Protocolo para Negociação de Atributos e Ambiente de Execução na Identidade Fiduciária e Análise Formal de Segurança

Victor do Valle Cunha

[1]Departamento de Informática – Universidade Federal de Santa Catarina (UFSC)

victor.valle@grad.ufsc.br

Abstract. *The Fiduciary Identity model represents an innovative approach to developing identification systems, focusing on addressing two central challenges of conventional models: usability and the protection of user data privacy. The inclusion of a new actor, responsible for meeting these demands, enables the definition of properties that facilitate the achievement of these goals. However, defining these properties alone is not sufficient. To apply them effectively, it is essential to rely on secure communication protocols among the three entities involved in the user authentication process: the Issuer, the Fiduciary, and the Verifier. Additionally, it is crucial that these protocols incorporate specific security properties capable of validating them across different use scenarios. In light of these considerations, this thesis adapts and expands the OpenID Connect for Verifiable Presentation (OIDC4VP) protocol for the context of this new identity model, enabling the negotiation of sensitive data in the authentication process and the handling of such data within a trusted environment for the user. This approach seeks to provide a smoother experience, where the user does not need to constantly interact with authorization forms, while still preserving the essential security properties for the proposed scenario.*

Resumo. *O modelo de Identidade Fiduciária representa uma abordagem inovadora para o desenvolvimento de sistemas de identificação, com foco em solucionar dois desafios centrais dos modelos convencionais: a usabilidade e a proteção da privacidade dos dados do usuário. A inclusão de um novo ator, responsável por atender a essas demandas, permite a definição de propriedades que facilitam o alcance dessas metas. No entanto, a definição dessas propriedades, por si só, é insuficiente. Para que sejam aplicadas de maneira eficaz, é imprescindível contar com protocolos de comunicação seguros entre as três entidades envolvidas no processo de autenticação do usuário: o Emissor, o Fiduciário e o Verificador. Além disso, é essencial que esses protocolos incluam propriedades de segurança específicas, capazes de validá-los em diferentes cenários de uso. Diante dessas considerações, este artigo adapta e expande o protocolo OpendID para Apresentações Verficáveis (OIDC4VP) para o contexto deste novo modelo de identidade, possibilitando a negociação de dados sensíveis no processo de autenticação e a manipulação desses dados em um ambiente de confiança para o usuário. Dessa forma, busca-se uma experiência mais fluida, em que o usuário não precise interagir constantemente com formulários de autorização, mas que ainda preserve as propriedades de segurança essenciais para o cenário proposto.*

1. Introdução

A identificação de pessoas no ambiente digital é um processo essencial para plataformas online, desempenhando um papel crucial tanto na segurança quanto na personalização das interações dos usuários. Conforme o número de serviços digitais e transações online cresce exponencialmente, garantir que os indivíduos sejam corretamente identificados e autenticados tornou-se uma prioridade para empresas e governos. Esse processo não só assegura que o acesso aos sistemas seja restrito a usuários autorizados, protegendo dados sensíveis e prevenindo fraudes, mas também possibilita a entrega de experiências personalizadas, adaptadas às necessidades e preferências de cada usuário, tornando a interação mais fluida e satisfatória.

Ao longo dos anos, diferentes modelos de identidade – representações abstratas que descrevem formas de gerenciar as identidades dos usuários em sistemas computacionais [El Jaouhari et al. 2017] – foram propostos e desenvolvidos para atender às crescentes demandas por segurança, privacidade, escalabilidade e usabilidade no ambiente digital. Entre esses modelos, o Modelo Terceirizado destaca-se como o mais amplamente adotado atualmente. Esse modelo é caracterizado pela intermediação de grandes empresas que funcionam como elos entre o usuário e o Provedor de Serviços, do inglês Service Provider (SP), encarregado de fornecer o serviço desejado. Empresas como Google e Facebook atuam como Provedores de Identidade, do inglês Identity Provider (IdP), simplificando o processo de autenticação e aumentando a usabilidade ao permitir que os usuários utilizem as mesmas credenciais em múltiplos serviços. Esse processo é viabilizado pelo uso de tokens, que são dados digitais gerados para representar a identidade de um usuário. Em vez de transmitir diretamente as credenciais, os IdP emitem tokens que podem ser validados pelos Provedores de Serviços, garantindo que a identidade do usuário seja confirmada de forma segura e eficiente.

No entanto, essa conveniência envolve a transferência do armazenamento e gerenciamento de identidades virtuais para corporações, o que levanta preocupações quanto à privacidade e ao controle de grandes volumes de dados sensíveis por essas entidades. Para superar essas limitações, novos paradigmas estão sendo explorados, como o da Identidade Auto-Soberana, conhecida como Self-Sovereign Identity (SSI), que busca oferecer aos próprios titulares um controle mais direto sobre seus dados [Dock.io 2024]. Em vez de confiar em uma entidade terceira para armazenar suas informações, essa abordagem permite que os dados sejam mantidos de forma segura em dispositivos pessoais, como smartphones, utilizando tecnologias de criptografia e blockchain para assegurar sua integridade e autenticidade. Assim, o SSI inevitavelmente transfere a responsabilidade de gerenciar dados pessoais do IdP para o próprio usuário, o que pode gerar frustração em usuários com pouca familiaridade tecnológica.

Em resposta às limitações do modelo Terceirizado — que apresenta desafios quanto ao controle e à privacidade das informações — e do modelo de SSI, que transfere ao usuário a responsabilidade de gerenciar seus próprios dados, surge o modelo de identidade Fiduciário. Um novo padrão de gestão de identidades que introduz uma entidade de confiança, denominada Fiduciário [Schardong and Custodio 2024], que atua de forma transparente para garantir a autenticação e a autorização nas infraestruturas web desejadas pelo usuário, aliviando-o da tarefa de administrar seus dados pessoais.

Para que o modelo Fiduciário funcione de maneira eficaz e segura, torna-se es-

sencial a criação de um protocolo de comunicação que estabeleça normas claras para a interação entre o Fiduciário, os Provedores de Serviço e os usuários. Esse protocolo precisa definir como as informações de identidade devem ser trocadas de forma segura e garantir que os dados pessoais mantidos pelo Fiduciário estejam acessíveis somente aos serviços autorizados pelo usuário. A ausência de tal protocolo poderia gerar inconsistências e vulnerabilidades, comprometendo a privacidade e a confiabilidade do modelo.

Dentro desse escopo, o objetivo desta monografia é desenvolver um esquema que descreva de forma detalhada a comunicação entre o Fiduciário e o SP, buscando alcançar um equilíbrio eficaz entre usabilidade e segurança. A pesquisa visa aprofundar o nível de detalhamento necessário para uma possível implementação prática, explorando as diferentes possibilidades de comunicação entre essas entidades sem comprometer a privacidade dos usuário.

2. Modelos de Identidades

Modelo de identidade refere-se a uma representação abstrata que descreve como as identidades são gerenciadas, autenticadas e autorizadas em um sistema de computação. A identidade, predominantemente representada por meio de uma **credencial** contendo afirmações acerca do usuário, conhecidas por **claims**, e são de importância crucial para o controle de acesso e a segurança em ambientes computacionais. Essa representação possibilita a determinação de quais entidades possuem permissão para acessar recursos específicos e realizar ações designadas. Portanto, os modelos de identidade são estruturados em entidades que desempenham papéis específicos ao longo do ciclo de vida da credencial [Bertino and Takahashi 2009].

Os modelos são categorizados em três tipos distintos [Schardong 2022]: centralizado, terceirizado e auto-soberano. Cada uma dessas classificações representa uma abordagem única em termos de interação e organização dos elementos mencionados anteriormente.

2.1. Fiduciário

Esse modelo representa um conjunto de esforços destinados a superar as limitações observadas nos modelos anteriores. Conhecido como Fiduciário, esse novo paradigma busca encontrar uma solução para a dicotomia entre a experiência do usuário durante os procedimentos de autenticação e manipulação de dados, e o fortalecimento da preservação da privacidade desses dados. Enquanto sistemas de gerenciamento de identidades baseados em modelos terceirizados oferecem uma experiência de navegação satisfatória, facilitada pelos mecanismos de Single Sign-On (SSO), eles enfrentam dificuldades em evitar a divulgação desnecessária de dados em certos contextos. Em contrapartida, as propostas existentes de SSI conseguem solucionar esse problema, mas apresentam uma experiência de usuário insatisfatória [Schardong and Custodio 2024], com interações complexas e demoradas com os provedores de serviços e as carteiras.

O modelo fiduciário propõe que a gestão das identidades do usuário deve ser realizada por um novo agente de confiança, conhecido como fiduciário. Este, por sua vez, estabelece um relacionamento que dá origem a seu nome, fiduciário, que estabelece uma relação de confiança e responsabilidade em que uma parte (o fiduciário) tem o dever legal e ético de agir no melhor interesse da outra parte (o beneficiário). Esse tipo de relacionamento é comum em várias áreas, como no direito, na administração de empresas,

saúde e na gestão de investimentos. Este modelo também se caracteriza por inúmeras características de segurança e usabilidade, porém elas podem agrupadas nos seguintes princípios fundamentais: *Consent by Default*, *Transparency for Accountability* e *Non-Disclosure as a Goal*. Neste artigo, apenas o último é relevante.

Figura 1. Pilares do Modelo Fiduciário.

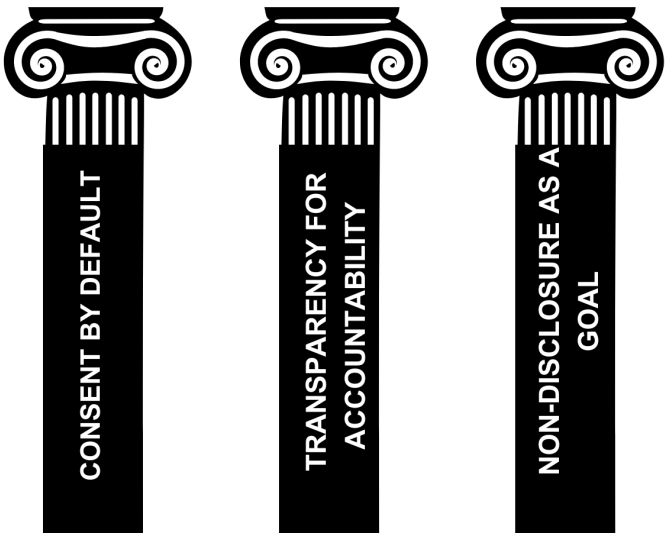

Figura 2. O Autor

Non-Disclosure as a Goal determina que o principal objetivo do modelo é manter a confidencialidade e evitar a revelação de determinadas informações, seja na construção de novos protocolos ou interfaces que utilizam o modelo como estrutura. Para alcançar esse objetivo, três mecanismos podem ser empregados: Divulgação Seletiva, Prova de Conhecimento Zero (ZKP) e Seleção de Ambiente de Execução. Embora cada um desses mecanismos possa ser aplicado individualmente, a combinação deles potencializa significativamente o grau de preservação das informações dos beneficiários.

A primeira estratégia fundamenta-se na capacidade do Fiduciário de revelar apenas partes específicas das informações do usuário, sem expor a totalidade de seus dados. Para alcançar esse objetivo, busca-se a utilização de Credenciais Verificáveis (VC) que disponibilizem mecanismos para a construção de Apresentações Verificáveis (VP) que contenham apenas um subconjunto de atributos, em vez de incluir todos os atributos disponíveis. A segunda estratégia é baseada na utilização de ZKP, pois apesar de também ser uma solução de divulgação seletiva, essa abordagem reduz de forma significativa a exposição de dados em comparação à seleção de atributos para uma VP. Independentemente do mecanismo adotado, o modelo visa proporcionar ao indivíduo maior controle sobre seus dados pessoais, permitindo-lhe fornecer apenas as informações estritamente necessárias e, assim, minimizar a exposição de dados supérfluos ou potencialmente invasivos.

A terceira estratégia explora a possibilidade de processamento de dados fora do ambiente do SP. Tradicionalmente, o processamento de dados relacionados à identidade

ocorre em um ambiente de confiança controlado pelo SP, sem oferecer ao usuário a opção de escolher o ambiente que considera mais seguro, resultando na transferência obrigatória de dados de um ambiente para outro. No entanto, o modelo proposto introduz duas alternativas: a primeira permite que o processamento ocorra em um ambiente de confiança do próprio titular dos dados, em consonância com a proposta de [Hardjono and Pentland 2019]. A segunda possibilita a computação colaborativa entre diferentes entidades por meio de técnicas seguras de Computação Multipartidária, conhecida em inglês como Multiparty Computation (MPC), assegurando que os dados de cada participante permaneçam privados e confidenciais ao longo de todo o processo.

2.2. OAuth 2.0, OpendID Connect e OIDC4VP

OAuth 2.0 é um protocolo de autorização que fornece à aplicações a capacidade de acessar um recurso de usuário por meio de tokens, evitando que o indivíduo precise compartilhar a sua credencial de acesso com aplicação. Dessa forma, não há necessidade de compartilhar credenciais sensíveis [Hardt 2012]. Por exemplo, um leitor pode autorizar um aplicativo de notícias à acessar sua conta ou realizar postagens em seu nome na sua conta de sua rede social sem revelar para o aplicativo qual é a senha. Caso o aplicativo de notícias sofra algum tipo de vazamento de dados, a senha da rede social desse leitor continua resguardada no IdP, a rede social.

De maneira similar, OpenID Connect (OIDC) é um protocolo de autenticação baseado na família de especificações OAuth 2.0, permitindo que as aplicações autentiquem usuários e obtenham informações sobre eles, proporcionando uma experiência de SSO [Sakimura et al. 2014]. Ele é amplamente adotado por grandes provedores de identidade, como Google, Microsoft e Facebook, proporcionando uma interoperabilidade robusta e simplificada entre diversas plataformas e serviços, facilitando a integração e melhorando a segurança na autenticação de usuários em aplicações web e móveis.

A partir desses protocolos, OpendID para Apresentações Verificáveis (OIDC4VP) ampliou as capacidades do OpenID ao permitir a apresentação de Credenciais Verificáveis na forma de Apresentações Verificáveis. Essa extensão oferece uma série de funcionalidades que reforçam a flexibilidade e a interoperabilidade do sistema, como a compatibilidade com todos os fluxos do OpenID Connect, o suporte a diferentes formatos de VCs e VPs, a capacidade de utilizar múltiplos métodos de transporte e o reaproveitamento do parâmetro claims para definir a sintaxe de solicitações.

O protocolo é projetado para ser compatível com todos os fluxos do OpenID Connect. Ademais, a sintaxe das solicitações reaproveita o parâmetro claims do OIDC e utiliza a especificação DIF Presentation Exchange [DIF 2024] para formatar VPs.

3. Um Protocolo Para Identidade Fiduciária

A Identidade Fiduciária é uma proposta recente e promissora, conforme discutido anteriormente. Entretanto, faltava uma solução de protocolo que atendesse aos seus requisitos específicos. Foi nesse contexto que surgiu a necessidade de detalhar uma abordagem viável. O foco recaiu sobre a comunicação entre o Fiduciário e o Verificador, buscando uma solução que não apenas permitisse a apresentação de credenciais no formato adequado, no caso as VPs, mas também oferecesse flexibilidade para expansão e incorporação dos pilares fundamentais do modelo.

A alternativa que melhor cumpria esses requisitos foi o OIDC4VP, que se destacou por ser uma solução simplificada, segura e amigável para desenvolvedores. Além disso, sua capacidade de aproveitar a infraestrutura já existente e o acesso facilitado a uma ampla gama de códigos e bibliotecas baseadas no OpenID Connect tornam sua adoção e integração com sistemas atuais muito mais simples.

Com base nessa escolha, foram desenvolvidas duas extensões alinhadas ao princípio *Non-Disclosure as a Goal*: a **Negociação de Atributos** e a **Negociação do Ambiente de Execução**. A primeira, utiliza a estrutura de Definição de Apresentação em conjunto com a definição de novas mensagens e endpoints, com o objetivo de mitigar o risco de uso indevido de dados, assegurando que apenas as informações estritamente necessárias sejam compartilhadas. A segunda, também fundamenta-se em mensagens e endpoints, permitindo que o processamento de dados sensíveis ocorra preferencialmente em ambientes confiáveis ao usuário, reforçando a segurança e a privacidade na interação.

4. Integração do OIDC4VP com o Modelo Fiduciário

A transição do OIDC4VP, originalmente concebido no modelo SSI, para o Modelo Fiduciário não demandou alterações. Isso ocorre porque a especificação já se concentra na redução da exposição de dados, alinhando-se plenamente ao princípio *Non-Disclosure as Goal*. Dessa forma, no OIDC4VP dentro do Modelo Fiduciário, o SP continua desempenhando seu papel como a entidade responsável por solicitar, receber e validar as VP, sendo assim, visto como Cliente OAuth.

O Fiduciário tem capacidade de desempenhar as mesmas funções que uma Carteira Digital, incluindo receber, armazenar, apresentar e gerenciar as VCs, além de administrar as chaves criptográficas associadas. Nesse sentido, como os pilares *Consent by Default* e *Transparency for Accountability* não foram abordados, as mensagens trocadas entre o Fiduciário e o SP, anteriormente direcionadas à Carteira Digital, permanecem equivalentes e o Fiduciário assume o papel de Servidor de Autorização, que no modelo original era desempenhado pela Carteira. Dessa forma, o modelo incorpora um mecanismo para a entrega de VPs na forma de tokens e também viabiliza a integração com as extensões de Negociação de Atributos e Negociação do Ambiente de Execução.

5. Uso de Provas de Conhecimento Zero

Uma das principais razões para a utilização do OIDC4VP, no Modelo Fiduciário é a implementação da segunda estratégia proposta, que identifica as ZKP como uma alternativa eficaz para reduzir a exposição de dados. Isso porque a especificação distingue-se por sua flexibilidade e abordagem agnóstica em relação ao formato das VCs utilizadas. Em outras palavras, sua adaptação ao modelo não exige, necessariamente, o uso de ZKP; no entanto, oferece um ambiente compatível com a integração de VCs que possuam essa funcionalidade. Como o OIDC4VP não impõe requisitos específicos para a implementação de ZKP, cabe aos implementadores optarem por VCs que oferecem suporte a essa tecnologia. O emprego de ZKP é especialmente recomendado em cenários que demandam altos níveis de privacidade e segurança na verificação de credenciais, garantindo que apenas as informações estritamente necessárias sejam divulgadas.

6. Negociação de Atributos

Quando um Fiduciário recebe uma Requisição de Autorização é possível que a Definição de Apresentação solicite informações do usuário as quais não estejam de acordo com as Políticas de Consentimento definidas previamente. Por exemplo, é comum que Provedores de Serviço realizem a solicitação da data de nascimento dos usuários apenas para verificar sua maioridade. Este tipo de requisição é invasiva e pode estar contra os desejos do indivíduo. Dessa forma, a extensão **Negociação de Atributos** permite definir quais informações sobre os usuários são relevantes para a plataforma que está oferecendo serviços sem comprometer a privacidade e o sigilo dos dados do dono desses recursos.

Esse mecanismo está centrado na estrutura de Definição de Apresentações (DA), que no modelo Fiduciário são chamados de **Declarações de Requisitos** (Requirements Statements). Neste texto, os termos DAs e Declarações de Requisitos serão tratados como sinônimos. Com essa estrutura, os Fiduciários conseguem definir quais informações requisitadas não estão em conformidade com as Políticas de Consentimento e propor novos que satisfaçam as intenções. Para que isso seja possível, são definidos o endpoint Negociação (Negotiation Endpoint), a Solicitação de Negociação (Negotiation Request) e sua respectiva resposta, Resposta de Negociação (Negotiation Response). Essa nova requisição, iniciada pelo Fiduciário e respondida pelo SP, ocorre entre a Requisição de Autorização e sua respectiva resposta, conforme destacado na 4.

Figura 3. Fluxo para Negociação de Atributos.

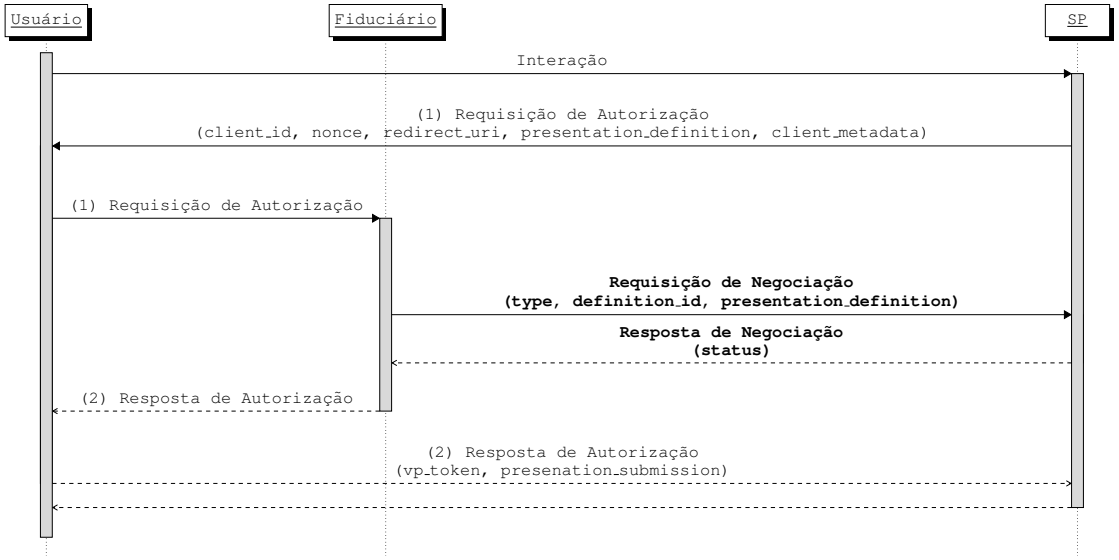

Figura 4. O Autor

6.1. Endpoint de Negociação

Este endpoint é utilizado para negociar Declaração de Requisitos previamente encaminhada pelo SP em sua Requisição de Autorização e que o Fiduciário identifica alguma divergência do que foi requisitado e o que é autorizado dentro das Políticas de Consentimento. A comunicação com o Endpoint de Neogicação deve utilizar TLS.

6.2. Solicitação de Negociação

A Solicitação de Negociação constitui uma requisição HTTP POST enviada pelo Fiduciário ao SP, com o tipo de mídia `application/json`.

- **type**: OBRIGATÓRIO. Este parâmetro deve ser utilizado na Solicitação de Negociação para especificar o tipo de negociação que está sendo realizada. No contexto da *Negociação de Atributos*, deve-se utilizar o valor `attribute`.
- **definition_id**: OPCIONAL. Trata-se de uma string que identifica uma Declaração de Requisitos previamente encaminhada ao Fiduciário. Após o acordo ser firmado entre o Fiduciário e o Provedor, o SP deve invalidar o `definition_id`. Este parâmetro se torna OBRIGATÓRIO quando o valor de `type` é `attribute`.
- **presentation_definition**: OPCIONAL. Este parâmetro contém um objeto JSON de Declaração de Requisitos, conforme a sintaxe estabelecida na [DIF 2024]. Este parâmetro se torna OBRIGATÓRIO quando o valor de `type` é `attribute`.

6.3. Resposta de Negociação

O Provedor de Serviços possui a prerrogativa de aceitar ou recusar a proposta de alteração da Declaração de Requisitos. Em determinados casos, o SP aceita a proposta que contém a nova Declaração de Requisitos e responde com uma mensagem com tipo de mídia `application/json` contendo em seu corpo o JSON com o parâmetro `status` indicando a aceitação da sugestão de declaração e o código de status HTTP 202 sinalizando está pronto para receber vp_token referente a nova declaração acordada, que é diferente daquele que o SP enviou em sua Requsição de Autorização.

- **status**: OBRIGATÓRIO. A semântica desse campo irá depender da negociação definida em type. Nesse exemplo, indica se o SP aceitou ou recusou a nova proposta para a Declaração de Requisitos. O código de status em formato ASCII selecionado entre as duas opções abaixo:
  - **accepted**: Indica que a proposta para a nova Declaração de Requisitos foi aceita pelo SP e está pronto para receber VP em um `vp_token` com os parâmetros relativos a nova declaração acordada.
  - **refused**: Indica que a proposta para a nova Declaração de Requisitos foi recusada pelo SP. Nesse caso, nenhuma alteração será realizada, e a Declaração de Requisitos permanecerá como está. O motivo da recusa pode ser detalhado em um campo adicional de erro chamado `error_description`, descrito a seguir.

Em outros casos, o SP pode rejeitar a proposta por não concordar com os novos parâmetros sugeridos. Nessas situações, a resposta HTTP deve utilizar o parâmetro status rejeitando a declaração e o código de status HTTP 400 (Bad Request) incluindo os seguintes parâmetros no corpo da resposta codificada em JSON.

Resposta de Recusa de Negociação

Se a Solicitação de Negociação não for aceita por não atender aos requisitos da aplicação, ou for considerada inválida devido à sintaxe incorreta da requisição, o SP deverá definir o campo de status como `refused` e incluir os campos `error` e `retry_after`.

- **Error**: OBRIGATÓRIO. O parâmetro de erro deve conter um único código de erro em formato ASCII selecionado a partir da lista a seguir:
  - **negotiation_request_denied**: A semântica desse campo irá depender da negociação definida em `type`. Nesse contexto, indica que a Solicitação de Negociação com a nova Declaração de Requisitos não foi aceita pelo provedor de serviços.
  - **invalid_negotiation_request**: A Solicitação de Negociação está incompleta, como a falta de um parâmetro obrigatório, inclusão de um parâmetro ou valor de parâmetro não suportado, repetição do mesmo parâmetro ou está malformada.
  - **unsupported_definition**: A Declaração de Requisitos contida na Solicitação de Negociação contém um formato que não é reconhecido ou suportado pelo SP. Esse erro acontece quando é utilizada uma versão desatualizada ou por não seguir a sintaxe e os padrões estabelecidos.
  - **expired_definition_id**: Esse erro indica que o `definition_id` fornecido na Solicitação de Negociação já expirou. Isso ocorre quando a negociação associada a esse `definition_id` já foi concluída e o identificador pode ser invalidado para impedir que novas solicitações sejam feitas com base em um estado anterior.
- **error_description**: OPCIONAL. O parâmetro `error_description` deve ser um texto ASCII, fornecendo informações adicionais para ajudar os implementadores do Fiduciário a entender o erro ocorrido. Ele pode incluir espaços, caracteres de pontuação e símbolos comuns, mas não pode incluir caracteres como o caractere aspas duplas (”), barra invertida (\), ou qualquer caractere de controle esteja fora dos intervalos `%x20-21 / %x23-5B / %x5D-7E`, onde o prefixo `%` indica valores hexadecimais da tabela ASCII.
- **retry_after**: OBRIGATÓRIO. A resposta de erro deve também conter o parâmetro `retry_after`, que determina o tempo mínimo em segundos que o Fiduciário deve aguardar antes de enviar uma nova solicitação ao Endpoint de Negociação.

Se o Fiduciário e o Provedor de Serviços não chegarem a um acordo sobre a Declaração de Requisitos, ambos continuarão a trocar mensagens até atingirem um limite estipulado por uma das partes. Assim, recomenda-se fortemente que o Provedor de Serviços adote práticas flexíveis, solicitando apenas as informações estritamente necessárias. Caso contrário, a ausência de um acordo pode resultar na perda de acesso ao serviço pelo usuário, o que pode afetar negativamente a percepção pública da marca, gerando comentários desfavoráveis e reclamações.

7. Negociação do Ambiente de Execução

As arquiteturas tecnológicas vigentes, de modo geral, adotam a lógica de que a manipulação dos dados dos usuários deve ocorrer exclusivamente no lado do SP, centralizando o processamento e o armazenamento das informações. No entanto, novas alternativas estão surgindo e possibilitam uma mudança nesse paradigma, como é o caso da proposta de execução dentro do Fiduciário e o uso de MPC. Essas soluções permitem que o processamento de dados sensíveis seja realizado de forma descentralizada e segura, protegendo a privacidade do usuário ao restringir o acesso direto aos dados pelo provedor de serviços e permitindo que o processamento ocorra de maneira distribuída e controlada.

Nesse sentido, a especificação de **Negociação do Ambiente de Execução** viabiliza esses dois paradigmas dentro do Modelo Fiduciário, possibilitando que o beneficiário decida o local (Fiduciário ou SP) e forma (tradicional ou MPC) apropriada para o processamento de suas informações privadas. O objetivo dessa mudança é manter o paradigma atual em operação, enquanto abre espaço para a integração dessas novas abordagens em serviços compatíveis. A especificação está centrada nas solicitações e endpoints mencionados na 6.2 e possibilita o usuário decidir em qual local é o melhor para que suas informações sejam processadas, conforme pode ser visto na 6.

Figura 5. Fluxo para Negociação de Ambiente de Execução.

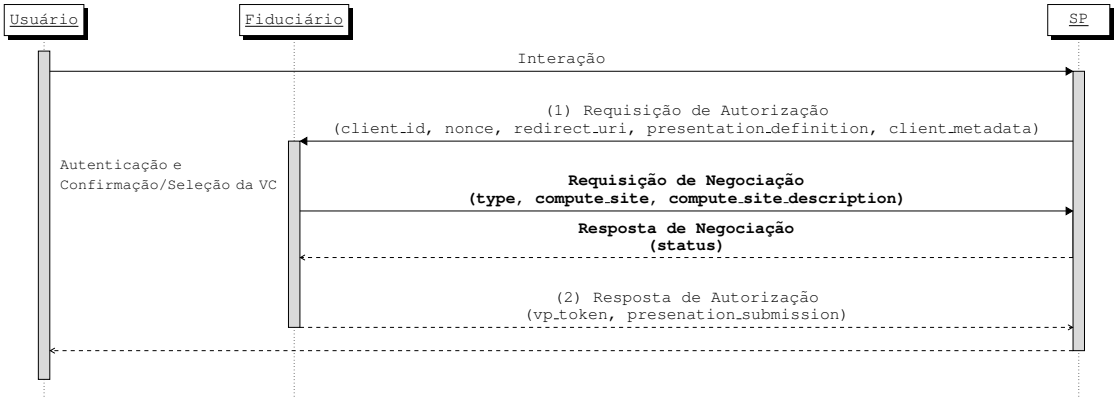

Figura 6. O Autor

7.1. Solicitação de Negociação

Essa extensão fará uso da Solicitação de Negociação descrita em 6.2. Nesse contexto, o valor atribuído a `type` será **env**. Consulte o exemplo apresentado abaixo.

- **compute_site**: OBRIGATÓRIO. Indica o local em que as manipulações de dados serão realizadas. O valor de `compute_site` deve estar no formato ASCII, escolhido entre as três opções a seguir:
  - **sp**: Indica que a manipulação de dados do usuário será realizada em um ambiente de confiança gerido pelo Provedor de Serviços. Caso o Fiduciário não faça uma proposta, este é o mecanismo padrão utilizado pelo provedor.
  - **fiduciary**: Indica que a manipulação de dados do usuário será realizada em um ambiente de confiança gerido pelo Fiduciário.
  - **both**: Indica que a manipulação de dados do usuário será realizada de forma colaborativa em um ambiente de confiança gerido tanto pelo Fiduciário quanto pelo Provedor de Serviços. Quando essa opção for selecionada, é necessário especificar o parâmetro `compute_site_description`.

7.2. Resposta de Negociação

Essa extensão utilizará a Resposta de Negociação, conforme descrito em 6.2. Nesse contexto, o campo `status` indica se o SP aceitou ou rejeitou a nova proposta sobre o local de execução. O valor do `status` é representado em formato ASCII e pode assumir uma das duas opções: `accepted` ou `refused`.

7.2.1. Resposta de Sucesso para Computação Multipartidária

Caso a Solicitação de Negociação inclua o parâmetro `compute_site` definido como `both` e seja validada pelo SP, este deverá adicionar o campo **compute_site_description** em sua Resposta de Negociação.

- **compute_site_description**: Esse parâmetro é um JSON que deve ser especificado de acordo com os requisitos estabelecidos pelo SP para disponibilizar a computação Multipartidária. A responsabilidade pela definição do formato deste JSON recai sobre os implementadores, que deverão garantir que ele atenda aos critérios e necessidades específicas do SP e do Fiduciário.

7.2.2. Resposta de Recusa para Execução em outros Ambientes

Se a Solicitação de Negociação não for aceita por não atender aos requisitos da aplicação, ou for considerada inválida devido à sintaxe incorreta da requisição, o SP pode reutilizar alguns dos campos estabelecidos 6.3.

- **negotiation_request_denied**: A interpretação deste campo dependerá do tipo de negociação especificado em `type`. Neste contexto, ele indica que a Solicitação de Negociação referente ao novo local de execução não foi aceita pelo Provedor de Serviços.
- **invalid_negotiation_request**: A Solicitação de Negociação está incompleta, como a falta de um parâmetro obrigatório, inclusão de um parâmetro ou valor de parâmetro não suportado, repetição do mesmo parâmetro ou está malformada.
- **not_supported**: Este parâmetro indica que o Provedor de Serviços carece de mecanismos necessários para realizar uma computação do tipo MPC.

8. Análise Formal de Segurança

Esta seção apresenta a prova formal de segurança para apenas para o módulo de Negociação de Atributos. Ela utiliza o modelo formal do protocolo OIDC4VP proposto em [Hauck 2023], para demonstrar que as propriedades de segurança definidas a seguir na 8.1 não podem ser comprometidas por um atacante. Essa prova oferece garantias de segurança para as novas extensões dentro protocolo OIDC4VP. Neste capítulo, as ideias principais das provas formais são explicadas de maneira resumida.

8.1. Propriedade de Segurança: Autenticação da Negociação

Em alto nível, a propriedade de Autenticação da Negociação garante que um atacante não seja capaz de realizar negociações com um Provedor de Serviços se passando por um Fiduciário. Essa propriedade é violada caso o atacante consiga utilizar o `definition_id` vinculado a um ID de Sessão.

8.2. Prova da Autenticação da Negociação

A prova de segurança *Autenticação da Neogciação* avalia que a segurança de uma definição de apresentação (pd) em um sistema de autenticação, mostrando que informações sensíveis são protegidas contra vazamentos para terceiros não autorizados,

inclusive potenciais atacantes. A análise foca em como as comunicações entre o navegador, o verificador e o fiduciário são protegidas. Utilizando conexões HTTPS, esses dados são criptografados e acessíveis apenas para as partes legítimas, conforme demonstrado pelo Lema 1 de [Fett et al. 2014].

O sistema garante que somente scripts confiáveis das origens autorizadas podem acessar pd, e esses scripts não manipulam ou divulgam a informação. Além disso, o fiduciário valida pd de acordo com suas restrições de segurança antes de criar qualquer nova definição de apresentação, protegendo a integridade dos dados ao longo do processo de autenticação. Assim, a prova conclui que pd permanece seguro e inacessível para qualquer agente malicioso, assegurando que apenas o Fiduciário honesto pode negociar Definições de Apresentações.

9. Conclusões

O presente trabalho iniciou abordando a complexa tarefa de garantir a identificação digital no cenário atual, ressaltando como a crescente dependência de sistemas digitais nas atividades cotidianas exige um aprofundamento em estudos na área de IAM. Nesse contexto, foram levantados e analisados diferentes modelos de gestão de identidades — centralizado, terceirizado e auto-soberano (SSI) —, culminando na recente proposição do Modelo Fiduciário. Além disso, discutiram-se os principais protocolos que sustentam esses modelos, como OAuth, OpenID Connect (OIDC) e OIDC4VC, destacando suas características e arquitetura. Por fim, foram apresentados os processos e a aplicação de uma análise formal de segurança, ilustrados por meio de um exemplo prático, reforçando a importância de metodologias rigorosas para a validação desses sistemas.

Nesse sentido, o texto apresentou uma adaptação do OpenID Connect com Apresentações Verificáveis para o Modelo Fiduciário, visando fornecer não apenas um mecanismo flexível para a transmissão de VPs, com suporte à divulgação seletiva de informações e a Zero-Knowledge Proofs (ZKP), mas também atender a um dos principais princípios do modelo, o *Non-Disclosure as a Goal*. Para atender a essa demanda, duas extensões foram propostas: *Negociação de Atributos* e *Negociação do Ambiente de Execução*.

A *Negociação de Atributos* é uma solução projetada para reduzir a transmissão de informações desnecessárias do usuário durante o processo de autenticação que utiliza os `vp_token`. Para alcançar esse objetivo, a adaptação do OIDC4VP no Modelo Fiduciário foi ajustada, permitindo que o Fiduciário realize requisições ao Provedor de Serviços para modificar o artefato que define as entradas do `vp_token` antes de encaminhá-lo para a autenticação do beneficiário. Nesse contexto, foram definidos os formatos das requisições e respostas, bem como os endpoints necessários para viabilizar a implementação dessa funcionalidade de forma eficiente e segura.

A *Negociação do Ambiente de Execução*, por sua vez, é uma solução semelhante à Negociação de Atributos no que diz respeito aos formatos de requisições e respostas, mas com um propósito distinto. Seu objetivo é modificar a lógica referente ao local onde ocorre a manipulação dos dados do usuário, tradicionalmente realizada no serviço web. Essa abordagem permite que o processamento seja executado diretamente no Fiduciário ou de forma colaborativa, utilizando técnicas de Computação Multipartidária para garantir maior controle, privacidade e segurança no tratamento dos dados sensíveis.

Por fim, foi realizada uma análise formal de segurança na extensão proposta para o protocolo OpenID para Apresentações Verificáveis. Para essa análise, definiu-se a propriedade de Autenticação da Negociação e provou-se que ela é mantida dentro dos limites do modelo formal. Essa análise demonstrou que a extensão proposta oferece uma segurança no contexto do modelo definido. Considerando que esses protocolos contribuem significativamente para o avanço do Modelo Fiduciário, esta pesquisa também se torna um incentivo relevante para o desenvolvimento de novas soluções no campo da IAM.

Trabalhos Futuros

Uma das direções promissoras é investigar o suporte dos protocolos para incorporar regras de consentimento específicas utilizadas pelos Fiduciários. Esse aprimoramento permitiria que os protocolos reconhecessem e aplicassem automaticamente políticas de consentimento conforme definidas por diferentes Fiduciários, promovendo um maior alinhamento com o Modelo Fiduciário.

Outra possibilidade é expandir a análise formal para cobrir propriedades relacionadas à negociação do ambiente de execução. Isso incluiria a definição e validação de propriedades que assegurem a integridade e a conformidade do ambiente em que a negociação ocorre, garantindo que todos os elementos envolvidos na transação sigam as especificações de segurança e privacidade necessárias. Esse tipo de análise pode fortalecer a proteção contra ambientes inseguros ou comprometidos, proporcionando garantias adicionais para as partes envolvidas e consolidando a segurança dos protocolos frente a cenários de execução complexos e distribuídos.

Referências

Bertino, E. and Takahashi, K. (2009). *Identity Management: Concepts, Technologies, and Systems*. Artech House, Norwood, MA, USA.

DIF (2024). Fpresentation exchange 2.1.0. `https://github.com/decentralized-identity/presentation-exchange`.

Dock.io (2024). Self-sovereign identity: The ultimate guide 2024. Acessado: 18 de julho de 2024.

El Jaouhari, S., Bouabdallah, A., and Bonnin, J.-M. (2017). Chapter 14 - security issues of the web of things. In *Managing the Web of Things*. Morgan Kaufmann.

Fett, D., Kusters, R., and Schmitz, G. (2014). The web infrastructure model (wim). In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Hardjono, T. and Pentland, A. (2019). MIT Open Algorithms. In *Trusted Data: A New Framework for Identity and Data Sharing*. The MIT Press.

Hardt, D. (2012). The oauth 2.0 authorization framework. RFC 6749.

Hauck, F. (2023). Openid for verifiable credentials: Formal security analysis using the web infrastructure model. Master's thesis, Informatik.

Sakimura, N., Bradley, J., Jones, M., and de Medeiros, B. (2014). Openid connect core 1.0 incorporating errata set 2.

Schardong, F. (2022). Self-sovereign identity: A systematic review, mapping and taxonomy. *Sensors 22(15)*.

Schardong, F. and Custodio, R. (2024). From self-sovereign identity to fiduciary identity: A journey towards greater user privacy and usability. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, SAC '24, page 687–694, New York, NY, USA. Association for Computing Machinery.