** Corresponding author at: Center for Cybersecurity, Fondazione Bruno Kessler, Trento, Italy.
E-mail addresses: gsassetti@fbk.eu (G. Sassetti), asharif@fbk.eu (A. Sharif), g.sciarretta@fbk.eu (G. Sciarretta), carbone@fbk.eu (R. Carbone), ranise@fbk.eu (S. Ranise).

https://doi.org/10.1016/j.cose.2026.104934
Received 4 December 2025; Received in revised form 27 February 2026; Accepted 19 April 2026
Available online 29 April 2026

privacy is shared between IdP, SP, and the user, and falls mainly on the first two. IdPs and SPs have to cooperate and enforce appropriate privacy measures to preserve the user privacy, and the user has to make informed decisions based on the information provided by the other parties. In decentralized architectures, instead, the user directly controls their digital identity and makes privacy-influencing decisions. While decentralized architectures empower users, they also place more responsibilities on them. IdPs and SPs can provide limited assistance to the user.

Decentralized Identity Management systems (IdMs) are currently being designed and developed, and more years will pass until we transition to these newer systems. In the meantime, our day-to-day online operations are possible only through centralized and federated IdMs. Being shared responsibility models, it is pivotal that we ensure that IdPs and SPs implement and enforce the right privacy mechanisms.

Among the things one could look into, when trying to increase the baseline privacy of an identity system, is the authentication protocol used between IdPs and SPs. The authentication protocol is used to request a user authentication, transmit identity attributes and claims about the outcome and assurance of the authentication process, and retrieve user data. In this context, the most used authentication protocol is OIDC (Sakimura et al., 2023a). The protocol enables Relying Parties (RPs), equivalent to SPs, to request the authentication of users at an OP, that is an IdP where the user is enrolled. RPs thus off-load all identity-related functionalities to an OP; this also allows users to use one OP to log into multiple RPs.

OIDC is currently implemented for services supported by private organizations, e.g., financial and banking applications, as well as public administrations (World Economic Forum, 2021). In the latter case, OIDC is used to support national and international digital identity infrastructures, which are of paramount importance as they empower citizens to exercise their rights by accessing online services provided by public administrations.

In both the private and the public (i.e., national) scenarios, OPs and RPs collect, share, and process large volumes of sensitive personal data. The disclosure or unauthorized modification of this data may have serious consequences for all the system's stakeholders and, in particular, the users. Besides substantially hardening the security postures of OPs to prevent massive exfiltration of personal data and other similar threats, it is crucial to guarantee that OIDC deployments are privacy-preserving to prevent specific privacy threats such as linkability. In fact, simply enhancing security measures will not resolve all the privacy principles violations (i.e., linkability), as insufficient privacy controls and guarantees can also lead to this problem. Linkability and many more threats to privacy can be mitigated only through the support of privacy controls by OPs (i.e., support of pseudonyms) and RPs (i.e., implementation of pseudonyms). Furthermore, preserving user privacy through controls is key to complying with data protection laws, such as the GDPR (European Parliament and Council of the European Union, 2016), and trust frameworks, such as the eIDAS (European Parliament and Council of the European Union, 2014).

Unfortunately, it is unclear what the current state of the art is for OIDC deployments with respect to the degree of privacy they offer. The problem is twofold. First, there is a lack of a coherent set of BCPs to help in configuring and implementing privacy-aware OIDC deployments. Instead, considerations and suggestions to use data protection mechanisms are scattered in several official OIDC specifications or are emerging as de facto standards because they are adopted by a large number of OPs. The second issue is a lack of privacy compliance analysis for private and public (i.e., national) sector OPs; thus, the extent to which privacy-enhancing features are implemented is unknown.

This gap in our knowledge leads to creating a fragmented OIDC landscape, where, without extensive guidance, OP developers implement their own solutions to improve user privacy or just meet the bare-minimum requirements for OIDC to work. The lack of implementation of privacy controls may prove greatly detrimental for user privacy, as the large number of identity transactions may eventually lead to unwarranted privacy violations.

Other works in the past have tried to improve the baseline privacy of OIDC by extending the protocol (Zhang et al., 2020; Fett et al., 2015; Hammann et al., 2020a). Despite its effectiveness, this approach is much more burdensome on OP developers and hinders interoperability; therefore, it is rarely ever adopted in practice. Our goal is to understand how to improve the privacy of OIDC without modifying the protocol and the standards, hereby preserving interoperability. We want to focus on controls that can be easily implemented by OP developers and are commonly put in practice in many OPs.

To address the aforementioned problems, our work provides the following contributions. Firstly, we contribute with updated list of privacy BCPs to reflect the latest changes within the OIDC specifications. This list consists of a set of actionable privacy-preserving features from different OIDC specifications (Sakimura et al., 2023a; Varley and Grassi, 2024; Lodderstedt et al., 2024) and the latest implementation trends. Each BCP is an easy-to-implement mitigation that improves the baseline privacy of OPs and helps meet privacy requirements defined through the GDPR (European Parliament and Council of the European Union, 2016). In particular, the BCPs support the main privacy principles covered by the GDPR, with the exception of storage and purpose limitation that cannot be verified without direct access to the OPs and their inner workings. The BCPs shall be implemented by OPs; RPs, however, must integrate them in their OIDC flows to actually enhance user privacy. It is therefore fundamental that users, in their interest, pressure both OPs and RPs into adopting the BCPs.

Secondly, we contribute two surveys that shed light on the state of the art of OIDC deployments. In one we survey the adoption of the BCPs in the wild, and in the other we find instances of non-compliance with respect to OIDC Core (Sakimura et al., 2023a) and Discovery (Sakimura et al., 2023b) specifications among OPs. In 2023 (Sassetti et al., 2023) we presented a survey of the BCPs adoption in the wild on a set of 14 private and 13 national OPs. In that first work, we adopted a manual methodology to test OPs and verify the adoption of BCPs. Since then, we have developed a new methodology, automated and scalable, to test OPs.

Through this new methodology we update the results of the previous survey and propose a new survey on a much larger dataset of 10,000 OPs. Thus, the BCPs survey we present was conducted at two different times, with two different methodologies and datasets. Updating the results of the 2023 survey aids in comprehending how OPs evolve and introduce new features, while the survey conducted on the new dataset offers a more precise representation of the current state of the art. By leveraging the new methodology, we also report on instances of non-compliance among OPs in the wild.

To summarize, in our previous work (Sassetti et al., 2023), we propose a well-defined and actionable set of BCPs to increase the OPs baseline privacy, and we provide a preliminary BCP adoption survey. The current paper extend (Sassetti et al., 2023) with the following main contributions:

- We update our list of well-defined and actionable set of privacy BCPs to reflect the latest changes within the considered OIDC specifications;
- We provide a methodology for the automated compliance check of OPs with our identified set of privacy BCPs. In addition, we highlight how the introduced methodology can be generalized to perform the OPs compliance with respect to the OIDC standards, focusing on OIDC Core and the OIDC discovery specifications;
- We propose a snapshot of the state of the art through a survey of the adoption of the privacy BCPs in the wild conducted over a dataset of 10,000 OPs worldwide, and a survey of instances of non-compliance among OPs in the wild with respect to OIDC Core and Discovery (Sakimura et al., 2023a,b).

Our findings show that the privacy BCPs are vastly overlooked by private OPs, having implementation rates as low as 2%. On top of that, 84% of them implement only mandatory features and no further BCPs that could increase their baseline privacy. 13% of private OPs in the wild do not comply with OIDC specifications, with that being due, in a limited number of instances, to the lack of support of HTTPS in OIDC endpoints. Securing the communication channels with HTTPs is pivotal for ensuring the security and privacy of all parties involved in the protocol and without that it can result in risks such as breach of confidentiality, as well as RP impersonation. Together these elements are alarming, as user privacy is safeguarded only in rare cases, and instances of non-compliance with OIDC standards can hamper the interoperability of the OPs. Fortunately, we also witness a sharp contrast between private OPs and national OPs, with the latter demonstrating a much higher baseline privacy profile. The average implementation rate of the privacy BCPs in national OPs is 52% against the 15% shown by private OPs. The privacy-preserving BCPs that had an implementation rate between 2% and 5% in private OPs have implementation rates between 60% and 85% in national OPs. The remarked difference between national and private OPs brings a spark of hope amid a multitude of privacy threats.

Our findings increase an understanding of the state of the art in the privacy of OIDC deployments and show what the more common practices are among OPs. However, they also raise concerns about user privacy. We hope our contribution can shed light on the OIDC landscape and help developers and policymakers improve the privacy of their solutions. It also seeks to empower users by raising awareness, enabling them to demand stronger privacy protections.

Paper structure. Section 2 details some relevant previous work that deals with privacy issues in OIDC. Section 3 presents some notions needed to understand this work. In Section 4, we provide the list of features analyzed for this work alongside our recommended privacy BCPs to provide privacy-preserving OPs implementations. In Sections 5 and 6 we present our surveys (BCP surveys and non-compliance survey), illustrating methodologies, data sources, and results. In Section 7 we discuss observations deriving from the surveys' results, and Section 8 concludes this work.

2. Related work

In the past, the security of Open Authorization (OAuth) (Hardt, 2012) and OIDC (Sakimura et al., 2023a) protocols has been widely studied, both theoretically and practically. The research has mostly focused on concrete attacks to the protocols (Chari et al., 2011; Fett et al., 2016, 2017), and a number of solutions and mitigations have been proposed to tackle their vulnerabilities (Zhou and Evans, 2014; Li et al., 2019; Calzavara et al., 2018). Despite that, little effort has been put into studying how OPs protect user privacy by integrating privacy-by-design principles within their implementations. In the following, we summarize some of the available works in the literature that deal with privacy issues of the OIDC protocol.

Fett et al. proposed a privacy-preserving Single-Sign-On (SSO) system for the web called "SPRESSO" (Fett et al., 2015) that decouples the direct communication between RP and OP by using a forwarder agent at the user's side with the aim to avoid user linkability by the OP at various RPs. Asghar et al. in Asghar et al. (2018) introduced a privacy-preserving solution that is a modified version of the cryptographic construction presented in Oblivion (Backes et al., 2016). Their solution decouples the interaction between the OP and RP by separating the credential issuance by the OP from its usage by the user at RP. Navas and Beltrán provide a comprehensive threat model for the OIDC in Navas and Beltrán (2019) that highlights the following privacy threats: lack of control over required personal data, personal data leakage, user profiling, and location tracking. The authors also proposed mitigations that include encryption to minimize the risk of personal data leakage and using flow-specific user identifiers to avoid user profiling. In 2020, Apple introduced its SSO solution based on OIDC called "Sign In with Apple" (Apple, 2020) that uses randomized (per RP) identifiers in place of a user email address to avoid user linkability across RPs. Zhang et al. proposed a privacy-preserving system based on OIDC called "EL PASSO" (Zhang et al., 2020) that implements anonymous credentials to enable selective disclosure and avoid user linkability. Hammann et al. (2020b) and Li and Mitchell (2020) proposed solutions to address the problem of user linkability by decoupling the interaction between the OP and RP in obtaining and using credentials. Most recently, Morkonda et al. described a browser extension called "SSOPrivateEye" (Morkonda et al., 2022) that provides a privacy comparison where users have multiple choices of OPs to login into an RP. Exploiting this information, users can choose the one that shares less amount of personal data with RPs. This solution provides some privacy insights only for Google, Facebook, and Apple as OPs.

Most of the aforementioned research works demand either major changes to the OIDC protocol or the installation of a browser plugin within the user's device to partially increase the user's privacy. Indeed, none of them provide some easy-to-implement privacy-preserving features by leveraging the features already available in OIDC (Sakimura et al., 2023a; Lodderstedt et al., 2024; Varley and Grassi, 2024). Furthermore, no study has assessed the privacy of eIDAS solutions. Given that, our work can be used to complement and enhance plug-in-based solutions by providing more informative data, e.g. privacy principles satisfied by each OP that make the users more aware of the privacy level of the OP. In addition, our research work can be integrated into a stand-alone tool to automate the procedure of privacy compliance analysis.

3. Background

In this section, we characterize privacy with respect to the goals extracted from the GDPR (European Parliament and Council of the European Union, 2016) (Section 3.1) and provide a concise description of OAuth and OIDC (Section 3.2).

3.1. Privacy principles

The characterization of privacy with different properties has been the subject of academic debate over the years (Koops et al., 2017). We define privacy based on the GDPR's privacy goals (European Parliament and Council of the European Union, 2016). We focus on a subset of privacy principles sourced from Article 5 of the GDPR, selecting those that are most relevant to the privacy of OIDC deployments and that can be supported through OIDC. Other principles in the GDPR, such as accountability, storage limitation, and lawfulness, are outside the scope of this work, since they cannot be enforced through OIDC.

Below, we explain briefly the five selected privacy principles within the scope of this work.

Data Minimization: the parties involved in data exchanges should use and share only the minimum amount of user data necessary for their functionalities;

Confidentiality: personal data shall be protected from unauthorized, unlawful disclosure. Confidentiality has to be ensured through the implementation of multiple mechanisms at different levels of the implementation stack. Here, we focus on the aspects of confidentiality related to the controlled disclosure of personal information. For evaluating the privacy of OPs, we assume that other security mechanisms (Lodderstedt et al., 2025), i.e., TLS for securing communication channels, have been put in place and consider only the mechanisms to access and disclose user data;

Data Accuracy: data shall be exact and correct; parties collecting data should granted a minimum degree of confidence in the correctness of the data;

Transparency: parties collecting and processing data shall clearly state the purpose and extent of the data acquisition and allow the user to opt-in to the processing of their data. Also, the parties with which personal data will be shared need to be communicated to the user.

We also define one more privacy principle, which is not explicitly included in Art. 5 of the GDPR, but is part of the LINDDUN framework (Jensen et al., 2012) and upholds multiple privacy principles expressed in the GDPR.

Unlinkability: the user should not be identifiable and traceable across different platforms, without giving explicit consent. User data should be stored and shared in a way that would not allow other parties to identify the user and link their actions to a single account, thus granting a minimum level of anonymity. Unlinkability should hold even when colluding parties unlawfully share data.

Unlinkability is a pragmatic transposition of purpose limitation, for which data shall be collected and processed only for its explicit and legitimate purpose, as stated in Art. 5 par. 1 lit. b of the GDPR. Any unauthorized party that is able to identify the user extracts an additional quantity of data from the user's activity. Thus, it violates the principles of confidentiality and transparency, as the user has not acknowledged and consented to the use of their data. Given that this scenario is particularly relevant for OIDC, and unlikability is part of the LINDDUN framework, we have included it as a privacy principle.

3.2. OAuth and OIDC

We present here an introduction to OAuth and OIDC that is not meant to be exhaustive. Rather, we aim to illustrate the main concepts and elements of the protocol, with a focus on the features that will be discussed in this work. OAuth is an authorization framework with which an application can be granted access to user resources by first asking for consent from the user (Hardt, 2012). OIDC is an identity protocol that adds authentication to the OAuth framework (Sakimura et al., 2023a). Authentication is possible by distributing user-identifying data called claims.

OIDC flows start with an authorization request sent by an application, called RP, to the OP. Authorization requests include several parameters (Listing 1), among which we need to highlight `scope`, with which the RP requests a set of the user's resources. The RP's accesses are limited to the resources listed in `scope`, and thus the parameter is pivotal for access control. The list of parameters supported in the authorization request depends on the OP, based on the specifications it implements.

All OPs must implement OIDC Core and Discovery (Sakimura et al., 2023a,b) and may implement additional specifications tailored to their use cases, such as the Financial Grade Security API (OpenID Foundation, 2023a), the International Government Profile (Varley and Grassi, 2024), and OIDC for Identity Assurance (Lodderstedt et al., 2024). Each specification may add new parameters or data attributes to *protocol artifacts*, i.e., the requests, responses, and payloads produced at each step of the protocol, such as the authorization request. Specifications can also increase the security and privacy requirements of OPs.

Each parameter included in a protocol artifact is assigned a requirement level keyword (Bradner, 1997). OIDC specifications mainly use three keywords. *Required* denotes parameters that must always be supported. *Recommended* parameters must be supported unless under exceptional circumstances that have to be documented and justified. *Optional* parameters are not necessary for complying with the specifications and may or may not be supported. It is common that the requirement level keyword of parameters is increased in specifications designed to grant higher security and privacy (Varley and Grassi, 2024; Burgin and Clancy, 2025).

OIDC Core introduces three flows, namely: Authorization Code, Implicit, and Hybrid (Sakimura et al., 2023a). In this work, we mainly focus on the Authorization Code flow as the most widely used and recommended flow by the security best current practices (Lodderstedt et al., 2025). A simplified explanation of the OIDC authorization code flow is as the following. After processing the authorization request from the RP, the OP directly asks the user to sign in, without the intermediation of the RP. The user is thus directed to a *consent page*, also called *authorization page*. The consent page informs the user of what data has been requested and allows the user to consent to share their resources. The user interaction with the consent page is the main way in which OPs guarantee the transparency and ensure the user's awareness of data processing and sharing.

Once the user grants their consent, the OP sends an authorization code back to the RP, which can then be exchanged at the *token endpoint* for an *ID Token* and an *Access Token*. The ID Token is a security data structure consisting of a unique user identifier, user claims, and authentication context data. The Access Token can be used to access the user resources and obtain claims on the authenticated user at the *userinfo endpoint*.

```
https://<OP_URL>/PATH/TO/ENDPOINT/authorize?
response_type=code&
client_id=<my_app_id>&
redirect_uri=<callback_URI_to_my_app>&
scope=openid
```

Listing 1: Example of authorization request.

3.3. OIDC privacy challenges

This section elaborates on how privacy issues may arise during the authorization code flow.

The authorization code flow begins by sending of the authorization request from RP to OP. During this step, data minimization is a primary concern, as RPs frequently request more scopes and claims than are strictly necessary for their functionality. This behavior is common in real deployments and leads to systematic over-collection of personal data. In addition, confidentiality may be at risk if RPs request scopes or claims they are not authorized to access. The effectiveness of privacy protection at this step therefore heavily depends on the OP's ability to validate authorization requests against well-defined access control policies.

After receiving the request, the OP prompts the user to authenticate. This step is critical to preserve confidentiality. The OP must always ensure that the authentication process provides a level of assurance adequate to the sensitivity of the resources that are being requested. Clearly, inadequate authentication processes may lead to a loss of confidentiality. Following authentication, the OP typically presents a consent page to the user, where users acknowledge the extent of the data processing and express or deny their consent to it. This step is essential for ensuring transparency and, where applicable, consent. If consent pages are not displayed or incomplete, there is no way for a common user to know what data is going to be collected and for what purposes.

Finally, the `sub` field in the ID token represents the greatest threat to unlinkability, as different RPs federated to the same OP can use the `sub` field to link different ID tokens to the same user. This not only compromises unlinkability, since they can infer what services the user accesses through that OP, but it may also allow RPs to extract more data than they would be allowed to. This would be the case, for instance, of two RPs having access to different personal data of the user.

Table 1
Summary of privacy-supporting features we present as our BCPs.

Feature	Description	Source
`scope`	Request user resources	Core (Sakimura et al., 2023a)
`claims`	Request specific user claims	Core, iGov (Sakimura et al., 2023a; Varley and Grassi, 2024)
`verified_claims`	Request specific user claims along with evidence of the verification process	Id. Assurance (Lodderstedt et al., 2024)
`acr_values`	Request stronger authentication	Core, iGov (Sakimura et al., 2023a; Varley and Grassi, 2024)
`vtr`	Request stronger authentication, define the authentication context	iGov (Varley and Grassi, 2024)
`pairwise`	User cannot be identified with the subject type	Core, iGov (Sakimura et al., 2023a; Varley and Grassi, 2024)
Consent	Transparent, informative consent page, implementation of selective disclosure	Common practices
Access control policy	Controlled disclosure of user data	Common practices

3.4. Discovery endpoint

The OIDC Discovery specification (Sakimura et al., 2023b) requires OPs to publish their metadata in a *discovery document*. The metadata can be retrieved by querying the OP's Discovery endpoint, whose address is formed by concatenating `/.well-known/openid-configuration` to the OP address. The discovery endpoint simplifies the configuration process as it allows the RPs to fetch all necessary details to interact with the OPs without requiring manual setup. This metadata includes essential elements to enable communication within the RP and OP, such as supported authentication methods, token endpoint authentication mechanisms, signing algorithms for ID tokens, and the scopes available for authorization requests.

By leveraging the discovery endpoint, RPs can dynamically adapt to changes in the OP's metadata, ensuring interoperability and reducing maintenance overhead. The metadata shared by OPs is in machine-readable JSON format Bray (2014) where the JSON properties (i.e., key–value pairs) contain OIDC-specific configuration data. A simplified example of discovery document is provided below.

```
{
 "issuer": "https://example.com",
 "authorization_endpoint":
 ↪  "https://example.com/oauth2/authorize",
 "token_endpoint": "https://example.com/oauth2/token",
 "userinfo_endpoint":
 ↪  "https://example.com/oauth2/userinfo",
 "jwks_uri": "https://example.com/oauth2/jwks",
 "response_types_supported": ["code", "id_token", "token
 ↪  id_token"],
 "subject_types_supported": ["public", "pairwise"],
 "id_token_signing_alg_values_supported": ["RS256"],
 "scopes_supported": ["openid", "profile", "email"],
 "claims_supported": ["sub", "name", "email"],
 "grant_types_supported": ["authorization_code",
 ↪  "implicit", "refresh_token"],
 "token_endpoint_auth_methods_supported":
 ↪  ["client_secret_basic", "client_secret_post"]
}
```

3.5. eIDAS regulation and network

European regulation on eIDAS (European Parliament and Council of the European Union, 2014), established in 2014, creates a standardized framework for secure electronic transactions across European member states. It ensures that electronic identification and trust services, e.g., digital signatures, are legally recognized and interoperable throughout members of the European Union. The regulation establishes a network connecting national electronic identification schemes of different countries. It operates through a system of interconnected nodes, where each member state maintains an eIDAS node acting as the gateway between national OPs and RPs in other countries. When a user holding a digital identity of a European national identity scheme needs to access online services in another country, the eIDAS network enables secure cross-border authentication.

The European Committee maintains a list of national OPs (European Parliament and Council of the European Union, 2023) that are compatible with the eIDAS network and thus can be used for cross-border authentication (we refer to them in this work as eIDAS IdP, or eIDAS OPs). The OPs that want to become eIDAS OP undergo a vetting process and must demonstrate compliance with higher security and privacy standards.

4. OIDC Privacy Best Current Practices

The privacy BCPs we present in this section consist of mechanisms or functionalities that can be implemented by an OP (*features*) without making any change to the OIDC protocol. The BCPs are divided into two groups. A first group (OIDC features) has been extracted from OIDC specifications, whereas a second group (non-OIDC features) is not specified in OIDC standards but is commonly implemented by OPs. Each feature contributes to one or more of the privacy principles introduced in Section 3.1.

The BCPs comprise five authorization request parameters that the OPs can choose to support, one feature regarding subject identifiers (defined below), and two non-OIDC features. The reader can observe in Fig. 1 a depiction of the BCPs with their connection to the privacy principles of Section 3.1. Fig. 2 exemplifies how the BCP features are included and used during the OIDC code flow, i.e., the default OIDC flow recommended by the OAuth security best practices (Lodderstedt et al., 2025). Table 1 provides a short description of the BCPs together with the sources they were extracted from.

To extract the OIDC BCPs, we have selected three OIDC specifications taken from the official list published by the OpenID Foundation (OpenID Foundation, 2023b): OIDC Core, OIDC for Identity Assurance, and OIDC iGov profile (Sakimura et al., 2023a; Varley and Grassi, 2024; Lodderstedt et al., 2024). Besides OIDC Core, which introduces the protocol, the other specifications seek to increase the baseline privacy of OIDC. With OIDC for Identity Assurance, OPs can issue trustworthy user claims by providing evidence of a verification process. The iGov profile introduces new requirements suitable for national OPs. OIDC for Identity Assurance and iGov profile (Varley and Grassi, 2024; Lodderstedt et al., 2024) make up for the lack of privacy considerations in OIDC Core (Sakimura et al., 2023a) by addressing data minimization, accuracy, confidentiality, transparency, and unlinkability.

To select the three specifications, we scanned through the list of available OIDC specifications, looking for parameters, protocol artifacts (e.g., access code, data attributes, etc.), and privacy and security considerations. We have selected only the specifications that define privacy-relevant request parameters and subject identifier semantics central to our analysis. We deemed other specifications out of scope, as they apply to specific industry settings and use cases (e.g., native applications or wireless network operators) or provide no further improvement for user privacy.

To extract the non-OIDC BCPs, we have considered design patterns and features that we found commonly implemented by OPs in the

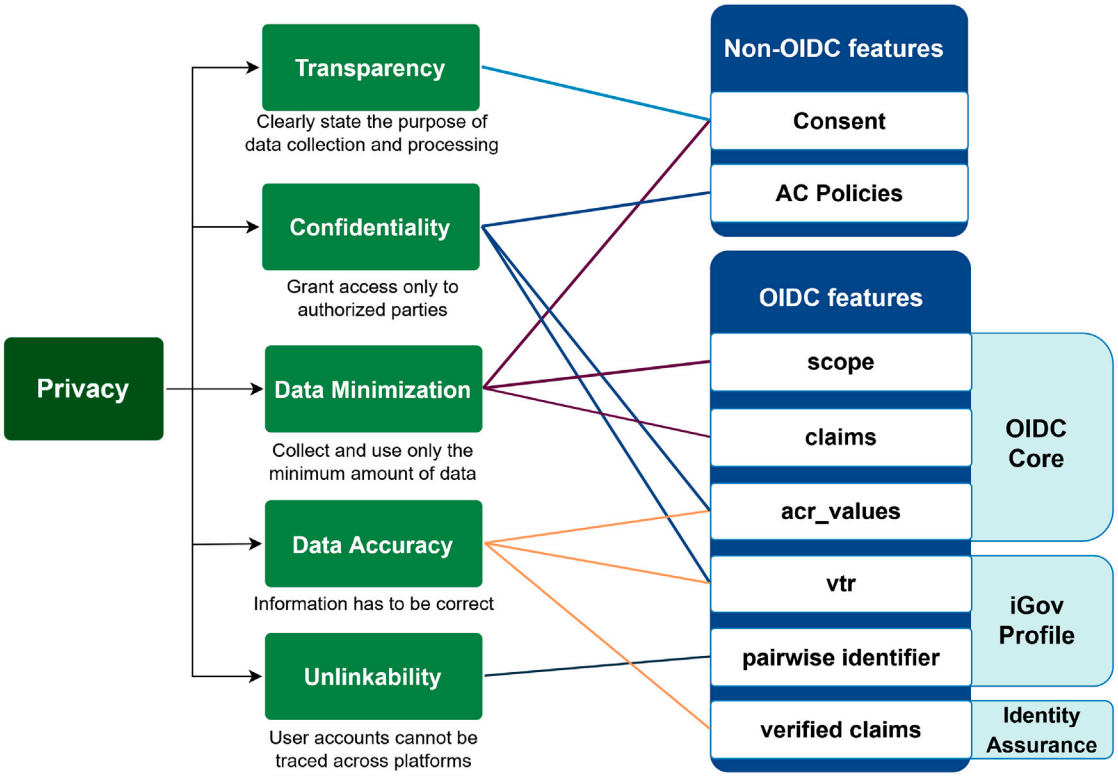

Fig. 1. Summary of our BCPs and their connection to privacy goals.

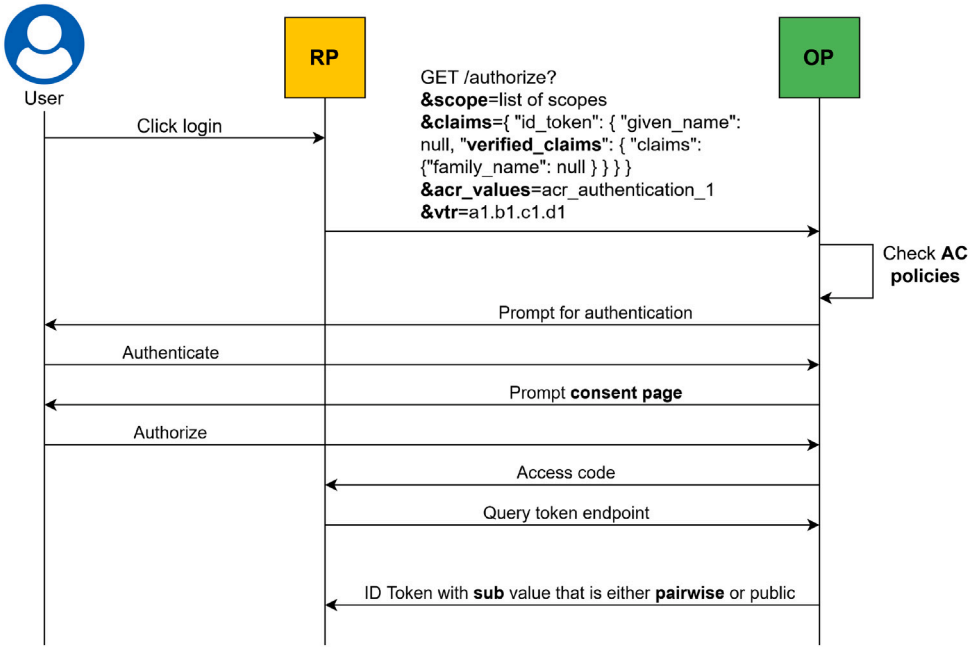

Fig. 2. Example of how the privacy BCPs are included in the OIDC authorization code flow. Note that this is not the entirety of the authorization code flow, but the section of the protocol where the BCPs are used. Also, the characters in the authorization request have not been encoded for readability.

wild during an analysis of the state of the art. We considered only non-OIDC features that contribute to privacy principles identified in Section 3.1. Non-OIDC features that cover aspects of privacy that are not already within the scope of OIDC are outside the scope of this work, e.g., policies for data storage and handling breaches.

The contributions to privacy of each selected specification have some overlaps but remain mostly distinct (e.g., both OIDC core and iGov include `acr_values`, but the latter emphasizes greatly on the necessity of using it to increase the authentication level. More on `acr_values` later in this section). Our BCPs connect the contributions of each specification and expand their coverage through the non-OIDC features, thus providing a complete and well-rounded set of recommendations. Moreover, the privacy goals, as well as the BCPs that enforce them, are interlinked. The risks deriving from the violation of

any privacy goals negatively impact all the others. We argue that our BCPs should be implemented altogether to minimize the overall risk to privacy and grant the highest degree of assurance to users.

Before continuing further with the BCPs presentation, we briefly digress on the interplay between the BCPs and assurance. Parties in a multi-party application trust each other with varying confidence based on the trustworthiness of the trustee application or, conversely, the degree of assurance provided by the trustee. To help increase its trustworthiness, and thus the assurance provided to other parties, an OP can demonstrate compliance with standards and the implementation of security and privacy measures. Higher assurance levels are required for regulated scenarios handling sensitive resources like national, financial, and healthcare systems. Our BCPs allow us to increase the level of assurance of OPs since they provide higher privacy to their users. On a separate note, privacy goals vary by use case, and Fig. 1 helps determine the necessary features to achieve the desired privacy posture of OPs. Tuning the required level of assurance is a prerequisite for establishing a suitably strong trust relationship between an OP and RPs for a certain use case scenario. In other words, Fig. 1 can be considered a high-level map to orient designers in the adoption of privacy features capable of yielding the right level of assurance and trustworthiness for the ecosystem (comprising OPs and RPs) of their use case scenario, be that finance, healthcare, or a simple web application.

We report below a detailed description of each feature.

Authorization request features.

`scope`: Through this parameter, the RP can request access to user-owned resources. Scopes are identifiers for sets of resources or permissions, and thus they are used for access control. The parameter impacts data minimization, as all user resources are requested through it. Although it is used in access control policies, the parameter does not directly support confidentiality because it can only be used to specify resources. `scope` was introduced in OIDC Core (Sakimura et al., 2023a) and is a mandatory parameter in authorization requests; it must contain the value `openid`, which allows OPs to distinguish between an OAuth authorization flow (Hardt, 2012) and an OIDC authentication flow. As an example of usage, an RP can include `scope=openid profile email address` in the authorization request. Later during the OIDC flow, the RP can obtain the data related to the user's profile, email, and address by querying the UserInfo endpoint.

`claims`: Through this parameter, the RP can request specific user claims to be returned in the ID Token or from the UserInfo endpoint after successful user authentication. If the parameter is missing, the OP will provide a default set of claims. `claims` can be used to request only the necessary resources, thus limiting the sharing of user data. The parameter is included as optional in OIDC Core (Sakimura et al., 2023a) but is defined as mandatory in the iGov profile (Varley and Grassi, 2024) because of its importance for data minimization.

`verified_claims`: This is a parameter introduced in OIDC for Identity Assurance (Lodderstedt et al., 2024) that allows requesting a set of verified user claims. On the OP side, user claims are associated with a verification method and a trust framework which they refer to. An example of such a framework is eIDAS, or a national eID scheme. The verification process usually happens upon registering the user's claims, e.g., via an electronic identity card. After successful authorization, user claims are returned alongside metadata containing evidence of the verification process. By providing trusted information, the parameter contributes to data accuracy.

`acr_values`: This parameter, introduced in OIDC Core (Sakimura et al., 2023a), allows the RP to request strong authentication methods. It defines a set of scalar values representing the minimum levels of identity proofing asserted during authentication. Depending on the value that the RP requests, the OP enforces different authentication methods. For instance, the first level is usually associated with a simple username and password login. Other levels may require Multi-Factor Authentication instead. `acr_values` allows to increase the authentication requirements; therefore, it contributes to privacy by providing greater data accuracy and confidentiality.

`vtr`: This parameter is introduced in the iGov profile (Varley and Grassi, 2024) with the purpose of requesting different authentication requirements, like `acr_values`, with the distinction that `vtr` leverages Vectors of Trust (VoT). `vtr` is a vector of scalar values, each of which defines a certain aspect of the authentication context. It is, therefore, more precise and flexible than `acr_values` (Richer and Johansson, 2018). It provides data accuracy and confidentiality.

Subject identifier feature. The `subject_type` defines how to generate the `sub` field, or subject identifier, a mandatory parameter of the ID Token that uniquely identifies the user. `subject_type` can have two values: `public` and `pairwise`, both defined in OIDC Core (Sakimura et al., 2023a). With `public`, all the RPs under the same OP receive the same identifier for a fixed user. Instead, with `pairwise`, the same user will have a different identifier for each RP. The purpose of `pairwise` is to reduce the risk of linkability between RPs. The implementation of `pairwise` identifiers for unlinkability has been questioned in the past (Hammann et al., 2020b; Li and Mitchell, 2020). As a matter of fact, `pairwise` identifiers grant unlinkability only if `sub` is the only identifying information contained in the ID Token. If data such as email or name is returned by the OP, RPs can easily identify and trace the user just by sharing the ID Token. There are available solutions in the wild that tackle this problem. However, except for the use of pseudonymization, the other solutions (e.g., Blind broker architecture (Boysen, 2021) or Zero-Knowledge Proofs (Hammann et al., 2020b)) demand either major changes to the OPs or to the OIDC protocol. Despite that, adopting `pairwise` identifiers is a necessary step to grant unlinkability, as stressed by the iGov profile (Varley and Grassi, 2024), and should therefore be considered a privacy BCP.

Non-OIDC features.

Consent page and selective disclosure: OPs should clearly communicate the purpose, means, and extent of the user data that is collected, processed, and shared. This is often done through a consent page with which users are shown the aforementioned information and can explicitly agree to the use of their data. Moreover, consent pages can support selective disclosure, a feature that allows the user to select which resources, among those requested, to grant access to. Selective disclosure helps make the user aware of which personal data is being processed while contributing to transparency and data minimization.

Access Control Policies for sensitive scopes: This feature includes any kind of policy for accessing high-risk or sensitive resources (e.g., users' health records, biometric data). Through `scope` or `claims`, OPs can enforce access control policies tailored to specific sensitive resources and restrict RPs access to them. For instance, OPs can initiate multi-factor authentication when sensitive resources are requested. Alternatively, OPs can restrict access to some scopes only to a subset of specifically authorized RPs.

Even though, access control policies are outside the scope of OIDC, we investigate them as they impact the confidentiality of user data. We consider the access control policies that are enforced on requested scopes and that concern the access and disclosure of user data.

Table 2
Key differences between the two surveys of the BCPs implementation.

		Sep 2022	Sep 2025
Methodology		Manual testing and documentation analysis	Automated discovery endpoint analysis
Data source	Private OPs	Alexa top ranked - 14 total OPs	Tranco list top ranked - 10000 total OPs, ForgeRock is no longer available
	National OPs	13 national eIDAS-notified OPs (all the available OPs in 2022)	11 national eIDAS-notified and 2 national no-eIDAs notified OPs. NemID and FranceConnect are no longer available. Added the NHS and Estonian government new OPs
BCPs			`purpose` is no longer supported by OIDC for Identity Assurance and is removed from the BCPs

Table 3
The BCPs considered during Sep2025 with their corresponding discovery document properties and requirement level keyword. The discovery document properties are tested to verify the implementation of the corresponding feature.

BCP	Discovery document property	Keyword	Test
`scope`	`scopes_supported`	Recommended	Verify implementation and inclusion of `openid`
`claims`	`claims_parameter_supported`	Optional	Verify content (true or false)
`verified_claims`	`verified_claims_supported`	Required	Verify implementation
`acr_values`	`acr_values_supported` or `claims_supported`	Optional and Recommended	Verify implementation or inclusion of `acr` inside `claims_supported`
`vtr`	`claims_supported`	Recommended	Verify inclusion of `vot` inside `claims_supported`
`pairwise` subject type	`subject_types_supported`	Required	Verify inclusion of `pairwise`

As a final remark, we observe that, despite not being included in the OIDC standards (the standard considers implementation details out of scope), these features seem to be crucial in providing adequate support for assurance and trustworthiness as they support the controlled sharing of personal data among OPs and RPs, thereby helping to achieve three of the five privacy goals in Section 3.1 (namely Data Minimization, Confidentiality and Transparency) and comply with the GDPR.

To the best of our knowledge, no other parameter in our BCPs has been redefined or removed from the specifications.

5. Survey of OPs on the BCPs

We performed two distinct surveys of the adoption rate of the privacy BCPs in the wild. One was performed in September 2022 (which we refer to as Sep2022) and another in September 2025 (Sep2025). The key distinctions between the two (Table 2) lie in (*i*) the methodology to check compliance of the OPs with the BCPs and (*ii*) in the data source. For Sep2022 we manually tested top-ranked private OPs and notified national ones. Instead, the Sep2025 survey uses an automated methodology and analyzes a much larger set of private OPs. The automated methodology offers a scalable and user-friendly assessment tool for evaluating the privacy posture of OPs, enabling practitioners to extract essential information regarding OP configurations necessary for data protection and privacy compliance. Thus, it solves the Sep2022 limitation in the number of surveyed OPs. Below, we illustrate in detail the methodology, data source, and results of each survey.

5.1. Analysis methodology

In the following sections, we first briefly explain the methodologies behind or manual and automated OPs analysis followed by the data sources for the 2022 and 2025 experiments. After that, we provide the results of our analysis.

5.1.1. Manual analysis of OPs

During Sep2022, we assessed the support for the privacy BCPs through (*i*) the analysis of the documentation and (*ii*) manual testing of the available OPs deployments.

OPs may offer their services in different ways. Namely, they may provide either (*i*) a proprietary OP server and a separate developer console for client registration, or (*ii*) the code of a standalone OP service that has to be deployed and run in your environment of choice.

To perform the tests, in case the OP was of the first type, we accessed the OP through an online developer console and created an RP. If the OP was of the second type, we deployed the OP locally, meaning that we installed the OP software in an isolated local environment, and then created an RP with the interfaces the software provides. Subsequently, we performed a `response_type=code` flow (Sakimura et al., 2023a; Hardt, 2012), querying the authorization and token endpoints and verifying their answers.

The OIDC code flow is the default protocol flow, the results of the tests are applicable to all other flows supported by the OP. To test the implementation of selective disclosure in consent pages, we performed a full login flow. To assess the presence of access control policies, we tested the OP's access control configuration through the developer console. While this works for private OPs, the majority of national OPs demo software does not allow for the creation of testing RPs nor provide full-fledged developer consoles. Thus, we cannot test non-OIDC features, i.e., access control policies and selective disclosure in consent pages, in the majority of national OPs. Instead, we rely only on the manual testing of the OIDC code flow.

5.1.2. Automated analysis of OPs

Our latest analyses are performed with an automated and scalable methodology based on the examination of the OP metadata, which is extracted from the OP's discovery endpoint (Sakimura et al., 2023b). We implement a Python script (that the reader can find in our complementary website (Sassetti, 2025)) that queries the OPs discovery endpoints (Section 3.4) and checks for compliance with respect to the privacy BCPs. The compliance checks are done using JSON Schemas, which the reader can find in Appendix B. The *properties* (i.e., JSON key–value pair) provided in the metadata are defined in OpenID Connect Discovery (Sakimura et al., 2023b). For each BCP, we have found one or more properties in the discovery document that prove its implementation. Table 3 shows the connection between the BCPs, discovery document properties, and the tests that are performed to verify that the BCPs are implemented.

Each property is associated with a requirement level keyword (Bradner, 1997). Our methodology relies mainly on properties marked as *REQUIRED* and *RECOMMENDED* (Table 3). While the former keyword defines the properties that must be supported by the OP, the latter defines properties that have to be provided by the OP unless there are valid reasons not to do so. Consequently, OIDC-compliant OPs must always share required properties, and they must share recommended ones in the overwhelming majority of the cases.

We have two exceptions for which we use *optional* properties: `claims` and `acr_values`. Optional properties, as their name implies, do not need to be implemented by OPs. For `claims`, the Discovery specification explicitly states that in case `claims_parameter_supported` is missing in the discovery document, it is assumed that the parameter is not implemented. Thus, all OPs that implement `claims` and want RPs to use the parameter must also advertise it in the discovery document. As such, we simply consider all OPs that do not implement `claims_parameter_supported` to not implement `claims` either.

Regarding `acr_values`, the Discovery specification lacks such a clause that implies the non-implementation of `acr_values` in case `acr_values_supported` is missing. This may lead to some false negative results, as OPs may implement `acr_values` but not publish them in the discovery document. To overcome this hurdle, we query `claims_supported`, a recommended property in the discovery document. In case the OP supports `acr_values`, it is common practice to include `acr` inside `claims_supported`. Experimental evidence supports this point; however, since this practice is non-standard, it may lead to a smaller number of false negative results. For this reason, we carried out an estimate of the false negative rate for `acr_values`. To do so, we have selected at random 100 OPs that, according to our tests, did not support `acr_values`. We examined their documentation to find out whether they are supporting `acr_values`, and thus estimated the number of false negatives by generalizing the result for the whole dataset. Since also the testing of `vtr` relies on `claims_supported`, we have performed the same tests for it.

Our automated methodology applies only to the BCPs in Table 3. That is because the remaining two BCPs, *selective disclosure in consent pages* and *access control policy*, can only be tested manually. Unlike other features that can be automatically checked for compliance, they both lack clear guidelines for their implementation. The absence of standardized guidelines makes it exceedingly difficult to automate their analysis, since the implementation varies from OP to OP. Furthermore, access control policies cannot be tested through the OIDC authentication flows (Hardt, 2012; Sakimura et al., 2023a). Instead, they are usually set up through a developer console, which puts managing access control outside the scope of OIDC. Consequently, we are unable to provide updated results for these two BCPs during out latest surveys.

5.2. Data sources

For this study, we used two data sources. The first (Data Source 2022) was mainly used in our previous work (Sassetti et al., 2023) and later re-examined with the automated methodology (Section 5.1.2). The second (Data Source 2025) was instead mainly employed with the automated methodology. Both data sources are comprised of a set of private OPs and a set of national OPs. Below, we describe the data sources in detail.

5.2.1. Data source 2022

The private OPs in Data Source 2022 are gathered from the top 100 websites in the Alexa[1] ranking, selecting those that were OIDC-certified,[2] for a total of 14 private OPs. Instead, the national OPs were selected from the list of notified and pre-notified eIDAS identity providers.[3] Since the eIDAS regulation is not binding for the choice of technologies, providers implement mainly either SAML 2.0 or OIDC (Campbell et al., 2015; Sakimura et al., 2023a). From the list, we selected only those implementing OIDC, for a total of 13 national OPs. The list of OPs can be found in Table 4.

5.2.2. Data source 2025

Since the Alexa ranking was discontinued in 2022, we gather the private OPs in Data Source 2025 from the Well-Known Resource Index,[4] which is based on the widely accepted Tranco list[5] (Le Pochat et al., 2019). The Well-Known Resource Index selects the top 10,000 OPs based on traffic analysis and provides their address and discovery endpoint. For our survey, failure to answer our queries due to connection issues or server errors led to the exclusion from the dataset. Out of the 10,000 starting private OPs, 9483 were reachable and thus included in this study. As for the national OPs, we selected the same OPs from the updated manual survey provided in Section 5.2.1. We could not include in the Data Source 2025 the OPs that do not have a public discovery endpoint. All private OPs that were in Data Source 2022 are also included in Data Source 2025, with the exception of OPs that were discontinued, i.e., ForgeRock (Forgerock, 2021).

The majority of the national OPs in Data Source 2025 are the same as in Data Source 2022, with some exceptions. NemID (SignaturGruppen, 2023) and FranceConnect (République française, 2019) have been discontinued and replaced with newer OPs, i.e., MitID (Digitaliseringsstyrelsen, 2022) and FranceConnect+ (République française, 2021). Moreover, we added two national OPs to the dataset, i.e., the UK's NHS Care Identity Service (NHS, 2023) and Estonia's GovSSO (Riigi Infosüsteemi Ameti, 2025a). We believe it is worth investigating both OPs since they are new OIDC projects of countries whose OPs were already included in Data Source 2022 and are based on earlier eIDAS-compliant projects. Other countries in the list have not developed new OPs since 2022. The reader should note that in Sassetti et al. (2023) we called national OPs eIDAS OPs, as we focused on eIDAS-interfacing OPs. Since the addition of the new OPs, it is no longer accurate to describe them as eIDAS OPs. Notwithstanding, the reader should consider the group of national OPs representative of eIDAS OPs since it includes all eIDAS-notified OIDC solutions.

5.3. Changes in the BCPs between our two studies

During our first study in 2022, we included `purpose` among the BCPs. `purpose` is an authorization request parameter included in OIDC for Identity Assurance (Lodderstedt et al., 2024) that enhances transparency. It allows RPs to elicit the reason for requesting specific claims, pass it to the OP, which then displays it on the consent page. The responsibility of making the consent page transparent usually falls on the OP alone. OPs add to the consent page the list of scopes and claims that have been requested, along with a short description of each. The implementation of `purpose` creates an alternative way of ensuring user awareness and allows RPs to contribute to transparency.

Unfortunately, `purpose` was removed from the final version of OIDC for Identity Assurance, most likely due to its overlap in functionalities with the OPs' consent page descriptions (Section 4). Since it is no longer included in any specification, we removed `purpose` from the BCPs, and we kept it only as part of the Sep2022 results. However, we argue that the removal of `purpose` does negatively impact users, as RPs do not have any other way of communicating in an easy-to-read manner their data collection purposes. Currently, the corpus

[1] https://en.wikipedia.org/wiki/Alexa_Internet
[2] https://openid.net/certification/
[3] https://ec.europa.eu/digital-building-blocks/sites/display/EIDCOMMUNITY/Overview+of+pre-notified+and+notified+eID+schemes+under+national
[4] https://well-known.dev/
[5] https://tranco-list.eu/

Table 4
Summary of the supported BCPs for private and eIDAS OPs selected during our manual analysis of OPs. We provide also the updated results of our analysis in 2025 to show changes over time. Note that Forgerock, NemID, and FranceConnect were discontinued (†), and `purpose`, access control policies, and consent pages could not be tested in 2025 (⋆). GovSSO and NHS CIS were release after Sep 2022 and were not in the dataset at the time (‡).

Private OPs

	`scope`		`claims`		`verified_claims`		`purpose`⋆	`acr_values`		`vtr`		`pairwise`		Access control policies ⋆	Consent pages ⋆
	2022	2025	2022	2025	2022	2025	2022	2022	2025	2022	2025	2022	2025	2022	2022
Cloudentity (SecureAuth, 2022)	✓	✓	✓	✓				✓	✓			✓	✓	✓	✓
Google (Google, 2022)	✓	✓												✓	✓
Facebook (Meta, 2022)	✓	✓										✓	✓	✓	
Yahoo (Yahoo, 2025)	✓	✓							✓						
Microsoft (Microsoft, 2022)	✓	✓							✓			✓	✓	✓	
OKTA (OKTA, 2022)	✓	✓							✓					✓	
Auth0 (Auth0, 2022)	✓	✓						✓	✓					✓	
Authlete (Authlete, 2022)	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓		
Ping (Ping Identity, 2025)	✓	✓						✓	✓			✓	✓	✓	
Amazon (Amazon Web Services, 2025)	✓	✓												✓	
WSO2 (WSO, 2024)	✓	✓	✓	✓				✓	✓				✓	✓	✓
Connect2id (Connect2Id, 2022)	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
IBM (IBM, 2019)	✓	✓												✓	✓
ForgeRock†(Forgerock, 2021)	✓	NA	✓	NA		NA		✓	NA		NA	✓	NA	✓	

National OPs

	`scope`		`claims`		`verified_claims`		`purpose`⋆	`acr_values`		`vtr`		`pairwise`		Access control policies ⋆	Consent pages ⋆
	2022	2025	2022	2025	2022	2025	2022	2022	2025	2022	2025	2022	2025	2022	2022
itsme (itsme, 2021)	✓	✓	✓	✓				✓	✓			✓	✓		
MitID (Digitaliserings-styrelsen, 2022)	✓	✓	✓					✓	✓			✓			
FranceConnect+ (République française, 2021)	✓	✓	✓	✓				✓	✓			✓	✓		
Pro Santé Connect (Agence du Numérique en Santé, 2022)	✓	✓		✓				✓	✓			✓	✓		
TARA (Smart-ID) (Riigi Infosüsteemi Ameti, 2025b)	✓	✓						✓	✓						
GovSSO‡ (Riigi Infosüsteemi Ameti, 2025a)	NA	✓	NA		NA		NA	NA	✓	NA		NA		NA	NA
ID Austria (A-Sit Plus, 2022)	✓	✓		✓								✓	✓		
ID-Porten (Digdir, 2022)	✓	✓	✓	✓				✓	✓			✓	✓		
NHS Login NHS (2022)	✓	✓								✓	✓				
NHS Care Identity Service ‡ (NHS, 2023)	NA	✓	NA		NA		NA	NA	✓	NA		NA		NA	NA
MojeId	✓	✓	✓	✓				✓	✓			✓	✓		✓
SPID (Ministero dell'Interno, AgID, 2019)	✓	✓	✓	✓				✓	✓			✓	✓		
CIE (Ministero dell'Interno, 2015)	✓	✓	✓	✓				✓	✓			✓	✓		
NemID†(SignaturGruppen, 2023)	✓	NA	✓	NA		NA		✓	NA		NA	✓	NA		
FranceConnect†(République française, 2019)	✓	NA		NA		NA		✓	NA		NA	✓	NA		

of OIDC specifications does not include more features that enhance transparency, to the possible detriment of user privacy. The lack of such a mechanism also comes at the expense of the shared-responsibility model OIDC adopts: only the OP has control over transparency, while the RP cannot help increase the level of assurance of the system.

5.4. Results of Sep2022

The reader can find the survey results in Table 4. The results reported in this section roughly organized the BCPs based on their impact on privacy principles (Fig. 1). This organization allow us to provide a broader understanding of the results and ease the operationalization of the BCPs in light of the privacy principles they support.

Private OPs. We would like to highlight the following results that apply to private OPs.

- The BCPs for data minimization, namely `scope` and `claims`, are supported, respectively, in 100% and 35% of private OPs.
- The BCPs for confidentiality, `acr_values` and `vtr`, have been implemented in 50% and 0% of OPs respectively. Thus, `acr_values` is preferred to `vtr` despite the greater precision and flexibility granted by the latter. Additionally, 85.7% of private OPs enforce an access control policy, making it the second most implemented BCP.
- `verified_claims` and `purpose` both have a 14% implementation rate. This result shows that during Sep2022, OPs that wanted to increase data accuracy rarely implemented OIDC for Identity Assurance, but relied on `acr_values` instead.
- Regarding unlinkability, `pairwise` identifiers have been implemented in 50% of private OPs. To understand the trend in the implementation of `subject_types`, we investigated the implementation of `public` subject identifiers. We discovered that 85% of OPs implement `public`, while 35% of them implement both subject types. Interestingly, 15% of OPs allow only the use of `pairwise`. This should be considered a privacy-preserving design choice.

Table 5
Implementation rate of BCPs in private and national OPs in the two surveys.

BCP	Implementation rate			
	Sep 2022		Sep 2025	
	Private OPs	National OPs	Private OPs	National OPs
`scope`	100%	100%	82.5%	100%
`claims`	35.7%	61.5%	2.0%	61.5%
`verified_claims`	14%	0.0%	0.01%	0.0%
`acr_values`	50%	84.6%	2.7%	84.6%
`vot`	0.0%	7.5%	0.0%	7.5%
`pairwise` subject type	50%	84.6%	5.2%	61.5%

- 35% of private OPs implement selective disclosure in consent pages. Despite that, the remaining OPs meet adequate transparency by showing short descriptions of the requested `claims` and `scope` in consent pages.

We discovered that access control policies are managed in a variety of ways, which change between each OP. We identified a significant trend in private OPs (71%) allowing for the customization of access control policies on scopes. The purpose of customizing access control policies is to share claims only with a subset of RPs, or under certain conditions, such as a higher level of assurance.

Some OPs give the possibility to create access control policies from scratch, while others allow modifying default policies. This highlights the tendency of OPs to deliver access control as part of their service and grant developers a high level of customization. An example of such customization is allowing to flag scopes with security levels, defining a set of authentication requirements for each. The OP would then match the highest authentication level among the requested scopes before granting access to the resources. Furthermore, OPs manage users with user groups and they define different authentication requirements for each user group.

The request for stronger authentication upon accessing protected resources, usually through Multi-Factor Authentication, is called step-up authentication (Wilson and Hingnikar, 2019). We found that all the OPs that implement access control customization allow step-up authentication, which highlights the importance of this feature. As part of the access control customization, OPs allow developers to define which resources start the step-up process. Although it can also be the case that sensitivity levels cannot be changed by developers and a default set of scopes will always start the step-up flow. This limits the degree of customization but ensures a baseline security level.

Another viable solution for OPs is to make developers submit their RP for review. In this case, a dedicated team checks the RP activity and purpose. After the RP is deemed conforming to the provider policies, it can access protected resources. This solution is preferred by OPs that allow quickly creating and setting up RPs. In this way, developers can easily access basic functionalities, while sensitive scopes are protected by granting access only to verified RPs. Nevertheless, this solution suffers from a lack of scalability and flexibility and is therefore adopted only in 2 (14%) cases.

National OPs. We would like to highlight the following findings.

- Regarding data minimization, `scope` is implemented in all OPs and `claims` is implemented in 61% of OPs, which marks a sharp increase in comparison with private OPs.
- The results for the features that enforce confidentiality reflect the higher assurance requirements of national OPs. `acr_values` has been implemented in 92% cases, a steep increase compared to private OPs. Despite that, `vtr` is implemented only in one OP, confirming the same trend witnessed in private OPs. Unfortunately, we cannot test national OPs' access control policies (Section 5.1.1).
- The higher implementation rate of `acr_values` determines the overall higher support for data accuracy with respect to private OPs. That being said, `verified_claims` was never implemented, mirroring another trend of private OPs.
- `pairwise` is implemented in 84% of OPs, compared to 50% of private OPs. Interestingly, 46% of OPs implement `public`, marking a reduction in the implementation rate. At the same time, the number of OPs implementing `public` only, without `pairwise`, decreased to 14%, compared to 50% of private OPs, while those implementing only `pairwise` increased to 46%. The results could derive from the emphasis given to `pairwise` identifiers in the OIDC iGov profile (Varley and Grassi, 2024), and the higher privacy assurance requirements of national OPs.
- Transparency is frequently overlooked since `purpose` is never implemented and selective disclosure in consent pages is in only one OP. It is noteworthy that, while 30% of OPs provide a consent page with short `scope` descriptions, 46% of OPs forgot this practice. Instead, the user is redirected to the RP with the access code right after sending their credentials. This seemingly non-GDPR-compliant behavior can be explained with the following reasons: (*i*) The eIDAS regulation (European Parliament and Council of the European Union, 2014) defines the minimum set of user data that can be shared with RPs and that is needed for user accountability; (*ii*) Since it is required to share at least the minimum set of data with the RPs to access their services, logging in implies the user's consent for sharing the aforementioned data. This is possible only for eIDAS OPs; private OPs would otherwise violate the principles of transparency and purpose specification (European Parliament and Council of the European Union, 2016).

5.4.1. Survey update with automated methodology

It is worth examining how the support for our BCPs has changed since 2022 in the OPs of Data Source 2022. To this end, we showcase the results obtained with the automated methodology 5.1.2 on the OPs included in Sep2022. In cases in which the OPs did not provide public discovery endpoints, we used the endpoints of local OP deployments. Table 4 presents the results of the updated analysis, below we highlight some notable findings.

Besides the slight differences in the OPs included in this new analysis (Section 5.2), we have detected only a few changes in the supported BCPs between Sep2022 and Sep2025. Namely, we found that three private OPs (Yahoo (Yahoo, 2025), Microsoft (Microsoft, 2022), and OKTA (OKTA, 2022)) now implement `acr_values`, and WSO (WSO, 2024) supports `pairwise` subject identifiers. As for national OPs, we found that Pro Santé Connect and AustriaID now support `claims`. Additionally, we have detected significant change in the BCPs support for MitID. In particular, our results show that MitID does not support `claims` and `pairwise` anymore. This should be attributed to the changes in *brokers*. Being a *brokered* eID scheme (Digital ID and Authentication Council of Canada (DIACC), 2020; OpenID Foundation, Elizabeth Garber and Mark Haine (editors), 2023), MitID uses intermediary services to handle authentication requests (Digitaliseringsstyrelsen, 2023). Different brokers may offer different support for OIDC features (SignatureGruppen, 2023; Signicat, 2025). During the Sep2025 survey we referred to the same broker (Signicat) as in Sep2022, detecting a reduction in the support of our BCPs.

5.5. Results of sep2025

We present here the results of the survey obtained with our automated methodology (Table 5). Overall, we witness a slight decrease in the average implementation rate of the BCPs in national OPs between 2022 and 2025. However, this should be attributed to the dataset size reduction, rather than a decline of the national OPs' baseline privacy.

Remarkably, results show (*i*) the same trend witnessed in the 2022 survey of avoiding the implementation of OIDC for Identity Assurance

and `vtr`, and (*ii*) the low privacy standards of private OPs in the wild proven by the low implementation rate of all BCPs. We highlight the following notable results.

- In private OPs, the `scope` parameter is not implemented in 17.5% of the cases, meaning that the same number of OPs are not OIDC Core-compliant. This can cause issues in setting up communications with RPs, limiting the usability of the OPs. This may lead to violating the data minimization privacy goal, as it allows the RPs to obtain more user data than it is needed. Instead, all national OPs implement `scope` correctly. Regarding the `claims` parameter, private OPs show a remarkably low 2.0% implementation rate, while national OPs maintain the same implementation rate.
- Regarding confidentiality, national OPs maintain the same high implementation rate for `acr_values`, whereas that drops in private OPs to 2.7%. No private OP supports `vtr`, and only one national OP does.
- No national OP and almost no private OP implements `verified_claims`, meaning that they do not support the OIDC for Identity Assurance specification (Lodderstedt et al., 2024). Combined with the aforementioned results regarding `vtr`, that confirms the trends witnessed in the 2022 survey.
- Regarding unlinkability, we see a decrease in the support of `pairwise` in national OPs, going from 84.6% down to 61.5%. The drop in support is instead much sharper in private OPs, going from 50% down to 5.2%
- Unfortunately, we cannot provide any results for transparency BCPs in Sep2025, as our methodology does not allow to test consent pages.

To conclude, the implementation rate of `claims`, `acr_values`, and `pairwise` in private OPs is particularly noteworthy in light of the fact that they are 10 times lower than those observed during the 2022 survey. On top of that, the implementation rate of privacy-preserving BCPs is lower than 8% for 6 out of 7 BCPs, suggesting that private OPs may frequently overlook crucial privacy aspects such as unlinkability, data accuracy, confidentiality, and data minimization. While the observed gap between private and national OPs is substantial in the 2022 survey results, it is even more remarked in the 2025 survey. These findings highlights potential risks for users and underscores the need for enhanced baseline privacy measures in private OPs.

5.5.1. False negative rates

We estimated the number of false negative results for `acr_values` and `vtr` by randomly selecting 100 private OPs from Data source 2025 that resulted as not implementing `acr_values` and `vtr`. In order to verify whether these results were false negatives, we manually analyzed their documentation, as in Section 5.1.1. In the documentation, we specifically looked for statements of the implementation of either parameter or their mention in the API documentation. We found no false negative result for both `acr_values` and `vtr` features.

6. Specification compliance

We have performed an additional OPs survey aimed at finding instances of non-compliance with respect to OIDC Core and Discovery (Sakimura et al., 2023a,b). This means, for instance, not respecting the conditions and constraints specified in an OIDC specification. Below, we report the methodology and data source of this survey, however the reader can notice they are similar to the ones used for Sep2025 (Section 5.1.2).

6.1. Methodology

The automated methodology presented in Section 5.1.2 can be generalized and extended to survey other parameters and test their compliance with OIDC specifications. Testing the discovery endpoint is part of the compliance verification with other OIDC specifications, and it allows us to check that OPs implement specific parameters and associate the right values with them. For instance, as per OIDC Core, all OPs must support the value `openid` in their `scopes_supported`, which can be verified through the discovery document.

During our 2025 analysis, we verify the OPs' compliance with the OIDC Core and OIDC Discovery specifications. We first identify the properties in the discovery document that are required for complying with the aforementioned specifications. Additionally, we verify whether the OPs implement both the recommended and optional properties. To define tests to verify that the properties are non-empty and contain correct values. Table 6 reports the full list of selected properties with the tests we performed.

The automated methodology is thought to find instances of non-compliance since it verifies that all required parameters are included in the discovery document and configured correctly. Its main limitation lies in the fact that it only accesses the metadata published in the discovery endpoint. For this reason, it is not suited to assess the complete compliance with respect to OIDC Core and is limited to finding non-compliant and misconfigured OPs.

6.2. Data source

For this survey, we use Data Source 2025 (Section 5.2.2).

6.3. Results of the specification compliance analysis

In Table 7, we report the results of the OPs compliance with the OIDC Core and Discovery specifications. The reader can find in the table both the implementation rate of the single discovery document parameters and the results of the tests following what is defined in Table 6. We did this to demonstrate that there are cases in which an OP does include a certain parameter inside the discovery document but fails to implement it correctly. In particular, we highlight the following results.

- Private and national OPs have similar implementation rates for all properties. However, national OPs passed the tests (Table 6) in almost all cases. In contrast, private OPs implemented the properties, but failed the tests for `response_types_supported`, `id_token_signing_alg_values_supported`, and `scopes_supported` in approximately 12% of the cases.
- Overall, 22.5% of private OPs and 30% of national OPs are found non-compliant with OIDC Core and Discovery, while only 42% and 40% respectively, correctly implement the recommended properties. Almost no OP correctly implemented all the optional features, however this result was to be expected given that optional properties are designed to handle specific use cases and scenarios, and hardly any OP would need to implement them all.
- 15.8% of private OPs provide no subject type and consequently are non-compliant. This can seriously hamper the functionality of the protocol, as RPs have to assign identifiers to the users themselves, leading to inconsistent identity information.
- 0.2% of private OPs provide at least one URI using HTTP instead of HTTPS and consequently are non-compliant, while this is never the case for national OPs. We must remark that using HTTPS is fundamental to ensure the user security and privacy. We also found that among the HTTP URIs, 9 were token endpoints and 10 were userinfo endpoints. This exposes the endpoints to data leak and impersonation threats, which can have severe impact on the user's privacy.

Table 6
Tests performed during the compliance survey (Section 6). Test failure implies non-compliance with OIDC Core and Discovery. † marks required properties, ‡ marks recommended properties. *`token_endpoint` is required unless only the Implicit Flow is used.

Attribute	Type of test	Test valid for
`scopes_supported`‡	Verify presence of `openid` value	OIDC Core, Discovery
`claims_supported`‡	Verify it is a JSON list	OIDC Discovery
`subject_types_supported`†	At least one value between `public` and `pairwise` must be supported	OIDC Core, Discovery
`token_endpoint`†*	If not implemented, `response_types_supported` must support the value `id_token token`	OIDC Core, Discovery
`id_token_signing_alg_values_supported`†	Must allow RSA Signature with SHA-256 (identified with RS256)	OIDC Core, Discovery
`response_types_supported`†	If the OP implements `registration_endpoint`, then it must include the values `code`, `id_token`, and `id_token token`	OIDC Core, Discovery
`issuer`† `authorization_endpoint`† `jwks_uri`† `token_endpoint`†* `userinfo_endpoint`‡ `registration_endpoint`‡	Must be in HTTPS format	OIDC Core, Discovery
Additional tests		
Required properties	Must all be implemented	OIDC Discovery
Recommended properties	Must all be implemented	OIDC Discovery
Optional properties	Must all be implemented	OIDC Discovery

Table 7
Results of the specification compliance analysis performed through the discovery document parameters applying the tests in Table 6 (survey conducted in 2025). Note that we are investigating both whether parameters are present in the discovery document (implementation rate) and also the results of the tests.

Property	Implementation rate		Test passed	
	Private OPs	National OPs	Private OPs	National OPs
`issuer`	100%	100%	99.98%	100%
`authorization_endpoint`	96.3%	100%	96.2%	100%
`token_endpoint`	96.7%	100%	93.2%	100%
`jwks_uri`	99.7%	100%	99.6%	100%
`response_types_supported`	95.8%	100%	85.9%	84.6%
`id_token_signing_alg_values_supported`	95.4%	100%	91.6%	69.2%
`subject_types_supported`	84.2%	100%	84.1%	100%
`scopes_supported`	93.1%	100%	89.1%	100%
`userinfo_endpoint`	52.2%	84.6%	52.2%	84.6%
`registration_endpoint`	33.2%	30.7%	33.2%	30.7%
`claims_supported`	86.6%	84.6%	86.5%	84.6%
OIDC Discovery Compliance by keyword (test only)				
Correctly implement all **required** properties	NA	NA	87.2%	61.5%
Correctly implement all **recommended** properties	NA	NA	67.2%	30.7%
Correctly implement all **optional** properties	NA	NA	0.0%	0.0%

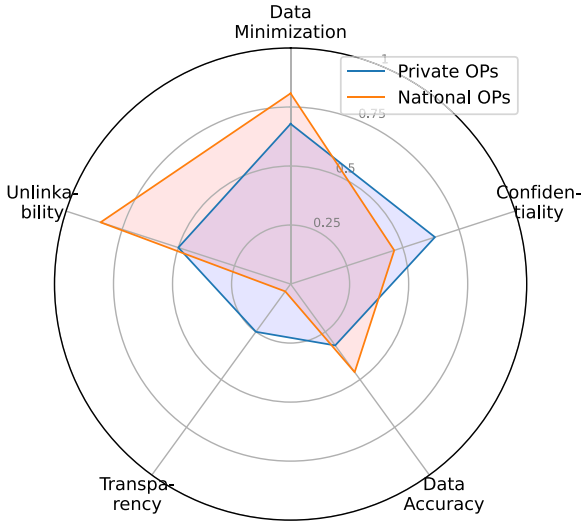

Fig. 3. Sep 2022 - Average number of BCPs implemented for each privacy principle, normalized by the number of BCPs for privacy principle.

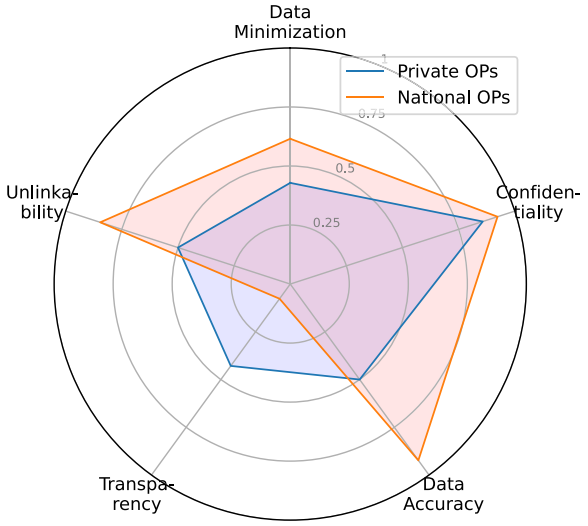

Fig. 4. Sep 2022 - Percentage of OPs implementing at least one BCP for each privacy principle.

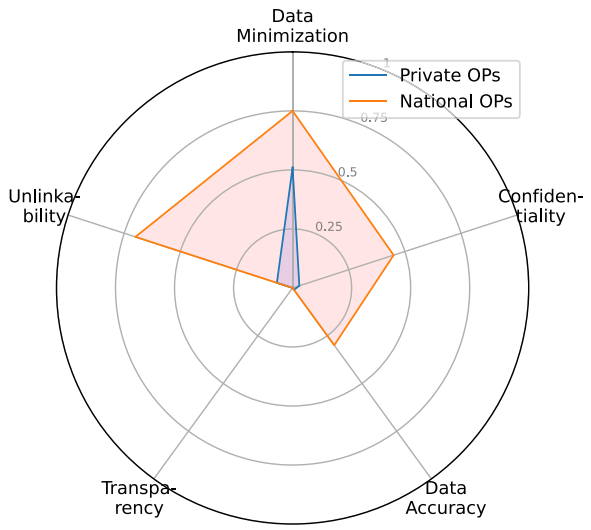

Fig. 5. Sep2025 - Average number of BCPs implemented for each privacy principle, normalized by the number of BCPs for privacy principle.

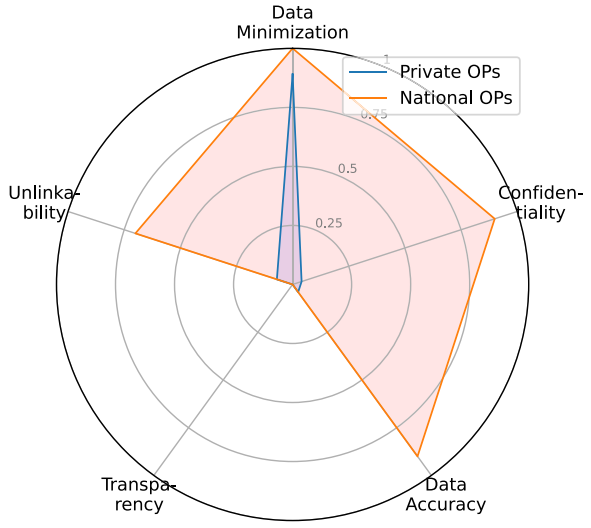

Fig. 6. Sep2025 - Percentage of OPs implementing at least one BCP for each privacy principle.

7. Discussion and further considerations

We present below some further analysis of the data retrieved during the surveys, alongside the plausible reasons behind our findings, using experimental evidence to corroborate our claims.

7.1. Further analysis of Sep2022

During Sep2022 we have found a significant discrepancy between private and national OPs, with the latter having on average higher baseline privacy and implementing more of our OIDC BCPs. 42% of private OPs and 61% of national OPs implement half or more of our BCPs. The difference between the two groups is further highlighted if we consider only the OIDC BCPs. In this case, 42% private OPs implement at least half of the features, whereas 84% national OPs do.

The same conclusions can be drawn by comparing the average implementation rate of the BCPs for a given privacy principle (Fig. 3) and the percentage of OPs implementing at least one BCP for each privacy principle (Fig. 4) between the two groups.

The figures also show that private and national OPs prioritize the privacy principle differently. On average, national OPs offer higher data minimization, unlinkability, and data accuracy (Fig. 3). Conversely, private OPs implement the non-OIDC BCPs more frequently, thus providing higher transparency and confidentiality than national OPs. However, this is only due to the higher implementation rate of non-OIDC features (Section 4). Taking only the OIDC BCPs into consideration, national OPs would outperform private OPs in the support of all privacy principles.

We can also see that a subset of OIDC features, namely `purpose`, `verified_claims`, and `vtr`, are implemented by almost no private and national OP. Moreover, 35% of private OPs have implemented only the mandatory features (`scope` and `public` subject types), whereas all national OPs implement at least one non-mandatory BCP.

The results reveal a very low adoption rate for OIDC for Identity Assurance (Lodderstedt et al., 2024) in both private and national OPs. This can be attributed to the fact that the specification was finalized only in late 2024, and underwent significant changes during its development. It is plausible that OPs were waiting for its finalization before adoption. Nonetheless, the updated results (Table 5) raise questions about the likelihood of widespread implementation OPs in the near future. We believe it is more likely that OPs designed for users with high data accuracy needs, such as trust services, will adopt the OIDC for Identity Assurance.

As for the OIDC iGov profile, we found that 35% private OPs and 61% national OPs comply with it. However, no OP implements all the BCPs introduced in the specification. That is due, in most cases, to the failure to adopt `vtr` Interestingly, new specifications for public infrastructures have been derived from the iGov profile, such as the Dutch Gov Assurance Profile (Logius, 2023) and the Italian SPID and CIE OIDC profile (Ministero dell'Interno, AgID, 2019). This represents an opportunity for development and improvement of privacy posture within their OIDC deployments, applied to specific use cases, such as public administration that demands a higher level of privacy.

During Sep2022 we found that often private OPs enforce access control policies on scopes rather than claims (Section 5.4). Additionally, they associate a default set of claims with each scope, thus not requiring `claims` for requesting resources. This may explain the low implementation rate of `claims`. Despite that, we recommend its implementation as it is pivotal for enhancing data minimization.

With our first survey (Sassetti et al., 2023), we proposed a methodology for grading the general level of assurance granted by each OP. For each OP, we provided a grade for each privacy principle based on the BCPs that were implemented. The results of some selected OPs are shown in Fig. 7 and 8. The results of our entire survey can be found in our complementary website (Sassetti, 2025). The results show a homogeneity in the assurance level of national OPs, caused by their similar requirements, and a greater diversity for private OPs, which often do not refer to the same set of requirements. Moreover, we can differentiate between two groups of private OPs, one of which provides a much higher assurance level than the other. We cannot make such a distinction for national OPs.

7.2. Further analysis of sep2025

The results of the automated survey (Table 5) indicate that private OPs demonstrate remarkably low implementation rates for all BCPs, thereby raising concerns for the achievement of the privacy principles. Consequently, the overwhelming majority of private OPs have a very low privacy profile. More precisely, we highlight that 84% of private OPs have implemented only the mandatory OIDC features (`scope` and `public` subject type). Considering that the same analysis on the Sep2023 dataset yielded 35% of OPs implementing only mandatory features (Section 7.1), we witness a sharp increase in the number of OPs implementing only the bare minimum OIDC features and thus

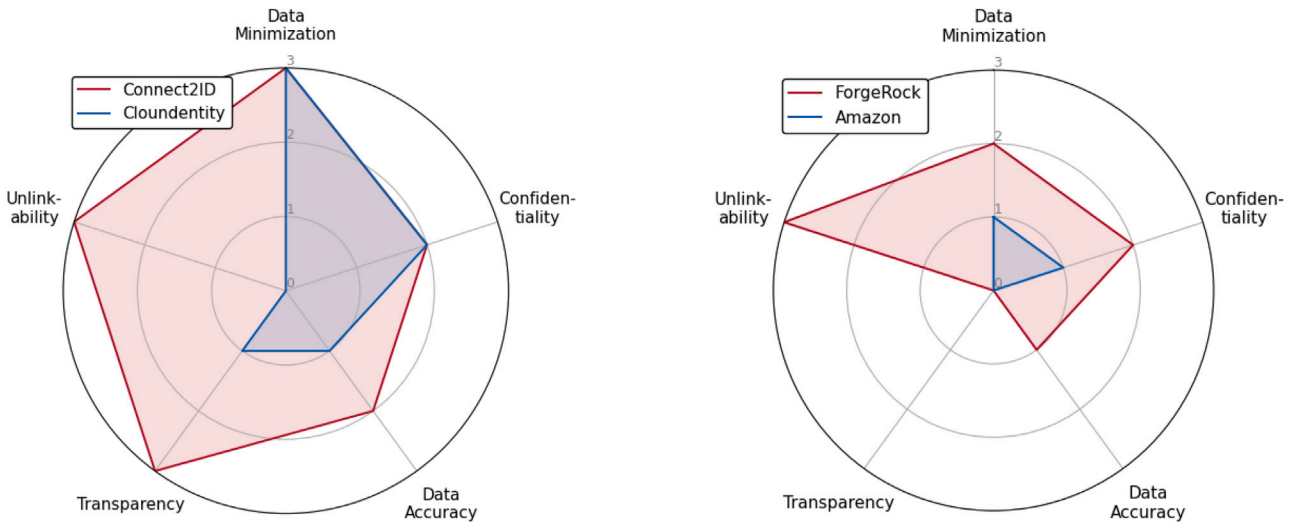

Fig. 7. Comparison of different private OPs with respect to the assurance level observed during Sep2022.

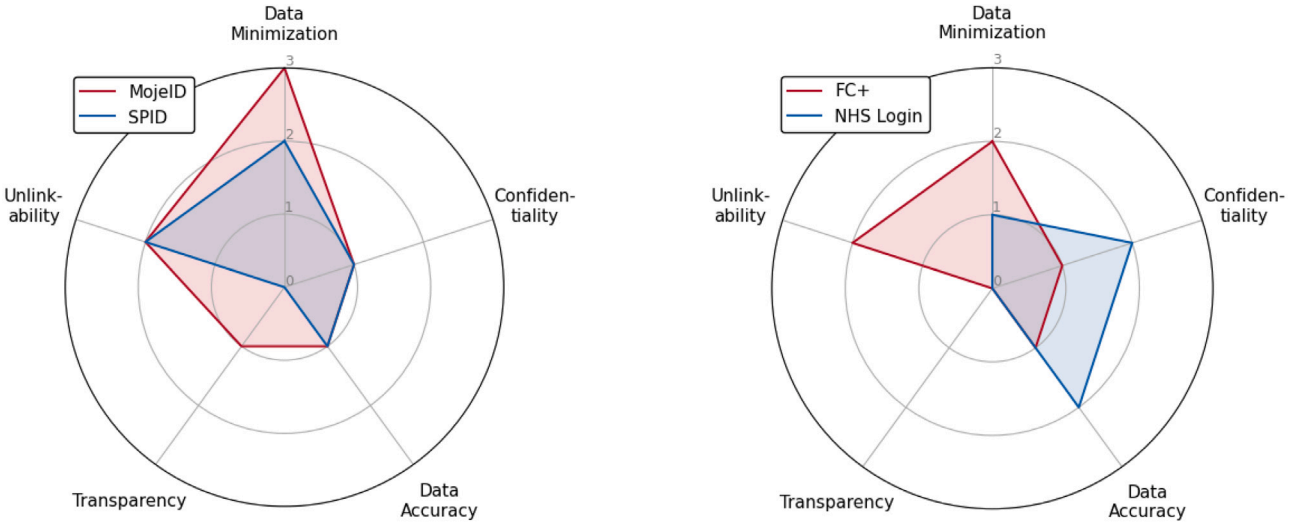

Fig. 8. Comparison of different national OPs with respect to the assurance level observed during Sep2022.

maintaining the lowest possible privacy profile for an OIDC-compliant OP.

Furthermore, we observe in private OPs a remarkably low investment in confidentiality, data accuracy, and unlinkability (Figs. 5 and 6). These findings underscore the suboptimal baseline privacy of private OPs and highlight a widespread oversight of privacy principles among them.

On top of that, we witness a substantial difference between the results of Sep2022 and Sep2025, which the reader can see by comparing Figs. 3 to 5, and Figs. 4 to 6. The reasons behind such a sharp contrast are two. First, the dataset used in Sep2025 is much larger than that of Sep2022. As such it captures a more accurate snapshot of the OIDC privacy landscape. Second, the private OPs in Sep2022 are all leading identity providers, certified by the OpenID Foundation.[6] The 2025 dataset is a superset of that of 2022 but includes many OPs that do not uphold the same standards. We consider this an added value, as the latter 2025 dataset allows us to understand that only a few OPs in the wild invest in user privacy and OIDC compliance.

In terms of transparency, we have no available data for Sep2025 as the `purpose` authorization request parameter has been removed from OIDC for Identity Assurance 5.3, and the adopted methodology does not include the testing of consent pages.

We encourage OPs to adopt our BCPs, as they are designed to enhance privacy, and, to the best of our knowledge, no other set of OIDC features offers this functionality. Additionally, increased user awareness may act as a complementary driver of adoption. By increasing transparency around privacy practices in the OIDC ecosystem, our contribution enables users to make more privacy-conscious choices, thereby creating incentives for OPs to align with higher privacy protections through the adoption of the BCPs.

7.3. Further analysis of the compliance survey

The results in Section 6 (see Table 7) indicate that a large subset of both private and national OPs are non-compliant with OIDC Core and Discovery. The prevalent causes of non-compliance are the lack of required properties (16%) and misconfiguration of properties (17%). Non-compliance decreases interoperability and hampers the ability of RPs to establish connections with the OPs and may raise security and privacy issues.

Furthermore, 0.16% of OPs are non-compliant due to the use of HTTP for their endpoints; this issue is not witnessed in any national

[6] https://openid.net/developers/certified-openid-connect-implementations/

OP. Initiating a connection with HTTP poses significant risks to all parties involved in the protocol, i.e., RP, OP, and the user. Connecting to the authorization endpoint with HTTP can be exploited to steal the RP credentials and impersonate it; using HTTP for token endpoint and user info endpoint requests can result in a breach of confidentiality, as well as RP impersonation. Thus, securing the communication channels with HTTPS is pivotal for ensuring security and privacy and cannot be overlooked by any OP. We investigated this issue further to understand whether these vulnerabilities could be exploited. We have found that two OPs have endpoints vulnerable to SSL stripping, effectively allowing for a man-in-the-middle attack. The remaining OPs perform an HTTP redirect on HTTPS but fail to enforce HTTP Strict Transport Security preloading. Thus, the first request to the endpoint is in HTTP, enabling an attacker to access its contents. This may allow for RP impersonation and data leak if the attacker can read the `client_id`, the `authorization_code`, or the `access_token`. Additionally, an attacker may force the connection to remain in HTTP or redirect the user to a third-party website, exposing the user to phishing and spoofing threats. We remark that the use of HTTPS is required in both OIDC Core (Sakimura et al., 2023a) and the OAuth Security Best Practices (Lodderstedt et al., 2025) and is fundamental to prevent such exploits.

The findings also show that less than half of private and national OPs (see Table 7) have implemented the OIDC Discovery recommended properties. OPs must implement recommended properties unless they are subject to particular conditions for which they provide *valid reasons*, which have to be weighted against the drawbacks of the property omission, as stated in RFC 2119 (Bradner, 1997). The main reason for not choosing to support a recommended attribute, such as the `registration_endpoint`, is that its policy does not permit dynamic registration of RPs because a vetting process is required. Similarly, OPs may not support the userinfo endpoint, as the user data can be retrieved through the ID token as well. Although that should be carefully weighted in light of the tradeoffs with data minimization, which is not enforced through the ID token. This practice is therefore discouraged unless OPs share an identical, static set of data with the ID token and the userinfo endpoint, which is itself discouraged.

The results indicate that the majority of OPs do not adhere to the definition of recommended properties and opt to not implement them. A considerable portion of both private and national OPs do not implement registration and userinfo endpoint (Table 7). More significantly, a similarly large group of OPs does not implement `scopes_supported` and `claims_supported`. While it can be argued that an OP may not want to implement the registration and userinfo endpoints as mentioned earlier, the same does not hold for the latter two properties. Instead, they are necessary for RPs to understand what data the RPs can request, thus helping RPs in establishing connections and avoiding instances of data oversharing. We remark that in no instance is it unsafe to state through the discovery document what data properties can be requested from the OPs. Therefore, we believe that all OPs should implement `scopes_supported` and `claims_supported`. Our point is further confirmed by the fact that the OIDC iGov profile requires the implementation of both.

We also investigated the implementation rate of `acr_values_supported` and `acr` inside `claims_supported`. In particular, `acr_values_supported` is an optional attribute with which OPs elicit the different kinds of values associated with authentication levels that can be requested through `acr_values`. `acr_values_supported` has no implementation guidelines and, together with the presence of `acr` inside `claims_supported`, has no impact on the compliance of the OP with the OIDC specifications. Their presence in the discovery document, however, provides further insight on the OPs' practices.

We found that only 46.7% of OPs implementing `acr_values` implement `acr_values_supported`, whereas 75.3% implements `acr` inside of `claims_supported`. This shows a consolidated practice in OPs to avoid publishing the supported acr values in their discovery document, which forces developers to find them manually, hampering the ease of use of the OP. As with the recommended properties, implementing `acr_values_supported` does not pose any risk to any stakeholder in the system, and we see no reason to avoid its implementation. Along with that, we found a gap in the OIDC Core and Discovery specifications: they do not require OPs that implement `acr_values` to include `acr_values_supported` in their discovery document. This gap has led to the aforementioned practice, which we discourage.

Interestingly, we found that only 46.7% of OPs implementing `acr_values` implement `acr_values_supported`, whereas 75.3% implements `acr` inside of `claims_supported`. This shows a consolidated practice in OPs to not publish the supported acr values in their discovery document, which forces developers to find them manually, hampering the ease of use of the OP. As with the recommended properties, implementing `acr_values_supported` does not pose any risk to any stakeholder in the system, and we see no reason to avoid its implementation. Along with that, we found a gap in the OIDC Core and Discovery specifications: they do not require OPs that implement `acr_values` to include `acr_values_supported` in their discovery document. This gap has led to the aforementioned practice, which we discourage.

During Sep2025, we have also collected additional data regarding the scopes supported by different OPs of Data Source 2025. In total we have found 12,321 unique scopes, with 10,528 (85%) being supported only by one OP. Examples of the scopes used by only one OP include `wallets`, `shields`, `wallets`, `debicheck-service`, and `client-config:read`, which are non-standard and are clearly created to access platform-specific resources. Although this result does not impact privacy or compliance, it showcases an interesting trend in OPs creating their own customized set of scopes.

We conclude by suggesting the following practices for all OPs based on the results discussed in this section. We suggest:

- the implementation of privacy-enhancing features, as the vast majority of private OPs meet only the minimum OIDC requirements. We must remark that implementing features such as `claims` and `pairwise` is undemanding and straightforward. The reader can find code snippets for their implementation in Appendix A;
- a thorough testing of all properties and parameters before the deployment of an OP. In particular, tests should be performed to check that required properties are implemented and that all properties are correctly configured. In Appendix B we provide JSON schemas that can be used to validate the configuration of discovery documents;
- the implementation of recommended properties in the discovery document, and specifically `scopes_supported` and `claims_supported`;
- the implementation of `acr_values_supported` when the OP implements `acr_values`, and discourage the practice of including `acr` in `claims_supported` without implementing `acr_values_supported`.

8. Conclusion and future work

This work presents an updated series of privacy BCPs, which are extracted from the official OIDC specifications and implementation trends, identifying easy-to-deploy privacy-enhancing features that strengthen the OIDC deployments baseline privacy without altering protocol or compromising interoperability and survey their adoption in real-world scenarios.

The BCPs offer actionable solutions to address the privacy requirements identified in the GDPR and eIDAS. To evaluate the adoption rate of our identified BCPs in the wild, we conducted two surveys: one in September 2022 and another September 2025.

During the first survey, we perform a manual review and testing of the OPs, whereas we automate the process and expand the pool of surveyed OPs to a dataset of 10,000 in the wild. The results show that some BCPs, i.e., `verified_claims` and `vtr`, are rarely implemented, indicating a significantly low baseline privacy for private OPs in the wild. While the situation for national OPs are quite better as in average, they implement a higher number of identified BCPs and consequently provide a much higher baseline privacy.

Furthermore, we present how our automated methodology can be generalized to facilitate compliance checks with other OIDC specifications. Thus, we conduct a third survey aimed at finding instances of non-compliance with OIDC Core and OIDC Discovery. The results show that a large portion of private OPs are not OIDC-compliant, often due to misconfiguration. We have also discovered a number of bad practices, such as the disregard for the implementation of recommended discovery document properties, or the use of `acr` inside of `claims_supported` instead of `acr_values_supported`. This may lead to not implementing the privacy features by RPs as a result of these bad practices that made the features not discoverable, and we provided actionable solutions to fix them.

Our findings highlight the critical role of combining actionable BCPs with automated framework for compliance assessment to enhance baseline privacy and support the ongoing privacy compliance assessment of OIDC deployments.

Future work. We are currently exploring a new research avenue where RPs' privacy and security practices are considered. In particular, we want to broaden the scope of this survey by including RPs and provide a snapshot of BCPs compliance within their implementation. In order to be effective, most of our BCPs must be supported by OPs but also used by RPs. In other words, OPs and RPs share responsibility for privacy.

We also plan on developing a browser extension that would enable the survey of RPs. The extension will capture the RP's requests to verify that the BCPs are used and show users a combined privacy assessment of the RP and OP. While the development of this solution is still in the early stages, the first prototype shows promising results. The extension may contribute to raising user awareness of privacy, enabling them to demand stronger privacy protection from both OPs and RPs.

We believe that in this way we can help provide further insights on the OIDC privacy landscape and help preserve user privacy in the shared responsibility model.

CRediT authorship contribution statement

Gianluca Sassetti: Writing – original draft, Methodology, Investigation, Data curation, Conceptualization. **Amir Sharif:** Writing – review & editing, Supervision, Project administration, Methodology. **Giada Sciarretta:** Writing – review & editing. **Roberto Carbone:** Writing – review & editing. **Silvio Ranise:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gianluca Sassetti reports financial support was provided by Italian Government Printing Office and Mint. Gianluca Sassetti reports financial support was provided by SERICS. Gianluca Sassetti reports financial support was provided by Ministero delle Imprese e del Made in Italy. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported in part by a joint laboratory between FBK and the Italian Government Printing Office and Mint, by the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union – Next Generation EU, and by the Ministero delle Imprese e del Made in Italy (IPCEI Cloud DM 27 giugno 2022 – IPCEI-CL-0000007) and European Union (Next Generation EU).

Appendix A. Code snippets

```
"""
Pairwise Subject Identifier (PPID) generator for
     OpenID Connect Core.

Implements the recommended algorithm from OIDC
    Core 8.1 (Pairwise Subject
Identifier Algorithm):

    sub = base64url( SHA-256( sector_identifier
    || 0x00 || local_sub || 0x00 || salt ) )

Notes:
- sector_identifier:
    If client metadata includes `
    sector_identifier_uri`, use its host.
    Otherwise, use the common host of all
    registered redirect_uris (must be identical)
    . Normalize to lower-case host, punycode-
    encoded, no port, no trailing dot.
- local_sub: the OP's stable, internal subject
    identifier for the end-user.
- salt: a secret, unguessable per-user (or per-
    OP) salt of at least 16 bytes.
- Output is base64url without padding, 43 chars
    for SHA-256.

Security guidance:
- Generate salt with `secrets.token_bytes(32)`
    and store it securely; reuse it for
  the same user to keep PPIDs stable across time
    .
- Do not share salts with RPs. Treat salts like
    credentials.
- If you need salt rotation, version salts and
    support re-issuance/lookup.
"""

from __future__ import annotations

import base64
import hashlib
import secrets
from typing import Iterable, Optional
from urllib.parse import urlparse


def _base64url_no_pad(data: bytes) -> str:
    return base64.urlsafe_b64encode(data).rstrip
    (b"=").decode("ascii")


def _canonicalize_sector_host(value: str) -> str
    :
    """
    Return the canonical sector identifier host:
      - extract host (ignore scheme, path, query
    , port)
```

```
      - lower-case
      - IDNA (punycode) for internationalized names
      - strip trailing dot
    Accepts either a full URI or a bare host[:port].
    """
    v = value.strip()
    parsed = urlparse(v if "://" in v else f"//{v}")
    host = parsed.hostname
    if not host:
        raise ValueError(f"Invalid sector identifier input: {value!r}")
    # IDNA encode then decode to ASCII, lower-case
    host_ascii = host.encode("idna").decode("ascii").lower().rstrip(".")
    return host_ascii


def derive_sector_identifier(
    redirect_uris: Iterable[str],
    sector_identifier_uri: Optional[str] = None,
) -> str:
    """
    OIDC Core sector identifier derivation:
      - If sector_identifier_uri is present, use its host.
      - Else, all redirect_uris must share the same host; use that host.
    Returns the canonicalized host string (e.g., 'example.com').
    """
    if sector_identifier_uri:
        return _canonicalize_sector_host(sector_identifier_uri)

    hosts = { _canonicalize_sector_host(uri) for uri in redirect_uris }
    if len(hosts) != 1:
        raise ValueError(
            "All redirect_uris must have the same host when sector_identifier_uri is absent."
        )
    return next(iter(hosts))


def generate_pairwise_sub(
    sector_identifier_host: str,
    local_sub: str,
    salt: bytes,
) -> str:
    """
    Compute pairwise 'sub' for a (sector, user) using the OIDC-recommended hash construction.

    Inputs:
      - sector_identifier_host: canonical sector host (use derive_sector_identifier or _canonicalize_sector_host)
      - local_sub: OP-internal stable subject identifier for the end-user
      - salt: secret, unguessable bytes (>= 16 bytes). Keep and reuse for the same user.

    Returns:
      - base64url (no padding) string suitable for 'sub' claim.
    """
    if not isinstance(salt, (bytes, bytearray)) or len(salt) < 16:
        raise ValueError("salt must be >= 16 random bytes")

    sector_b = sector_identifier_host.encode("utf-8")
    local_b = local_sub.encode("utf-8")
    # Use explicit 0x00 delimiters to avoid ambiguity
    material = sector_b + b"\x00" + local_b + b"\x00" + bytes(salt)
    digest = hashlib.sha256(material).digest()
    return _base64url_no_pad(digest)


# -------------------------
# Example usage
# -------------------------
if __name__ == "__main__":
    # Example client metadata (typical registration)
    redirect_uris = [
        "https://rp.example.com/cb",
        "https://rp.example.com/oidc/return",
    ]
    # If present, use sector_identifier_uri instead:
    sector_identifier_uri = None  # e.g., "https://rp-sector.example.org/sector.json"

    sector_host = derive_sector_identifier(redirect_uris, sector_identifier_uri)

    # Your OP's stable, internal user id
    local_sub = "1e7f6c3a-8b8b-4d2d-9f81-8e6b3b5da5b2"

    # Generate or look up a per-user salt stored in your database (reuse for that user)
    user_salt = secrets.token_bytes(32)  # store this; do not regenerate each call

    pairwise_sub = generate_pairwise_sub(sector_host, local_sub, user_salt)
```

Listing 2: Function for generating pairwise subject identifiers.

```
import json
from urllib.parse import parse_qs

def parse_authorization_request(authorization_request):
    """
    Parse the authorization request and extract the claims parameter.

    Args:
        authorization_request: The authorization request as a query string.
    Returns:
        A dictionary containing the requested claims.
    """
    # Parse the query string into a dictionary
    params = parse_qs(authorization_request)

    # Extract the 'claims' parameter if it exists
    claims_json = params.get('claims', ['{}'])[0]  # Default to '{}' if not present
```

```
    # Parse the claims JSON into a Python dictionary
    claims = json.loads(claims_json)

    return claims

# We cannot implement this function fully since it is depends on the OP's backend implementation, this is mockup code that can be used as an example
def get_user_claims(requested_claims, authorization_request, local_user_id):
    """
    Return the user's claims based on the requested claims.

    Args:
        requested_claims: A dictionary containing the requested claims
        authorization_request: The authorization request sent by the RP
        local_user_id: Unique identifier of the user at the OP, must not be shared with RP
    Returns:
        A dictionary containing the filtered user claims.
    """

    # Defining enforce_access_control is outside the scope of this work, it is important to note that it MUST NOT allow access to claims the RP is not authorized to receive
    authorized_claims = enforce_access_control(requested_claims, authorization_request)

    return retrieve_claims(authorized_claims, local_user_id)
```

Listing 3: Code for parsing claims in the authorization request and returning the desired claims.

Appendix B. JSON schemas for testing discovery document implementation.

```
{
    "type" : "object",

    "properties" : {

        "issuer": {"type" : "string", "pattern" : "^https:\/\/(www.)?[a-zA-Z0-9]{2,}(.[a-zA-Z0-9]{2,})(.[a-zA-Z0-9]{2,})?"},

        "authorization_endpoint": {"type" : "string", "pattern" : "^https:\/\/(www.)?[a-zA-Z0-9]{2,}(.[a-zA-Z0-9]{2,})(.[a-zA-Z0-9]{2,})?"},

        "token_endpoint": {"type" : "string", "pattern" : "^https:\/\/(www.)?[a-zA-Z0-9]{2,}(.[a-zA-Z0-9]{2,})(.[a-zA-Z0-9]{2,})?"},

        "jwks_uri": {"type" : "string", "pattern" : "^https:\/\/(www.)?[a-zA-Z0-9]{2,}(.[a-zA-Z0-9]{2,})(.[a-zA-Z0-9]{2,})?"},

        "response_types_supported": {"type" : "array",
                                     "contains": {
                                         "const": "code"
                                     }},

        "subject_types_supported": {"type" : "array",
                                    "contains": {
                                        "anyOf": [
                                            { "const": "public" },
                                            { "const": "pairwise" }
                                        ]
                                    }},

        "id_token_signing_alg_values_supported": {"type" : "array",

        "contains": {

            "const": "RS256"

        }}
        },

        "required": ["issuer","authorization_endpoint","token_endpoint","jwks_uri","scopes_supported","response_types_supported","subject_types_supported","id_token_signing_alg_values_supported"]
}
```

Listing 4: JSON schema for checking that required attributes are correctly configured in the discovery document.

```
{
    "type" : "object",

    "properties" : {

        "userinfo_endpoint": {"type" : "string", "pattern" : "^https:\/\/(www.)?[a-zA-Z0-9]{2,}(.[a-zA-Z0-9]{2,})(.[a-zA-Z0-9]{2,})?"},

        "registration_endpoint": {"type" : "string", "pattern" : "^https:\/\/(www.)?[a-zA-Z0-9]{2,}(.[a-zA-Z0-9]{2,})(.[a-zA-Z0-9]{2,})?"},

        "claims_supported": {"type" : "array"},

        "scopes_supported": {"type" : "array",
                             "contains": {
                                 "const": "openid"
                             }
        },

    "required": ["userinfo_endpoint","registration_endpoint", "claims_supported", "scopes_supported"]
}
```

Listing 5: JSON schema for checking that recommended attributes are correctly configured in the discovery document.

Data availability

I have included links to all data and material in the manuscript.

References

A-Sit Plus, 2022. AUSTRIA ID OIDC documentation. https://eid.egiz.gv.at/wp-content/uploads/2021/10/ID-Austria-Technisches-Whitepaper-fuer-Service-Owner-1.pdf. (Accessed 30 July 2025).

Agence du Numérique en Santé, 2022. Pro santé connect OIDC documentation. https://industriels.esante.gouv.fr/produits-services/pro-sante-connect/documentation-technique. (Accessed 30 July 2025).

Amazon Web Services, 2025. AWS cognito OIDC documentation. https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-userpools-server-contract-reference.html, https://docs.aws.amazon.com/cognito/latest/developerguide/federation-endpoints.html. (Accessed 30 July 2025).

Apple, 2020. Sign in with apple. https://developer.apple.com/sign-in-with-apple/. (Accessed 23 November 2025).

Asghar, M.R., Backes, M., Simeonovski, M., 2018. PRIMA: Privacy-preserving identity and access management at internet-scale. In: 2018 IEEE International Conference on Communications. ICC, IEEE, pp. 1–6.

Auth0, 2022. Auth0 API documentation. https://auth0.com/docs/api/authentication. (Accessed 30 July 2025).

Authlete, 2022. Authlete API documentation. https://docs.authlete.com/en/shared/latest. (Accessed 30 July 2025).

Backes, M., Marnau, N., Druschel, P., 2016. Oblivion: Mitigating privacy leaks by controlling the discoverability of online information. In: Applied Cryptography and Network Security: 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers, vol. 9092, Springer, p. 431.

Boysen, A., 2021. Decentralized, self-sovereign, consortium: The future of digital identity in canada. Front. Blockchain 11.

Bradner, S., 1997. Key words for use in RFCs to Indicate Requirement Levels, BCP 14. RFC Editor, URL http://www.rfc-editor.org/rfc/rfc2119.txt.

Bray, T., 2014. The JavaScript Object Notation (JSON) Data Interchange Format, RFC 7159. RFC Editor, URL http://www.rfc-editor.org/rfc/rfc7159.txt.

Burgin, K., Clancy, T., 2025. International Government Assurance Profile (iGov) for OpenID Connect 1.0. Tech. Rep., The OpenID Foundation, Available at: https://openid.net/specs/openid-igov-oauth2-1_0.html.

Calzavara, S., Focardi, R., Maffei, M., Schneidewind, C., Squarcina, M., Tempesta, M., 2018. WPSE: Fortifying web protocols via browser-side security monitoring. In: 27th USENIX Security Symposium. USENIX Security 18, USENIX Association, Baltimore, MD, pp. 1493–1510, URL https://www.usenix.org/conference/usenixsecurity18/presentation/calzavara.

Campbell, B., Mortimore, C., Jones, M., 2015. Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants, RFC 7522. RFC Editor.

Chari, S., Jutla, C., Roy, A., 2011. Universally composable security analysis of OAuth v2.0. Cryptology ePrint Archive, Paper 2011/526. URL https://eprint.iacr.org/2011/526.

Connect2Id, 2022. Connect2id API documentation. https://connect2id.com/products/server/docs/api. (Accessed 30 July 2025).

Digdir, 2022. ID-porten OIDC documentation. https://docs.digdir.no/docs/idporten/oidc_old/oidc_protocol_authorize.html. (Accessed 30 July 2025).

Digital ID and Authentication Council of Canada (DIACC), 2020. Making sense of identity networks. URL https://diacc.ca/2020/05/13/making-sense-of-identity-networks/. (Accessed 13 February 2024).

Digitaliserings-styrelsen, 2022. MitID documentation. https://digst.dk/it-loesninger/mitid/. (Accessed 30 July 2025).

Digitaliserings-styrelsen, 2023. MitID broker list. https://www.mitid.dk/en-gb/broker/current-brokers/. (Accessed 30 July 2025).

European Parliament and Council of the European Union, 2014. Regulation (EU) No 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eIDAS). Official Journal of the European Union, L 257/73. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014R0910.

European Parliament and Council of the European Union, 2016. Regulation (EU) 2016/679 - general data protection regulation. Official Journal of the European Union, L 257/73. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

European Parliament and Council of the European Union, 2023. EU/EEA trusted list browser. https://eidas.ec.europa.eu/efda/tl-browser/. (Accessed 23 January 2025).

Fett, D., Küsters, R., Schmitz, G., 2015. Spresso: A secure, privacy-respecting single sign-on system for the web. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1358–1369.

Fett, D., Küsters, R., Schmitz, G., 2016. A comprehensive formal security analysis of OAuth 2.0. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16, Association for Computing Machinery, New York, NY, USA, pp. 1204–1215. http://dx.doi.org/10.1145/2976749.2978385.

Fett, D., Küsters, R., Schmitz, G., 2017. The web sso standard openid connect: In-depth formal security analysis and security guidelines. In: 2017 IEEE 30th Computer Security Foundations Symposium. CSF, IEEE, pp. 189–202.

Forgerock, 2021. ForgeRock API documentation. Note: Forgerock has since been acquired by Ping Identity, who now maintains its their old documentation. URL https://backstage.pingidentity.com/docs/am/7.1. (Accessed 30 July 2025).

Google, 2022. Google identity API documentation. https://developers.google.com/identity/openid-connect/openid-connect. (Accessed 30 July 2025).

Hammann, S., Sasse, R., Basin, D., 2020a. Privacy-preserving OpenID connect. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security.

Hammann, S., Sasse, R., Basin, D., 2020b. Privacy-preserving openid connect. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. In: ASIA CCS '20, Association for Computing Machinery, New York, NY, USA, pp. 277–289. http://dx.doi.org/10.1145/3320269.3384724.

Hardt, D., 2012. The OAuth 2.0 Authorization Framework, RFC 6749. RFC Editor, URL http://www.rfc-editor.org/rfc/rfc6749.txt.

IBM, 2019. IBM OIDC documentation. https://www.ibm.com/docs/en/sva/9.0.7?topic=methods-openid-connect-oidc-authentication. (Accessed 30 July 2025).

itsme, 2021. itsme API documentation. https://belgianmobileid.github.io/slate/login.html. (Accessed 30 July 2025).

Jensen, M., Santos, N., Assis, L., Wuyts, M., 2012. LINDDUN: A privacy threat analysis framework for software architectures. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. ASE'12, IEEE, pp. 589–592, URL https://ieeexplore.ieee.org/document/6494345.

Koops, B., Newell, B., Timan, T., Skorvánek, I., Chokrevski, T., Galič, M., 2017. A typology of privacy. University Pa. J. Int. Law 38 (2), 483–575.

Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., Joosen, W., 2019. Tranco: A research-oriented top sites ranking hardened against manipulation. In: Proceedings 2019 Network and Distributed System Security Symposium. In: NDSS 2019, Internet Society, http://dx.doi.org/10.14722/ndss.2019.23386, URL http://dx.doi.org/10.14722/ndss.2019.23386.

Li, W., Mitchell, C.J., 2020. User access privacy in OAuth 2.0 and OpenID connect. In: 2020 IEEE European Symposium on Security and Privacy Workshops. EuroS&PW, IEEE, pp. 664–6732.

Li, W., Mitchell, C.J., Chen, T., 2019. OAuthGuard: Protecting user security and privacy with oauth 2.0 and openid connect. In: Proceedings of the 5th ACM Workshop on Security Standardisation Research Workshop. SSR '19, Association for Computing Machinery, New York, NY, USA, pp. 35–44. http://dx.doi.org/10.1145/3338500.3360331.

Lodderstedt, T., Bradley, J., Labunets, A., Fett, D., 2025. Best Current Practice for OAuth 2.0 Security, RFC 9700. Internet Engineering Task Force, URL https://www.rfc-editor.org/info/rfc9700.

Lodderstedt, T., Fett, D., Haine, M., Pulido, A., Lehman, K., Koiwai, K., 2024. OpenID Connect for Identity Assurance 1.0. Tech. Rep., The OpenID Foundation, Available at: https://openid.net/specs/openid-connect-4-identity-assurance-1_0.html.

Logius, 2023. OpenID NLGov 1.0.1 documentation. https://logius.gitlab.io/oidc/#authorization-endpoint. (Accessed 30 July 2025).

Meta, 2022. Facebook login OIDC documentation. https://developers.facebook.com/docs/facebook-login/guides/advanced/manual-flow/. (Accessed 30 July 2025).

Microsoft, 2022. Microsoft OIDC documentation. https://learn.microsoft.com/it-it/entra/identity-platform/v2-protocols-oidc. (Accessed 30 July 2025).

Ministero dell'Interno, 2015. Carta di identità elettronica - caratteristiche del documento. URL https://www.cartaidentita.interno.gov.it/cose-la-carta/caratteristiche-del-documento/. (Accessed 20 November 2024).

Ministero dell'Interno, AgID, 2019. SPID oidc documentation. https://docs.italia.it/AgID/documenti-in-consultazione/lg-openidconnect-spid-docs/it/bozza/index.html. (Accessed 30 July 2025).

Morkonda, S.G., Chiasson, S., van Oorschot, P.C., 2022. SSOPrivateEye: Timely disclosure of single sign-on privacy design differences. arXiv preprint arXiv:2209.04490.

Navas, J., Beltrán, M., 2019. Understanding and mitigating OpenID connect threats. Comput. Secur. 84, 1–16.

NHS, 2022. NHS login OIDC documentation. https://nhsconnect.github.io/nhslogin/oidc-login-flow/. (Accessed 30 July 2025).

NHS, 2023. NHS care identity service 2 OIDC documentation. https://digital.nhs.uk/services/care-identity-service/applications-and-services/cis2-authentication/guidance-for-developers/detailed-guidance/authorization-code-flow. (Accessed 30 July 2025).

OKTA, 2022. OKTA API documentation. https://developer.okta.com/docs/api/. (Accessed 30 July 2025).

OpenID Foundation, 2023a. Financial-grade api (fapi) 2.0 security profile. Tech. Rep., The OpenID Foundation, Available at: https://openid.net/specs/openid-financial-api-part-2-1_0.html.

OpenID Foundation, 2023b. List of openid specifications. https://openid.net/developers/specs/. (Accessed 06 March 2023).

OpenID Foundation, Elizabeth Garber and Mark Haine (editors), 2023. Human-centric digital identity: for government officials. URL https://openid.net/wp-content/uploads/2023/10/Human-Centric_Digital_Identity_Final-v1.1.pdf. (Accessed 12 September 2024).

Ping Identity, 2025. Ping federation SSO documentation. https://docs.pingidentity.com/bundle/pingone/page/gbj1632772285136.html. (Accessed 30 July 2025).

République française, 2019. Franceconnect identity provider documentation. https://partenaires.franceconnect.gouv.fr/fcp/fournisseur-identite. (Accessed 30 July 2025).

République française, 2021. Franceconnect+ OIDC documentation. https://docs.partenaires.franceconnect.gouv.fr/fi/openid-connect/oidc-presentation/. (Accessed 30 July 2025).

Richer, J., Johansson, L., 2018. Vectors of Trust, RFC 8485. RFC Editor, URL https://www.rfc-editor.org/info/rfc8485.

Riigi Infosüsteemi Ameti, 2025a. GovSSO OIDC documentation. https://e-gov.github.io/GOVSSO/. (Accessed 30 July 2025).

Riigi Infosüsteemi Ameti, 2025b. TARA OIDC documentation. https://e-gov.github.io/TARA-Doku/TechnicalSpecification. (Accessed 30 July 2025).

Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., Mortimore, C., 2023a. OpenID Connect Core 1.0 Incorporating Errata Set 2. Tech. Rep., The OpenID Foundation, Available at: https://openid.net/specs/openid-connect-4-identity-assurance-1_0.html.

Sakimura, N., Bradley, J., Jones, M., Jay, E., 2023b. OpenID Connect Discovery 1.0 Incorporating Errata Set 2. Tech. Rep., The OpenID Foundation, Available at: https://openid.net/specs/openid-connect-4-identity-assurance-1_0.html.

Sassetti, G., 2025. Best current practices for privacy-preserving openid connect: A study of their adoption in the wild. https://st.fbk.eu/complementary/OIDCPrivacy2025.

Sassetti, G., Sharif, A., Sciarretta, G., Carbone, R., Ranise, S., 2023. Assurance, consent and access control for privacy-aware OIDC deployments. In: Atluri, V., Ferrara, A.L. (Eds.), Data and Applications Security and Privacy XXXVII. Springer Nature Switzerland, Cham, pp. 203–222.

SecureAuth, 2022. Cloudentity OAuth API documentation. Note: Cloudentity has since rebranded to SecureAuth. https://docs.secureauth.com/ciam-apis/oauth2.html. (Accessed 30 July 2025).

SignatureGruppen, 2023. MitID documentation by SignatureGruppen. https://signaturgruppen-a-s.github.io/signaturgruppen-broker-documentation/oidc-integration.html. (Accessed 30 July 2025).

SignaturGruppen, 2023. Nemid identity provider documentation. https://www.signaturgruppen.dk/download/broker/docs/Nets%20eID%20Broker%20Technical%20Reference.pdf. (Accessed 30 July 2025).

Signicat, 2025. MitID documentation by signicat. https://developer.signicat.com/identity-methods/mitid/integration-guide/oidc-mitid/. (Accessed 30 July 2025).

Varley, M., Grassi, P., 2024. International Government Assurance Profile (iGov) for OpenID Connect 1.0. Tech. Rep., The OpenID Foundation, Available at: https://openid.net/specs/openid-igov-openid-connect-1_0.html.

Wilson, Y., Hingnikar, A., 2019. Solving Identity Management in Modern Applications: Demystifying Oauth 2.0, Openid Connect, and SAML 2.0. Springer.

World Economic Forum, 2021. Digital identity ecosystems - unlocking new value. https://www3.weforum.org/docs/WEF_Guide_Digital_Identity_Ecosystems_2021.pdf. (Accessed 12 September 2024).

WSO, 2024. WSO2 identity server documentation. https://is.docs.wso2.com/en/latest/. (Accessed 30 July 2025).

Yahoo, 2025. Yahoo OIDC documentation. https://developer.yahoo.com/oauth2/guide/openid_connect/. (Accessed 30 July 2025).

Zhang, Z., Król, M., Sonnino, A., Zhang, L., Rivière, E., 2020. El passo: privacy-preserving, asynchronous single sign-on. arXiv preprint arXiv:2002.10289.

Zhou, Y., Evans, D., 2014. SSOScan: Automated testing of web applications for single sign-on vulnerabilities. In: 23rd USENIX Security Symposium. USENIX Security 14, USENIX Association, San Diego, CA, pp. 495–510, URL https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/zhou.