UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

Eduardo Dani Perottoni

Desenvolvimento de um Barramento de Interoperabilidade Acadêmico Utilizando o Protocolo *OpenID Federation 1.0*

Florianópolis
2026

Eduardo Dani Perottoni

Desenvolvimento de um Barramento de Interoperabilidade Acadêmico Utilizando o Protocolo *OpenID Federation 1.0*

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Ciência da Computação.
Orientador: Prof. Frederico Schardong, Dr.
Coorientador: Prof. Ricardo Felipe Custódio, Dr.

Florianópolis
2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Perottoni, Eduardo Dani
Desenvolvimento de um Barramento de Interoperabilidade Acadêmico Utilizando o Protocolo OpenID Federation 1.0 / Eduardo Dani Perottoni ; orientador, Frederico Schardong, coorientador, Ricardo Felipe Custódio, 2026.
170 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2026.

Inclui referências.

1. Ciências da Computação. 2. Identidade Digital. 3. Identidade Federada. 4. CAFe. 5. OpenID Federation. I. Schardong, Frederico. II. Custódio, Ricardo Felipe. III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Eduardo Dani Perottoni

Desenvolvimento de um Barramento de Interoperabilidade Acadêmico Utilizando o Protocolo *OpenID Federation 1.0*

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 12 de julho de 2026.

Profª. Lúcia Helena Martins Pacheco, Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Frederico Schardong, Dr.
Orientador
Instituto Federal do Rio Grande do Sul

Shirlei Aparecida de Chaves, Ma.
Avaliadora
Instituto Federal de Santa Catarina

Brendon Vicente Rocha Silva
Avaliador
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais e inspirações, Airton Perottoni e Luciane Dani Perottoni, que me educaram e nunca mediram esforços para me proporcionar o melhor ensino possível. Deixo registrada aqui a imensa gratidão e admiração aos dois. Pelo nome deles, estendo o agradecimento a toda minha família, que me apoiou e incentivou durante essa jornada acadêmica. Especialmente, devo agradecer pelos quatro anciãos das famílias Dani e Perottoni, os meus avós, os quais tenho o privilégio de ter em vida e poder prestar essa pequena homenagem.

Agradeço à RNP, que, por meio do Programa de Serviços Avançados, proporcionou a pesquisa e o desenvolvimento desse trabalho. Deixo um agradecimento especial à toda a equipe do GT-BAITA, conduzida pelos professores Frederico Schardong e Ricardo Felipe Custódio, respectivamente, orientador e coorientador do presente trabalho. Aproveito para agradecer por suas brilhantes orientações e ensinamentos. Aos demais membros do GT, Brendon, Giulia, Alison, Charlie, Raissa, Rayane e Leonardo, agradeço por toda a colaboração e companheirismo durante o desenvolvimento do projeto. Aprendi muito com todos.

Preciso destacar quem acompanhou a jornada de perto. Às minhas irmãs, Ana Carolina e Maria Eduarda, agradeço pela ajuda durante toda a graduação e por virem comigo para essa aventura longe de casa.

À minha namorada Joice, agradeço pela cumplicidade, companhia em todos os momentos e suporte, principalmente emocional. Essa jornada foi mais leve com ela ao meu lado e com certeza não teria conseguido chegar até aqui sem ela.

Agradeço também aos meus amigos, que me apoiaram e incentivaram durante toda a graduação. Não vou citar nomes, com medo de esquecer alguém, mas eles sabem quem são. Agradeço também aos meus colegas de curso, que compartilharam comigo momentos de aprendizado e crescimento.

Se eu vi mais longe, foi por estar sobre ombros de gigantes.
- Isaac Newton

RESUMO

As federações de identidade digital internacionais estão em um período de atualização tecnológica, migrando do protocolo *Security Assertion Markup Language* (SAML) 2.0 para o OpenID Federation 1.0. Nesse cenário, a Comunidade Acadêmica Federada (CAFe), federação brasileira de identidade para educação e pesquisa, enfrenta desafios significativos por sua dependência do SAML. Embora ainda amplamente adotado pelas federações acadêmicas, esse protocolo é desatualizado para demandas atuais de interoperabilidade, impõe uma integração complexa para novos provedores e opera sobre uma estrutura centralizada e rígida. Tais limitações resultam em práticas de gestão de identidade subótimas e desestimulam a adesão de novos serviços à federação. Para responder a esses desafios, este trabalho propõe uma arquitetura de barramento de interoperabilidade acadêmico que não apenas moderniza a infraestrutura da CAFe, alinhando-a a tendências globais, mas também soluciona problemas específicos do ecossistema de instituições brasileiras. A arquitetura proposta é composta por: um *proxy* para facilitar a adoção por parte das instituições com esforço mínimo, permitindo a integração de Provedores de Identidade (PI) que adotam protocolos modernos sem a necessidade de manter uma instância separada baseada no *software Shibboleth*; um *Where Are You From* (WAYF) aprimorado, concebido para, além de descobrir o PI do usuário, atuar como normalizador de atributos e garantidor de múltiplos fatores de autenticação; um *proxy* de tradução de protocolos para o suporte gradual ao SAML; um sistema de gestão de entidades do barramento, com a possibilidade de gestão de subfederações para diferenciar provedores públicos e privados dentro das instituições; e um portal de benefícios para divulgação dos Provedores de Serviço (PS) integrados ao barramento, voltado aos usuários finais. Para comprovar a viabilidade da arquitetura proposta, uma prova de conceito (PoC) funcional do barramento foi implementada e implantada no ambiente de *testbeds* da Rede Nacional de Ensino e Pesquisa (RNP). Essa PoC implementa os componentes supracitados. Para avaliação da PoC, exercitou-se, de ponta a ponta, o fluxo de autenticação federada iniciando em PSs integrados ao barramento até PIs que implementam tanto OpenID Connect (OIDC) quanto SAML, comprovando a interoperabilidade entre os protocolos. Além disso, contatou-se a facilidade proposta de adesão de provedores que não implementam o protocolo de federação ao barramento. Parte das funcionalidades projetadas, como a gestão de subfederações e a normalização de atributos, permanece em estágio de projeto e é registrada entre os trabalhos futuros. Os resultados indicam a viabilidade de um ecossistema de identidade mais flexível, escalável e seguro, que simplifica a gestão para operadores e instituições, incentiva a adesão e prepara a CAFe para futuras inovações, como a Identidade Digital Descentralizada.

Palavras-chave: Identidade Digital. Identidade Federada. CAFe. OpenID Federation.

ABSTRACT

International digital identity federations are undergoing a technological update, migrating from the *Security Assertion Markup Language* (SAML) 2.0 protocol to OpenID Federation 1.0. In this scenario, the Brazilian Federated Academic Community (CAFe), the national identity federation for education and research, faces significant challenges due to its dependence on SAML. Although still widely adopted by academic federations, this protocol is outdated for current interoperability demands, imposes a complex integration process on new providers, and operates over a centralized and rigid structure. Such limitations result in suboptimal identity management practices and discourage the adoption of new services within the federation. To address these challenges, this work proposes an academic interoperability bus architecture that not only modernizes the CAFe infrastructure, aligning it with global trends, but also solves specific problems of the Brazilian institutional ecosystem. The proposed architecture is composed of: a *proxy* to facilitate adoption by institutions with minimal effort, enabling the integration of Identity Providers (IdP) that adopt modern protocols without the need to maintain a separate instance based on the *Shibboleth software*; an enhanced *Where Are You From* (WAYF), conceived to, beyond discovering the user's IdP, act as an attribute normalizer and an enforcer of multi-factor authentication; a protocol translation *proxy* for gradual support of SAML; an entity management system with the possibility of subfederation management to distinguish public and private providers within institutions; and a benefits portal, aimed at end users, for the dissemination of the Service Providers (SP) integrated into the bus. To demonstrate the feasibility of the proposed architecture, a functional proof of concept (PoC) of the bus was implemented and deployed on the *testbeds* environment of the Brazilian National Education and Research Network (RNP). This PoC implements the aforementioned components. To evaluate the PoC, the federated authentication flow was exercised end to end, starting from SPs integrated into the bus up to IdPs that implement both OpenID Connect (OIDC) and SAML, demonstrating interoperability between the protocols. In addition, the proposed ease of adoption of providers that do not implement the federation protocol into the bus was verified. Part of the designed functionalities, such as subfederation management and attribute normalization, remains at the design stage and is recorded as future work. The results indicate the feasibility of a more flexible, scalable, and secure identity ecosystem that simplifies management for operators and institutions, encourages adoption, and prepares CAFe for future innovations, such as Decentralized Digital Identity.

Keywords: Digital Identity. Federated Identity. CAFe. OpenID Federation.

LISTA DE FIGURAS

Figura 1 – Representação da identidade digital do indivíduo como identidades parciais. 24
Figura 2 – Formulário de autorização do protocolo OAuth 2.0. 26
Figura 3 – Legenda para os diagramas que serão apresentados. 27
Figura 4 – Modelo de Identidade Isolada. 28
Figura 5 – Modelo de Identidade Centralizada. 29
Figura 6 – Modelo de Identidade Federada. 30
Figura 7 – Modelo de Identidade Centrada no Usuário. 31
Figura 8 – Modelo de Identidade Autossoberana. 32
Figura 9 – Componentes da especificação SAML2. 33
Figura 10 – Fluxo de autenticação no *Shibboleth*. 35
Figura 11 – Interface do serviço WAYF da CAFe. 37
Figura 12 – Fluxo de autenticação do OIDC. 40
Figura 13 – Exemplo de federações ancoradas em A e B. 42
Figura 14 – Cadeia de Confiança do OID Fed. 43
Figura 15 – Mesclagem da política de metadados entre dois SSs. 48
Figura 16 – Obtenção dos metadados resolvidos. 49
Figura 17 – Fluxo de registro automático do cliente no OP. 54
Figura 18 – Fluxo de registro explícito do cliente no OP. 55
Figura 19 – Arquitetura simplificada da eduGAIN proposta para o piloto. 61
Figura 20 – Arquitetura proposta no projeto SPID/CIE. 62
Figura 21 – BMC resultante do processo de prospecção do GT-BAITA. 66
Figura 22 – Arquitetura geral do BAITA. 77
Figura 23 – Arquitetura básica baseada na compatibilidade com a eduGAIN. 80
Figura 24 – Emissão de TMs para a federação. 81
Figura 25 – Hierarquia considerando subfederação privada por instituição 84
Figura 26 – Hierarquia considerando subfederação por grupo específico. 85
Figura 27 – Casos de uso do SGB. 86
Figura 28 – Arquitetura detalhada do SGB. 88
Figura 29 – Arquitetura interna do *Proxy-OID-Fed*. 90
Figura 30 – *Endpoints* expostos pelo Proxy-OID-Fed abstraindo federação para uma RP. 92
Figura 31 – *Endpoints* expostos pelo Proxy-OID-Fed abstraindo a federação para um OP. 92
Figura 32 – Sequência de registro e configuração de um SP e sua subsequente representação na federação pelo Proxy-OID-Fed. 94
Figura 33 – Fluxo de registro e posterior comunicação entre OPs e RPs OIDC por meio do Proxy-OID-Fed. 95
Figura 34 – Posicionamento do WAYF na topologia da federação. 96
Figura 35 – Arquitetura detalhada do WAYF. 97
Figura 36 – Resultado da busca em largura na federação executada pelo WAYF. 98

Figura 37 – Problema de interoperabilidade sintática e semântica de atributos. 100
Figura 38 – Fluxo de autenticação passando pelo WAYF. 102
Figura 39 – Arquitetura do Proxy-SAML. 103
Figura 40 – Representação da topologia com suporte ao legado no nível 1. 104
Figura 41 – Representação da topologia com suporte ao legado nos níveis 2 e 3. 105
Figura 42 – Fluxo de autenticação com PI legado no nível 2 de compatibilidade. 106
Figura 43 – *Endpoints* expostos pelo Proxy-SAML no nível 2 de compatibilidade. 107
Figura 44 – Fluxo de autenticação com PS legado no nível 3 de compatibilidade. 108
Figura 45 – *Endpoints* expostos pelo Proxy-SAML no nível 3 de compatibilidade. 109
Figura 46 – Casos de uso do PoB. 110
Figura 47 – Posicionamento do PoB na topologia da federação. 111
Figura 48 – Arquitetura interna do *Portal de Benefícios*. 112
Figura 49 – Página inicial do SGB. 114
Figura 50 – Tela de Lista de Entidades do SGB. 115
Figura 51 – Página de detalhes de uma organização (UFSC) no SGB. 116
Figura 52 – Aba de *Histórico* de uma entidade no SGB. 118
Figura 53 – Tela de configurações de perfil e segurança do usuário no SGB. 119
Figura 54 – Página inicial do PoB. 120
Figura 55 – Página de Benefícios do PoB, com busca e filtros sobre o catálogo. 121
Figura 56 – Página de detalhes de um serviço no PoB. 123
Figura 57 – Página inicial da RPflask. 124
Figura 58 – Descoberta no WAYF: seleção de um provedor SAML2. 125
Figura 59 – Descoberta no WAYF: seleção do OP OIDC registrado KPR. 126
Figura 60 – Página de *login* do provedor SAML2 da UFSC. 127
Figura 61 – Exibição dos atributos recebidos do IdP SAML2 pelo RPflask. 127
Figura 62 – Página de *login* do OP OIDC de teste KPR. 128
Figura 63 – Exibição dos atributos recebidos do OP OIDC pelo RPflask. 128
Figura 64 – Criação e delegação da organização SBC ao gestor da instituição. 142
Figura 65 – Cadastro do gestor no SGB. 143
Figura 66 – Configuração do segundo fator de autenticação (*passkey* ou *OTP*). 143
Figura 67 – Configuração do aplicativo autenticador. 144
Figura 68 – Configuração do aplicativo autenticador. 144
Figura 69 – Primeiro acesso do gestor à sua página inicial. 145
Figura 70 – Preenchimento das informações da organização SBC. 146
Figura 71 – Confirmação após o preenchimento do formulário da organização. 147
Figura 72 – Início da criação de um OP subordinado à SBC. 148
Figura 73 – Formulário de criação do PI da SBC. 148
Figura 74 – Página inicial indicando o novo PI com status *Pendente*. 149
Figura 75 – Preenchimento da configuração técnica do PI. 150
Figura 76 – Página de detalhes do PI Ativo. 151

Figura 77 – Página inicial do gestor após a ativação do PI. 152
Figura 78 – O PI listado como subordinado à SBC na Lista de Entidades. 152
Figura 79 – Início da criação de um PS subordinado à UFSC. 153
Figura 80 – Formulário de cadastro do serviço More UFSC. 155
Figura 81 – Definição do tipo de acesso do serviço: público ou restrito. 156
Figura 82 – Preenchimento da redirect URI do serviço. 156
Figura 83 – Credenciais OIDC fornecidas ao PS. 157
Figura 84 – Confirmação do cadastro do serviço e o seu endereço no PoB. 157
Figura 85 – O serviço More UFSC já catalogado no PoB. 158

LISTA DE TABELAS

Tabela 2 – Comparação entre federação SAML e OpenID Federation. 57
Tabela 3 – Principais tecnologias utilizadas no BAITA e suas finalidades. 70

LISTA DE ALGORITMOS

Algoritmo 1 – Pseudoalgoritmo de avaliação da TC. 43
Algoritmo 2 – Metadados autodeclarados publicados pela $\texttt{RP}_1$ em sua EC. 44
Algoritmo 3 – Metadados e políticas de metadados publicados pelo $\texttt{IM}_1$ no SS sobre o subordinado $\texttt{RP}_1$. 45
Algoritmo 4 – Política de metadados publicados pela `TA` no SS sobre o subordinado $\texttt{IM}_1$. 46
Algoritmo 5 – Representação de uma EC com o *claim* `trust_marks`. 50
Algoritmo 6 – *Payload* do TM. 50
Algoritmo 7 – *Claims* referentes aos emissores e proprietários de TMs na EC da TA da federação. 51

LISTA DE ACRÔNIMOS

2FA	*Two-Factor Authentication* (Autenticação de Dois Fatores)
AAI	*Authentication and Authorisation Infrastructure* (Infraestrutura de Autenticação e Autorização)
AARC	*Authentication and Authorisation for Research and Collaboration*
API	*Application Programming Interface* (Interface de Programação de Aplicações)
AS	*Authorization Server* (Servidor de Autorização)
BAITA	Barramento de Interoperabilidade Acadêmico
BFF	*Backend For Frontend*
BFS	*Breadth-First Search* (Busca em Largura)
BMC	*Business Model Canvas* (Quadro de Modelo de Negócios)
CAFe	Comunidade Acadêmica Federada
CICD	*Continuous Integration and Continuous Delivery* (Integração e Entrega Contínuas)
CIE	*Carta di Identità Elettronica*
CPF	Cadastro de Pessoa Física
CRUD	*Create, Read, Update and Delete* (Criar, Ler, Atualizar e Excluir)
DID	*Decentralized Identifier* (Identificador Descentralizado)
DLT	*Distributed Ledger Technology* (Tecnologia de Registro Distribuído)
DS	*Discovery Service* (Serviço de Descoberta)
EC	*Entity Configuration* (Configuração da Entidade)
eduGAIN	*EDUcation Global Authentication INfrastructure*
eIDAS2	*electronic IDentification, Authentication, and trust Services*
ES	*Entity Statement* (Declaração de Entidade)
FID	Federação de Identidade Digital
GDPR	*General Data Protection Regulation*
GId	Gestão de Identidade
GT	Grupo de Trabalho
GT-BAITA	Grupo de Trabalho Barramento de Interoperabilidade Acadêmico
HPC	*High Performance Computing* (Computação de Alto Desempenho)
HTTP	*Hypertext Transfer Protocol*
ICP	Infraestrutura de Chaves Públicas
ID	Identidade Digital
IDD	Identidade Digital Descentralizada
IDF	Identidade Digital Federada
IdP	*Identity Provider* (Provedor de Identidade)
IM	*Intermediate Entity* (Entidade Intermediária)
JSON	*JavaScript Object Notation*

JWE	*JSON Web Encryption*
JWS	*JSON Web Signature*
JWT	*JSON Web Token*
LDAP	*Lightweight Directory Access Protocol*
LE	*Leaf Entity* (Entidade Folha)
LGPD	Lei Geral de Proteção de Dados
MFA	*Multi-Factor Authentication* (Múltiplos Fatores de Autenticação)
MVP	*Minimum Viable Product* (Produto Mínimo Viável)
NREN	*National Research and Education Network* (Rede Nacional de Educação e Pesquisa)
OAuth 2.0	*Open Authorization*
OID Fed	*OpenID Federation 1.0*
OIDC	OpenID Connect
OP	*OpenID Provider* (Provedor OpenID)
OTP	*One-Time Password*
PAD	*Personal Authentication Device* (Dispositivo de Autenticação Pessoal)
PI	Provedor de Identidade
PIN	*Personal Identification Number*
PKCE	*Proof Key for Code Exchange*
PoB	Portal de Benefícios do Barramento de Interoperabilidade Acadêmico
PS	Provedor de Serviço
R&E	*Research and Education* (Pesquisa e Ensino)
REFEDS	*Research and Education FEDerations group*
REST	*Representational State Transfer*
RM	*Resolved Metadata* (Metadados Resolvidos)
RNP	Rede Nacional de Ensino e Pesquisa
RP	*Relying Party* (Parte Confiante)
RS	*Resource Server* (Servidor de Recursos)
SAML2	*Security Assertion Markup Language 2.0*
SBC	Sociedade Brasileira de Computação
SCHAC	*SCHema for ACademia*
SFA	*Single-Factor Authentication* (Autenticação de Fator Único)
SGB	Sistema de Gestão do Barramento de Interoperabilidade Acadêmico
SIRTFI	*Security Incident Response Trust Framework for Federated Identity*
SOAP	*Simple Object Access Protocol*
SP	*Service Provider* (Provedor de Serviço)
SPA	*Single Page Application* (Aplicação de Página Única)
SPI	*Service Provider Interface*
SPID	*Sistema Pubblico di Identità Digitale*

SS	*Subordinate Statement* (Declaração de Subordinado)
SSO	*Single Sign-On* (Autenticação Única)
TA	*Trust Anchor* (Âncora de Confiança)
TC	*Trust Chain* (Cadeia de Confiança)
TI	Tecnologia da Informação
TM	*Trust Mark* (Marca de Confiança)
USB	*Universal Serial Bus*
VC	*Verifiable Credential* (Credencial Verificável)
VP	*Verifiable Presentation* (Apresentação Verificável)
WAYF	*Where Are You From*
WS-Federation	*Web Services Federation*
WSGI	*Web Server Gateway Interface* (Interface de Gateway de Servidor Web)
XML	*Extensible Markup Language*
ZKP	*Zero Knowledge Proof* (Prova de Conhecimento Zero)

SUMÁRIO

	Sumário	**16**
1	**INTRODUÇÃO**	**18**
1.1	CONTEXTO DE DESENVOLVIMENTO	21
2	**OBJETIVOS**	**22**
2.1	OBJETIVO GERAL	22
2.2	OBJETIVOS ESPECÍFICOS	22
3	**FUNDAMENTAÇÃO TEÓRICA**	**23**
3.1	IDENTIDADE DIGITAL E GESTÃO DE IDENTIDADE	23
3.2	MODELOS DE GESTÃO DE IDENTIDADE	27
3.3	ECOSSISTEMA SAML E CAFe	32
3.4	ECOSSISTEMA OPENID CONNECT E OPENID FEDERATION	38
3.5	PROXY DE IDENTIDADE	57
4	**TRABALHOS CORRELATOS**	**60**
4.1	PILOTO OPENID FEDERATION DA EDUGAIN	60
4.2	PROJETO SPID/CIE	61
4.3	DISSERTAÇÃO DO INSTITUTO POLITÉCNICO DE TURIM	63
4.4	ANÁLISE COMPARATIVA	63
5	**METODOLOGIA**	**65**
5.1	FASE DE PROSPECÇÃO DO GT-BAITA	65
5.2	DESIGN DA ARQUITETURA	68
5.3	DESENVOLVIMENTO DO CÓDIGO-FONTE	69
6	**PROPOSTA**	**71**
6.1	PREMISSAS PARA O DESENVOLVIMENTO	71
6.2	*PROXY*-OID-FED	72
6.3	WAYF	72
6.4	*PROXY*-SAML	73
6.5	GESTÃO DE SUBFEDERAÇÕES	73
6.6	SISTEMA DE GESTÃO DO BAITA	74
6.7	PORTAL DE BENEFÍCIOS	75
7	**ARQUITETURA**	**77**
7.1	VISÃO GERAL DA ARQUITETURA	77
7.2	MAPEAMENTO DO OPENID FEDERATION E SEUS ARTEFATOS	78

7.3 SISTEMA DE GESTÃO DO BAITA 85
7.4 *PROXY*-OID-FED 90
7.5 WAYF 95
7.6 *PROXY*-SAML 101
7.7 PORTAL DE BENEFÍCIOS 107

8 RESULTADOS E DISCUSSÕES 113
8.1 SISTEMA DE GESTÃO DO BARRAMENTO 113
8.2 PORTAL DE BENEFÍCIOS 119
8.3 FLUXO DE AUTENTICAÇÃO COM WAYF E *PROXIES* 122

9 CONCLUSÃO 129
9.1 TRABALHOS FUTUROS 129
9.2 CONTRIBUIÇÕES EM ESPECIFICAÇÕES TÉCNICAS INTERNACIONAIS 130

REFERÊNCIAS 132

APÊNDICES 140

APÊNDICE A – FLUXO DE INTEGRAÇÃO DE UM PROVEDOR DE IDENTIDADE OIDC 141
A.1 CRIAÇÃO DA ORGANIZAÇÃO E DELEGAÇÃO 141
A.2 REGISTRO DO PROVEDOR DE IDENTIDADE 141

APÊNDICE B – FLUXO DE INTEGRAÇÃO DE UM PROVEDOR DE SERVIÇO OIDC 153

APÊNDICE C – ARTIGO SOBRE O TCC NO FORMATO SBC . . . 159

1 INTRODUÇÃO

Identidade Digital (ID) pode ser definida como um conjunto de atributos que descrevem unicamente uma entidade e sua relação com as demais (Windley, 2005, p. 8). A ID não é apenas uma representação digital de uma identidade física. Ela é criada, usada e destruída segundo o desejo do usuário e permite o compartilhamento dos atributos essenciais para uma transação específica, preservando o anonimato de dados irrelevantes (Schardong et al., 2024). Nos sistemas de gerenciamento de ID atuais, o Provedor de Identidade (PI) é utilizado pelo Provedor de Serviço (PS)[1] para autenticar e autorizar usuários (Bhargav-spantzel; Squicciarini; Bertino, 2005).

Autenticação é o ato do sujeito provar que é quem ele diz ser, enquanto a autorização indica a concessão ou revogação de permissão para compartilhamento de atributos acerca da identidade com uma terceira parte. (Lopez; Oppliger; Pernul, 2004). Dessa forma, o PS, mediante devida autorização, pode ter acesso aos dados do usuário, fornecidos pelo PI. Nesse contexto, vários protocolos de autenticação e autorização foram desenvolvidos. Pöhn e Hommel (2023) apresentam um histórico destes protocolos, como o *Lightweight Directory Access Protocol* (LDAP), o *Security Assertion Markup Language 2.0* (SAML2), o *Open Authorization* (OAuth 2.0) e o OIDC.

Geralmente a comunicação entre PS e PI ocorre dentro do escopo de um domínio, espaço delimitado por regras específicas e acordos administrativos compartilhados por entidades deste espaço. Quando múltiplas organizações, ou domínios, criam uma relação de confiança e permitem que indivíduos usem as mesmas informações de identificação e credenciais para obter acesso a serviços oferecidos por elas, tem-se uma Federação de Identidade Digital (FID) (Aldosary; Alqahtani, 2021). O conceito chave em uma federação é o estabelecimento de confiança entre suas partes (Kylau et al., 2009), o que possibilita a troca de informações sobre uma identidade digital de forma segura e confiável. Nesse cenário, várias especificações de gerenciamento de FID foram desenvolvidas, como o SAML2 (OASIS, 2005) e o *Web Services Federation* (WS-Federation) (OASIS, 2009).

FIDs são empregadas especialmente no ambiente educacional, onde a troca de dados entre instituições é frequente. No contexto brasileiro, a Comunidade Acadêmica Federada (CAFe), serviço oferecido pela Rede Nacional de Ensino e Pesquisa (RNP), conta com mais de 320 instituições conectadas e figura entre as cinco maiores federações do mundo (RNP, 2024a). A CAFe, assim como outras grandes FIDs mundiais, utiliza o protocolo SAML2 para comunicação e estabelecimento de confiança entre as entidades da federação.

Discussões sobre FID no grupo *Research and Education FEDerations group* (REFEDS), fórum que reúne as federações acadêmicas e de pesquisa de todo o mundo para ar-

[1] Ao longo deste trabalho, os termos PI, Provedor de Identidade (IdP, *Identity Provider*) e Provedor OpenID (OP, *OpenID Provider*) são empregados de forma intercambiável para designar o provedor de identidade. Da mesma forma, PS, Provedor de Serviço (SP, *Service Provider*) e Parte Confiante (RP, *Relying Party*) designam indistintamente o provedor de serviço. Embora OP e RP sejam as denominações técnicas exatas no contexto do OpenID Connect (OIDC) essa distinção não é preservada ao longo do texto.

ticular políticas e boas práticas comuns de identidade federada, apontam o SAML2 como um protocolo desatualizado e que não atende às novas necessidades de gestão de identidade (Flanagan, 2024). As discussões também citam que, devido ao seu modelo de estabelecimento de confiança, o SAML2 não é capaz de suportar os novos requisitos de Identidade Digital Descentralizada (IDD). Esses requisitos estão surgindo principalmente no contexto do *electronic IDentification, Authentication, and trust Services* (eIDAS2) (European Union, 2024), regulamentação da União Europeia que estabelece um marco legal para a identificação eletrônica e os serviços de confiança em transações eletrônicas no mercado interno europeu. Em sua segunda versão, o eIDAS2, introduz a Carteira Europeia de Identidade Digital, apoiada em modelos de identidade descentralizada.

Além dessas restrições do SAML2, a CAFe possui políticas rígidas de operação. Uma delas é a exigência de que os PIs sejam aplicações *Shibboleth* (Consortium, 2025), um conjunto de *software* de código aberto que implementa o SAML2 e é amplamente adotado por federações acadêmicas para prover autenticação federada e login único (*Single Sign-On*). Outra é a exigência de que, para integrar um PS à federação, seja necessário um processo de aprovação longo e manual por parte do operador da federação. Essas políticas, embora visem garantir a segurança e a confiabilidade da federação, acabam por criar barreiras à integração de novos serviços e dificultar a adoção de práticas modernas de gestão de identidade. Todo esse contexto resulta em (i) complexidade na integração de novos PIs e PSs; (ii) hierarquia rígida devido à centralização; e (iii) dificuldade de alinhamento com padrões internacionais atuais.

Para acatar aos padrões impostos, muitas instituições federadas não adotam boas práticas de Gestão de Identidade (GId), mantendo um PI para serviços oferecidos a usuários internos e outro PI habilitado para SAML2, especificamente para a CAFe, usando o *software Shibboleth*. O primeiro, em muitos dos casos, é construído com base em protocolos da família OIDC. Ademais, há, na CAFe, uma baixa adoção de PSs, o que limita a oferta de serviços federados aos usuários finais. No momento da escrita deste texto, a CAFe conta com 330 PIs, mas apenas 48 PSs registrados na CAFe[2]. Destes, apenas 9 são PSs registrados por instituições-membro fora do domínio da RNP. Um dos fatores que contribuem para essa baixa adoção é a dificuldade de integração, apontada como uma das dores mais recorrentes na fase de prospecção deste trabalho, detalhada na Seção 5.1. Caso a PS utilize protocolos modernos como o OIDC, por exemplo, seu gestor precisa configurar e manter um *proxy* que traduza as mensagens para o SAML2, o que demanda recursos e conhecimento técnico.

Considerando os problemas do SAML2 e as necessidades atuais de FIDs, a *OpenID Foundation*, entidade que desenvolve padronizações acerca de ID, está desenvolvendo um novo protocolo de FID, o *OpenID Federation 1.0* (OID Fed). Atualmente no seu *draft* 43 (Hedberg et al., 2026), esse novo protocolo proporciona uma série de benefícios e mitiga problemas citados acima. Além disso, permite à federação o emprego de protocolos atuais e amplamente utilizados em processos de autenticação e autorização, como o OAuth 2.0 e o OIDC (Sakimura

[2] https://ds.cafe.rnp.br/metadata/cafe-metadata.xml. Acesso em: 20 jun. 2026.

et al., 2023a).

O presente trabalho propõe a implementação do Barramento de Interoperabilidade Acadêmico (BAITA), um barramento que permite integração de PIs e PSs baseado no protocolo OID Fed e considerando o contexto da CAFe e suas problemáticas, que resultam numa baixa adoção à federação, principalmente por PSs. O BAITA é composto por um conjunto de componentes, dentre eles um sistema de descoberta de PIs, um sistema de descoberta de PSs, um sistema de gestão da federação e um proxy de tradução de mensagens entre protocolos.

O BAITA permite que as instituições que já possuem um PI empregando o protocolo OIDC integrem-se ao barramento sem a necessidade de um segundo PI *Shibboleth*, economizando recursos na gestão e manutenção e facilitando a integração à federação. Outro benefício notável desse protocolo é o registro automático de PSs nos PIs da federação, o que reduz a latência de disponibilidade do serviço federado. Além disso, a proposta prevê a utilização de *proxies* mantidos pelo operador da federação, focando na facilidade de adoção de provedores[3] à federação com esforço mínimo por parte dos seus mantenedores.

Esse trabalho busca atualizar tecnologicamente a CAFe, seguindo o rumo de federações internacionais, como a *EDUcation Global Authentication INfrastructure* (eduGAIN), serviço de interfederação que interconecta as federações acadêmicas nacionais de diversos países, entre elas a CAFe, permitindo a troca confiável de informações de identidade entre suas instituições. A eduGAIN já possui projetos direcionados à migração da tecnologia atual para o OID Fed (Vaghetti, 2024a) e vem trabalhando na definição das regras e topologias que regerão essa adoção (eduGAIN, 2025). Outro exemplo é a Itália, cuja infraestrutura nacional de identidade digital, composta pelo *Sistema Pubblico di Identità Digitale* (SPID), que dá aos cidadãos uma identidade digital única para acessar serviços públicos e privados, e pela *Carta di Identità Elettronica* (CIE), a carteira de identidade eletrônica nacional, adotou o OID Fed em produção e em larga escala (Agenzia per l'Italia Digitale (AgID), 2024).

Adicionalmente, o OID Fed é compatível com modelos de IDD, como as carteiras digitais europeias estabelecidas no eIDAS2 2.0. Especificações sobre IDD já estão sendo desenvolvidas, como o *OpenID Federation for Wallet Architectures 1.0* (Marco et al., 2024a), o *OpenID for Verifiable Credential Issuance* (Lodderstedt; Yasuda; Looker, 2024) e o *OpenID for Verifiable Presentations* (Terbu et al., 2024), todas compatíveis com o OID Fed. Isso possibilita futuro suporte a IDDs dentro da federação.

Os resultados esperados são abrangentes e envolvem o atendimento à necessidade latente de modernização tecnológica da CAFe. Espera-se que as funcionalidades do protocolo *OID Fed* tragam benefícios para operadores da federação, instituições membro e para todos os usuários finais, que utilizarão o BAITA para acessarem serviços providos pela federação.

No restante do texto, o Capítulo 2 apresenta os objetivos do trabalho, enquanto o Capítulo 3 traz os fundamentos teóricos que embasam o desenvolvimento. O Capítulo 4 mostra esforços relacionados com a presente iniciativa e o Capítulo 5 descreve a metodologia adotada.

[3] Neste texto, o termo "provedores" é usado para designar tanto PSs quanto PIs.

O Capítulo 6 expõe a proposta e as premissas do projeto considerando a fase de prospecção, e o Capítulo 7 detalha a arquitetura e os componentes técnicos desenvolvidos. Em seguida, o Capítulo 8 apresenta e discute os resultados obtidos com o protótipo implementado. Por fim, o Capítulo 9 conclui o trabalho e delineia perspectivas de trabalhos futuros.

1.1 CONTEXTO DE DESENVOLVIMENTO

Este trabalho foi desenvolvido no âmbito da Fase 1 do Programa de Serviços Avançados da RNP, que fomenta Grupos de Trabalho (GTs) com foco na inovação de seus serviços. O autor deste trabalho atuou como assistente de inovação do Grupo de Trabalho Barramento de Interoperabilidade Acadêmico (GT-BAITA), participando das atividades de *design* da arquitetura, coordenação da equipe de desenvolvimento e implementação do código-fonte do BAITA.

O GT-BAITA foi estabelecido com a missão de projetar e validar uma evolução substancial da CAFe, orientada a sanar os gargalos operacionais e tecnológicos acumulados ao longo dos anos de operação sobre o SAML2. Para isso, o grupo conduziu uma fase de prospecção, descrita com mais detalhes na Seção 5.1, que incluiu entrevistas com *stakeholders*, mapeamento de dores e a definição de um Produto Mínimo Viável (MVP, *Minimum Viable Product*) que delimita o escopo deste trabalho.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Propor um barramento de interoperabilidade entre PSs e PIs no contexto da CAFe utilizando o protocolo *OpenID Federation 1.0*.

2.2 OBJETIVOS ESPECÍFICOS

- Revisar as principais especificações acerca de Identidade Digital Federada (IDF), para entender os requisitos para implantação do barramento de interoperabilidade;
- Estudar com profundidade o protocolo *OpenID Federation 1.0*, com o intuito de elencar quais das suas funcionalidades devem ser utilizadas no contexto da CAFe;
- Estudar as implementações do protocolo *OpenID Federation* a fim de reutilizá-las na implementação da prova de conceito;
- Desenvolver uma arquitetura para a federação CAFe utilizando o protocolo *OpenID Federation 1.0*, contendo o WAYF, o sistema de gestão do barramento e o sistema de descoberta de serviços propostos;
- Instanciar a arquitetura em um protótipo, que comprove a possibilidade de utilização do *OpenID Federation 1.0* e dos sistemas propostos no contexto da CAFe.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 IDENTIDADE DIGITAL E GESTÃO DE IDENTIDADE

A identidade, em seu sentido mais amplo não é inata e imutável, mas sim uma construção social dinâmica. Segundo Hall (2006), a identidade é formada e transformada continuamente a partir das interações do indivíduo e da forma como ele é representado nos sistemas sociais que o rodeiam. Na prática, a identidade de um sujeito preenche o espaço entre o seu mundo pessoal e o mundo público e é composta por uma multiplicidade de papéis, características e relações.

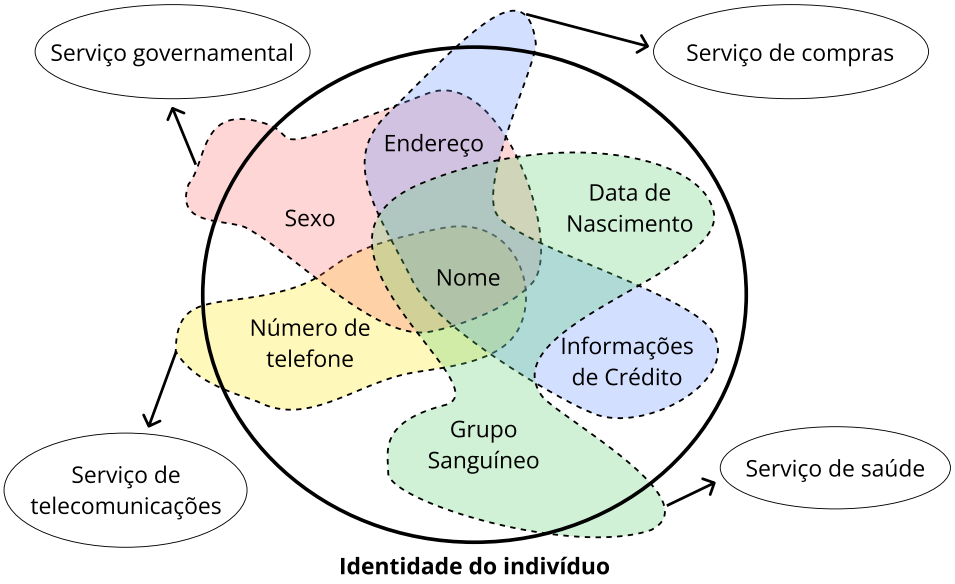
No mundo físico, a necessidade de organizar interações sociais e legais deu origem a mecanismos práticos de gestão de identidades (Schardong; Custódio, 2022). Cotidianamente, as pessoas utilizam credenciais emitidas por entidades de confiança para comprovar atributos, conforme a situação exige (Jøsang; Pope, 2005; Schardong; Custódio, 2022). Por exemplo, apresenta-se um passaporte para cruzar fronteiras internacionais, uma Carteira Nacional de Habilitação para comprovar a aptidão legal de conduzir um veículo e uma Carteira de Nacional de Identidade para comprovação de atributos frente a um serviço específico, como abrir uma conta em um banco. Em cada um desses casos, o documento revela atributos específicos para a situação em que é usado e também pode conter atributos gerais inúteis para a ocasião (Pfitzmann; Hansen, 2010). Em todos os documentos citados, por exemplo, o nome do indivíduo é um atributo presente.

Com a expansão das atividades humanas para o mundo digital, surgiu o desafio de transpor e padronizar esses mecanismos físicos de identificação e autorização para o ambiente computacional. É nesse cenário que a identidade digital pode ser definida como um conjunto de atributos permanentes ou temporários associados com uma entidade (Camp, 2004). Segundo Windley (2005, p. 8), "uma identidade digital contém dados que descrevem unicamente uma pessoa ou coisa (chamada de sujeito ou entidade) mas também contém informação sobre as relações do sujeito com outras entidades". Jøsang e Pope (2005) mencionam a dependência da identidade digital com o domínio de aplicação específico. Afinal, uma entidade pode ter diferentes identidades para diferentes contextos de utilização, assim como ocorre com as credenciais físicas.

Clauß e Köhntopp (2001) introduzem o conceito de "identidades parciais", que são subconjuntos dos atributos de uma identidade. Em diferentes contextos, um indivíduo pode ser representado por diferentes identidades parciais. A Figura 1 exemplifica esse conceito. Nela, vê-se que cada PS pode requerer determinada identidade parcial.

A gestão dessas identidades no mundo digital é realizada por um ecossistema, isto é, um ambiente formado por múltiplos sistemas que cooperam para administrar a coleta, a autenticação e o uso das identidades e das informações a elas associadas ao longo de seu ciclo de vida (Clauß; Köhntopp, 2001; Hansen; Schwartz; Cooper, 2008; Bhargav-spantzel et al., 2006). Esse ecossistema é composto por programas, *frameworks* e protocolos que, em conjunto,

Figura 1 – Representação da identidade digital do indivíduo como identidades parciais.



Fonte: Adaptado de (Clauß; Köhntopp, 2001).

lidam com a identidade digital. Protocolos, como o OAuth 2.0, e *softwares* que implementam PIs, como o *Shibboleth*, são exemplos de componentes que participam desse ecossistema.

Um ecossistema de GId, assim como tratam Bhargav-Spantzel et al. (2006), envolve três elementos principais: o usuário, o PI e o PS. O usuário é o indivíduo representado pelos atributos da identidade digital. O PI é o *software* que (i) gerencia os dados acerca da identidade do usuário; (ii) executa os procedimentos de autenticação e autorização; e (iii) emite credenciais, que atestam que o usuário possui uma identidade digital no PI. Por fim, o PS é o serviço que oferece as mais diversas funcionalidades para o usuário final e utiliza informações providas pelo PI acerca do usuário. Na Figura 1 são ilustrados exemplos de serviços que podem ser oferecidos por PSs.

O acesso a serviços com controle de acesso consiste em (i) identificar a entidade a partir de um conjunto único de atributos; (ii) autenticar a identidade e (iii) autorizar o compartilhamento de atributos com o PS (Schardong et al., 2024). Os itens (i) e (ii) são fundamentais, enquanto (iii) é opcional.

A identificação consiste na associação entre identificadores únicos e um indivíduo que está apresentando seus atributos (Camp, 2004). Esse processo requer que, dentre toda a população de atributos gerenciada pelo PI, exista um atributo ou um conjunto deles que mapeie unicamente uma identidade digital para um único indivíduo. No mundo digital, esse identificador único pode ser solicitado ao usuário por meio de um campo `Nome de usuário` em um formulário de autenticação, por exemplo. O identificador pode ser criado em um formulário de registro ou utilizar um identificador já conhecido no mundo físico, como o Cadastro de Pessoa Física (CPF), por exemplo. No mundo físico, para criar uma conta bancária, por exemplo, exige-se que o titular registre seu CPF.

A autenticação se refere à prova da associação da entidade com seu identificador único (Camp, 2004). No mundo físico, esse processo geralmente consiste na comprovação pela posse do documento físico. Para comprar uma bebida alcoólica, por exemplo, permitido apenas para maiores de 18 anos, o comprador se autentica no estabelecimento comercial mostrando o documento emitido pelo órgão de confiança com a data de nascimento de 18 anos atrás. O vendedor verifica se o documento realmente pertence ao comprador, normalmente pela semelhança entre a foto que está no documento e a aparência do comprador. No mundo digital, esse processo tem sido executado por meio da verificação de igualdade entre o segredo, geralmente uma senha, informado pelo usuário no formulário de autenticação com o segredo previamente relacionado com a identidade digital no momento do registro no PI. Atualmente, como mostram Ometov et al. (2018), uma senha apenas não garante segurança em muitos casos e, por isso, múltiplos fatores de autenticação estão sendo utilizados.

A autorização é o processo de garantia de que políticas de acesso a recursos são seguidas (Bhargav-spantzel; Squicciarini; Bertino, 2005). Após a autenticação, o PI pode conceder ou revogar o compartilhamento de atributos com o PS. Protocolos atuais, como o OAuth 2.0 (Hardt, 2012), viabilizam que essa decisão seja delegada ao próprio usuário, que consente quais atributos requisitados pelo PS podem ser compartilhados. O protocolo, contudo, apenas faculta a coleta desse consentimento, sem prescrever como ela deve ser apresentada ao usuário. A materialização dessa interação fica a cargo de cada implementação. A Figura 2 ilustra um exemplo dessa coleta de consentimento em uma aplicação que utiliza o OAuth 2.0.

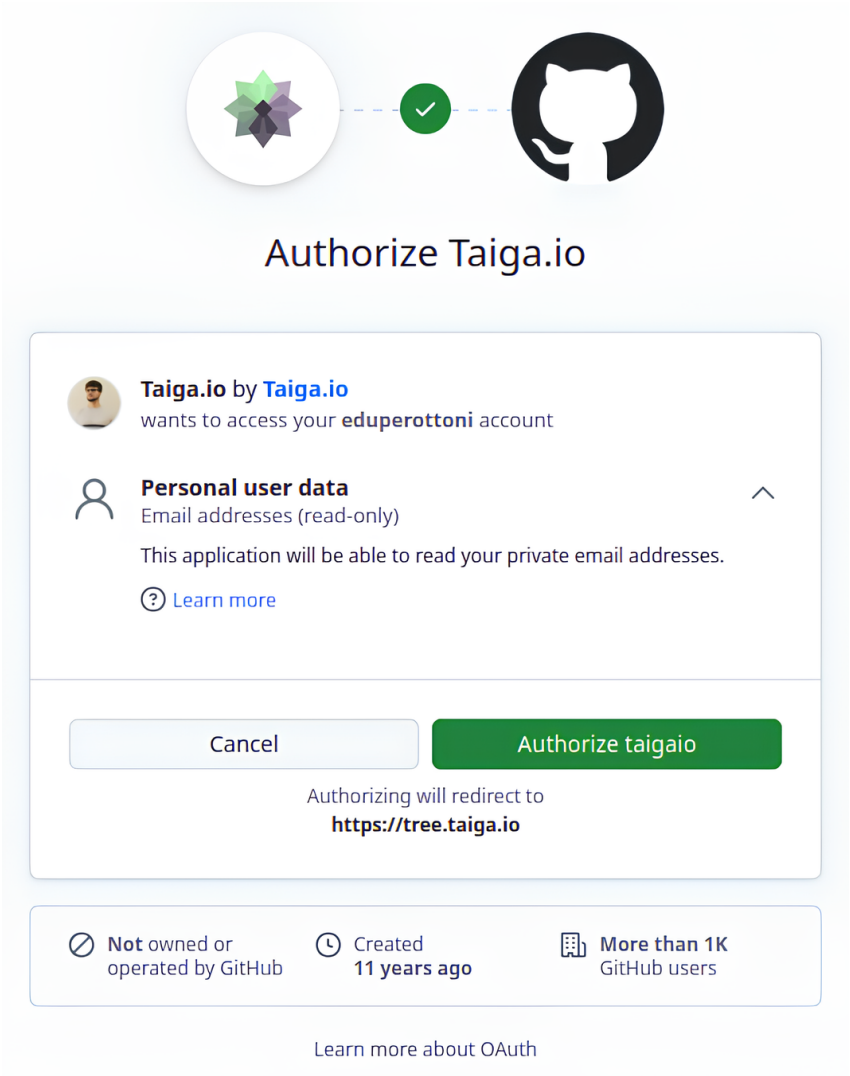
3.1.1 Múltiplos fatores de autenticação (MFA)

Métodos de autenticação são categorizados baseados em três categorias (Suleski et al., 2023):

- **Conhecimento (o que se sabe)**: Informações que o usuário deve lembrar. Exemplos dessa categoria são senhas, combinações de pergunta e resposta e códigos do tipo *Personal Identification Number* (PIN);
- **Posse (o que se tem)**: Se baseiam em aparelhos físicos que o usuário deve provar a posse no momento da autenticação. Chaves físicas, *Universal Serial Bus* (USB), aparelhos *mobile*, *smart cards* e *One-Time Password* (OTP) são exemplos de fatores de autenticação desta classe;
- **Inerência (o que se é)**: Baseada em biometria ou elementos que dizem respeito a padrão de comportamento humano. Exemplos são impressão digital, reconhecimento facial, de voz ou de íris e reconhecimento de padrão de digitação no teclado.

O conceito de MFA evoluiu a partir da Autenticação de Fator Único (SFA, *Single-Factor Authentication*) e da Autenticação de Dois Fatores (2FA, *Two-Factor Authentication*) (Ometov et al., 2018). Tradicionalmente, SFA utilizava um fator baseado em conhecimento.

Figura 2 – Formulário de autorização do protocolo OAuth 2.0.



Fonte: o autor.

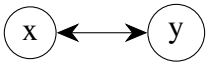
Posteriormente, como um passo adicional, passaram-se a utilizar fatores de posse, como *smart cards*. MFA foi proposto para prover um nível maior de segurança, como uma forma de proteger sistemas críticos de acessos não autorizados.

A utilização de múltiplos fatores, especialmente de diferentes categorias, aumenta radicalmente a segurança da autenticação e garante o nível de segurança requerido pela aplicação (Suleski et al., 2023; Ometov et al., 2018). Porém, como estudam Das et al. (2019), Khaskheli, Sherbaz e Shaikh (2022) e Baseer e Charumathi (2024), problemas de usabilidade surgem dependendo do arranjo entre número de fatores e a combinação dos mesmos. Em outras palavras, há um *trade-off* entre segurança e usabilidade: cada fator adicional acrescenta uma etapa ao processo de autenticação, elevando o tempo, o esforço e a carga cognitiva exigidos do usuário. Logo, combinações pouco adequadas, como o uso de vários fatores de posse ou de métodos impraticáveis em determinados dispositivos, podem aumentar o atrito e a taxa de falhas de autenticação, desestimulando o uso do sistema.

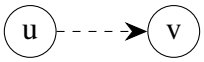
3.2 MODELOS DE GESTÃO DE IDENTIDADE

A literatura classifica modelos de gerenciamento de identidade, segundo a autenticação e o acesso do usuário a serviços e de como se dá a interação entre PS e PI. Apesar de haver conflitos de nomenclatura entre trabalhos, como entre Allen (2016) e Wangham et al. (2010), de forma geral, os modelos são tratados cronologicamente, de acordo com seu desenvolvimento e utilização. A seguir, apresentam-se esses modelos. A Figura 3 apresenta uma legenda para os diagramas que serão mostrados para descrição dos modelos.

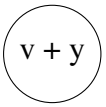
Figura 3 – Legenda para os diagramas que serão apresentados.



x interage com y e vice-versa



u confia em v



v e y são executados no mesmo serviço



i identifica a entidade v



Fonteiras de um domínio

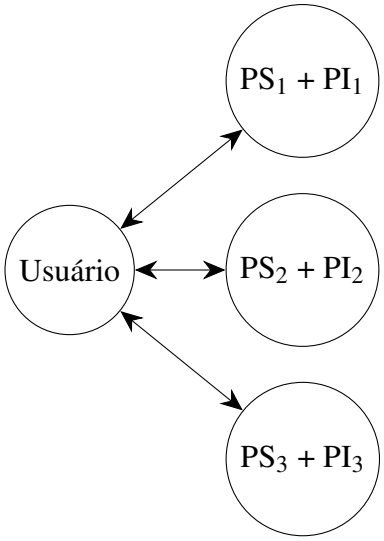
Fonte: o autor.

3.2.1 Modelo de Identidade Isolado

O modelo primordial é o modelo isolado. Nele, a identificação do usuário é tratada de forma isolada por cada PS, o qual também atua como PI (Wangham et al., 2010). Nesse caso, cada PS armazena as credenciais de cada um dos seus usuários e é responsável pela autenticação dos mesmos. A Figura 4 ilustra o modelo de identidade isolada. Nela, nota-se que PSs e PIs estão no mesmo ambiente.

O modelo isolado traz uma série de problemas, tanto do ponto de vista do usuário quanto dos provedores. O usuário deve gerenciar credenciais para cada um dos serviços acessados, o que gera uma enorme complexidade no quesito usabilidade, visto o crescimento de serviços online. Para resolver, ou (i) o usuário utiliza senhas parecidas de baixa entropia para

Figura 4 – Modelo de Identidade Isolada.



Fonte: o autor.

diferentes sistemas (Schardong; Custódio, 2022), ou (ii) escolhe senhas diferentes e não consegue memorizar todas elas. Ambos os casos geram problemas de segurança. (i), por exemplo, aumenta a superfície para ataques de *credential stuffing* (OWASP Foundation, 2024), técnica em que o atacante, de posse de credenciais vazadas em um serviço, as testa automaticamente em larga escala contra diversos outros serviços, na expectativa de que o usuário tenha reutilizado a mesma senha.

Do ponto de vista dos provedores, sem padronizações definidas, não há como garantir que eles implementem fluxos de autenticação seguros ou mesmo que armazenem dados sensíveis seguindo as boas práticas e respeitando leis de privacidade, como a Lei Geral de Proteção de Dados (LGPD) e a *General Data Protection Regulation* (GDPR). A responsabilidade sobre esses sistemas isolados é considerável, pois cada provedor precisa custodiar e proteger, de forma independente, dados que podem ser sensíveis. Como os mesmos dados de um indivíduo acabam replicados em diversos provedores, amplia-se a superfície de exposição e a gravidade de um eventual vazamento.

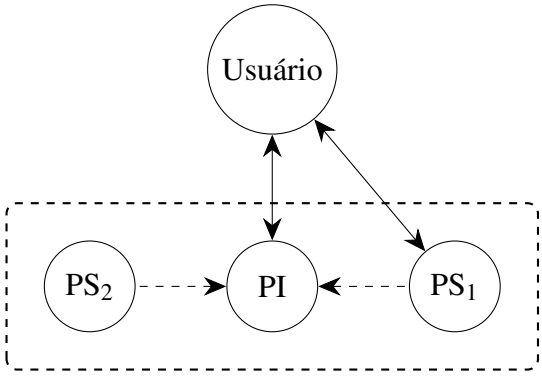
3.2.2 Modelo de Identidade Centralizado

Os problemas apresentados no modelo isolado levaram ao desenvolvimento do modelo centralizado. Nesse modelo, um único PI é utilizado por todos os provedores de serviço, tanto de forma exclusiva, quanto em adição a outros PIs (Jøsang; Pope, 2005) dentro da mesma hierarquia ou domínio. Assim como explicado por Schardong e Custódio (2022), o modelo centralizado é marcado pela separação de responsabilidades entre PS e PI e a comunicação entre ambos e o usuário é padronizada por protocolos como o SAML2 (OASIS, 2005), OAuth 2.0 (Hardt, 2012) e OIDC (Sakimura et al., 2023a).

Nesse modelo surge o conceito de Autenticação Única (SSO, *Single Sign-On*), um

mecanismo que usa uma única ação de autenticação, permitindo que o usuário autorizado, durante uma sessão, possa acessar vários serviços sem executar a ação de autenticação novamente (Radha; Reddy, 2012). O SSO traz uma grande facilidade para os usuários, pois a estes só é necessário realizar o processo de autenticação uma única vez e usufruir das credenciais obtidas em todos os PSs que utilizar, até que estas credenciais expirem (Wangham et al., 2010). A Figura 5 ilustra esse modelo.

Figura 5 – Modelo de Identidade Centralizada.



Fonte: o autor.

Porém, com o número de organizações crescendo, o usuário ainda deve lidar com diversas credenciais, uma para cada hierarquia organizacional que utiliza na internet. Outra consequência desse esquema é que o PI ganha poder, por gerenciar as credenciais e realizar a autenticação de todos os usuários. Segundo Allen (2016), "centralização inatamente dá poder às entidades centralizadas e não aos usuários".

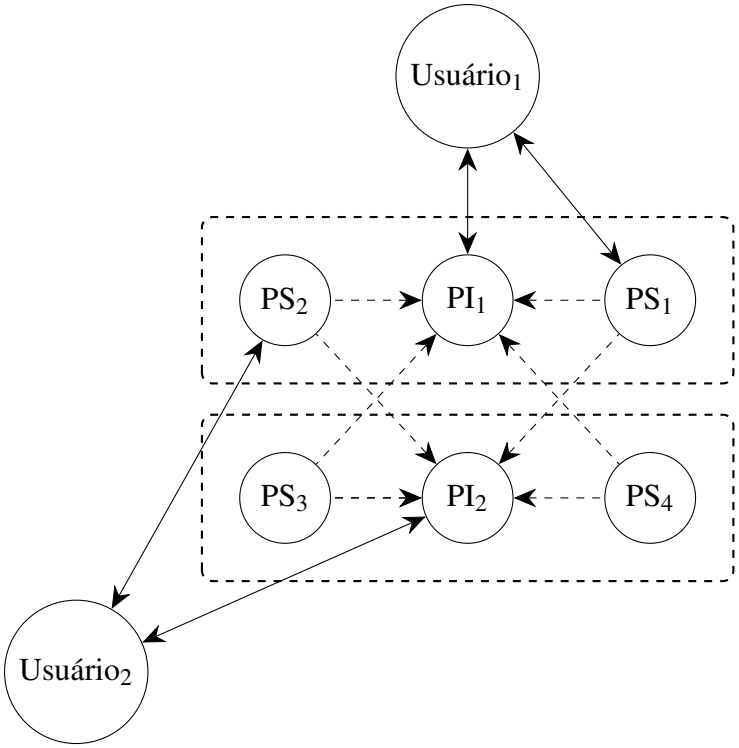
3.2.3 Modelo de Identidade Federada

O próximo avanço para a identidade digital ocorreu quando uma variedade de organizações comerciais se moveram além da hierarquia do próprio domínio (Allen, 2016). No modelo de IDF, acordos são estabelecidos entre PIs e PSs de forma que identidades de diferentes domínios são reconhecidas por todos os domínios da federação (Jøsang; Pope, 2005).

Nesse modelo ainda vale o conceito de SSO, assim como os protocolos de comunicação citados anteriormente. Ainda, é essencial manter um mecanismo de estabelecimento de confiança entre os provedores da federação: PS deve confiar no PI e nas identidades autenticadas, assim como na integridade das informações emitidas por ele acerca dos usuários. A Figura 6 ilustra o paradigma federado. Na imagem, o $\texttt{Usuário}_2$ é autenticado no $\texttt{PI}_2$ e utiliza os serviços do $\texttt{PS}_2$, sendo que os provedores em questão pertencem a domínios diferentes.

Esse modelo foi aplicado no desenvolvimento de federações acadêmicas e de pesquisa ao redor do mundo, dada a necessidade de compartilhamento de recursos entre usuários de diferentes instituições (Wangham et al., 2010). A federação norte-americana *Incommon* (InCommon Federation, 2024) e a CAFe (RNP, 2024a) são exemplos de implementações do modelo de IDF para academia. Desde então, o SAML2 foi adotado como protocolo oficial para

Figura 6 – Modelo de Identidade Federada.



Fonte: o autor.

o desenvolvimento dessas federações, sendo o *Shibboleth* a implementação mais utilizada para implantação dos PIs.

3.2.4 Modelo Centrado no Usuário

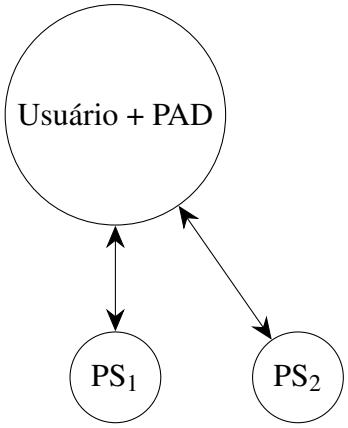
Apesar da evolução dos modelos com o intuito de melhorar a experiência do usuário, nenhum deles aumenta o poder do usuário sobre suas credenciais. É isso que o modelo centrado no usuário propõe. Como explicam Jøsang e Pope (2005), neste modelo, o usuário tem a posse de um Dispositivo de Autenticação Pessoal (PAD, *Personal Authentication Device*), utilizado para autenticação, cumprindo o papel de PI. Uma vez autenticado, o usuário pode enviar suas credenciais diretamente ao PS. A Figura 7 ilustra esse modelo.

Contudo, o modelo centrado no usuário não ganhou tração e é atualmente visto como uma extensão dos modelos de PI com maior controle do usuário (Schardong; Custódio, 2022). Segundo elucida Allen (2016), as implementações de modelos centrados no usuário focaram em consenso do usuário e interoperabilidade. Com essas funcionalidades, o usuário tem controle sobre os dados que compartilha com os PSs acessados.

3.2.5 Modelo de Identidade Autossoberana

O modelo centrado no usuário foi um passo importante rumo ao controle do usuário sobre sua identidade, mas uma real mudança precisava considerar a autonomia do usuário (Al-

Figura 7 – Modelo de Identidade Centrada no Usuário.



Fonte: o autor.

len, 2016). O modelo de Identidade Autossoberana promete que apenas os indivíduos sejam proprietários de sua identidade digital, podendo criar e controlar suas credenciais (Pöhn; Hommel, 2023). Isso distingue esse modelo dos anteriores, os quais viam o indivíduo como usuário (Schardong; Custódio, 2022). Nesse modelo, o indivíduo é o centro do sistema, ele tem a soberania sobre seus dados e decide com quem irá compartilhá-los.

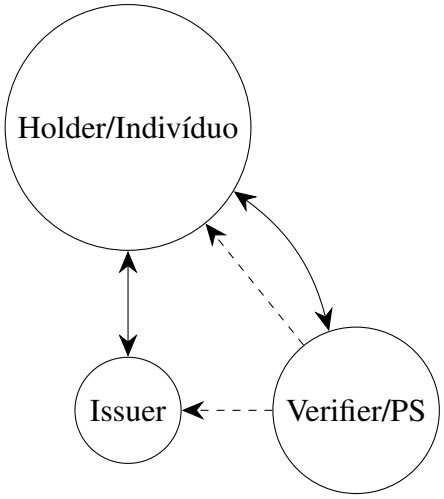
A arquitetura da Identidade Autossoberana pode ser construída com o suporte de uma Tecnologia de Registro Distribuído (DLT, *Distributed Ledger Technology*), estrutura de armazenamento distribuído de dados, como uma *Blockchain* (Mühle et al., 2018). Nesse caso, esse registro pode armazenar Identificadores Descentralizados (DIDs, *Decentralized Identifier*s) (Soltani; Nguyen; An, 2021). O identificador é pareado com um meio de autenticação e isso permite a verificação do identificador por qualquer entidade apenas fazendo a leitura no registro e um desafio ao indivíduo (Mühle et al., 2018).

Nesse esquema há alguns papéis específicos, como (i) *holder*, o indivíduo ou entidade que controla credenciais verificáveis, suas ou de terceiro; (ii) *issuer*, o criador das credenciais verificáveis (agência governamental ou bancos, por exemplo); e (iii) *verifier*, entidade que obtém as credenciais verificáveis do *holder* para verificar e utilizá-las (provedores de serviço em geral cumprem esse papel). A Figura 8 ilustra a interação entre essas entidades.

Como fluxo geral do modelo, o *issuer* emite Credenciais Verificáveis (VCs, *Verifiable Credential*s), uma estrutura de dados adequada para representar atributos sobre o usuário criptograficamente verificáveis e à prova de adulteração (Soltani; Nguyen; An, 2021). Um ou mais atributos, do mesmo ou de diferentes VCs, podem ser organizados em uma Apresentação Verificável (VP, *Verifiable Presentation*), compartilhada com o PS (Schardong; Custódio, 2022).

Graças à tecnologia de Prova de Conhecimento Zero (ZKP, *Zero Knowledge Proof*), o *holder* pode provar que ele possui uma VC contendo um valor sem revelar o valor real (Sporny et al., 2025). Sporny et al. (2025) ainda citam três propriedades importantes garantidas pela ZKP: (i) compartilhar apenas as informações necessárias com o *verifier* por meio da VP (divul-

Figura 8 – Modelo de Identidade Autossoberana.



Fonte: o autor.

gação seletiva); (ii) técnicas como assinaturas cegas possibilitam que o titular utilize diferentes valores de assinatura em apresentações distintas, dificultando a correlação entre elas (divulgação não vinculável); e (iii) comprovar a posse ou a emissão de uma credencial sem revelar identificadores persistentes, permitindo combinar múltiplas credenciais de diferentes emissores em uma única apresentação sem expor a identidade dos emissores ou do sujeito.

3.3 ECOSSISTEMA SAML E CAFe

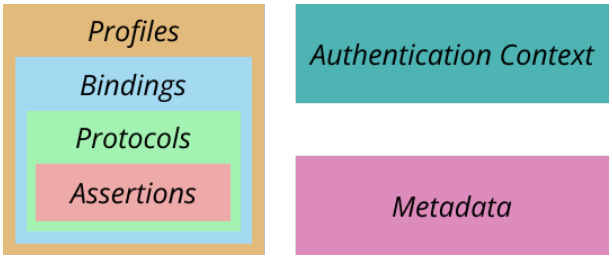
3.3.1 Protocolo SAML

O SAML2 é um protocolo que define uma estrutura baseada em *Extensible Markup Language* (XML) para descrever e trocar informações de segurança entre PSs e PIs. As mensagens entre ambos são transmitidas na forma de asserções SAML2 portáteis que podem ser trocadas por aplicações que operam entre limites de domínios de segurança. A especificação OASIS (2005) define a sintaxe e regras precisas para solicitar, criar, comunicar e usar essas asserções.

Essa especificação foi adotada mundialmente por apresentar uma solução de autenticação SSO para serviços *web* de diferentes domínios, impulsionando sua adoção para o gerenciamento de identidades federadas. A especificação é formada por vários componentes, que, quando combinados, permitem o suporte a vários casos de uso (Hughes; Maler, 2005). A Figura 9 ilustra os componentes e a relação entre os mesmos.

- **Asserções (*Assertions*):** Pacotes de informação contendo declarações sobre um sujeito (o *principal*) emitidos por uma autoridade SAML2 (*asserting party*). A estrutura e o conteúdo de uma asserção são definidas por um *Schema* de Asserção SAML2. Uma asserção pode ter três tipos de declarações: (i) declarações de autenticação, criadas pela entidade

Figura 9 – Componentes da especificação SAML2.



Fonte: Adaptado de (Hughes; Maler, 2005).

que autentica o usuário, com informações referentes à autenticação; (ii) declarações de atributo, que contém informações que identificam o sujeito; e (iii) declarações de decisão de autorização, que definem o que o usuário está autorizado a fazer;

- **Protocolos (*Protocols*):** Mensagens de solicitação/resposta definidas pelo SAML2, usadas para solicitar e retornar asserções e outras informações. Exemplos incluem o Protocolo de Solicitação de Autenticação, Protocolo de Single Logout, Protocolo de Consulta e Solicitação de Asserção e Protocolo de Resolução de Artefato;

- **Mapeamentos (*Bindings*):** Detalham como as mensagens do protocolo SAML2 podem ser transportadas por protocolos de comunicação subjacentes, como *Hypertext Transfer Protocol* (HTTP) ou *Simple Object Access Protocol* (SOAP). Exemplos incluem *HTTP Redirect Binding*, *HTTP POST Binding*, *HTTP Artifact Binding*, *SAML SOAP Binding* e *Reverse SOAP (PAOS) Binding*.

- **Perfis (*Profiles*):** Definem conjuntos específicos de regras para usar e restringir a sintaxe do SAML2, combinando asserções, protocolos e mapeamentos, para resolver casos de uso de negócio específicos de maneira interoperável. Exemplos de perfis são o Perfil *Web Browser* SSO, *Single Logout* e Perfis de Federação de Identidade.

- **Metadados (*Metadata*):** Esse componente define maneiras de expressar e compartilhar informações de configuração entre partes do protocolo, como URLs de *endpoints* de serviço, papéis operacionais (PI, PS) e chaves para verificação de assinaturas. Esses metadados são armazenados em um arquivo XML com um formato específico, que pode ser trocado entre as entidades do protocolo.

- **Contexto de Autenticação (*Authentication Context*):** Usado em uma declaração de autenticação para carregar informações detalhadas sobre o tipo e a força da autenticação usada.

Em se tratando de entidades do protocolo, o SAML2 define entidades gerais como a Autoridade SAML2 (ou *asserting party*), que emite asserções e a *Relying Party*, que utiliza as asserções. Além disso, há o sujeito, uma entidade que pode se autenticar. Para o caso de

SSO, a especificação também define os papéis de PS e PI. Ainda, há a Autoridade de Atributos (Attribute Authority), ente que produz asserções em resposta a consultas de atributos de identidade (Hughes; Maler, 2005).

O padrão define mecanismos de segurança como assinaturas digitais e criptografia XML, usadas para garantir a integridade das mensagens e proteger contra ataques como *man-in-the-middle*. Ademais, o SAML2 é projetado para ser flexível. *Schemas* de asserções e protocolos podem ser estendidos, assim como novos perfis podem ser definidos.

A troca de mensagens entre as entidades, no SAML2, assume que as partes possuem uma relação de confiança pré-estabelecida (Hughes; Maler, 2005). No SAML2, a confiança é construída por meio da troca de metadados entre provedores. PS e PI devem obter os metadados da outra parte e registrar esses metadados como sendo confiáveis. Esse acordo deve ser feito antes de qualquer interação do usuário com os provedores (Ferdous; Poet, 2013). Isso se torna complexo à medida que o número de IdPs e SPs cresce, impactando a escalabilidade de soluções que utilizam o SAML2.

3.3.2 Shibboleth

O *Shibboleth* é um projeto criado pelo consórcio *Internet2* (Internet2, 2025) em 2000. Ele entrega uma solução de gerenciamento FID, inicialmente pensada para tratar os desafios de identidade e controle de acesso em instituições acadêmicas, mas atualmente visa ser uma solução genérica de FID (Wangham et al., 2010). O sistema permite realização de SSO e usa o SAML2 2.0 para lidar com a troca de mensagens entre entidades. Os principais componentes do *Shibboleth* são listados abaixo (Shibboleth, 2025):

- **PI**: Também referido como *origin*. É responsável pela autenticação do usuário e provê informações sobre o mesmo para o PS. Está localizado na organização de origem, a qual mantém as credenciais e dados do usuário. Comumente, as credenciais do usuário são armazenadas em uma base de dados LDAP, um protocolo de aplicação, de padrão aberto na indústria, concebido para aceder e gerir serviços de informação de diretórios distribuídos sobre redes IP (Sermersheim, 2006). O servidor LDAP atua tipicamente como o repositório central autorizado onde são armazenadas as credenciais e os atributos dos usuários, tais como alunos, professores e pesquisadores (Howes; Smith; Good, 2003).

- **PS**: Também referido como *target*. Protege um recurso *web*, consome informações do PI e concede acesso ao usuário após autenticação. Pode estar localizado em qualquer domínio da federação.

- **WAYF**: Também referido como Serviço de Descoberta (DS, *Discovery Service*). É opcional e utilizado para que o PS descubra qual a organização de origem e, consequentemente, qual o PI do usuário que requisita o recurso.

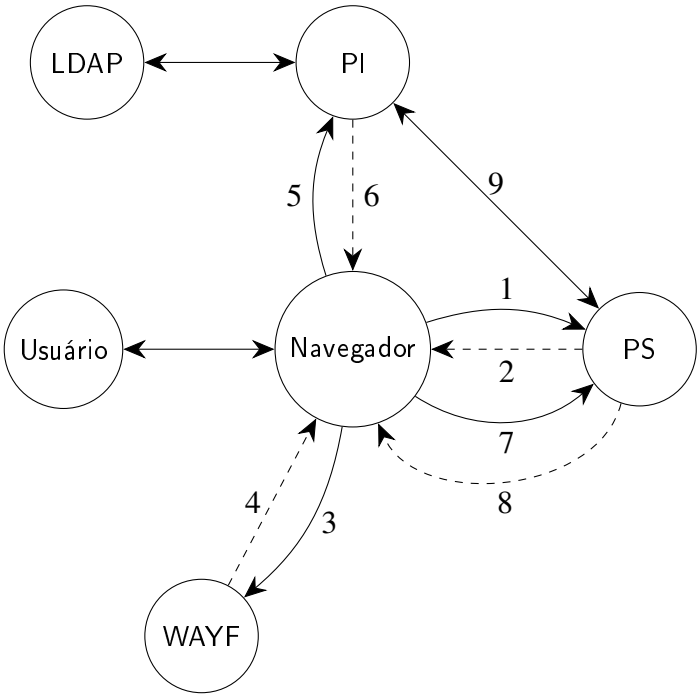
A Figura 10 mostra o fluxo de autenticação de um usuário ainda não autenticado que acessa determinado serviço. Primeiramente, o usuário acessa o serviço por meio de seu navegador *web*, requisitando um recurso protegido, *i.e.*, que exige autenticação do usuário (1). O PS verifica que o usuário não está autenticado. A princípio, o PS não sabe qual dos PIs da federação contém as credenciais do usuário e, para descobrir, redireciona o usuário para o WAYF (2 e 3). Na interface do WAYF são enumerados apenas os PIs confiados pelos PSs da federação (Wangham et al., 2010).

Após selecionar sua instituição de origem, o usuário é redirecionado novamente, agora para o PI selecionado no WAYF (4 e 5). Caso o usuário tenha selecionado sua instituição previamente, a seleção manual entre uma lista de PIs pode ser ignorada. Nesse caso, utilizam-se *cookies* de navegador para armazenar a instituição e o usuário deve apenas confirmar a seleção (Switchaai, 2025).

Já no PI, o usuário executa o processo de autenticação. Depois, baseado no nome do sujeito, no PS e no perfil SAML2 específico configurado, o PI decide quais informações enviará ao PS. Essas informações são empacotadas em uma asserção SAML2. Essa asserção é assinada com a chave privada do PI, para garantir autenticidade. No caso do SAML2 2.0, a asserção é cifrada com a chave pública do PS, para garantir confidencialidade. A asserção (ou uma referência a ela) é, então, enviada como uma resposta HTTP ao navegador (6) para ser entregue ao PS, em um *endpoint* chamado *Assertion Consumer Service* (7) (Wiki, 2019).

Finalmente, caso a decifragem e checagens de segurança da resposta estiverem corretas, o PS redireciona o navegador ao recurso *web* desejado (8). O PS ainda pode requisitar outros atributos do usuário ao PI (9).

Figura 10 – Fluxo de autenticação no *Shibboleth*.



Fonte: Adaptado de (Wangham et al., 2010).

O estabelecimento de confiança entre PS e PI, no caso do *Shibboleth*, é definido pela troca de metadados entre as partes. Esse tipo de modelo de confiança leva a desvantagens conhecidas: dificuldade de implantação e alta dependência de autoridades centrais (Almenárez et al., 2009). Uma federação implica a agregação de metadados em uma grande lista de provedores que concordam em seguir regras e políticas pré-estabelecidas (Cabarcos et al., 2009).

Para inclusão de um provedor na federação, é necessário que um bloco de metadados referentes ao provedor seja inserido em um grande arquivo de metadados da federação, gerenciado pelo operador da federação. Esse grande arquivo de metadados serve para informar aos seus participantes quais são os servidores reconhecidos e confiáveis (RNP, 2020).

3.3.3 CAFe

A RNP disponibiliza a CAFe desde 2009 às instituições de ensino e pesquisa brasileiras. A organização que adere à federação e concorda em cumprir com a sua política é denominada "membro" da federação. Por meio da CAFe é possível que os usuários vinculados aos membros da federação, utilizando suas contas institucionais, possam acessar serviços fora do perímetro administrativo da sua instituição (RNP, 2021). No momento da escrita deste trabalho, a CAFe agrega 384 membros (RNP, 2025).

Segundo as diretrizes da CAFe, existem duas categorias de membros da federação (RNP, 2021):

- **Pleno**: instituição qualificável como Organização Usuária do Sistema RNP. Possui o direito de registrar entidades do tipo Provedor de Identidade ou Provedor de Serviço;
- **Parceiro**: instituição que deseja apoiar a federação e possui serviços de interesse do Sistema RNP. Possui o direito de registrar apenas entidades do tipo Provedor de Serviço.

A CAFe é construída com base no SAML2/*Shibboleth* e, portanto, reflete as características e limitações do *framework*. Para registrar um provedor na federação, longos processos são necessários, sendo eles executados pelo operador da federação, de forma centralizada. O documento de Declaração de Práticas de Registro de metadados explica as práticas de adesão de um PS ou PI à CAFe (Belotto; Rocha; Ribeiro, 2020).

Os metadados de todos os provedores (PSs e PIs) da CAFe são listados em um grande arquivo XML[1], gerenciado pelo operador da CAFe. O documento RNP (2020)[2] explica como isso é operacionalizado entre membros da federação e operador.

Em um ambiente federado, para prover interoperabilidade entre provedores, é essencial que atributos referentes ao usuário emitidos pelos PIs da federação estejam conforme esperado pelos PSs. A CAFe impõe que PIs de membros da federação entreguem atributos específicos nas asserções SAML2 que são enviadas aos PSs. Porém, na prática, essa padronização não

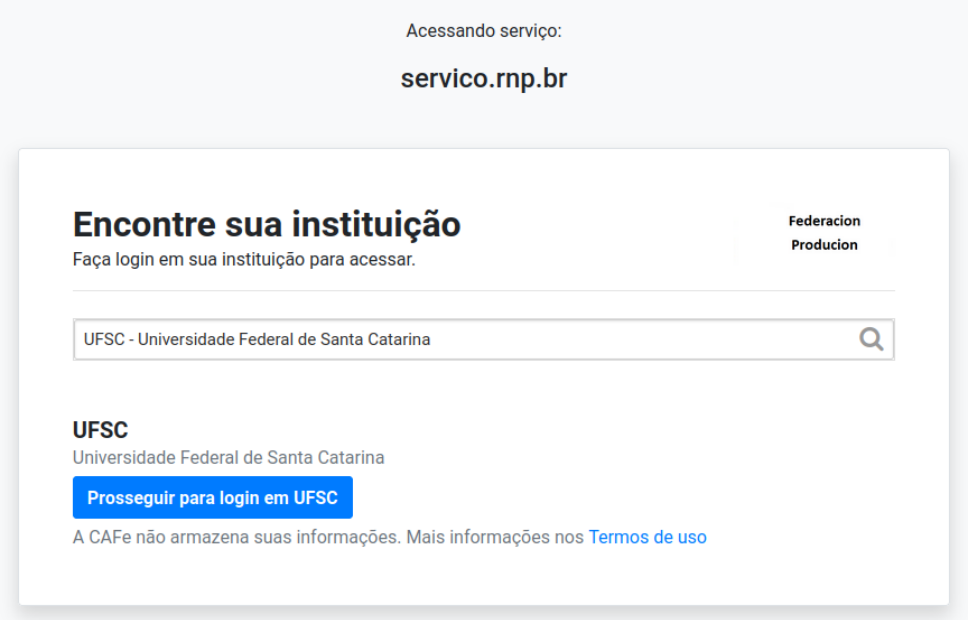
[1] https://ds.cafe.rnp.br/metadata/cafe-metadata.xml
[2] https://ajuda.rnp.br/cafe/processo-de-adesao/adesao-idp/inclusao-de-metadados-na-federacao-chimarrao

acontece e alguns PIs geram asserções semanticamente incorretas, com valores de atributos sendo entregues em campos inadequados, como, por exemplo, o valor do *eduPersonPrincipalName* sendo entregue no campo *uid* (RNP, 2021). Isso pode causar falhas de autenticação e acesso a serviços, além de dificultar o gerenciamento da federação.

Ademais, alinhada com padrões internacionais definidos pela REFEDS e utilizados por interfederações como a eduGAIN, a CAFe adota *schemas* LDAP padronizados de definição de atributos, como o *SCHema for ACademia* (SCHAC), o *eduPerson* (Refeds, 2020) e o *inetOrgPerson* (Smith, 2000). Adicionalmente, define o *schema brEduPerson* (Cafe, 2020), uma extensão com atributos necessários no contexto brasileiro de federação.

No caso da CAFe, durante o fluxo de autenticação do usuário, o WAYF lista todos os PIs previamente presentes no arquivo de metadados da federação. A Figura 11 ilustra a interface atual desse serviço.

Figura 11 – Interface do serviço WAYF da CAFe.



Fonte: o autor.

Em se tratando de autenticação, na CAFe, todo o procedimento é responsabilidade do PI da instituição. As regras que PIs da federação devem seguir são listadas em um documento de políticas (Dagser, 2011). Qualquer modificação necessária na etapa de autenticação, como a garantia de MFA na federação, exige adoção por cada um dos membros, que precisam atualizar sua instância *Shibboleth*. Esse processo tende a ser lento, o que dificulta o gerenciamento e a atualização da federação.

3.4 ECOSSISTEMA OPENID CONNECT E OPENID FEDERATION

3.4.1 OAuth 2.0 e OpenID Connect

OAuth 2.0 é um *framework* de autorização que permite a uma aplicação de terceiro obter acesso limitado a recursos protegidos, tipicamente dados de um usuário expostos por um serviço *web* sobre HTTP, sem que o usuário precise compartilhar suas credenciais com essa aplicação. O acesso ocorre em nome do proprietário do recurso, isto é, o usuário a quem os dados pertencem, que autoriza a aplicação por meio de uma interação de aprovação intermediada pelo serviço. Alternativamente, o *framework* também contempla o cenário em que a própria aplicação acessa recursos em seu próprio nome, sem um usuário envolvido (Hardt, 2012). A especificação define diferentes tipos de fluxos para obtenção e utilização de *access tokens*, credenciais utilizadas para acesso a recursos protegidos com escopo e duração específicos e geralmente, opacos para o cliente.

O OAuth 2.0 define quatro papéis principais (Hardt, 2012):

- **Dono do recurso (*Resource Owner*)**: A entidade que pode conceder acesso a um recurso protegido (geralmente o usuário final quando se trata de indivíduos);
- **Aplicação Cliente (*Client*)**: Uma aplicação que faz requisições a recursos protegidos em nome do proprietário do recurso e com sua autorização;
- **Servidor de Autorização (AS, *Authorization Server*)**: O servidor que emite *access tokens* para o cliente após autenticar com sucesso o proprietário do recurso e obter autorização;
- **Servidor de Recursos (RS, *Resource Server*)**: O servidor que hospeda os recursos protegidos, capaz de aceitar e responder a requisições usando *access tokens*.

O fluxo abstrato do OAuth 2.0 envolve (i) o cliente requisitando autorização do proprietário do recurso; (ii) recebendo uma concessão de autorização (*authorization grant*); (iii) trocando essa concessão por um *access token* no servidor de autorização; e (iv) usando o *access token* para acessar recursos no servidor de recurso. O protocolo define tipos de concessão de autorização (*grant types*), incluindo *Authorization Code*, *Implicit*, *Resource Owner Password Credentials*, e *Client Credentials*.

Ele utiliza recursos disponíveis via Interfaces de Programação de Aplicações (APIs, *Application Programming Interface*s), como o *endpoint* de autorização para interagir com o proprietário do recurso e obter a concessão e *endpoint* de obtenção de *token* para trocar a concessão por um *access token*. No entanto, o OAuth 2.0 por si só não define métodos padrões para fornecer informações de identidade e é incapaz de fornecer informações sobre a autenticação de um usuário final (Sakimura et al., 2023a).

O OIDC é uma camada de identidade construída sobre o protocolo OAuth 2.0. Ele estende o processo de autorização do OAuth 2.0 para incluir funcionalidades de autenticação.

O principal propósito do OIDC é permitir que aplicações-cliente, no contexto OIDC, chamados de RP, verifiquem a identidade do usuário final com base na autenticação realizada por um AS, chamado de OP.

Esse protocolo também permite obter informações básicas de perfil sobre o usuário final de forma interoperável e *REST-like*. Ele adiciona uniformidade para atributos comuns (como nome, e-mail, endereço, número de telefone) e permite que PSs solicitem MFA aos PIs.

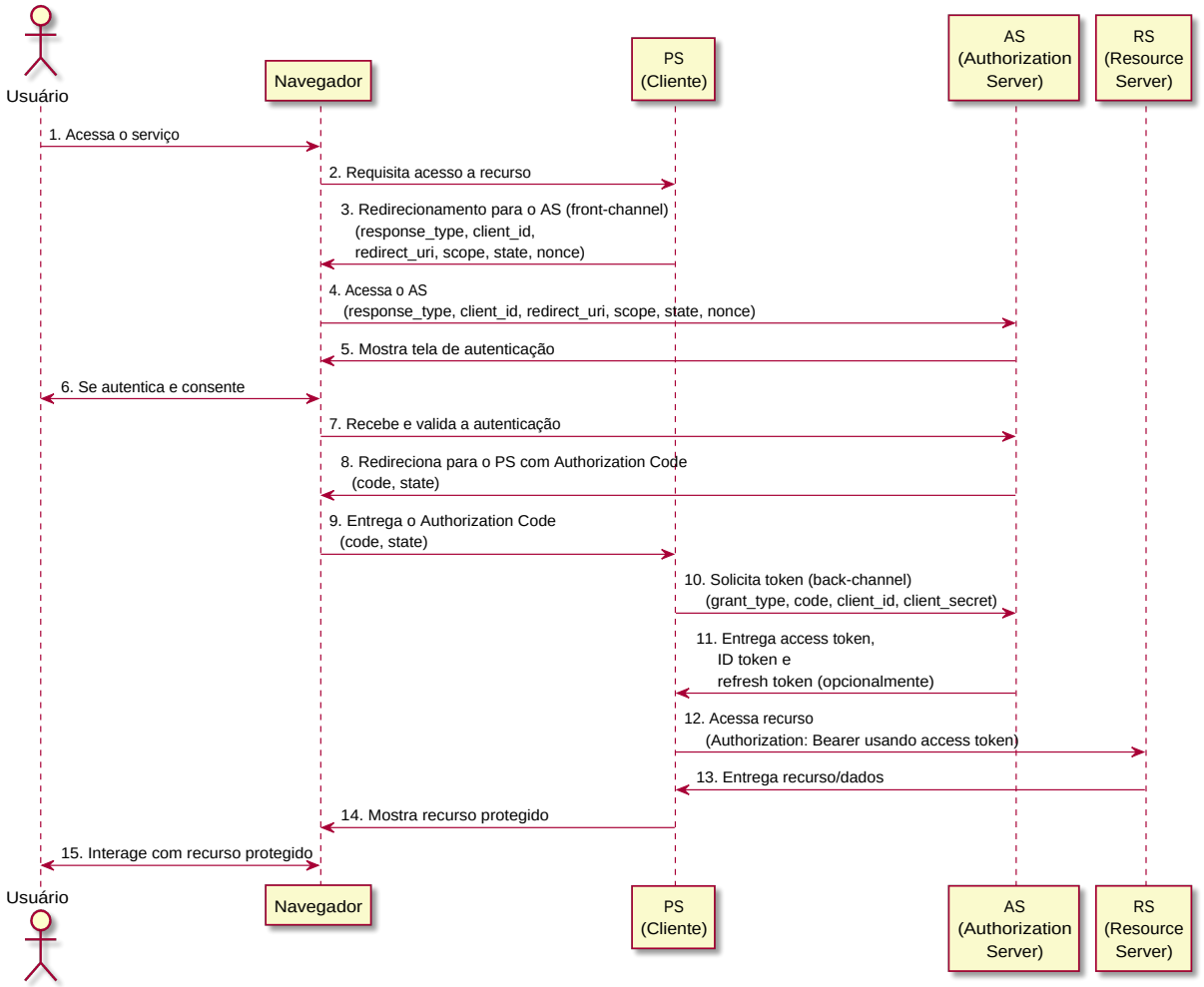
A principal extensão do OIDC em relação ao OAuth 2.0 é o *ID Token*. Trata-se de um token de segurança, representado como um *JSON Web Token* (JWT) (Jones; Bradley; Sakimura, 2015), contendo *claims*, que consistem em asserções estruturadas no formato de par chave-valor, utilizadas para transmitir de forma padronizada os atributos de identidade do usuário. O *ID Token* contém *claims* sobre o evento de autenticação de um usuário pelo OP e *claims* solicitadas na requisição de autenticação, como o tempo de autenticação (`auth_time`) e o destinatário do *token* (`aud`). Opcionalmente, pode conter informações sobre o nível de autenticação (*Authenication Context Class Reference*, no *claim* `acr`) e os métodos utilizados durante a autenticação (*Authentication Mehods Reference*, no *claim* `amr`). O *ID Token* deve ser assinado usando *JSON Web Signature* (JWS) e, opcionalmente, criptografado usando *JSON Web Encryption* (JWE) (Sakimura et al., 2023a).

O OIDC utiliza os *endpoints* de Autorização e de Token do OAuth 2.0. Além disso, define o *endpoint userinfo*, um recurso protegido que retorna *claims* autorizados sobre o usuário quando um *access token* é apresentado pelo cliente. O cliente obtém as *claims* via escopos especificados na norma, como *profile*, *email*, *address* ou *phone*.

A autenticação no OIDC pode seguir um dos três fluxos principais, determinados pelo parâmetro `response_type` na requisição de autenticação: (i) *Authorization Code Flow*, (ii) *Implicit Flow* e (iii) *Hybrid Flow*. O fluxo (ii) e certas variações do (iii) apresentam problemas conhecidos de segurança, como a exposição de *tokens* diretamente no navegador do usuário (comunicação via *front-channel*), o que os torna vulneráveis a ataques de injeção e vazamento no histórico (Lodderstedt; Mcgloin; Hunt, 2013; Lodderstedt et al., 2023). Por este motivo, as melhores práticas atuais recomendam o uso exclusivo do fluxo (i), preferencialmente aliado ao *Proof Key for Code Exchange* (PKCE) (Lodderstedt et al., 2023). A Figura 12 ilustra o fluxo de autenticação do usuário para o modo (i).

O usuário requisita acesso a recurso protegido (1). A aplicação cliente detecta que o usuário não está autenticado ou não tem as permissões necessárias para o recurso. Então, ela constrói uma URL de autorização e redireciona o navegador do usuário para o AS (2). Essa URL inclui parâmetros cruciais como `response_type`, `client_id`, `redirect_uri`, `scope`, `state`. O AS recebe a requisição de autorização e, se o usuário não estiver autenticado, apresenta uma tela de *login*. Após a autenticação, ou se o usuário já estiver logado, o AS exibe uma tela de consentimento (3), informando ao usuário quais informações o cliente está solicitando acesso. O usuário insere suas credenciais e/ou concede permissão para o cliente acessar as informações e recursos solicitados (4). Após, o AS redireciona o navegador do usuário de volta para a `redirect_uri`. Esta URL inclui os parâmetros `code` e `state` (5). Ao receber o *Authorization*

Figura 12 – Fluxo de autenticação do OIDC.



Fonte: o autor.

Code, cliente faz uma requisição direta via *back-channel* para o AS, no *endpoint* de token, trocando *code* por *tokens* (6). O AS valida a requisição do cliente e retorna ao cliente o *access token*, *id token* e opcionalmente o *refresh token*, um token de longa duração que o cliente pode usar para obter novos *access tokens* (7). Com o *access token* em mãos, o cliente pode fazer requisições para o RS. O *access token* é incluído no cabeçalho *Authorization* da requisição (8). O RS recebe a requisição, valida o *access token* e retorna os recursos ou dados solicitados ao cliente (9). Finalmente, o cliente pode utilizar os dados do usuário para determinada operação em nome do usuário (10).

3.4.2 OpenID Federation 1.0

Esse protocolo descreve métodos e ferramentas para que a confiança entre duas entidades seja estabelecida por meio da confiança de ambas em uma terceira parte, chamada Âncora de Confiança (TA, *Trust Anchor*) (Hedberg et al., 2026). O método de estabelecimento de confiança carrega ideias do padrão de Infraestrutura de Chaves Públicas (ICP) X.509, porém

com várias adaptações considerando um mundo conectado e escalabilidade da rede de confiança (Marco et al., 2024b).

O protocolo é genérico, pois além de permitir entidades que utilizam OIDC, também habilita clientes e servidores OAuth 2.0 ou provedores de carteiras digitais ou VCs (Connect2id, 2024). Uma FID pode, portanto, ser estabelecida a partir desse protocolo, onde as duas entidades são PI e PS e a terceira parte confiável é o operador da federação (Hedberg et al., 2026). Como resultado, o provisionamento manual de entidades confiáveis, por meio de um portal *web*, como é feito pelo SAML2/*Shibboleth*, não é mais necessário (Connect2id, 2024).

3.4.2.1 Entidades do Protocolo

O OID Fed define alguns papéis específicos no protocolo:

- **TA**: Entidade que tem o propósito de emitir declarações sobre entidades subordinadas. É a terceira parte confiável no estabelecimento de confiança entre dois outros membros. No caso de uma FID, o operador administra a TA raiz da federação;
- **Entidade Intermediária (IM, *Intermediate Entity*)**: Entidade que aparece em algum lugar entre a TA e um provedor na hierarquia da federação. Também emite declarações sobre seus subordinados;
- **Entidade Folha (LE, *Leaf Entity*)**: Entidade que não possui subordinados. Tipicamente desempenha um papel no protocolo de autenticação. Em uma FID baseada nos protocolos OIDC ou OAuth 2.0, ela se refere a (i) PS (ou RP); (ii) PI, (ou OP); (iii) *OAuth Client*; (iv) AS; ou (v) RS. No contexto de VCs, uma LE pode ser um verificador ou emissor de VCs.

A Figura 13 ilustra duas federações com os papéis explicados acima. Nela, uma seta da entidade *x* para a entidade *y* significa que *x* emite declarações sobre *y*.

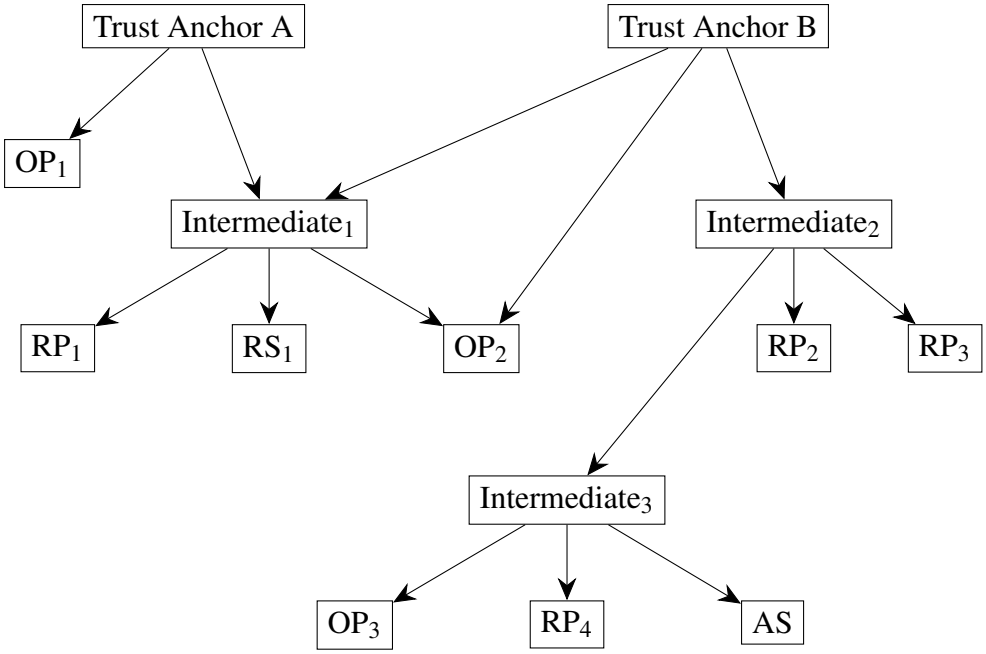
3.4.2.2 Declarações de Entidade

Uma Declaração de Entidade (ES, *Entity Statement*), é o componente básico do protocolo. Ele consiste em um JWT assinado e contém as informações necessárias para que a entidade identificada como sujeito (*claim* `sub`) do JWT participe da federação. O ES é emitido e assinado pela entidade identificada como emissora (*claim* `iss`) do JWT.

No OID Fed, todas as entidades emitem um ES sobre si mesmas, *i.e.* autoassinado, chamado de Configuração da Entidade (EC, *Entity Configuration*). Além disso, entidades superiores (TAs e IMs) publicam um ES para cada uma das entidades imediatamente subordinadas, chamado de Declarações de Subordinados (SSs, *Subordinate Statement*s).

No exemplo da Figura 13, a TA `A` emite SSs para $\texttt{OP}_1$ e para $\texttt{Intermediate}_1$. A TA `B` também emite SSs para $\texttt{Intermediate}_1$, $\texttt{OP}_2$ e $\texttt{Intermediate}_2$. $\texttt{Intermediate}_1$ publica um

Figura 13 – Exemplo de federações ancoradas em A e B.



Fonte: Fonte: Adaptado de (Hedberg et al., 2026).

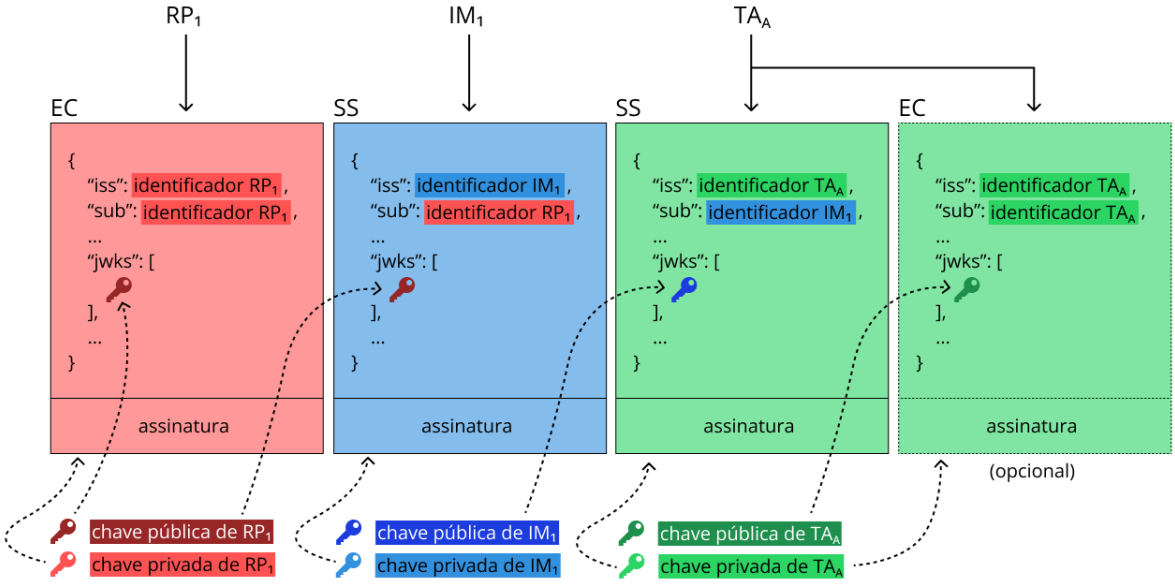
SSs sobre $\texttt{RP}_1$, $\texttt{RS}_1$ e $\texttt{OP}_2$, assim como $\texttt{Intermediate}_2$ e $\texttt{Intermediate}_3$ fazem em relação aos seus subordinados imediatos. Todas as entidades publicam ECs sobre si mesmas.

3.4.2.3 Cadeia de Confiança

Uma Cadeia de Confiança (TC, *Trust Chain*) é uma sequência de ESs assinadas que estabelece uma relação de confiança entre entidades em uma federação. Ela começa com a EC do sujeito da cadeia (geralmente uma LE), segue com zero ou mais SSs emitidos por IMs e o SS emitido pela TA sobre o IM imediatamente subordinado a ela, caso exista. A cadeia termina com a EC da TA, embora essa declaração seja opcional na cadeia. (Hedberg et al., 2026). A Figura 14 ilustra a TC relativa ao $\texttt{RP}_1$ do exemplo de federação da Figura 13. As setas de uma entidade para um ES indicam que a entidade emitiu o ES. Todos os objetos vermelhos na imagem indicam que o objeto pertence a ou foi emitido por $\texttt{RP}_1$. Da mesma forma, objetos azuis para o $\texttt{Intermediate}_1$ e os verdes para a `Trust Anchor A`.

A especificação do OID Fed define um procedimento para verificar se uma sequência de ESs forma uma TC criptograficamente válida desde o sujeito até uma TA, confiada pela entidade que executa a validação. É um passo essencial para que uma entidade firme confiança com outra na federação. O Algoritmo 1 ilustra um pseudoalgoritmo de execução dessa avaliação. Nele, considera-se que `ES[0]` é a autodeclaração do sujeito e `ES[i]` é a declaração emitida pela TA.

Figura 14 – Cadeia de Confiança do OID Fed.



Fonte: Adaptado de (Kawasaki, 2023).

Algoritmo 1 – Pseudoalgoritmo de avaliação da TC.

```
para cada j = 0,..,i:
    verifique a sintaxe de ES[j]
    verifique que o claim "iat" tem um valor no passado
    verifique que o claim "exp" tem um valor no futuro

verifique "iss" == "sub" para ES[0]
verifique a assinatura de ES[0] usando a chave pública de ES[0]["jwks"]

para cada j = 0,...,i-1:
    verifique ES[j]["iss"] == ES[j+1]["sub"]
    verifique a assinatura de ES[j] usando a chave pública de ES[j+1]["jwks"]

verifique se "iss" == "sub" para ES[i]
verifique a assinatura de ES[i] usando a chave pública de ES[i]["jwks"]
```

Fonte: Adaptado de (Hedberg et al., 2026).

3.4.2.4 Metadados (Metadata) e Política de Metadados (Metadata Policy)

O OIDC e o OAuth 2.0 já possuem o conceito de metadados de entidade, um conjunto de informações sobre características e capacidades da entidade expressos em formato *JavaScript Object Notation* (JSON). No OIDC, esses metadados são padronizados pela especificação *OpenID Connect Discovery*, que define os metadados do OP (Sakimura et al., 2023b), enquanto no OAuth 2.0 os metadados do servidor de autorização são definidos pela *OAuth 2.0 Authorization Server Metadata* (Jones; Sakimura; Bradley, 2018). Essa funcionalidade foi incorporada ao OID Fed (Connect2id, 2024). Os metadados de uma entidade são publicados na *claim* `metadata` de sua EC ou no SS emitido pela entidade imediatamente superior.

Esse campo é composto por entradas que dizem respeito ao tipo de entidade (*claim*

`entity_type`) em questão e suas características na federação, capacidades e como outras entidades devem interagir com ela. Os tipos de entidades podem ser `federation_entity` (IMs ou TAs), `openid_relying_party` (RPs), `openid_provider` (OPs), `oauth_authorization_server` (ASs), `oauth_client` e `oauth_resource` (RSs). No contexto de IDD, a especificação *OpenID Federation for Wallet Architectures 1.0* também define os tipos `openid_wallet_provider`, `openid_credential_issuer` e `openid_credential_verifier` (Marco et al., 2024a).

O Algoritmo 2 ilustra a *claim* `metadata` na EC emitida pelo $\texttt{RP}_1$ sobre si. O tipo de entidade especificado é `openid_relying_party`. Dos metadados definidos, alguns são herdados do OIDC, como `grant_types` e `redirect_uris` e outros são definidos pelo OID Fed, como `logo_uri` e `client_registration_types`.

Algoritmo 2 – Metadados autodeclarados publicados pela $\texttt{RP}_1$ em sua EC.

```
{
  "iss": "https://rp1.taa.edu",
  "sub": "https://rp1.taa.edu",
  ...
  "metadata": {
    "openid_relying_party": {
      "application_type": "web",
      "redirect_uris": [ "https://rp1.taa.edu/rp/callback" ],
      "organization_name": "RP1",
      "logo_uri": "https://rp1.taa.edu/images/32x32.png",
      "grant_types": ["authorization_code"],
    }
  },
  ...
}
```

Fonte: Adaptado de (Hedberg et al., 2026).

Diferente do anterior, o Algoritmo 3 ilustra a *claim* `metadata` em um SS, emitido pelo $\texttt{IM}_1$ sobre seu subordinado $\texttt{RP}_1$. Nesse caso, temos duas entidades afirmando metadados sobre $\texttt{RP}_1$: $\texttt{IM}_1$ e ela mesma. A entidade superior tem preferência durante o algoritmo de resolução da cadeia de confiança e, portanto, sobrescrevem-se os metadados da entidade subordinada. Em outras palavras, as características de uma entidade definidas por sua entidade imediatamente superior valem mais do que as características definidas pela própria entidade sobre si mesma.

TAs e IMs podem definir políticas que se aplicam aos metadados dos seus subordinados, podendo assim validar, adicionar, definir ou restringir um valor de metadado. Essas políticas são definidas na *claim* `metadata_policy` do SS. Um exemplo de política de metadados pode ser visto no Algoritmo 3.

Nesse contexto, uma federação pode utilizar essas imposições para garantir interoperabilidade entre PSs e PIs ou um perfil específico de segurança (Hedberg et al., 2026). Uma TA, por exemplo, pode definir quais algoritmos de assinatura digital são permitidos na federação (Jones; Bradley, 2024). A política de metadados possui alguns pilares no protocolo (Hed-

Algoritmo 3 – Metadados e políticas de metadados publicados pelo $\texttt{IM}_1$ no SS sobre o subordinado $\texttt{RP}_1$.

```
{
  "iss": "https://im1.taa.edu",
  "sub": "https://rp1.taa.edu",
  ...
  "metadata": {
    "openid_relying_party": {
      "organization_name": "Portal do Estudante",
      "logo_uri": "https://im1.taa.edu/oficial.png"
    }
  },
  "metadata_policy": {
    "openid_relying_party": {
      "application_type": { "value": "web" },
      "contacts": { "add": "im1.contact@taa.edu.br" }
    }
  }
  ...
}
```

Fonte: o autor.

berg et al., 2026):

- **Hierarquia**: Políticas fluem da entidade superior para a inferior na TC. Uma entidade subordinada só pode tornar uma política mais restritiva, nunca mais permissiva.
- **Contribuição Equitativa**: Qualquer entidade não folha (TAs ou IMs) pode adicionar políticas, desde que não crie conflitos. Um conflito invalida a cadeia.
- **Especificidade e Granularidade**: As políticas são aplicadas de forma isolada conforme o tipo de entidade e a metadados individuais, sem afetar outros *claims*.
- **Operação**: Uma política atua ativamente sobre um metadado, realizando uma verificação ou uma modificação.
- **Aplicação Integral**: A validação das políticas é uma etapa obrigatória na resolução de uma TC. Se os metadados finais não cumprirem as políticas, a cadeia é considerada inválida.
- **Determinismo**: O processo de aplicação de políticas é previsível. Dada a mesma TC, o resultado da validação será sempre o mesmo.

Uma política é definida por operadores de política, parametrizados com valores de metadados e podem ser combinados. A especificação define alguns operadores, listados abaixo, e permite que novos sejam criados.

- `value`: Atribui o parâmetro ao valor do metadado;
- `default`: define um valor padrão se o metadado estiver ausente;

- `one_of`: requer que o valor do metadado seja um de uma lista pré-definida;
- `subset_of`: requer que o valor seja um subconjunto de uma lista permitida;
- `superset_of`: requer que o valor seja um superconjunto de uma lista obrigatória;
- `add`: adiciona parâmetros ao metadado.
- `essential`: Requer a presença de um metadados;

Nesse contexto, seguindo o exemplo do Algoritmo 3, $\texttt{IM}_1$ está definindo uma política válida para todos os seus subordinados do tipo RP e afirma que todos devem ser aplicações *web* e devem adicionar o contato `im1.contact@taa.edu.br` nos seus metadados.

O Algoritmo 4 ilustra a política de metadados declarada pela `Trust Anchor A` no SS emitido para $\texttt{IM}_1$. `Trust Anchor A` não dita exatamente quais `grant_types` do OIDC usar, mas garante que nenhuma RP use opções fora da lista segura, como o fluxo *implicit*, que é considerado inseguro para muitas aplicações *web* modernas. Ademais, garante que publicar o metadado que diz respeito ao nome da organização (`organization_name`) é obrigatório para RPs, assim como ser uma aplicação *web* ou *mobile* (nativa).

Algoritmo 4 – Política de metadados publicados pela `TA` no SS sobre o subordinado $\texttt{IM}_1$.

```
{
  "iss": "https://taa.edu",
  "sub": "https://im1.taa.edu",
  ...
  "metadata_policy": {
    "openid_relying_party": {
      "organization_name": { "essential": true},
      "grant_types": {
        "subset_of": ["authorization_code", "refresh_token"]
      }
      "application_type": { "one_of": ["web", "native"] },
    }
  },
  ...
}
```

Fonte: Adaptado de (Hedberg et al., 2026).

As políticas são combinadas em toda a TC, da entidade de mais alto nível (TA) para a de mais baixo nível (LE). O resultado é um conjunto de Metadados Resolvidos (RM, *Resolved Metadata*), que reflete todas as políticas aplicadas. Ainda no exemplo, os Algoritmos 2, 3 e 4, nesta ordem, formam uma TC. Uma entidade que precisa estabelecer a confiança no $\texttt{RP}_1$ ($\texttt{OP}_1$, por exemplo), deve resolver essa cadeia de confiança.

3.4.2.5 Resolução da TC

Para validar uma TC, dois passos são necessários: avaliação da TC (detalhada no Algoritmo 1) e resolução dos metadados do sujeito da TC. Este último ainda pode ser separado em resolução da política de metadados e aplicação da política resolvida, passos que são explicados a seguir.

O processo de **resolução da política de metadados** deve começar pela SS mais alta na hierarquia e descer sequencialmente até o SS emitido pelo superior imediato ao sujeito da cadeia. Em cada passo da iteração, a *claim* `metadata_policy` do SS é validada. Isso garante que a estrutura de dados esteja correta, que apenas combinações de operadores permitidos sejam usadas e que não existam operadores críticos (`metadata_policy_crit`) desconhecidos. Uma falha na validação resulta em um erro de política. O processo consiste na mesclagem das políticas. Iniciando pela *claim* `metadata_policy` do SS mais acima, essa fusão ocorre em três níveis, do mais geral para o mais específico:

1. **Políticas para Tipos de Entidade**: Se um tipo de entidade já existe na política consolidada, suas regras são mescladas no nível inferior. Se não existe, ele é simplesmente adicionado.

2. **Políticas de Parâmetros de Metadados**: Da mesma forma, se uma política para um parâmetro específico (`application_type`, por exemplo) já existe, seus operadores são mesclados. Se não existe, a política do parâmetro é adicionada.

3. **Operadores**: Se um operador já existe para um parâmetro, seus valores são combinados conforme a especificação do operador. Se não existe, o operador e seu parâmetro são adicionados.

O resultado de todas as iterações de fusão é a política de metadados resolvida para a TC em questão. Usando como exemplo a cadeia de confiança dos exemplos anteriores, o primeiro passo para resolver a política de metadados é considerar a política emitida pela `TA` no Algoritmo 4 e mesclá-la com a política definida no próximo SS, do Algoritmo 3. A Figura 15 ilustra a mesclagem, mostrando o resultado da etapa.

Nota-se que as políticas (`organization_name` e `grant_types`) foram definidas apenas pela `TA` e, como o $\texttt{IM}_1$ não as modificou, elas permanecem inalteradas. O mesmo ocorreu com `contact`, sendo declarado apenas por $\texttt{IM}_1$. Já a política `application_type` foi definida nos dois SSs. Como o operador definido pelo $\texttt{IM}_1$ (`value`) é compatível com o operador definido pela TA (`one_of`) e mais restritivo, então a regra definida pelo subordinado prevalece. O resultado da mesclagem da Figura 15 deve ser utilizado para a próxima mesclagem com as políticas do SS subsequente na TC. Como não há mais SSs na TA em questão, então a política foi resolvida para essa TC.

O próximo passo é a **aplicação da política resolvida** considerando os metadados do sujeito da cadeia, obtidos de sua própria EC:

Figura 15 – Mesclagem da política de metadados entre dois SSs.



Fonte: o autor.

1. Primeiramente, aplica-se quaisquer metadados definidos diretamente pelo seu superior;
2. A política resolvida é então aplicada a esses metadados base. Para cada regra, os operadores definidos são executados em uma ordem específica para modificar ou validar os valores dos metadados.
3. Se a aplicação da política resultar em metadados inválidos ou incorretos, a resolução da cadeia de confiança deve retornar um erro.

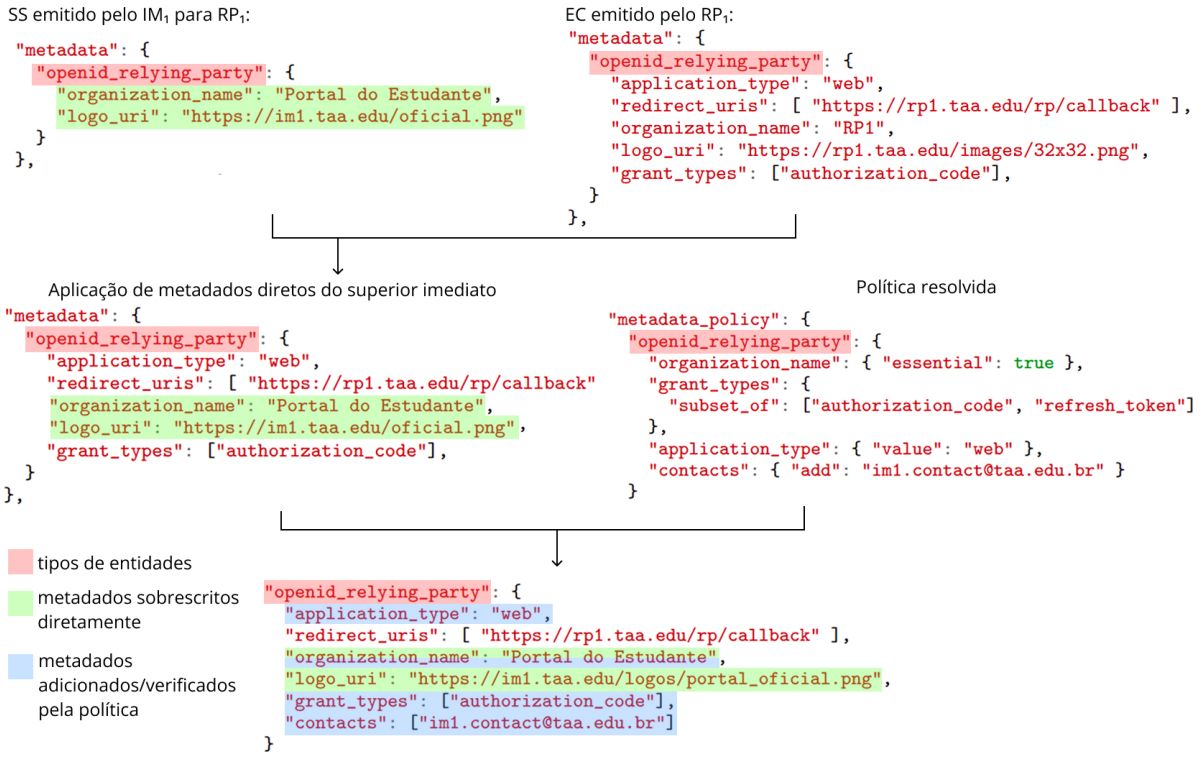
Ainda utilizando o exemplo, considerando a EC do Algoritmo 2, o primeiro passo é aplicar os valores de metadados definidos pelo `IM1` no SS do Algoritmo 3. Depois, aplicam-se as políticas resolvidas sobre os metadados resultantes da aplicação direta. Esse processo resulta no RM para o sujeito da TC. Isso é ilustrado pela Figura 16.

3.4.2.6 Marcas de Confiança (Trust Marks)

A Marca de Confiança (TM, *Trust Mark*), é uma declaração de conformidade com um conjunto de critérios ou requisitos, determinados por uma autoridade de acreditação. Assim como os demais artefatos da especificação, é representado como um JWT. Além das informações básicas como emissor, sujeito e validade (opcional para este artefato), possui as *claims*:

- `trust_mark_type`: identificador do TM, geralmente uma URL.
- `logo_uri`: Uma URL que referencia um logotipo para o TM (opcional).
- `ref`: Uma URL que aponta para informações legíveis por humanos sobre a emissão do TM (opcional).

Figura 16 – Obtenção dos metadados resolvidos.



Fonte: o autor.

- `delegation`: Um JWT de delegação, usado quando o TM é emitido em nome de um proprietário de TM diferente.

Os TMs podem ser incluídos em ECs, na *claim* `trust_marks`. Esse *claim* consiste em uma lista em que cada elemento representa um TM e contém o tipo do TM e sua representação em formato JWT. O Algoritmo 5 ilustra essa *claim* em uma EC emitida por uma RP do projeto italiano SPID/CIE.

O Algoritmo 6 ilustra o *payload* do TM. Além das *claims* definidas pelo protocolo, esse TM conta com *claims* definidas pelo próprio *framework* italiano específico, que descrevem metadados da RP em questão, como `tos_uri` e `service_documentation`.

Os TMs são assinados por uma autoridade de acreditação federada, chamada de emissor de TM, que deve ser representado por uma entidade na federação. Além disso, a emissão de TMs pode ser delegada. Um emissor de TM pode emití-lo em nome do proprietário do TM, que é a entidade que tem o direito sobre um tipo de TM.

Na federação, quem define a confiança nos emissores de TMs é a TA, que publica os emissores confiados para cada identificador de TM no *claim* `trust_mark_issuers` de sua EC. Se o operador sabe que o identificador possui um proprietário que é diferente do emissor, então esse conhecimento deve ser expresso no *claim* `trust_mark_owner` da mesma EC. O Algoritmo 7 ilustra esses *claims* em um EC emitida pela TA da federação. Nesse caso, o TM de identificador `https://refeds.org/wp-content/uploads/2016/01/Sirtfi-1.0.pdf` é

Algoritmo 5 – Representação de uma EC com o *claim* `trust_marks`.

```
{
  "iss": "https://rp.example.it/spid/",
  "sub": "https://rp.example.it/spid/",
  "iat": 1516239022,
  "exp": 1516298022,
  "trust_marks": [
    {
     "trust_mark_type": "https://www.spid.gov.it/certification/rp",
     "trust_mark":
     ↪ "eyJ0eXAiOiJ0cnVzdC1tYXJrK2p3dCIsImFsZyI6IlJTMjU2Iiwia2lkIjoia29zR20yd3VaaDlER2
10eEF0a3VPNDBwUGpwTDMtakNmMU4tcVBPLVllVSJ9.eyJpc3MiOiJodHRwczovL3d3dy5hZ2lkLmdvdi5pdCIsI
nN1YiI6Imh0dHBzOi8vcnAuZXhhbXBsZS5pdC9zcGlkIiwiaWF0IjoxNTc5NjIxMTYwLCJ0cnVzdF9tYXJrX3R5c
GUiOiJodHRwczovL3d3dy5zcGlkLmdvdi5pdC9jZXJ0aWZpY2F0aW9uL3JwIiwibG9nb191cmkiOiJodHRwczovL
3d3dy5hZ2lkLmdvdi5pdC90aGVtZXMvY3VzdG9tL2FnaWQvbG9nby5zdmciLCJyZWYiOiJodHRwczovL2RvY3Mua
XRhbGlhLml0L2RvY3Mvc3BpZC1jaWUtb2lkYy1kb2NzL2l0L3ZlcnNpb25lLWNvcnJlbnRlLyJ9.L_pSh1InEiFA
cs3E-1HBM7fNZYwF5ru3UGA_8yc80dGS3sszfA_sbj4AoW_zAJWQBdZpjxnHBBmybYXFrfZBcqxcedsrvUYrmbt1
nPYxbUE54fRRoZK-sJmVqh1GzSan5nOmkxuAtMinU8k_-aWnPWj83sYe2AzT2mMgkXiz8zhda3jZm8hoxZ4jR6B0
YAvbMlq2pPWO5OWKdZhiFRMSprwh0GYluQkK0j1aLNMGXD3keMJd2zEoWX9D7w2fXShAA48W3cNhuXyBVnCoum1K
4IWK3s_fx4nIkp6W-V4jCBOpxp7Yo8LZ3Oo_xpEOzGTIECGWVR86azOAlwVC8XSiAA"
    }
  ],
  ...
}
```

Fonte: Adaptado de (Hedberg et al., 2026).

Algoritmo 6 – *Payload* do TM.

```
{
  "trust_mark_type":"https://spid.federation.example.com/openid_relying_party/public/",
  "iss": "https://spid.tm-issuer.example.it",
  "sub": "https://rp.example.it/spid/",
  "iat": 1579621160,
  "organization_name": "Example of RP",
  "policy_uri": "https://rp.example.it/privacy_policy",
  "tos_uri": "https://rp.example.it/info_policy",
  "service_documentation": "https://rp.example.it/api/v1/get/services",
  "ref": "https://rp.example.it/documentation/manuale_operativo.pdf"
}

```

Fonte: Adaptado de (Hedberg et al., 2026).

de propriedade da entidade `https://refeds.org/sirtfi` e é emitido pela federação nacional `https://swamid.se`.

Para validar um TM, é necessário validar sua assinatura e verificar se ele não expirou. Se o identificador do TM estiver listado na *claim* `trust_mark_owners` da TA, é preciso verificar se o TM contém o *claim* `delegation`. Se for o caso, deve-se validar o JWT de delegação. Entidades também podem usar o *endpoint* de *status* de TM para verificar se um TM continua ativo, como será explicado na Seção 3.4.2.7.

De forma geral, os TMs permitem atestar conformidade com perfis de segurança específicos, como para gerenciamento de dados de usuários menores de idade ou perfis de segurança

Algoritmo 7 – *Claims* referentes aos emissores e proprietários de TMs na EC da TA da federação.

```
{
  "iss": "https://edugain.example.edu",
  "sub": "https://edugain.example.edu",
  "iat": 1516239022,
  "exp": 1516298022,
  "trust_mark_owners": {
    "https://refeds.org/wp-content/uploads/2016/01/Sirtfi-1.0.pdf": {
      "sub": "https://refeds.org/sirtfi",
      "jwks" : {
        "keys": [{
            "alg": "RS256",
            ...
        }]
      }
    }
  },
  "trust_mark_issuers": {
    "https://refeds.org/wp-content/uploads/2016/01/Sirtfi-1.0.pdf":
      ["https://swamid.se"]
  }
  ...
}
```

Fonte: Adaptado de (Hedberg et al., 2026).

nacionais. Eles adicionam uma camada suplementar de relações de confiança. Ademais, podem servir como evidência da confiabilidade de um provedor.

3.4.2.7 Endpoints das entidades

O OID Fed é baseado na troca de informações por meio de requisições a APIs via HTTP. Por isso, as entidades devem expor *endpoints* que possibilitem essa comunicação. A seguir, são descritos os principais *endpoints* definidos pela especificação, suas finalidades e quais entidades os publicam (Hedberg et al., 2026).

- **Endpoint de configuração da entidade** (`/.well-known/openid-federation`): Deve ser exposto por todas as entidades da federação. Ele retorna a EC da entidade.
- **Endpoint de SS emitidos** (`/federation_fetch_endpoint`): IMs e TAs devem disponibilizar esse *endpoint*, o qual, dado um subordinado (parâmetro `sub` da requisição), retorna o SS emitido pela entidade em relação ao seu subordinado. Caso o SS não tenha sido emitido, deve retornar um erro.
- **Endpoint de lista de subordinados** (`/federation_list_endpoint`): Publicado por TAs, IMs e Emissores de TMs. Retorna uma lista contendo os *entity id*s dos seus subordinados. Uma extensão a esse *endpoint* está sendo escrita (Marco et al., 2025). Ela cria novos parâmetros, permitindo uma maior filtragem dos subordinados e adiciona mais informações acerca dos mesmos na resposta.

- **Endpoint de resolução da cadeia de confiança (**`/federation_resolve_endpoint`**)**: Usado para obter os metadados resolvidos e os TMs para uma entidade específica. Este *endpoint* abstrai a complexidade de montar, verificar e aplicar políticas a uma TC, atuando como um resolvedor confiável. Qualquer entidade da federação pode publicá-lo. Para construir e resolver a TC, precisa dos parâmetros `sub` (informa o sujeito) e `trust_anchor` (TA a ser utilizada para a resolução).

- **Endpoint de status de TM (**`/federation_trust_mark_status_endpoint`**)**: Emissores de TMs devem expor esse *endpoint*, que indica se um TM específico emitido para uma entidade (informado via parâmetro `trust_mark` da requisição) é válido. Durante a construção do protocolo OID Fed, o autor do presente trabalho propôs que mais detalhes acerca da revogação do TM fossem incluídos na resposta deste *endpoint*[3]. Como resultado da discussão, os autores optaram por definir esses detalhes em uma extensão do protocolo.

- **Endpoint de Lista de Entidades com TMs (**`/federation_trust_mark_list_endpoint`**)**: O objetivo desse *endpoint* é consultar um Emissor de TMs sobre todos os destinatários de um tipo específico de TM que ele emitiu e que ainda são válidos. Deve ser exposto por emissores de TMs.

- **Endpoint de Trust Mark (**`/federation_trust_mark_endpoint`**)**: Exposto por um emissor de TMs para fornecer os próprios TM aos sujeitos. A requisição por um TM deve informar o tipo de TM desejado (por meio do parâmetro `trust_mark_type`) e quem é o sujeito (parâmetro `sub`). Esse último serve para que entidades possam requisitar TMs em nome de outras.

- **Endpoint de Chaves Históricas (**`/federation_historical_keys_endpoint`**)**: O propósito desse *endpoint* é publicar chaves de assinatura utilizadas anteriormente pela entidade. Isso garante a não-repudiação de declarações antigas mesmo após a rotação de chaves. Pode ser publicado por todas as entidades da federação.

3.4.2.8 Registro de clientes no PI

Pela especificação do OID Fed, existem duas formas de registro de RPs em OPs na federação. O metadado `client_registration_types_supported` é utilizado para expressar o suporte a cada uma delas.

O **registro automático** permite que uma RP faça requisições de autenticação sem uma etapa de registro prévia com o OP. Obrigatoriamente a RP informa seu identificador de entidade como valor do parâmetro `client_id` da requisição de autenticação. O OP utiliza esse valor para resolução da cadeia de confiança para o RP. Como o OP não atribui um `client_secret` para a

[3] https://github.com/openid/federation/issues/290.

RP, então a requisição de autenticação deve ser assinada pela chave privada do RP, para garantia de autenticidade. Ademais, a TC em relação à própria RP pode ser enviada na requisição para auxiliar o OP na descoberta e validação.

A Figura 17 ilustra simplificadamente o processo de registro automático de um cliente no OP da federação. O fluxo se inicia quando um usuário tenta acessar um recurso protegido no RP. Como o usuário não está autenticado, o RP cria uma requisição de autenticação OIDC. Crucialmente, o RP define seu próprio identificador de entidade na federação (uma URL) como o `client_id` da requisição. Toda a requisição é empacotada em um JWT e assinada com a chave privada do RP, sendo incluída no parâmetro `request`. O RP redireciona o navegador do usuário para o OP, passando os parâmetros necessários.

O OP recebe a requisição de autenticação. O valor de `client_id` não está nos seus clientes registrados, mas o OP percebe que se trata de uma URL, indicando um fluxo de federação. Logo, ele usa a URL contida em `client_id` para buscar a EC do RP, no *endpoint* `/.well-known/openid-federation`. A partir dessa declaração, o OP busca as declarações dos superiores hierárquicos, montando e validando a TC até encontrar uma TA que ele conheça. Com a confiança no RP estabelecida, o OP usa a chave pública do RP (obtida da cadeia de confiança) para validar a assinatura do objeto `request` original.

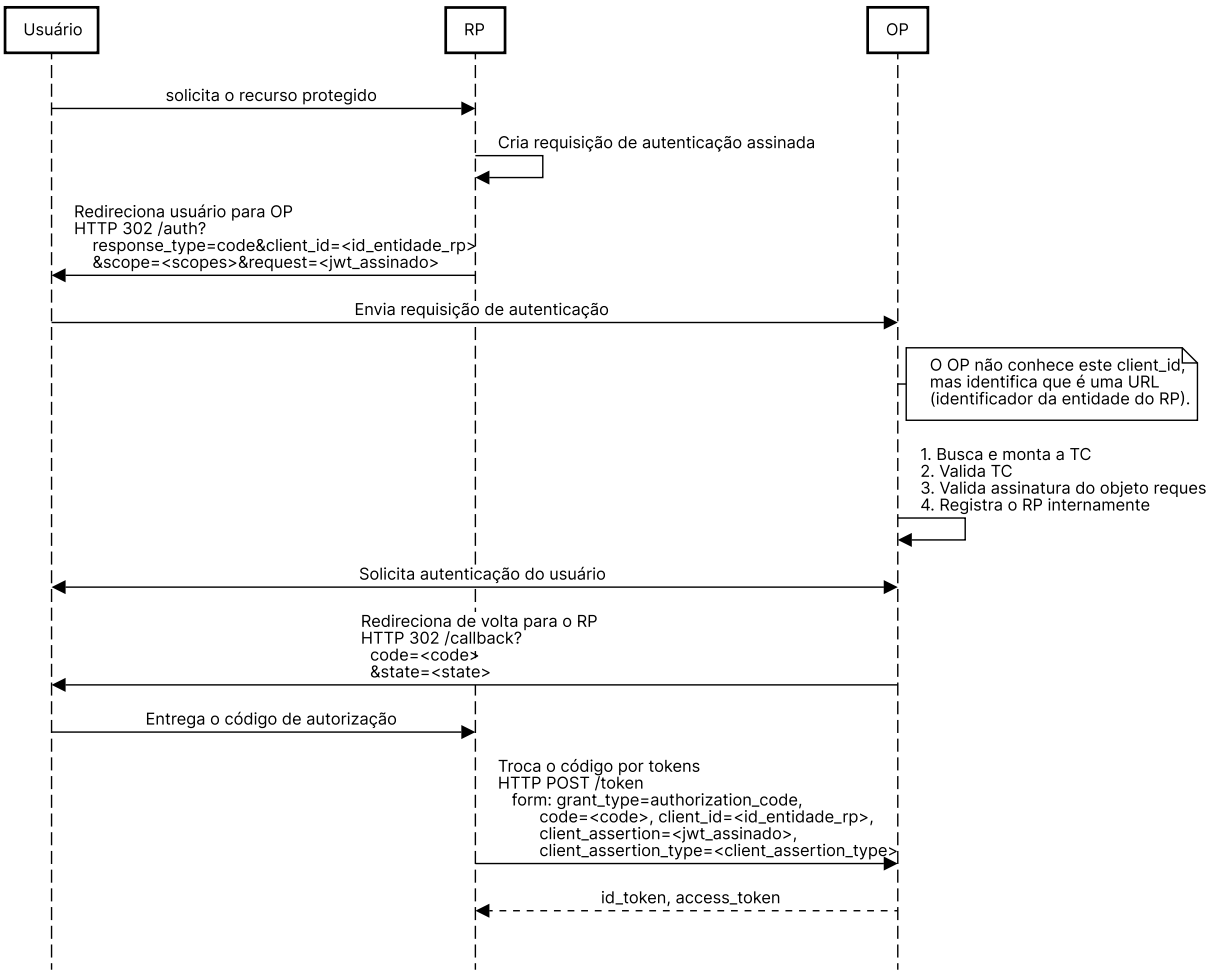
Se todas as validações forem bem-sucedidas, o OP cria um registro interno para o RP naquele momento. Com o RP registrado, o OP prossegue com a autenticação interativa do usuário final. Após a autenticação bem-sucedida, o OP redireciona o navegador do usuário de volta para o *endpoint* `redirect_uri` do RP, entregando um código de autorização de uso único. De posse do código, o RP pode obter *tokens* de acesso e de identificação com o OP diretamente, via *back-channel*. Para se autenticar nesta etapa, RP não usa um `client_secret`. Em vez disso, ele cria e assina uma `client_assertion`, um JWT provando sua identidade para aquela transação específica.

Diferentemente do registro automático, o **registro explícito** a RP estabelece seu registro de cliente com a OP através de uma requisição de registro dedicada, semelhante à especificação *OpenID Connect Dynamic Client Registration* (Sakimura; Bradley; Jones, 2023). Este é um processo de configuração deliberado e único, executado antes de qualquer interação do usuário. É ideal para RPs que precisam ou preferem operar com credenciais tradicionais, como `client_id` e `client_secret`. Após o registro, a RP pode fazer requisições de autenticação regulares. Nesse caso, o OP atribui um `client_id` à RP e o fornece na resposta de registro, opcionalmente enviando também um `client_secret`.

O OP que suporta registro explícito deve definir o metadado `federation_registration_endpoint` para a URL onde recebe as requisições desse tipo de registro. O RP usa essa URL para enviar a requisição de registro, passando como parâmetro sua EC ou uma TC da própria RP em relação a uma TA também confiável pelo OP. A Figura 18 ilustra o fluxo de registro explícito de uma RP.

Um administrador ou um processo automatizado no lado da RP aciona a funcionalidade de registro para se registrar em um OP específico. A RP obtém sua própria EC e a envia

Figura 17 – Fluxo de registro automático do cliente no OP.

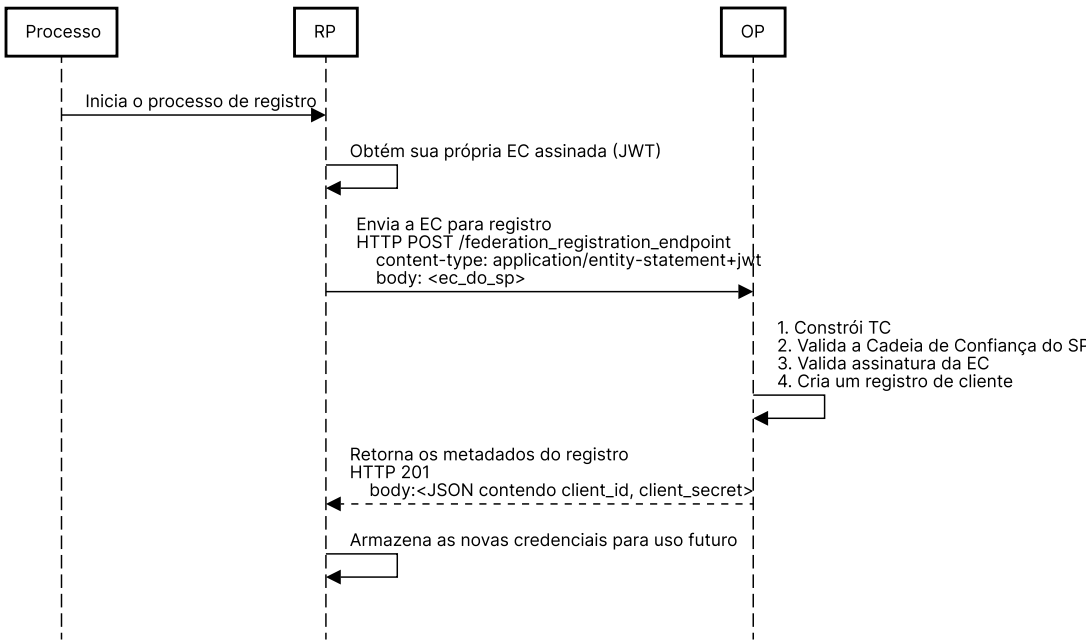


Fonte: o autor.

em uma requisição diretamente para o *endpoint* de registro explícito do OP. O corpo da requisição é a própria EC. Este ato constitui a apresentação formal e criptograficamente segura da identidade da RP ao OP.

Ao receber a requisição, o OP realiza uma série de validações críticas: (i) constrói TC do RP até uma TA reconhecida por ele; (ii) valida a TC; (iii) valida a assinatura da EC para garantir sua autenticidade e integridade. Se todas as validações forem aprovadas, o OP cria um registro de cliente permanente para o RP em sua base de dados, respondendo com um objeto JSON com os metadados do cliente registrado, incluindo um `client_id` tradicional e, opcionalmente, um `client_secret`. A RP recebe essas credenciais e as armazena de forma segura. A partir deste momento, o RP pode iniciar fluxos de autenticação OIDC padrão com o OP, utilizando o `client_id` e `client_secret`, recém adquiridos, como faria em um ambiente não federado.

Figura 18 – Fluxo de registro explícito do cliente no OP.



Fonte: o autor.

3.4.3 Comparativo entre SAML e OpenID Federation

Embora tanto o OID Fed quanto o SAML2/*Shibboleth* sejam protocolos para federação de identidade, eles apresentam abordagens distintas em termos de arquitetura, gerenciamento de confiança e flexibilidade. O SAML2, por exemplo, foi concebido em um cenário tecnológico que não antecipava os atuais requisitos de conectividade e alta disponibilidade para serviços *web* (Vaghetti, 2024a).

Uma das diferenças fundamentais reside no formato das mensagens. O SAML2 baseia-se em XML e em protocolos como o SOAP, cuja adoção no mercado é atualmente baixa. Em contrapartida, o OID Fed adota padrões modernos, priorizando o uso de APIs e estruturas de dados em JSON, que são mais leves, simples e alinhadas às práticas de desenvolvimento contemporâneas.

O modelo de estabelecimento de confiança também é um ponto central de divergência. No SAML2, a confiança é gerenciada de forma centralizada. Operadores de federação (como o eduGAIN) consultam os metadados de cada participante, os consolidam em um único e extenso arquivo, e o distribuem periodicamente a todos os membros (Connect2id, 2024). A confiança é estabelecida apenas se uma entidade constar nesse arquivo de metadados (Jones, 2020a). Este modelo é considerado frágil e pouco escalável. Em contraste, o OID Fed implementa uma TC hierárquica. A confiança é estabelecida dinamicamente ao verificar uma sequência de ESs que conecta uma LE a uma TA comum (Hedberg et al., 2026). Nesse sistema, o operador da federação precisa registrar apenas as organizações diretamente subordinadas, resultando em um modelo escalável e de fácil manutenção.

A abordagem para validação de metadados segue a mesma lógica. No ecossistema

SAML2/*Shibboleth*, o operador da federação deve aceitar o conjunto completo de metadados de uma entidade para que ela seja integrada à federação (Jones, 2020b). Já o OID Fed oferece um controle mais granular, permitindo que as entidades publiquem metadados tipados sobre si mesmas. As entidades não-folha (TAs e IMs) podem não apenas afirmar parâmetros sobre seus subordinados, mas também definir Políticas de Metadados para garantir a interoperabilidade e a segurança. Um exemplo é exigir dos seus subordinados tipos específicos de concessão OAuth 2.0 ou algoritmos de assinatura de *tokens*. Essas políticas são aplicadas de cima para baixo ao longo da cadeia de confiança.

Essa diferença arquitetural impacta diretamente o registro de novas instituições e o reconhecimento entre elas. No SAML2, o processo é centralizado e pode ser demorado. Uma entidade só é considerada parte da federação após seus metadados serem incluídos no arquivo central (Jones, 2020b), mas a propagação desse arquivo para todos os participantes pode levar um tempo considerável. Consequentemente, para um PI confiar em um novo PS, ele precisa primeiro receber o arquivo de metadados atualizado da autoridade central. Em contraste, o OID Fed adota um modelo descentralizado com confiança transitiva. O registro de um PS junto a um PI ocorre de forma dinâmica: quando o PS inicia uma requisição de autenticação, o PI pode construir e validar a cadeia de confiança em tempo real, permitindo o registro imediato, seja ele automático ou explícito.

Adicionalmente, no SAML2, a participação de uma mesma entidade em múltiplas federações é incomum e, em alguns casos, desaconselhada, como na eduGAIN, cujo modelo prevê que cada entidade seja registrada em uma única federação de origem (*home federation*), a partir da qual seus metadados são exportados e republicados às demais federações participantes (UK federation, s.d.). No OID Fed, por outro lado, o *design* nativo do protocolo não impõe desvantagens para a participação em múltiplas federações.

Em termos de evolução tecnológica, o SAML2 não possui especificações nativas para os desafios de segurança de aplicações *mobile* nativas e Aplicações de Página Única (SPAs, *Single Page Application*s). Aplicações-cliente executadas nesses ambientes são inerentemente mais expostas, pois não podem armazenar segredos de forma segura. O OID Fed, por ser construído sobre o OAuth 2.0, herda soluções robustas para esses cenários, como o padrão PKCE (Sakimura; Bradley; Agarwal, 2015) e a especificação para aplicações nativas (Denniss; Bradley, 2017). Além disso, uma especificação baseada em OIDC está em desenvolvimento para padronizar o SSO entre múltiplas aplicações nativas em um mesmo dispositivo (Fletcher, 2025).

Outra característica notável do OID Fed é sua capacidade de criar um modelo de confiança escalável, onde os participantes podem descobrir e verificar a autenticidade uns dos outros dinamicamente. Essa arquitetura torna o OID Fed uma ferramenta fundamental para estabelecer confiança em ecossistemas de IDD. Especificações como *OpenID Federation Wallet Architectures 1.0* (Marco et al., 2024a), *OpenID for Verifiable Credential Issuance* (Lodderstedt; Yasuda; Looker, 2024) e *OpenID for Verifiable Presentations* (Terbu et al., 2024) definem os protocolos para que os usuários obtenham e gerenciem credenciais digitais em suas carteiras (*wallets*),

apresentando-as a terceiros com total controle e consentimento.

A Tabela 2 resume as principais diferenças apresentadas nesta Seção entre os protocolos SAML2 e OID Fed.

Tabela 2 – Comparação entre federação SAML e OpenID Federation.

Característica	**SAML/Shibboleth**	**OpenID Federation**
Suporte para aplicações mobile	Limitado	Otimizado
Dinamicidade	Baixa	Alta
Interoperabilidade	Limitada	Alta
Integração	Complexa, manual	Automatizada, dinâmica
Escalabilidade	Limitada	Alta
Modelo de Confiança	Bilateral, centralizado	Descentralizado, Âncoras
Registro de instituições	Centralizado	Descentralizado
Registro de PSs em PIs	Demorado	Rápido
Compatibilidade com IDD	Não compatível	Compatível
Formato de Mensagens	XML/SOAP	APIs RESTful/JSON
Adaptação a Novas Tecnologias	Lenta	Rápida

Fonte: o autor.

3.5 PROXY DE IDENTIDADE

Em sistemas computacionais, um *proxy* é um intermediário que se posiciona entre duas partes e age em nome de uma delas perante a outra, recebendo, mediando e repassando as requisições trocadas (Shapiro, 1986). Aplicado ao domínio da identidade, um *proxy* de identidade, também referido como *proxy* de autenticação, é a entidade que se interpõe no fluxo de autenticação entre o PS e o PI (REFEDS, 2019; AARC, 2019). Para o PS, o *proxy* apresenta-se como um PI e para o PI de origem, apresenta-se como um PS (IdentityPython, 2019).

No contexto de uma FID, o *proxy* de identidade interpõe-se entre os PIs e os PSs da federação, e sua orientação depende de qual lado se deseja abstrair. Na configuração mais comum, descrita pela *Authentication and Authorisation for Research and Collaboration* (AARC) e pela REFEDS como *SP-IdP-Proxy*, o *proxy* dirige-se à federação como um PS, consumindo asserções dos PIs, e volta-se aos PSs *upstream* como um PI, emitindo asserções para os mesmos. Dessa forma, abstrai a federação para os serviços que atende (REFEDS, 2019).

O mesmo *proxy* pode, no entanto, ser orientado no sentido inverso e apresentar-se como PI perante a federação enquanto atua como PS diante de um PI *upstream*. Essa configuração permite integrar à federação um provedor de identidade que, de outro modo, não participaria dela, como um PI legado ou que utilize um protocolo distinto.

A motivação para a adoção desse padrão decorre da heterogeneidade do espaço federado. Uma instituição ou colaboração de pesquisa que precise integrar dezenas de serviços a uma interfederação de larga escala, como a eduGAIN, enfrentaria o custo de conectar e manter cada serviço individualmente a um ambiente diverso em protocolos, esquemas de atributos e

níveis de garantia. Ao posicionar um *proxy* entre a federação e os serviços, estes são protegidos dessa heterogeneidade. Eles passam a confiar em uma única entidade, e o *proxy* assume a complexidade de comunicação com o ecossistema federado externo (AARC, 2019). A REFEDS denominou essa evolução do modelo federativo de *Federation 2.0*, justamente por deslocar a integração de uma relação direta entre PIs e PSs para uma mediada por *proxies* (REFEDS, 2019).

3.5.1 Arquitetura de referência da AARC

O padrão *SP-IdP-Proxy* foi consolidado e formalizado pela AARC, iniciativa europeia que produziu uma Arquitetura de Referência (*Blueprint Architecture*, AARC-BPA) para a construção de Infraestruturas de Autenticação e Autorização (AAIs, *Authentication and Authorisation Infrastructure*s) de colaborações de pesquisa (AARC, 2017a; AARC, 2019). A AARC-BPA organiza uma AAI em camadas: identidades dos usuários, tradução de protocolos, serviços de atributos comunitários, autorização e serviços finais. Para isso, posiciona o *proxy* como o componente central que articula essas camadas, geralmente referido como *Community AAI*.

Na arquitetura, o *proxy* conecta-se à interfederação representando um único PS, escondendo os serviços da organização. Dessa forma, os PIs da federação enxergam apenas uma entidade e os serviços não precisam lidar diretamente com a federação. Essa centralização traz ganhos operacionais expressivos, como a redução do número de integrações mantidas, uniformização do tratamento de identidades e criação de um ponto central para a aplicação de políticas transversais. A AARC-BPA tornou-se a base para as AAIs de pesquisa europeias e influencia diretamente a interfederação eduGAIN (AARC, 2019).

3.5.2 Funções de um proxy de identidade

As responsabilidades que um *proxy* de identidade pode assumir, segundo as diretrizes da AARC e da REFEDS, incluem:

- **Tradução de protocolos**: o *proxy* pode receber requisições em um protocolo e repassá-las em outro, traduzindo, por exemplo, entre SAML2 e OIDC/OAuth 2.0. A tradução exige mapear corretamente as informações do *token* de origem para o de destino (REFEDS, 2019);
- **Agregação e normalização de atributos**: o *proxy* pode enriquecer a identidade autenticada com atributos provenientes de fontes externas e normalizar valores e esquemas heterogêneos antes de repassá-los aos serviços. As diretrizes da AARC sobre agregação de atributos detalham como agregadores devem reescrever escopos e gerir a validade dos dados agregados (AARC, 2017b);

- **Identificadores persistentes e níveis de garantia**: o *proxy* pode introduzir identificadores únicos, persistentes e não reatribuíveis para os usuários e anexar informação sobre o nível de garantia da autenticação, elementos essenciais para colaborações de pesquisa e nem sempre fornecidos pelos PIs de origem (AARC, 2019);

- **Centralização de políticas**: por ser um ponto central de passagem, o *proxy* é um local adequado para impor políticas transversais, como exigência de múltiplo fator de autenticação, autorização baseada em pertencimento às comunidades e vinculação de contas.

3.5.3 Implementações e exemplos de uso

A implementação de referência de código aberto é o SATOSA[4], mantido pela comunidade *IdentityPython*[5]. Trata-se de um *proxy* configurável que traduz entre SAML2 2.0, OIDC e OAuth 2.0. Sua arquitetura é modular, extensível e organizada em *frontends* (módulos que recebem requisições dos serviços), *backends* (módulos que se conectam aos PIs) e microsserviços intermediários para processamento de atributos (IdentityPython, 2019).

O SATOSA é a base de diversas AAIs de pesquisa. Um exemplo é o eduTEAMS[6], serviço da GÉANT que oferece uma solução pronta para que comunidades de pesquisa criem e operem suas próprias *Community AAI*s em conformidade com a AARC-BPA, permitindo o acesso a serviços por meio de identidades federadas da eduGAIN. Outro é o EGI Check-in[7], o *proxy* de identidade da infraestrutura de pesquisa europeia EGI, que funciona como um *hub* central conectando os PIs federados aos provedores de serviço da infraestrutura e combinando atributos de múltiplas fontes, igualmente baseado na AARC-BPA.

No contexto brasileiro, a própria RNP mantém o GIdLab, um laboratório de experimentação em GId que emprega *proxies* baseados no SATOSA para traduzir entre protocolos e integrar à federação provedores que, de outro modo, teriam limitações para ingressar nela diretamente (Dalmazo et al., 2024). Ademais, no ambiente de produção CAFe, o SATOSA é utilizado como *proxy* para permitir que PSs implementados em OIDC sejam integrados à federação.

Além dos supracitados, existem outros exemplos consolidados de *proxies* de identidade em operação. O serviço norte-americano *CILogon*, da federação *InCommon*, atua como *proxy* que recebe autenticações SAML2 das instituições federadas e emite *tokens* OIDC aos serviços conectados (CILogon, 2025). O *MyAccessID*, operado pela f, media o acesso a infraestruturas de computação de alto desempenho conectando PIs da eduGAIN, e foi desenvolvido conforme determina a AARC-BPA (GÉANT, 2025). A federação alemã de pesquisa adota o mesmo modelo em sua NFDI-AAI (IAM4NFDI, 2024).

[4] https://github.com/identitypython/satosa
[5] https://github.com/IdentityPython
[6] https://eduteams.org/
[7] https://docs.egi.eu/users/aai/check-in/

4 TRABALHOS CORRELATOS

A migração de SAML2 para OID Fed não é apenas uma atualização técnica para a CAFe, mas um passo estratégico para evitar o isolamento tecnológico. Iniciativas globais vêm sendo desenvolvidas no ínterim de adoção ao novo protocolo de federação (Dijk; Vaghetti, 2024; Marco, 2022; Morabito, 2023).

4.1 PILOTO OPENID FEDERATION DA EDUGAIN

O projeto diretamente análogo a uma potencial migração da CAFe é o piloto de OID Fed conduzido pela GÉANT, a rede pan-europeia de pesquisa e educação que opera a interfederação eduGAIN. Esta iniciativa representa o esforço da comunidade acadêmica global para modernizar sua infraestrutura de confiança, e seu estudo oferece um modelo valioso para a CAFe.

O objetivo central desse projeto é reter o valor fundamental da federação multilateral: permitir que uma entidade se conecte uma vez à sua federação nacional e ganhe acesso a serviços em todo o mundo, enquanto se adota uma base tecnológica moderna, flexível e mais segura (Dijk; Vaghetti, 2024). Para alcançar isso, o projeto visa desenvolver e validar um Perfil Tecnológico oficial da eduGAIN para o OID Fed. Este é um trabalho em andamento no momento da escrita deste texto e um grupo de trabalho foi criado para essa finalidade (Harris; Vaghetti, 2025). Este perfil servirá como um guia de implementação para cada Rede Nacional de Educação e Pesquisa (NREN, *National Research and Education Network*), garantindo a interoperabilidade dentro do novo ecossistema (Vaghetti, 2024b).

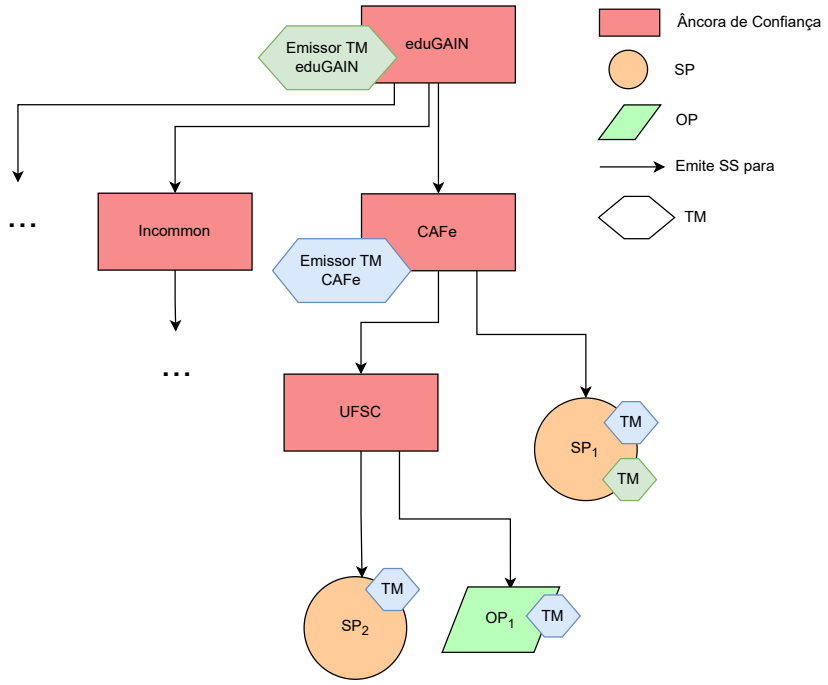
Os benefícios esperados incluem a simplificação da gestão da federação por meio de maior automação, o estabelecimento de confiança de ponta a ponta (do PS até o PI e vice-versa) de forma mais transparente e a criação de uma infraestrutura de confiança única que possa, eventualmente, suportar múltiplos protocolos, incluindo OIDC, SAML2 e outros (Dijk; Vaghetti, 2024).

A arquitetura definida para o piloto prevê a eduGAIN como uma TA, as NRENs como entidades intermediárias e os provedores como LEs. A resolução da TC de entidade é provida por um *endpoint* de resolução de TC, que será publicado pela TA da eduGAIN e que deve estar disponível também na TA da federação nacional.

Além disso, a arquitetura baseia-se na utilização de TMs para determinar pertencimento ou não de certa entidade à interfederação (Dijk; Vaghetti, 2024). Ou seja, a eduGAIN terá um emissor de TM associado e as entidades da federação devem solicitar sua participação na eduGAIN (*i.e.* a emissão do TM). A ideia é que o pedido seja recorrente e a validação da entidade e emissão do *token* feita automaticamente, com o emissor validando a entidade com base na presença do TM emitido pela NREN (Dijk, 2024). A Figura 19 ilustra a arquitetura proposta para a eduGAIN.

Na Figura 19, a TA da CAFe age como um IM em relação à interfederação, assim

Figura 19 – Arquitetura simplificada da eduGAIN proposta para o piloto.



Fonte: Adaptado de (Dijk; Vaghetti, 2024).

como as demais NRENs. A TA da UFSC, da mesma forma, age como IM no contexto da CAFe. Ainda, $\texttt{SP}_1$ possui tanto o TM que indica pertencimento à federação CAFe quanto à eduGAIN. Já o provedor $\texttt{SP}_2$, assim como o $\texttt{OP}_1$, possui apenas o TM que indica pertencimento à CAFe.

Apesar de ter como objetivo o suporte a múltiplos protocolos, isso ainda não está no escopo do projeto segundo o *workshop* apresentado por Dreef, Vaghetti e Kremers (2025), promovido pela própria eduGAIN. Além disso, o piloto também não trata da coexistência entre as federações atuais, baseadas em SAML/*Shibboleth* e as entidades novas que estão sendo criadas. O piloto teve seu início no segundo semestre de 2025, com reuniões quinzenais para acompanhamento dos esforços e engajamento das NRENs.

4.2 PROJETO SPID/CIE

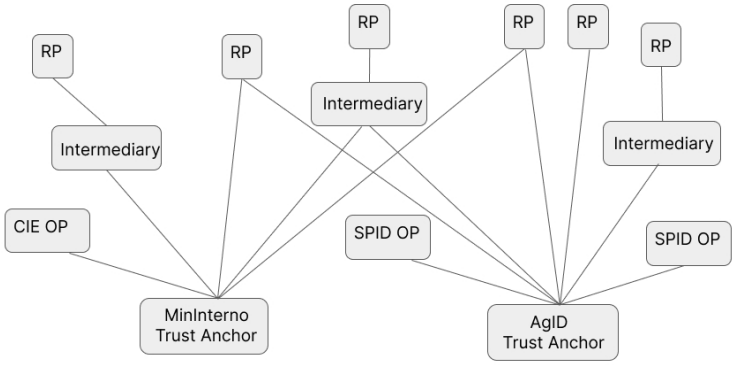
Enquanto o piloto da eduGAIN representa o esforço da comunidade acadêmica, o projeto do governo italiano para migrar seus sistemas nacionais de identidade digital, SPID e CIE, para OID Fed constitui a mais significativa implementação do padrão em ambiente de produção (Agenzia per l'Italia Digitale (AgID), 2024; Marco, 2022; Italia, 2024). Sua análise oferece lições valiosas sobre a viabilidade, a arquitetura e a estratégia de adoção do OID Fed em larga escala.

A principal força motriz por trás da iniciativa italiana foi uma decisão estratégica de modernizar a infraestrutura de ID do país. O objetivo era unificar os esquemas SPID e CIE, que operavam com bases tecnológicas distintas, sob um único protocolo moderno e alinhado com

os padrões globais da indústria (Marco, 2022).

Uma série de implementações de código aberto foram realizadas pelo Departamento de Transformação Digital e pela Agência para a Itália Digital (Agid)[1]. Foram desenvolvidos e disponibilizados *Software Development Kits* (SDKs) de código aberto para as linguagens de programação mais populares no mercado, incluindo `Python (Django)`, `Java`, `PHP`, `ASP.NET Core` e `Node.js`. Em se tratando de federação, a arquitetura é composta por duas TAs e pelos seus respectivos intermediários e provedores. A Figura 20 ilustra a arquitetura proposta.

Figura 20 – Arquitetura proposta no projeto SPID/CIE.



Fonte: (Italia, 2025).

A documentação fornecida ainda prevê que a utilização de TMs é obrigatória, estende a especificação original, definindo mais *claims* compondo a estrutura do TM e define um padrão para a identificação desses TMs como sendo `<TA domain>/<entity_type>/<tm_profile>/`, em que `<entity_type>` é o tipo de entidade, segundo a especificação do OID Fed e `<tm_profile>` define o perfil da entidade, podendo ser *public* para provedores do serviço público e *private* para provedores privados.

Outra contribuição do projeto italiano é a proposta de um IM que atua como *proxy* entre a federação e o PS. Ele abstrai as complexidades adicionadas ao OIDC pelo OID Fed. Esse IM centraliza a gestão de chaves criptográficas, metadados e TCs. A confiança é estabelecida entre ele e os demais OPs confiáveis da federação e os múltiplos serviços registrados nesse intermediário herdam essa confiança. Isso facilita a entrada de PSs que implementam OIDC na federação, uma vez que basta um registro prévio junto ao IM e passar a redirecionar os pedidos de autenticação para o mesmo, sem alterações relacionadas a recursos necessários para OID Fed.

[1] https://github.com/italia

4.3 DISSERTAÇÃO DO INSTITUTO POLITÉCNICO DE TURIM

A dissertação de Morabito (2023) tem como foco a integração dos protocolos OAuth 2.0 e OIDC na infraestrutura pan-europeia de identidade digital eIDAS2. O problema central abordado é a integração da rede eIDAS2 com os sistemas nacionais de identidade dos estados-membros. O *framework* eIDAS2, por regulamentação, exige o uso do protocolo SAML2 para a comunicação transfronteiriça. No entanto, vários países, como a Itália com seu sistema SPID/CIE, estão modernizando suas infraestruturas nacionais para adotar o OIDC e OID Fed. Isso cria uma lacuna de protocolo: como um serviço em um país A (usando eIDAS-SAML) pode autenticar um cidadão de um país B cujo PI nacional fala apenas OIDC?

A solução proposta se baseia na intervenção em um componente específico do nó eIDAS2 nacional, o *Specific Proxy Service*. Este componente é modificado para atuar como uma ponte de protocolo, um *proxy* de tradução. Dessa forma, ao receber uma requisição em formato SAML2, originada de um PS de um país membro, o *Proxy Service* do nó eIDAS2 intercepta a requisição. Neste ponto, ele assume o papel de uma RP para o protocolo OIDC. Ele traduz a requisição SAML2 recebida em uma requisição de autenticação OIDC, incluindo os atributos solicitados mapeados para *scopes* e *claims* OIDC. Esta nova requisição OIDC é enviada ao OP de origem do usuário. Após a autenticação bem-sucedida, o OP retorna um código de autorização para o *Proxy Service*. Este troca este código por um *access token* e um *ID token*, e então utiliza o *access token* para obter os atributos do usuário do *userinfo endpoint* do OP. Finalmente, o *Proxy Service* traduz os atributos do usuário (*claims* OIDC) de volta para o formato de uma SAML2 *Assertion* e a envia de volta pela rede eIDAS2 até o PI original.

A tese conclui que esta arquitetura é uma solução viável, performática e transparente. Ela permite a interoperabilidade entre SAML2 e OIDC, sem exigir que o *framework* eIDAS2 abandone o SAML2 ou que os provedores nacionais OIDC implementem SAML2. A complexidade da tradução de protocolo fica encapsulada em um único componente, tornando a migração tecnológica gradual e gerenciável. Esta abordagem representa a estratégia mais pragmática e de menor risco para uma federação grande e heterogênea como a CAFe, que enfrenta um desafio análogo de interoperabilidade entre um legado SAML2 e um futuro OIDC. Um modelo de ponte permitiria que o núcleo da federação fosse modernizado, enquanto uma camada de tradução garantiria a continuidade dos serviços para os participantes legados. Isso cria uma federação de múltiplos protocolos, onde novos serviços podem se integrar usando o moderno OID Fed, enquanto os serviços SAML2 existentes continuam a funcionar através do *proxy*. Com o tempo, os serviços podem migrar para fora do *proxy* à medida que os recursos permitirem.

4.4 ANÁLISE COMPARATIVA

A proposta apresentada neste trabalho não surge de forma isolada, mas sim como uma síntese das lições aprendidas e das arquiteturas validadas por iniciativas internacionais de grande relevância. Unem-se e adaptam-se as melhores práticas dos trabalhos correlatos para

construir um barramento de interoperabilidade que atenda às necessidades específicas do ecossistema da CAFe e do cenário brasileiro de pesquisa e ensino.

O piloto da eduGAIN, até o momento de escrita deste trabalho, não aborda a facilidade de adoção para provedores que não implementam nativamente o OID Fed. A presente proposta incorpora o conceito de uma entidade intermediária que atua como *proxy* completo, uma ideia central do projeto italiano SPID/CIE. Esse componente permite a redução do atrito e a facilidade de integração de novas instituições.

Ademais, o projeto em execução pela eduGAIN não trata sobre a coexistência com federações SAML2 legadas. Para preencher essa lacuna, propõe-se uma solução diretamente inspirada pela arquitetura de *proxy* detalhada na dissertação do Instituto Politécnico de Turim. Esse componente permite a tradução da comunicação entre provedores que implementam SAML2 e provedores que implementam OIDC, o exato cenário do barramento proposto em relação à federação em voga, construída em SAML2.

Em se tratando do projeto governamental italiano de carteiras digitais, apesar de ser o exemplo mais maduro de implementação do OID Fed, ele não reside no âmbito das federações de pesquisa e educação, que possuem requisitos e políticas específicos, como a necessidade de interoperabilidade com a eduGAIN. A definição de TMs específicos e de extensões da especificação formam uma barreira para a interoperabilidade, uma vez que implementações externas ao *framework* de confiança definido devem se adequar para adotar essas extensões.

Assim como o SPID/CIE não trata de especificidades acerca de federações de Pesquisa e Ensino (R&E, *Research and Education*), a dissertação do Instituto Politécnico de Turim, apesar de utilizar o OID Fed como base para os experimentos desenvolvidos, foi desenhada para a lacuna do protocolo no contexto do eIDAS2.

Por fim, não foi identificado nenhum projeto que trate da atualização tecnológica da infraestrutura de FID brasileira. É imperativo que a academia se posicione nesse cenário. O presente trabalho mostra-se pioneiro nesse contexto, propondo um barramento que preenche as lacunas dos projetos analisados. Além de atualização tecnológica, preocupa-se com facilidade de adoção, interoperabilidade com *frameworks* externos e suporte ao legado.

5 METODOLOGIA

O desenvolvimento do BAITA se deu por: (i) design da arquitetura, resultado de um processo de descoberta de problemas e pesquisa aplicada; e (ii) implementação do código-fonte, conduzida de forma colaborativa seguindo práticas consolidadas de engenharia de software. Este capítulo descreve (i) e (ii), explicando como as decisões arquiteturais foram tomadas e como o desenvolvimento foi organizado e executado no contexto do GT-BAITA.

5.1 FASE DE PROSPECÇÃO DO GT-BAITA

A fase 1 do Programa de Serviços Avançados da RNP conta com um período inicial de capacitação empreendedora, cujo objetivo é a identificação de dores reais dos *stakeholders* e a validação do valor da solução proposta.

A capacitação é estruturada com base na metodologia *Customer Discovery* de Steve Blank (Blank; Dorf, 2012), fundamento do movimento *Lean Startup* (Blank, 2013). Essa abordagem parte da premissa de que nenhum produto bem-sucedido nasce de suposições internas e deve-se *"sair do escritório"* e testar, junto aos *stakeholders* reais, as hipóteses formuladas sobre problemas e soluções, validando-as ou invalidando-as antes de investir em desenvolvimento.

Para execução prática dessa fase, utilizou-se o BMC (Osterwalder; Pigneur, 2010), ferramenta que representa o modelo de uma iniciativa em nove blocos: segmentos de clientes, proposta de valor, canais, relacionamento com clientes, fontes de receita, recursos-chave, atividades-chave, parcerias-chave e estrutura de custos. No contexto do GT-BAITA, cada bloco foi inicialmente preenchido com hipóteses sobre os problemas e necessidades dos *stakeholders* da CAFe. Essas hipóteses foram sistematicamente confirmadas ou refutadas com base nas evidências coletadas em campo, ajustando o BMC iterativamente ao longo das rodadas de entrevistas. A Figura 21 apresenta o BMC resultante do processo.

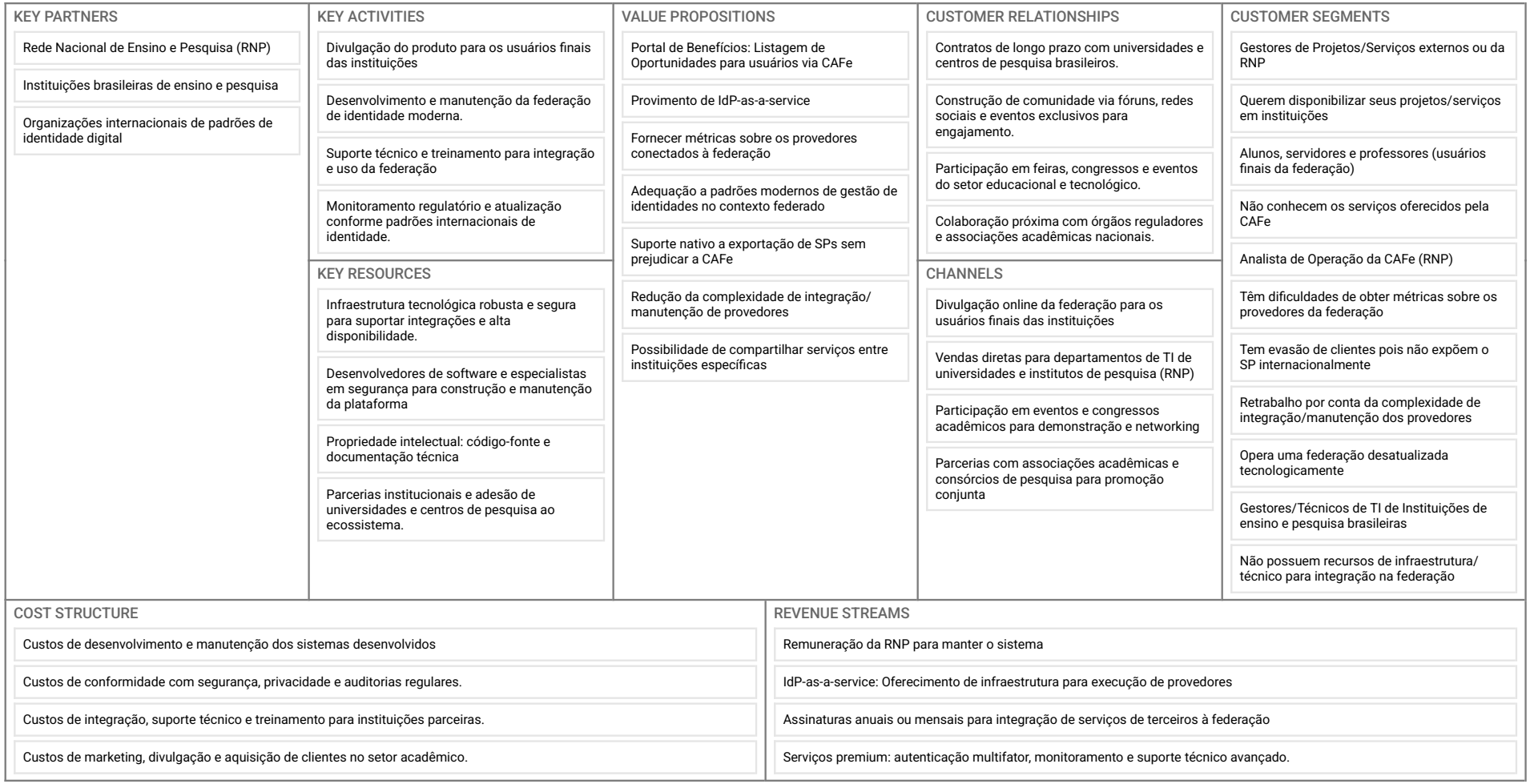
Para gerenciar esse processo, utilizou-se o GLIDR[1], plataforma digital especializada em *Customer Discovery*. O GLIDR permite organizar hipóteses por segmento de cliente, registrar e classificar as entrevistas realizadas, catalogar as evidências coletadas e acompanhar o progresso da validação ao longo do tempo. Cada entrevista era documentada na plataforma, indicando se a hipótese testada havia sido confirmada, refutada ou inconclusiva, o que permitiu ao grupo identificar padrões de dores de forma sistemática e auditável.

As entrevistas foram conduzidas entre os meses de julho e setembro de 2025. Ao longo desse período, o grupo realizou 61 entrevistas com *stakeholders* do ecossistema da CAFe (Perottoni et al., 2026). Os entrevistados foram segmentados em cinco grupos, cada um com dores distintas, de acordo com suas perspectivas em relação à federação. Um resumo acerca desses grupos e das dores identificadas é apresentado a seguir.

- **Operadores da (RNP)**: responsáveis pela operação e manutenção da CAFe. As entre-

[1] https://glidr.io

Figura 21 – BMC resultante do processo de prospecção do GT-BAITA.



Fonte: o autor.

vistas revelaram que a equipe operacional perde dezenas de horas semanais em suporte técnico manual às instituições, incluindo agendamento de reuniões individuais para depuração de configurações. Ferramentas paliativas como o *Moka*, um agregador de metadados SAML2 utilizado na CAFe (Dalmazo et al., 2024), e o *RPilot*, um executor de receitas automáticas para auxílio com a instalação do *Shibboleth* foram desenvolvidas internamente para reduzir esse volume de trabalho manual. A concepção dessas ferramentas evidencia que os problemas decorrentes do SAML2 são estruturais;

- **Gestores de Tecnologia da Informação (TI) das instituições**: responsáveis pela integração e manutenção dos provedores da instituição na CAFe. As entrevistas evidenciaram que as instituições, frequentemente, carecem de mão de obra técnica qualificada para operar o *software Shibboleth*, levando meses até que um novo provedor esteja integrado à federação. Muitas relataram a necessidade de manter dois PIs em paralelo: um para serviços internos e outro *Shibboleth* exclusivamente para a CAFe. O primeiro, muitas vezes baseado em OIDC.

- **Desenvolvedores de aplicações**: desenvolvedores que desejam federar seus serviços na CAFe ou que já o fizeram. Esses profissionais se deparam com dificuldades ao tentar integrar seus serviços à federação. A necessidade de um *proxy* de tradução de protocolo representa custo adicional de desenvolvimento que, em casos mais extremos, inviabiliza ou atrasa o lançamento de serviços federados;

- **Representantes de laboratórios de pesquisa**: grupos que desenvolvem aplicações de pesquisa e necessitam de acesso federado. Identificou-se que a invisibilidade dos serviços disponíveis na CAFe é um problema relevante: muitos recursos operados pela RNP ou desenvolvidos pelas universidades permanecem ociosos por falta de visibilidade e de um mecanismo de descoberta acessível;

- **Usuários finais**: estudantes, professores e pesquisadores que utilizam serviços federados. As entrevistas revelaram que a CAFe é percebida majoritariamente como uma barreira de autenticação (uma "tela de *login*"), e não como uma plataforma de colaboração digital. Os usuários desconhecem os serviços aos quais têm direito por meio da federação.

Consolidando as dores relatadas pelos cinco grupos, identificam-se os desafios centrais a seguir. Cada um deles é retomado individualmente no Capítulo 6, na seção que apresenta o componente proposto para abordá-lo.

- **Desafio 1: Alto atrito na integração de provedores**: as instituições carecem de mão de obra qualificada para operar o *Shibboleth* e os desenvolvedores arcam com o custo de manter *proxies* de tradução de protocolo, o que atrasa ou inviabiliza a integração de novos provedores.

- **Desafio 2: Ausência de interoperabilidade semântica e de MFA consistente**: os atributos de identidade não são entregues de forma padronizada pelos PIs e a adoção de MFA depende da atualização individual de cada instituição, comprometendo a interoperabilidade e o nível de garantia da autenticação.
- **Desafio 3: Necessidade de coexistência com a federação legada SAML2**: a CAFe opera centenas de provedores SAML2 que não podem ser descontinuados de imediato, exigindo que a nova infraestrutura coexista com a federação atual durante a transição tecnológica.
- **Desafio 4: Falta de controle de escopo dos provedores**: a federação não oferece um mecanismo nativo e escalável para restringir um provedor ao público da própria instituição ou de colaborações entre organizações específicas, levando muitos provedores a permanecer fora da CAFe.
- **Desafio 5: Operação manual e custosa da federação**: os operadores da RNP dedicam dezenas de horas semanais a suporte técnico manual às instituições e recorrem a ferramentas paliativas, evidenciando a necessidade de simplificar e automatizar a operação da federação.
- **Desafio 6: Invisibilidade dos serviços federados**: os serviços disponíveis permanecem ocultos aos usuários finais, e a federação é percebida como mera barreira de autenticação, o que exige um mecanismo de visibilidade e descoberta de serviços.

5.2 DESIGN DA ARQUITETURA

O *design* da arquitetura do BAITA foi um processo iterativo, que se desenvolveu em paralelo com a fase de prospecção. Esse processo contou com algumas etapas metodológicas. A primeira delas foi o estudo da especificação do OID Fed (Hedberg et al., 2026). Esse estudo incluiu a análise dos mecanismos de TC, políticas de metadados, TMs, *endpoints* e tipos de registro de clientes, assim como explicado em 3.4.2. Identificou-se que as características nativas do protocolo, como a topologia hierárquica e a resolução dinâmica de confiança eram adequadas para tratar as dores identificadas.

A segunda etapa foi a pesquisa sobre federações internacionais que já adotam o OID Fed, como o piloto da eduGAIN (Dijk; Vaghetti, 2024), que forneceu o modelo de referência para a hierarquia de entidades e o mecanismo de TMs para interfederação e o projeto italiano SPID/CIE (Agenzia per l'Italia Digitale (AgID), 2024), que inspirou a concepção de um *proxy* de identidade como entidade facilitadora da integração de provedores.

A terceira etapa consistiu na consolidação da arquitetura, cujas decisões tomadas nas etapas anteriores foram formalizadas em artefatos visuais. Foram produzidos diagramas de arquitetura, que representam a estrutura hierárquica da federação, o posicionamento de cada componente do barramento na topologia do OID Fed e as relações entre eles e diagramas de

fluxo de mensagens, que descrevem as trocas de mensagens entre os componentes para cada cenário de integração e nível de compatibilidade com a federação legada. O conjunto desses artefatos será utilizado nas seções seguintes para explicar a arquitetura.

5.3 DESENVOLVIMENTO DO CÓDIGO-FONTE

A implementação do BAITA é uma atividade colaborativa conduzida pelo GT-BAITA. O desenvolvimento do código-fonte é organizado em um repositório *GitLab* hospedado na infraestrutura da própria RNP. O repositório agrupa os diferentes componentes do barramento e aplicam-se processos consolidados de revisão de código, *pipeline* de testes automatizados e rastreabilidade de mudanças por meio de *merge requests*.

A escolha das tecnologias de implementação seguiu os critérios de maturidade das soluções no contexto de ID e domínio do time de desenvolvimento. As ferramentas específicas empregadas em cada componente, bem como suas responsabilidades, são descritas em detalhe no Capítulo 7. A Tabela 3 apresenta as principais ferramentas utilizadas no BAITA e a finalidade de cada uma.

O fluxo de trabalho adotado seguiu o modelo *feature branch* (Driessen, 2010), no qual cada nova funcionalidade ou correção é desenvolvida em um ramo dedicado, submetida como *merge request* e revisada por ao menos outro membro do grupo antes de ser integrada ao ramo principal. Esse processo garante que modificações não revisadas não alcancem o código de produção.

Aplicam-se ao código-fonte testes de unidade, integração e de vulnerabilidades. Os primeiros verificam o comportamento isolado de funções, classes e módulos. No contexto do presente trabalho, os testes são desenvolvidos tanto para o *back-end*, utilizando-se pytest[2] quanto para o *front-end*, utilizando-se Jest[3]. Os testes de integração garantem a interação correta entre os componentes do barramento e entre o barramento e sistemas externos, como fluxo completo de autenticação passando pelo WAYF e pelo Proxy-OID-Fed. Já os testes de vulnerabilidade garantem que as bibliotecas e dependências utilizadas não contenham vulnerabilidades conhecidas, utilizando-se ferramentas como o trivy[4] para análise de imagens Docker e o bandit[5] para análise de código Python.

O repositório GitLab é configurado com um *pipeline* de Integração e Entrega Contínuas (CICD, *Continuous Integration and Continuous Delivery*), executado automaticamente a cada *push* ao repositório. O *pipeline* inclui as etapas de execução dos testes citados acima, análise estática do código e construção das imagens Docker.

[2] https://docs.pytest.org
[3] https://jestjs.io
[4] https://trivy.dev
[5] https://bandit.readthedocs.io

Tabela 3 – Principais tecnologias utilizadas no BAITA e suas finalidades.

Tecnologia	**Finalidade**
React + Shadcn/UI[1]	Construção das interfaces de usuário dos componentes propostos.
Flask[2]	*Framework web Python* para as APIs REST dos componentes de *backend*.
PostgreSQL[3]	Banco de dados relacional para armazenamento persistente de metadados, configurações das entidades OID Fed e dos provedores integrados ao barramento e dados de usuários.
Redis[4]	Cache em memória para diminuição de latência e melhoria do desempenho dos componentes, além de armazenamento temporário de dados como as TCs.
Keycloak[5]	Plataforma de gerenciamento de identidade e acesso, responsável pela autenticação dos usuários do Sistema de Gestão do barramento e dos usuários que passam pelo WAYF.
SATOSA[6]	Motor de mediação de protocolos de autenticação nos *proxies* implementados.
LightHouse[7]	Resolvedor de metadados OID Fed, responsável pela descoberta e cache de cadeias de confiança.
Prometheus[8]	Coleta e agregação de métricas de observabilidade dos componentes em execução.
Grafana[9]	Visualização de métricas, *logs* e *dashboards* de monitoramento.
Docker Compose[10]	Orquestração dos contêineres nos ambientes de desenvolvimento, teste e produção.

[1] https://react.dev e https://ui.shadcn.com
[2] https://flask.palletsprojects.com
[3] https://www.postgresql.org
[4] https://redis.io
[5] https://www.keycloak.org
[6] https://github.com/IdentityPython/SATOSA
[7] https://github.com/go-oidfed/lighthouse
[8] https://prometheus.io
[9] https://grafana.com
[10] https://docs.docker.com/compose

Fonte: o autor.

6 PROPOSTA

O presente trabalho propõe o BAITA, um barramento de interoperabilidade acadêmico que adota o OID Fed 1.0 (Hedberg et al., 2026) como protocolo base. Além de modernizar a infraestrutura de IDF, o BAITA é projetado para resolver lacunas da CAFe, como simplificar a integração de provedores, viabilizar a coexistência com o SAML2 durante a transição de protocolos de federação, centralizar a normalização de atributos e de MFA, e aumentar a visibilidade dos serviços federados. Este capítulo une as premissas que guiaram as decisões arquiteturais com a proposta para mitigar essas problemáticas, introduzindo os componentes da solução em alto nível.

6.1 PREMISSAS PARA O DESENVOLVIMENTO

A análise das 61 entrevistas explicadas na Seção 5.1 revelou uma gama de desafios relacionados à interoperabilidade acadêmica. Esses desafios foram convertidos em premissas de *design* e desenvolvimento, que designam a tradução dos desafios levantados na fase de prospecção detalhados na Seção 5.1 em requisitos que orientaram as decisões arquiteturais do BAITA.

- **Compatibilidade**: A topologia do BAITA deve ser compatível com a topologia do piloto da eduGAIN (Dijk; Vaghetti, 2024), a fim de manter a interoperabilidade de operação com a interfederação em um ambiente de produção real;
- **Redução de atrito**: A integração de novos provedores à federação deve ser facilitada, reduzindo o atrito técnico e operacional para as instituições e gestores de serviços;
- **Interoperabilidade**: Os provedores devem comunicar-se de forma interoperável, com garantia de que suas mensagens e os atributos da ID do usuário serão entendidos pelos demais provedores;
- **Suporte ao legado**: A arquitetura deve suportar a comunicação entre provedores do BAITA e da federação CAFe, baseada no SAML2;
- **Fácil operação**: O operador da federação deve executar os procedimentos de manutenção do barramento de forma facilitada;
- **Gestão de subfederações**: O barramento deve suportar provedores públicos e privados das instituições, organizados em subfederações, controlando o acesso e a comunicação de cada provedor com o restante da federação;
- **Experiência do usuário**: A experiência do usuário ao utilizar o BAITA deve ser melhorada, com interações significativas e acesso facilitado aos serviços federados.

Baseado nessas premissas, apresentam-se a seguir os componentes propostos para o BAITA, cada um deles projetado para atender uma ou mais premissas.

6.2 *PROXY*-OID-FED

Participar nativamente de uma federação OID Fed exige que um provedor implemente funcionalidades específicas do protocolo (*i.e.*, publicação de ECs, gestão de chaves de federação e resolução de TCs), extensões que vão além do OIDC. Para instituições que já operam soluções baseadas em OIDC, esse requisito representa uma barreira técnica significativa. A fase de prospecção confirmou que muitas equipes de TI das instituições carecem de mão de obra qualificada para absorver essa complexidade. Esse atrito técnico constitui o Desafio 1.

Baseado na estratégia adotada pela federação italiana SPID/CIE (Agenzia per l'Italia Digitale (AgID), 2024), propõe-se o Proxy-OID-Fed, um *proxy* de identidade que abstrai as complexidades do OID Fed para os provedores sob sua tutela, permitindo que qualquer provedor OIDC ingresse na federação por meio de um simples registro prévio. A proposta é que esse componente seja hospedado na infraestrutura do operador da federação, eliminando a dependência de recursos técnicos das instituições. A arquitetura interna do componente e os detalhes de seu funcionamento são descritos na Seção 7.4.

A adoção de um *proxy* que abstrai a federação para os provedores que não implementam protocolos de confiança é um padrão arquitetural consolidado nas infraestruturas de pesquisa e educação, formalizado pela Arquitetura de Referência (*Blueprint Architecture*) da AARC. Tal arquitetura recomenda a delegação da responsabilidade de lidar com Provedores de Identidade e fontes de atributos heterogêneos, de modo a poupar os serviços finais desse esforço (AARC, 2017a). Esse padrão é conhecido como SP-IdP-*Proxy* e concentra no operador do *proxy* as tarefas de enriquecimento de atributos e tradução de protocolos em nome de todas as aplicações servidas (REFEDS, 2019).

Um exemplo de implementação desse padrão é o serviço norte-americano *CILogon*[1], da federação *InCommon*, atua como *proxy* que se comunica em SAML2 com as instituições federadas e emite *tokens* OIDC aos serviços conectados (CILogon, 2025). Outro exemplo é o EuroHPC[2], cujo acesso é mediado pelo *MyAccessID*, um *proxy* operado pela GÉANT que conecta PIs da eduGAIN, PIs de Computação de Alto Desempenho (HPC, *High Performance Computing*) e IDs do eIDAS2, desenvolvido em conformidade com os princípios da AARC (GÉANT, 2025). O Proxy-OID-Fed proposto neste trabalho instancia esse padrão, especializando-o para o protocolo OID Fed.

6.3 WAYF

Na CAFe atual, dois problemas estruturais limitam a qualidade da autenticação federada. O primeiro é a ausência de MFA consistente. Como cada instituição é responsável por manter seu próprio PI, a atualização de todos os PIs para suporte total dessa funcionalidade é lenta, fato corroborado pelas entrevistas explicadas na Seção 5.1. O segundo é a falta de intero-

[1] https://www.cilogon.org
[2] https://eurohpc-ju.europa.eu

perabilidade semântica dos atributos de identidade. Muitas instituições não seguem os padrões estabelecidos, obrigando cada PS a consumir individualmente uma API de mapeamento mantida pela RNP[3], o que gera atrito significativo. Em conjunto, esses dois problemas compõem o Desafio 2.

Para mitigar esses problemas, o BAITA propõe um WAYF estendido que, além de sua função convencional de listar os PIs disponíveis para o usuário, atua como *proxy* de identidade com três responsabilidades adicionais: (i) **normalização de atributos**, centralizando o mapeamento para os esquemas *eduPerson* e *brEduPerson*; (ii) **garantia de MFA**, verificando o nível de garantia retornado pelo PI e, se necessário, solicitando o segundo fator diretamente ao usuário; e (iii) **mapeamento semântico de métodos de autenticação**, traduzindo *frameworks* próprios de cada instituição para o padrão da federação. A proposta é que o WAYF seja executado na infraestrutura da RNP. Os detalhes de implementação de cada funcionalidade e a análise dos *trade-offs* arquiteturais são apresentados na Seção 7.5.

6.4 *PROXY*-SAML

A migração de uma federação do porte da CAFe, de SAML2 para OID Fed não pode ser realizada de forma abrupta. Centenas de provedores SAML2 em produção não podem ser descontinuados de imediato. A coexistência entre a nova infraestrutura baseada em OID Fed e a federação legada baseada em SAML2 é, portanto, um requisito fundamental durante o período de transição tecnológica, conforme o Desafio 3. O `Proxy-SAML` é o componente responsável por essa ponte de protocolo, interceptando e traduzindo a comunicação entre provedores SAML2 e provedores OID Fed/OIDC. Esse componente também é implementado sobre o SATOSA[4] e habilita a comunicação entre as duas federações durante o período de transição.

A estratégia de compatibilidade é definida em três níveis progressivos: no **Nível 1**, as duas federações operam em paralelo, sem interação; no **Nível 2**, PSs da nova federação delegam autenticação para PIs SAML2 via *proxy*; no **Nível 3**, PSs SAML2 passam a acessar PIs modernos via *proxy*. Os fluxos detalhados de cada nível são apresentados na Seção 7.6.

6.5 GESTÃO DE SUBFEDERAÇÕES

Atualmente, na CAFe e nas demais federações construídas com base no SAML2, a confiança é estabelecida de forma agregada, na qual o operador publica um único arquivo de metadados que atua como registro de confiança, de modo que toda entidade nele listada é tecnicamente confiável pelas demais, podendo validar assinaturas e comunicar-se mutuamente (RNP, 2020). Contudo, o protocolo não provê um mecanismo nativo e escalável para restringir o escopo de confiança ou a visibilidade de uma entidade a um subconjunto da federação. A restrição

[3] A documentação da API está disponível em https://hmg.api.mapeador.rnp.br/

[4] https://github.com/IdentityPython/SATOSA

de acesso efetiva recai sobre as próprias instituições. Para que um usuário acesse um serviço, basta que sua instituição tenha aderido e libere o acesso (RNP, 2024b).

Essa liberação é operacionalizada por políticas de liberação de atributos configuradas individualmente em cada PI e cada regra define quais PSs, sob quais condições, recebem quais atributos (ACOnet Identity Federation, 2023; Shibboleth Consortium, 2021), além de eventual filtragem de PIs no lado do PS. Tais abordagens são descentralizadas, manuais e onerosas de manter. Como consequência, instituições que mantêm serviços destinados apenas ao seu próprio público interno não os integram à CAFe, conservando-os em relações locais entre PI e PS, o que contribui para a baixa adesão de PSs à federação.

Para resolver essa limitação, que constitui o Desafio 4, a arquitetura proposta introduz uma distinção entre entidades públicas e privadas no contexto do barramento. Ao permitir que serviços de escopo restrito sejam federados sem se tornarem acessíveis a toda a CAFe, essa separação visa três objetivos: (i) ampliar a adoção do barramento, acomodando provedores que hoje permanecem fora da federação justamente por exigirem acesso restrito ao seu público interno; (ii) incentivar a atualização tecnológica dos membros da CAFe; e (iii) disseminar a interoperabilidade entre provedores.

A solução técnica para isso apoia-se no modelo de TCs do OID Fed. As entidades públicas são subordinadas indiretas da TA da CAFe. Como todos os provedores da federação confiam nessa TA, é sempre possível construir uma TC válida até ela, tornando-os confiáveis por toda a federação. As entidades privadas, por sua vez, não são alcançáveis a partir da TA da CAFe, sendo ancoradas em uma TA distinta, própria da instituição, de modo que somente os provedores que adotam essa TA como âncora de confiança, *i.e.*, aqueles no escopo da própria instituição, conseguem estabelecer uma TC válida com ela. Assim, a confiança com provedores privados fica restrita ao domínio institucional pela própria estrutura da federação, sem depender de configurações manuais de liberação de acesso em cada provedor.

À semelhança dos demais componentes, a implementação dessa separação é organizada em um plano de fases. Na primeira fase, apenas entidades públicas são aceitas, de forma análoga ao modelo atual da CAFe. Na segunda fase, provedores privados passam a ser suportados e cada instituição recebe uma TA Privada, superior à sua TA Pública, e os provedores sob ela são confiáveis apenas dentro do domínio da própria instituição. Já na terceira fase, grupos que necessitam de uma esfera de confiança própria, como um consórcio de laboratórios, recebem uma TA dedicada, aproveitando o suporte nativo do OID Fed à federações aninhadas com topologia arbitrária. Os detalhes arquiteturais de cada fase são apresentados na Seção 7.2.4

6.6 SISTEMA DE GESTÃO DO BAITA

A premissa de facilidade na operação da federação é favorecida pela adoção do OID Fed. Como a confiança é construída de forma hierárquica, os procedimentos de integração de novos provedores podem ser descentralizados. Dessa forma a carga do *onboarding* de entidades pode ser distribuída para IMs. Para integrar-se à federação, basta que uma entidade folha se

registre em uma IM subordinada à TA da federação, sem que nenhum procedimento precise ser executado pelo operador da CAFe. Adicionalmente, as entidades podem ter administradores diferentes de acordo com sua infraestrutura de execução.

Para tratar dessas questões e atender ao Desafio 5, propõe-se o Sistema de Gestão do Barramento de Interoperabilidade Acadêmico (SGB). Seu objetivo principal é fornecer aos gestores da federação uma interface amigável para os procedimentos de operação. O SGB é executado na infraestrutura da RNP e é utilizado tanto pelos operadores da RNP quanto pelos administradores dos domínios-membro da federação. Ele conta com um PI à parte, dedicado exclusivamente ao gerenciamento de identidade, permissões e grupos de gestores.

A administração pelo SGB é descentralizada por domínios. O gestor da TA da CAFe pode criar um domínio para cada instituição e delegá-lo a um administrador externo, concedendo-lhe permissões para criar e gerenciar entidades e subdomínios dentro dos seus limites. Quando essa entidade for executada na infraestrutura do SGB, deve ser possível editar suas configurações, enquanto entidades executadas fora dessa infraestrutura (*i.e.*, na infraestrutura da instituição), podem ser observadas na topologia da federação, mas não podem ter suas configurações alteradas pelo SGB. Os detalhes técnicos da arquitetura do sistema são apresentados no Capítulo 7.

6.7 PORTAL DE BENEFÍCIOS

Para resolver o Desafio 6, relativo à invisibilidade e ao desconhecimento dos serviços integrados à federação, propõe-se o Portal de Benefícios do Barramento de Interoperabilidade Acadêmico (PoB), uma aplicação executada na infraestrutura da RNP que cataloga automaticamente os PSs da federação e os apresenta ao usuário em uma interface de pesquisa e descoberta. O portal é ele próprio uma RP na federação, permitindo que o usuário se autentique para obter uma experiência personalizada. Os detalhes de implementação são apresentados na Seção 7.7.

O alcance do catálogo apresentado pelo PoB acompanha o escopo de confiança do usuário autenticado. Como o portal é uma RP da federação e pode identificar a instituição de origem do usuário, ele pode descobrir e listar não apenas os PSs públicos da CAFe, mas também os serviços pertencentes ao escopo da própria instituição. Esses serviços, organizados nas subfederações privadas descritas na Seção 6.5, são ancorados na TA privada da instituição e, portanto, não são visíveis à federação. Ao resolver as cadeias de confiança a partir dessa âncora institucional, o PoB expõe ao usuário os recursos de escopo restrito aos quais ele tem direito e que, de outro modo, permaneceriam ocultos. Assim, um estudante da `UFSC`, por exemplo, visualiza tanto os serviços abertos a toda a CAFe quanto aqueles destinados exclusivamente à sua instituição.

O escopo do catálogo pode ainda ser estendido para além do contexto nacional. O PoB pode utilizar a TA da eduGAIN como ponto de partida para a descoberta de provedores, apresentando ao usuário também os serviços da eduGAIN, isto é, recursos de outras NRENs que participam da interfederação. Dessa forma, o PoB unifica, em uma única interface de

descoberta, três escopos complementares de serviços: os públicos da CAFe, os privados da instituição do usuário e os internacionais da eduGAIN.

Espera-se que esta ferramenta melhore significativamente a experiência do usuário e impulsione a utilização dos recursos da federação. Ela foi pensada para ser a porta de entrada do usuário final para a federação.

7 ARQUITETURA

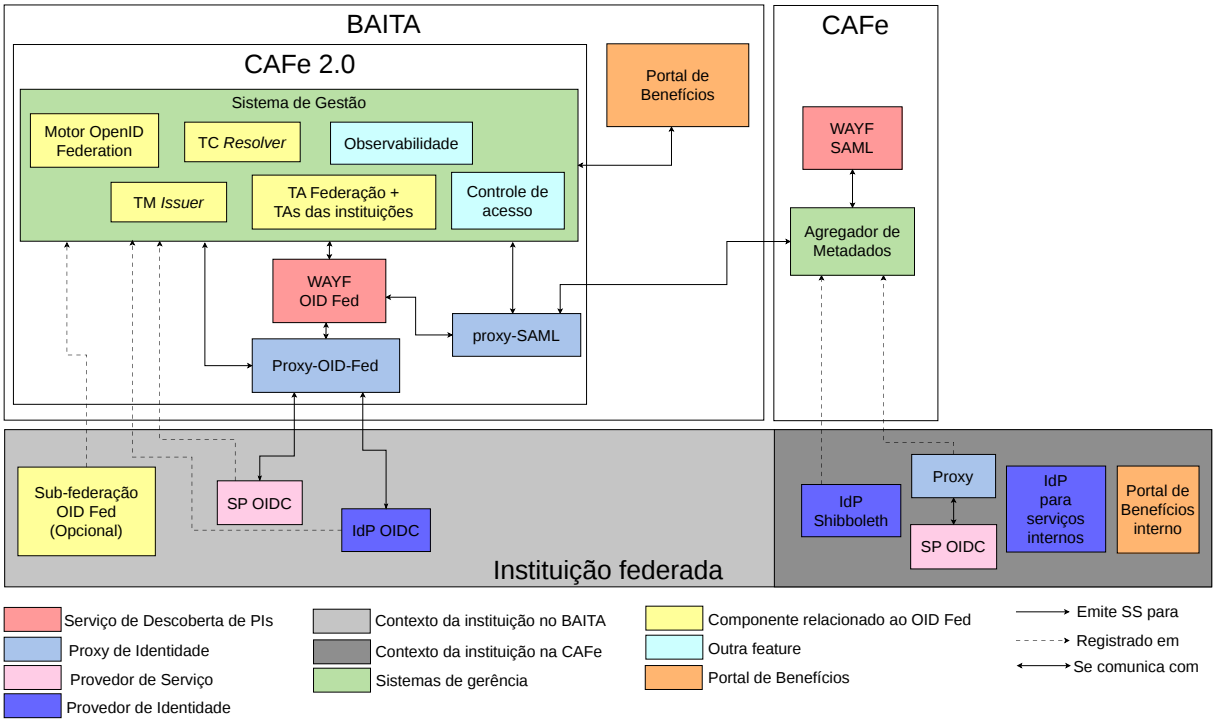
O presente capítulo detalha a arquitetura do BAITA, apresentando seus componentes seguindo uma abordagem *top-down*. Inicia-se com uma visão geral da solução e das relações entre seus sistemas e segue com a descrição individual de cada componente, cobrindo sua arquitetura interna, seus fluxos de operação e suas integrações.

7.1 VISÃO GERAL DA ARQUITETURA

O BAITA é composto por múltiplos componentes que operam em conjunto para atender as premissas explicadas na Seção 6.1. A Figura 22 apresenta esses componentes em alto nível, organizados nos três contextos envolvidos na proposta: (i) a infraestrutura da CAFe operada pela RNP e atualmente em produção; (ii) o BAITA, barramento de interoperabilidade, cujo objetivo é que seja mantido pelo operador da federação; e (iii) a instituição federada, que se integra à federação, gerencia e configura os provedores que participam da federação.

Nota-se que o BAITA não consiste apenas na implementação da federação OID Fed, mas sim em um ecossistema que provê outras facilidades, como o `Portal de Benefícios`. Dessa forma, a Figura 22 apresenta a `CAFe 2.0`, implementação da federação OID Fed, como uma parte componente do BAITA e o `Portal de Benefícios` como um componente a parte da `CAFe 2.0`.

Figura 22 – Arquitetura geral do BAITA.



Fonte: o autor.

Ainda, na Figura 22, o núcleo da `CAFe 2.0` é o SGB, que concentra as funcionalidades

do OID Fed e outras funcionalidades de gestão, como observabilidade e controle de acesso dos usuários. Integrados à `CAFe 2.0` estão ainda o `WAYF OID Fed` e os dois *proxies*. O `Proxy-OID-Fed` permite que provedores OIDC participem do BAITA sem implementar complexidades relacionadas ao protocolo e o Proxy-SAML viabiliza a comunicação entre a nova federação e a federação SAML2.

O contexto da CAFe, à direita na Figura 22, permanece inalterado durante a transição entre os protocolos, com o `WAYF SAML` e o `Agregador de Metadados` continuam operando e servindo os provedores que ainda não migraram para o OID Fed.

Na infraestrutura da `instituição federada`, a Figura 22 evidencia o contraste entre os dois modelos de confiança. Na CAFe SAML2, cada instituição precisa manter um PI *Shibboleth* para participar da federação e, caso possua serviços baseados em OIDC, é também responsável por operar seu próprio *proxy* de tradução entre SAML2 e OIDC. Além disso, a instituição geralmente mantém um segundo IdP usado para autenticação em serviços internos. Em alguns casos, ainda, a instituição desenvolve um `Portal de Benefícios interno` para divulgação dos mesmos para usuários finais, atualizado manualmente pela equipe de TI da instituição. Porém, grande parte das instituições não mantém um portal com essa funcionalidade e, consequentemente, não oferecem uma visão clara dos serviços que estão disponíveis para seus usuários, o que dificulta a adoção da federação pelos usuários finais.

Um dos objetivos do BAITA é agregar valor para as instituições federadas. Logo, a Figura 22 ilustra que, no contexto do BAITA, a instituição pode utilizar diretamente seus provedores OIDC na federação, registrando-os no SGB. Além disso, após fazê-lo, o serviço estará listado no `Portal de serviços` automaticamente. Opcionalmente, caso a instituição deseje manter a sua própria subfederação, tendo total controle, ela pode implementar o OID Fed e integrar sua sub-árvore ao BAITA, por meio da sua TA.

7.2 MAPEAMENTO DO OPENID FEDERATION E SEUS ARTEFATOS

Antes de detalhar cada componente, é necessário compreender como o modelo de confiança do OID Fed é mapeado para o contexto da CAFe. Esse mapeamento define o esqueleto da federação, sobre o qual os demais componentes operam. Essa Seção explica como se dá esse mapeamento.

7.2.1 Hierarquia da federação

A fim de manter a interoperabilidade com a interfederação, a hierarquia de entidades deve ser compatível com a arquitetura recomendada pelo piloto da eduGAIN (Dijk; Vaghetti, 2024). O modo de operação também deve ser compatível, para que os procedimentos definidos pela GÉANT sejam respeitados, consoante o Perfil Técnico da eduGAIN (Harris; Vaghetti, 2025), ainda em desenvolvimento no momento de escrita deste texto. Dessa forma, PSs da

CAFe podem se comunicar com PIs externos ao contexto brasileiro e vice-versa. O mapeamento das entidades do OID Fed para o contexto da CAFe é apresentado a seguir:

- **TA da CAFe**: raiz de confiança da federação, administrada pelo operador da RNP. Define as políticas de metadados aplicáveis a todos os provedores, publica os emissores de TMs confiáveis e expõe um *endpoint* de resolução de TC (Hedberg et al., 2026; Dijk; Vaghetti, 2024);
- **TA da Instituição**: para cada instituição membro, uma TA é criada como subordinada direta da TA da CAFe. Essa entidade representa o domínio institucional e garante a soberania da instituição sobre seus provedores. Caso, por algum motivo, a CAFe desative a instituição da federação (*i.e.*, pare de emitir SS para essa TA), a mesma ainda pode existir e ter a entidade TA como raiz de uma subfederação independente;
- **IMs**: entidades intermediárias que representam subdomínios administrativos da instituição, como laboratórios, grupos de pesquisa ou departamentos. Emitem SSs para os provedores sob sua responsabilidade e podem ser usados para organizar logicamente a hierarquia de administração;
- **LEs**: os provedores em si, que participam ativamente dos fluxos de autenticação definidos pelo OIDC (Sakimura et al., 2023a) e pelo OAuth 2.0 (Hardt, 2012).

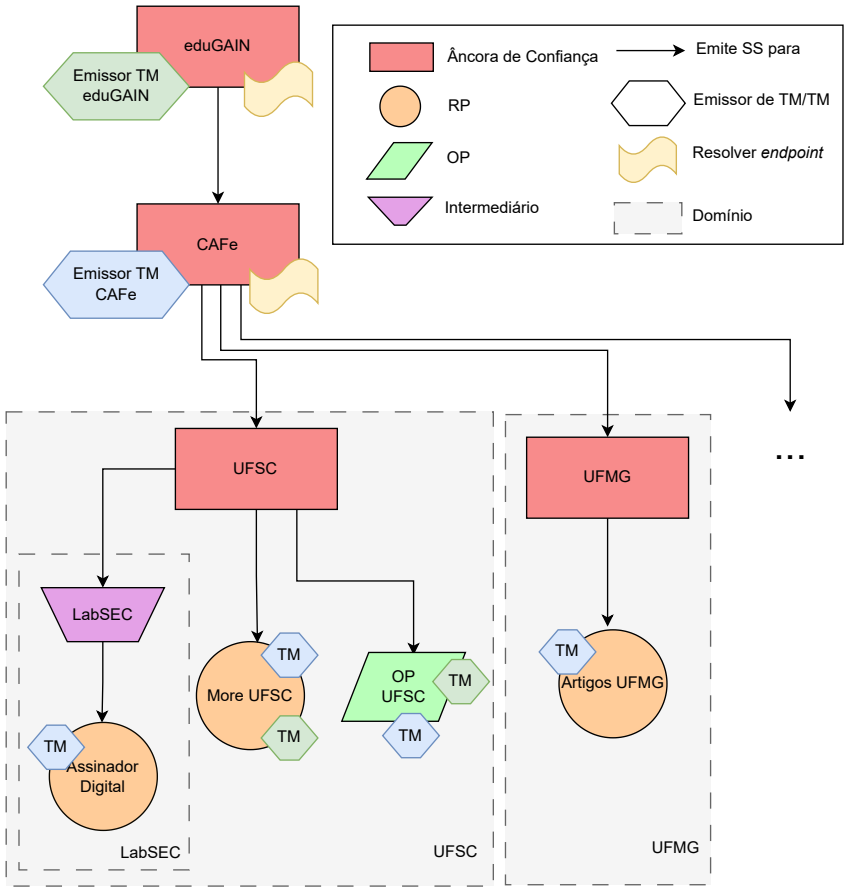
A Figura 23 ilustra a hierarquia proposta para a CAFe 2.0 considerando a compatibilidade com o piloto da interfederação eduGAIN. Nela, a `UFSC` representa a TA de uma instituição da federação, assim como a `UFMG`. O Laboratório de Segurança em Computação (`LabSEC`), que é uma suborganização da `UFSC`, por exemplo, pode ser representado como um IM e atuar como entidade superior de algum provedor na federação, caso do `Assinador Digital` na Figura. `More UFSC` e `OP UFSC` são subordinados diretos da TA da instituição. Por outro lado, `UFMG` apenas possui um serviço (`Artigos UFMG`), que participa da federação.

Observa-se que tanto os RPs do domínio `UFSC` quanto os do domínio `UFMG` conseguem estabelecer uma TC válida com `OP UFSC` por meio da confiança na TA `CAFe` e vice-versa, permitindo que os usuários da `UFSC` acessem todos os RPs da federação.

7.2.2 *Trust Marks*

Uma das premissas do piloto da eduGAIN é utilizar o recurso de TMs do OID Fed para atestar que um provedor pertence à interfederação. Na prática, o pertencimento à federação é atestado pela presença do TM na EC da entidade. Quando a TC for resolvida, esse TM é validado criptograficamente. Os TMs serão emitidos de forma automática pelo emissor da eduGAIN, que validará se o provedor já possui o TM da CAFe como pré-condição para emissão (Dijk, 2024).

Figura 23 – Arquitetura básica baseada na compatibilidade com a eduGAIN.



Fonte: o autor.

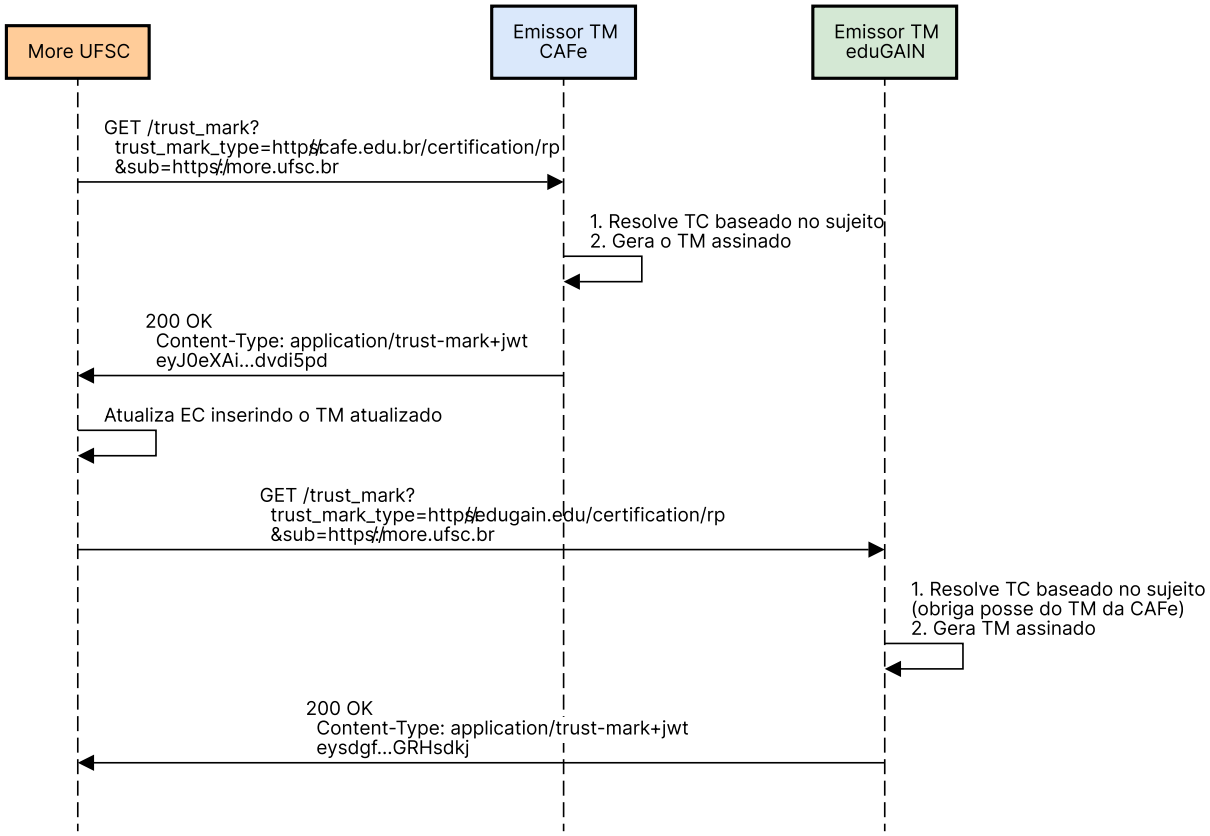
O diagrama de sequência da Figura 24 ilustra o processo de emissão de TMs requerido pela definição da eduGAIN. Nela, o `More UFSC` requisita TMs da CAFe e da eduGAIN. O intervalo de tempo para a atualização dos TMs deve ser definido pelo Perfil Técnico da eduGAIN. No exemplo da Figura 23, todos os provedores possuem TMs da CAFe, enquanto apenas os provedores subordinados diretamente da TA da UFSC possuem o TM da eduGAIN.

A renovação do TM é automatizada e periódica. Ainda, a Figura 24 pode sofrer alteração a depender do nível de controle que o operador do barramento quiser exercer sobre os provedores. O operador pode optar por emitir TMs apenas para provedores que respeitarem certa política de metadados, ou mesmo, autorizar ou revogar a emissão manualmente para cada provedor. Nesses casos, não basta apenas o `Emissor TM CAFe` executar os dois passos mostrados na Figura 24, sendo necessárias mais verificações precedendo a emissão.

Discussões atuais indicam que os TMs serão a ferramenta utilizada para atestar conformidade com padrões e políticas definidos por organismos como a REFEDS. A conformidade com o *Security Incident Response Trust Framework for Federated Identity* (SIRTFI)[1], por exemplo, pode ser representada por um TM (Harris; Vaghetti, 2025).

[1] https://refeds.org/sirtfi

Figura 24 – Emissão de TMs para a federação.



Fonte: o autor.

7.2.3 *Metadata Policy*

As políticas de metadados do OID Fed permitem que TAs e IMs imponham restrições que se propagam para baixo na hierarquia e são aplicadas automaticamente no momento da resolução de uma TC, assim como explicado em 3.4.2. O mecanismo representa essencialmente uma obrigatoriedade por requisitos. Na prática, ao invés de regras documentadas e verificadas manualmente, as restrições são expressas em artefatos criptograficamente assinados e verificáveis por qualquer entidade que resolva a TC. Dessa forma, o operador da federação pode impor comportamentos de segurança e interoperabilidade sem precisar intervir individualmente em cada provedor.

Cabe destacar que o Perfil Técnico da eduGAIN (Harris; Vaghetti, 2025) ainda não define quais políticas de metadados devem ser adotadas na interfederação. A consolidação dessas políticas é, portanto, responsabilidade do operador da CAFe e deverá ocorrer em trabalhos futuros. A título de sugestão, as seguintes políticas representam requisitos relevantes de segurança e interoperabilidade para o contexto da CAFe:

- **Algoritmos de assinatura**: restrição dos algoritmos criptográficos aceitos. Por exemplo, `RS256` e `ES256`, proibindo algoritmos obsoletos ou inseguros (Jones; Bradley, 2024);
- **Identificação obrigatória**: exigência do campo `organization_name` para todas as LEs,

garantindo rastreabilidade dos provedores na federação;

- ***Grant types* seguros**: exclusão do fluxo *implicit* do OAuth 2.0, considerado inseguro para aplicações *web* modernas (Lodderstedt; Mcgloin; Hunt, 2013);
- **PKCE obrigatório**: exigência de PKCE (Sakimura; Bradley; Agarwal, 2015) em fluxos com RPs públicas, mitigando ataques de interceptação do código de autorização;
- **Método de autenticação no *token endpoint***: incentivo ao uso de `private_key_jwt` em detrimento de `client_secret_basic`, aumentando a segurança na troca de credenciais entre RP e OP;
- ***Scopes* permitidos**: restrição dos escopos OIDC solicitáveis por uma RP, limitando a superfície de exposição de dados pessoais dos usuários e alinhando a federação com princípios de minimização de dados.

7.2.4 Gestão de Subfederações

A hierarquia descrita até aqui pressupõe que todos os provedores sejam públicos, isto é, subordinados direta ou indiretamente da TA da CAFe e, portanto, confiáveis por toda a federação. Esse é o comportamento da CAFe atual e das demais federações SAML2, nas quais a confiança é estabelecida de forma agregada, na qual toda entidade listada no arquivo de metadados torna-se tecnicamente confiável pelas demais (RNP, 2020).

Muitas instituições, contudo, mantêm serviços que deveriam ser acessíveis apenas no seu próprio escopo e, na ausência de um mecanismo nativo de restrição, não os integram à federação. Para atender à premissa de *gestão de subfederações* discutida na Seção 6.5, o BAITA introduz a distinção entre entidades públicas e privadas, organizando estas últimas em subfederações com âncoras de confiança próprias.

O mecanismo apoia-se inteiramente no modelo de TCs do OID Fed. As entidades públicas permanecem subordinadas à TA da CAFe e, como todos os provedores confiam nessa âncora, é sempre possível resolver uma TC válida até elas. As entidades privadas, por sua vez, são ancoradas em uma TA distinta, própria da instituição ou de um grupo, inalcançável a partir da TA da CAFe, de modo que apenas os provedores que compartilham essa mesma âncora conseguem estabelecer confiança com elas. O escopo de confiança passa, assim, a ser delimitado pela própria estrutura da federação, sem depender de configurações manuais de liberação de atributos em cada provedor.

Vale destacar que essa abordagem é mais robusta do que uma baseada em TMs. Como os TMs são opcionais segundo a especificação e poderiam ser ignorados por implementações que não os verificassem (Hedberg et al., 2026), a restrição via hierarquia de TAs é imposta pela própria resolução criptográfica da cadeia de confiança. A implantação é organizada em três fases progressivas.

Na primeira fase, somente entidades públicas são aceitas, em um arranjo análogo ao modelo atual da CAFe e idêntico à hierarquia da Figura 23. Toda entidade é alcançável a partir da TA da CAFe e, portanto, confiável por toda a federação.

A segunda fase introduz o acesso restrito ao escopo institucional. Para cada instituição que o deseje, cria-se uma `TA Privada`, posicionada como superior imediata da TA Pública da instituição. A Figura 25 ilustra esse arranjo, em que a `UFSC privado` é superior imediata da `UFSC` e do `WAYF`. Os provedores subordinados à `UFSC privado`, mas não à `UFSC` pública, tornam-se confiáveis apenas no escopo da instituição.

No exemplo, a RP `Serviço UFSC` resolve uma TC válida até o `OP UFSC` e até o `OP Privado LabSEC`, mas ambos os provedores da `UFSC` não conseguem estabelecer confiança com provedores da `UFMG` ou com qualquer outro subordinado direto da TA `CAFe`, pois não existe cadeia que os ligue àquela TA. Para que as facilidades do WAYF também estejam disponíveis nos fluxos privados, ele é igualmente subordinado à `UFSC privado`, garantindo uma TC válida entre ele e os provedores privados.

Por fim, mantém-se uma entidade de políticas, a `UFSC privado políticas`, operada pela RNP e posicionada como superior do ramo privado. Sua existência decorre do modelo de políticas de metadados do OID Fed. Como as políticas de metadados definidas por uma entidade só se propagam para os provedores que a consideram ao resolver a TC, as políticas declaradas na TA da CAFe não alcançam as entidades privadas. Sem uma entidade dedicada a impor regras no ramo privado, esse ramo seria excluído de todo o conjunto de requisitos de segurança e interoperabilidade que o barramento aplica aos provedores públicos.

A entidade de políticas preenche essa lacuna ao estabelecer um conjunto mínimo de regras que um provedor privado deve cumprir para integrar o barramento. Por estar acima deles na cadeia de confiança, suas políticas de metadados são aplicadas automaticamente na resolução de cada TC. Em outras palavras, ser privado restringe quem pode acessar o serviço, mas não o dispensa de atender a um patamar mínimo de segurança e conformidade.

Essa garantia é especialmente relevante para os provedores privados hospedados e configurados pela própria instituição, fora da infraestrutura do barramento. Nesses casos, a entidade de políticas é o único mecanismo de que o operador dispõe para assegurar, de forma verificável na própria resolução da TC, que tais provedores respeitem as regras mínimas do barramento.

A terceira fase cobre os casos mais excepcionais, em que um conjunto restrito de entidades precisa formar a sua própria esfera de confiança. A Figura 26 ilustra esse cenário. Suponha que a RP `Serviço Treinamento IA` seja computacionalmente custosa e que o `LabSEC` deseje torná-la acessível apenas a um OP específico, o `OP Privado LabSEC`. Por se tratar de um relacionamento pontual entre entidades, cria-se uma TA dedicada, a `Projeto de Pesquisa`, que une os dois provedores em uma federação aninhada independente, arranjo plenamente suportado pelo OID Fed, que admite federações aninhadas com topologia arbitrária (Hedberg et al., 2026).

À semelhança da fase anterior, mantém-se uma entidade de políticas (`Projeto de Pesquisa políticas`) sob responsabilidade do próprio SGB. Também assume-se que as RPs

Figura 25 – Hierarquia considerando subfederação privada por instituição

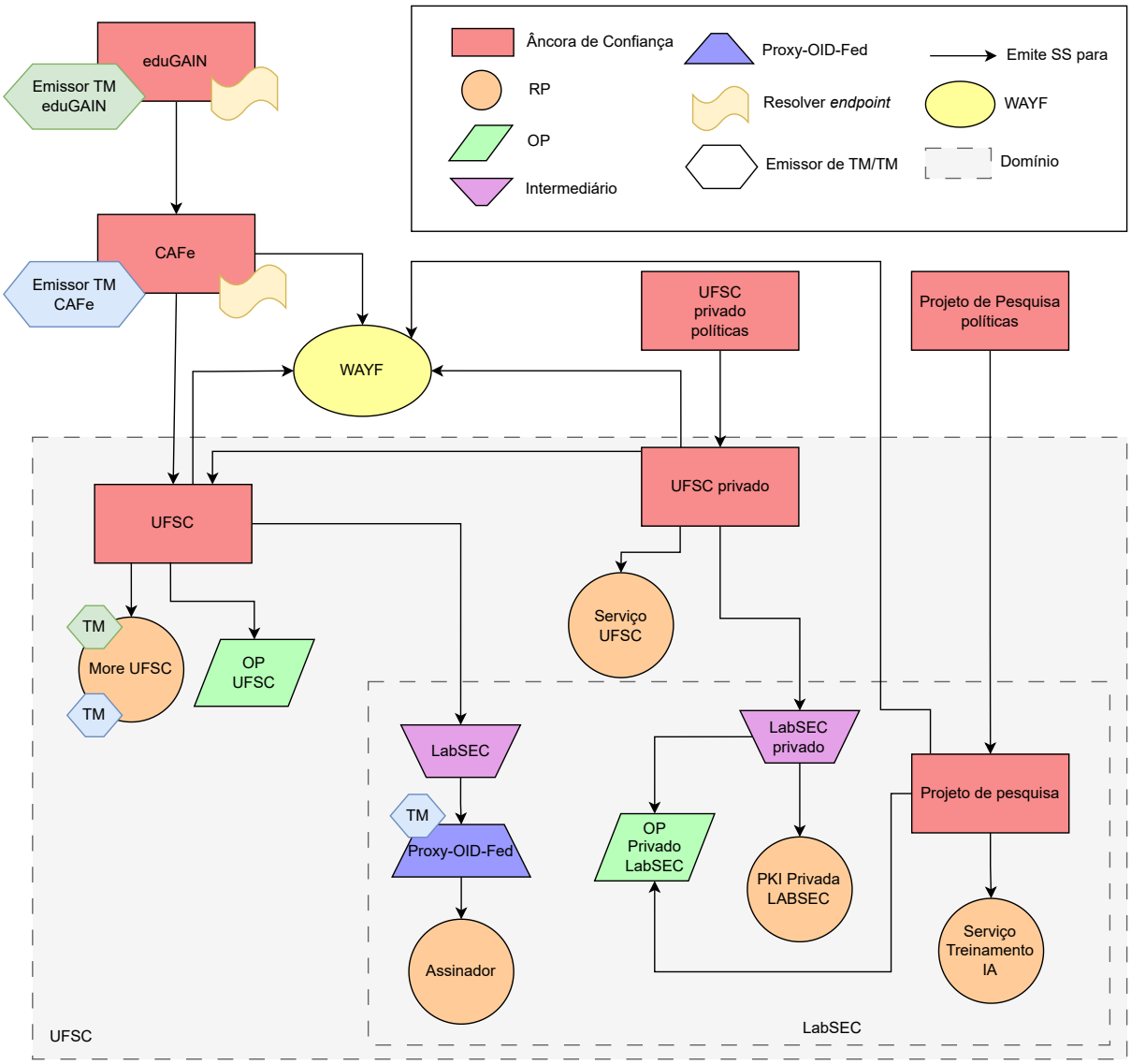


Fonte: o autor.

desse grupo, assim como as demais RPs da federação, não implementam por si mesmas a descoberta de OPs nem as outras funções do WAYF, das quais dependem para conduzir a autenticação. Para utilizá-lo, no entanto, é necessária uma TC válida entre elas e o WAYF. Como este é subordinado à TA da CAFe, ele é inalcançável a partir da TA `Projeto de Pesquisa`, que forma uma federação aninhada isolada do restante da federação. Por essa razão, a TA `Projeto de Pesquisa` também adota o WAYF como subordinado. Ao emitir uma SS para ele, traz o WAYF para dentro da sua esfera de confiança e possibilita a formação da TC que permite às RPs do grupo alcançá-lo e, por meio dele, descobrir e acessar o `OP Privado LabSEC`. É essa subordinação adicional que explica a aresta entre as duas entidades na Figura.

Em conjunto, as três fases permitem que uma instituição se integre ao BAITA mesmo quando parte dos seus serviços exige acesso restrito, com o escopo de confiança garantido pela

Figura 26 – Hierarquia considerando subfederação por grupo específico.



Fonte: o autor.

própria estrutura hierárquica da federação e não por configurações manuais ou por mecanismos opcionais do protocolo. Cabe ressaltar que a gestão de subfederações aqui descrita constitui uma proposta arquitetural validada conceitualmente, mas ainda não implementada no BAITA no âmbito deste trabalho. Sua realização é tratada como trabalho futuro na Seção 9.1.

7.3 SISTEMA DE GESTÃO DO BAITA

O SGB é o núcleo administrativo do barramento. Sua função primária é abstrair a operação do OID Fed para os administradores do barramento, fornecendo interfaces para o gerenciamento do ciclo de vida das entidades. O sistema é projetado com uma abordagem de descentralização administrativa: o operador da RNP delega a gestão de organizações a administradores específicos, que por sua vez delegam a gestão de provedores aos seus mantenedores,

criando uma cadeia de responsabilidade que espelha a hierarquia de confiança do OID Fed. O diagrama de casos de uso da Figura 27 ilustra os dois perfis de usuário do SGB e as interações possíveis com o sistema.

Figura 27 – Casos de uso do SGB.



Fonte: o autor.

O operador da RNP tem acesso aos casos de uso de maior privilégio: emitir e revogar TMs e gerenciar *metadata policies*. Tanto ele quanto os administradores de organizações podem gerenciar a hierarquia de entidades da federação, criar novas organizações ou provedores, delegar a administração de organizações a gestores específicos e gerenciar metadados e métricas acerca das entidades sob sua responsabilidade. Importante observar que cada um pode atuar apenas no escopo da sua organização. Um administrador da UFSC só pode criar e gerenciar entidades subordinadas à TA da UFSC, enquanto o operador da CAFe pode fazê-lo no escopo da TA da CAFe. Um mantenedor de SP ou IdP só pode editar os metadados do seu próprio provedor.

Para garantir a segurança e integridade das entidades do barramento, a maioria dos casos de uso requer autenticação. Daí, surge a necessidade de um mecanismo de autenticação e controle de acesso robusto, implementado por um PI dedicado. O SGB delega a autenticação

a esse PI, que emite tokens de acesso com escopos específicos para cada perfil de usuário, permitindo controle granular sobre as ações permitidas a cada tipo de usuário.

O SGB é uma aplicação *multi-tenant* que implementa múltiplas funcionalidades críticas para a operação do barramento. Em primeiro lugar, atua como implementação das entidades da federação, hospedando as entidades definidas pelo protocolo, como TAs e IMs. O SGB gerencia as configurações dessas entidades e emite os artefatos que permitem a resolução de TCs. Em segundo lugar, o sistema funciona como um *Backend For Frontend* (BFF) para a interface *web*, expondo uma API otimizada para as necessidades do *frontend*. Em terceiro lugar, serve como orquestrador de componentes. Ele coordena a configuração e operação dos demais componentes do barramento hospedados na infraestrutura da RNP, como o Proxy-OID-Fed, o Proxy-SAML, e os emissores de TMs. Em quarto lugar, implementa gestão *multi-tenant* do barramento, em que cada organização tem seu próprio domínio administrativo, com isolamento de dados, permissões e operações. Essa arquitetura *multi-tenant* permite que o operador da federação (a RNP) hospede múltiplas subfederações independentes, cada uma com sua própria hierarquia, sem custo operacional significativo ou complexidade de operação para os gestores das instituições. Finalmente, em quinto lugar, o sistema mantém registros de auditoria de todas as operações administrativas. A Figura 28 detalha a arquitetura interna do sistema, organizada em camadas de interface *web*, API, motores internos e armazenamento e integrações com serviços externos.

A Interface *Web* funciona como porta de entrada para gestores de entidades do barramento, sendo uma aplicação responsiva construída com *React*, *TypeScript* e usando a biblioteca de componentes *Shadcn*. A aplicação é compilada com *Vite* e é composta por páginas que permitem as interações de: gestão das entidades, com visualização, edição e histórico de edições das mesmas; *onboarding* das entidades, com a criação e delegação das mesmas para gestores; visualização da topologia da federação, com navegação em profundidade, mostrando todas as entidades integradas ao BAITA; e página de perfil do usuário, para gerenciamento de fatores de autenticação dos usuários.

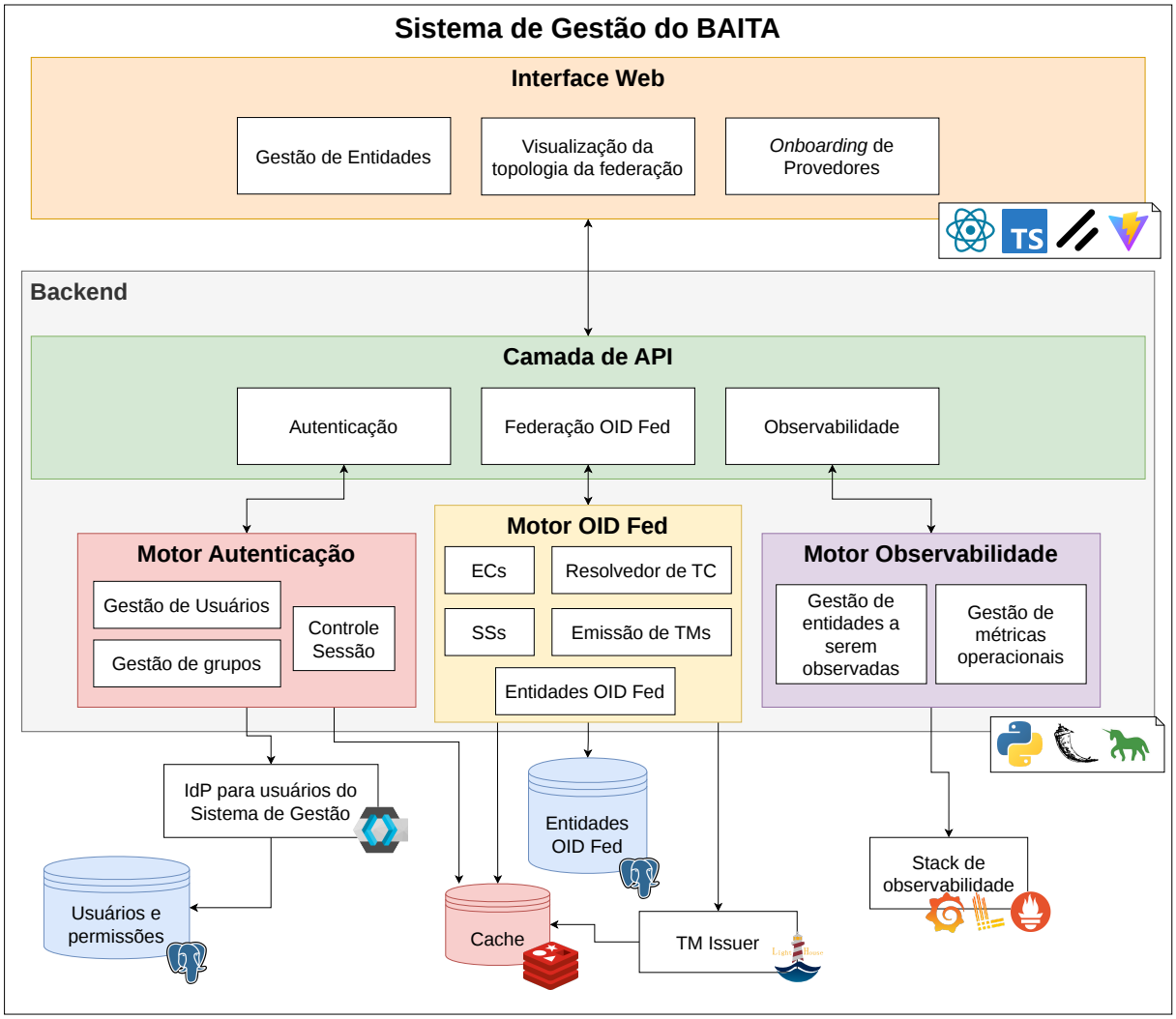
A camada de API, implementada em *Flask* e utilizado o Interface de Gateway de Servidor Web (WSGI, *Web Server Gateway Interface*) *gunicorn*[2], funciona como orquestrador entre a interface *web* e os motores internos do *backend*. Nessa camada, ferramentas de segurança são utilizadas, como o *Flask-CORS*[3] para controle de origem, e *Flask-Limiter*[4] para proteção contra ataques de negação de serviço. O roteamento é organizado em módulos principais:

- Rotas de autenticação, que lidam com a autenticação de usuários do sistema de gestão, interagindo com o motor de autenticação e sendo chamadas sempre que há necessidade de autenticar ou encerrar a sessão de autenticação de usuários.

- Rotas relacionadas com a federação OID Fed, que provém métodos de Criar, Ler, Atua-

[2] https://gunicorn.org/
[3] https://flask-cors.readthedocs.io/en/latest/
[4] https://pypi.org/project/Flask-Limiter/

Figura 28 – Arquitetura detalhada do SGB.



Fonte: o autor.

lizar e Excluir (CRUD, *Create, Read, Update and Delete*) das entidades do barramento e funcionalidades integradas de *onboarding* das entidades e descoberta de provedores. Todas as operações de CRUD são gravadas para auditabilidade das entidades. Além disso, segundo a especificação do OID Fed, o sistema expõe os *endpoints* de federação em nome dos *tenants* que gerencia.

- Rotas voltadas para observabilidade, que se comunicam com o motor de observabilidade para fornecer informações acerca das entidades que são observadas.

O motor autenticação é responsável por controlar a identidade dos administradores do barramento, implementando gestão de usuários, grupos e controle de sessão em um contexto separado da federação. Esse componente se comunica com o IdP que gerencia os usuários do SGB, que por sua vez é implementado como uma instância do *Keycloak*[5], que controla o banco de dados de usuários e permissões. Para fazer gestão de permissões dos usuários, o motor

[5] https://www.keycloak.org/

de autenticação se comunica com o *Keycloak* via API de administração e utiliza para isso a biblioteca *keycloak-admin-aio*[6].

O fluxo de autenticação dos usuários do sistema utiliza *Authorization Code Grant* com PKCE, protegendo contra ataques de interceptação. Tokens dos usuários são armazenados em uma cache *Redis*. Metadados de usuário como grupos (para controle de acesso baseado em papéis) e *claims* customizados são extraídos do *userinfo endpoint* do *Keycloak* e cacheados, evitando consultas repetidas. A separação entre o IdP de administração e os OPs da federação garante que indisponibilidade do OP não impeça a operação do barramento.

O motor OID Fed é o núcleo que implementa as operações do protocolo de federação, gerando, assinando e rotacionando artefatos criptográficos, e fornece funcionalidades para gestão das entidades mantidas pelo sistema. Dentre suas responsabilidades, estão: (i) a geração dinâmica de ECs, SSs e outros artefatos a serem expostos nos *endpoints* de federação em nome das entidades gerenciadas pelo sistema; (ii) geração de TMs para as entidades da federação; (iii) resolução da TC; (iv) mapeamento da topologia da federação; e (v) funcionalidades de *onboarding* e edição de entidades. (ii) e (iii) são delegadas ao *LightHouse*[7], ferramenta que está sendo utilizada pela eduGAIN em seu piloto. (i), (iv) e (v) usam o banco de dados de entidades do OID Fed para consulta e escrita. (iv) utiliza a cache para evitar percorrer todas as entidades da federação a cada requisição recebida.

O banco de dados, como mostrado na Figura 28, é um *PostgreSQL*[8]. A interface de persistência é construída com *SQLAlchemy*[9] via *Flask-SQLAlchemy*[10], com versionamento de modelos de dados gerenciado pela ferramenta *Flask-Migrate*[11]. Para a *cache*, utiliza-se a ferramenta *Redis*.

O motor de observabilidade coleta e expõe métricas para dois objetivos distintos. Um deles é a observabilidade da operação do SGB e o outro é a observabilidade dos provedores federados. Quanto à observabilidade do sistema em si, utiliza-se o *prometheus-flask-exporter*[12] para monitorara as requisições recebidas, coletando latência, contadores de *status* por código de resposta, e taxa de erros.

Em relação à observabilidade de provedores, o motor de observabilidade oferece a funcionalidade de checagem de saúde dos SPs integrados ao barramento. Para tal, utilizam-se consultas diretas ao *Prometheus* que avalia continuamente a saúde desses SPs, verificando disponibilidade de *endpoints* configurados na interface *web* pelo gestor do SP. Essas métricas permitem aos operadores identificar provedores degradados, anômalos ou potencialmente comprometidos, facilitando ações preventivas ou corretivas.

O *Grafana* é configurado para ler as métricas do *Prometheus* e exibir *dashboards* ope-

[6] https://github.com/V-Mann-Nick/keycloak-admin-aio
[7] https://go-oidfed.github.io/lighthouse/
[8] https://www.postgresql.org/
[9] https://www.sqlalchemy.org/
[10] https://flask-sqlalchemy.readthedocs.io/en/stable/
[11] https://flask-migrate.readthedocs.io/en/latest/
[12] https://pypi.org/project/prometheus-flask-exporter/

racionais. Essa instrumentação permite detectar degradações de desempenho no barramento, rastrear o comportamento de usuários administrativos, identificar provedores problemáticos, e gerar alerta baseado em limiares de disponibilidade e conformidade.

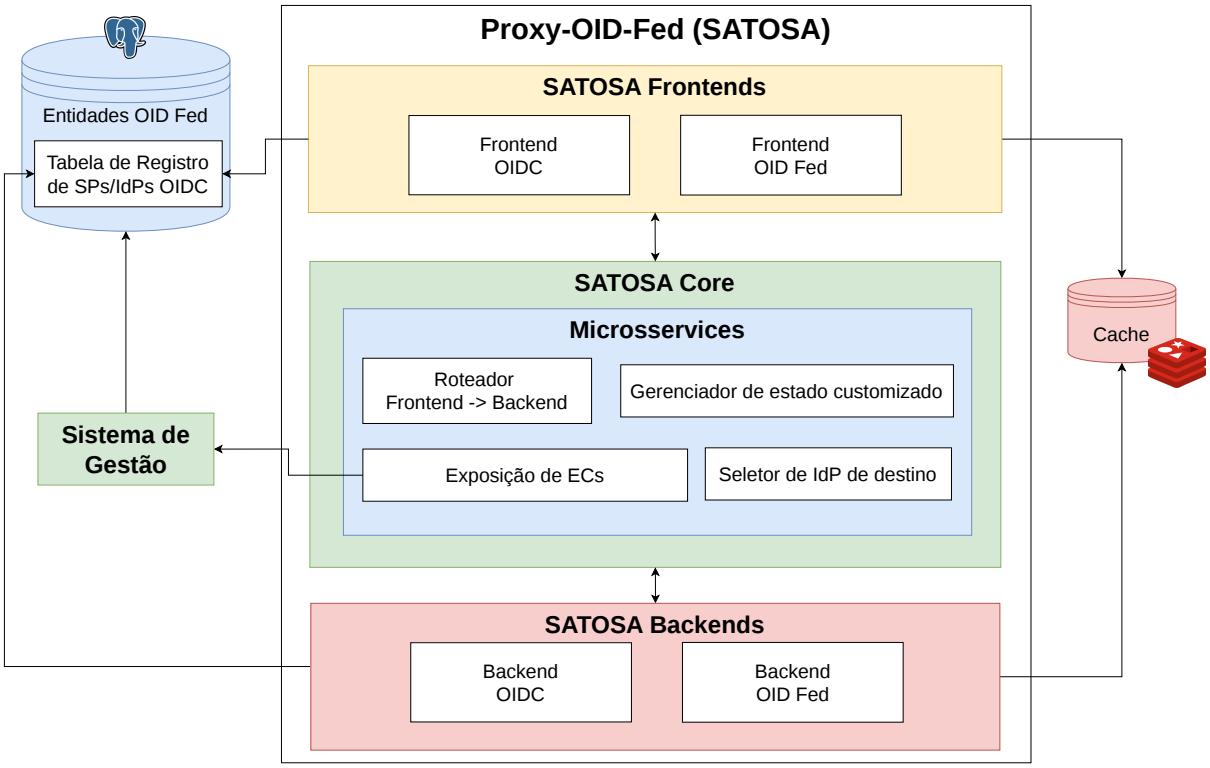
7.4 *PROXY*-OID-FED

O Proxy-OID-Fed é o componente central de redução de atrito para a integração de novos provedores ao barramento. Conforme apresentado na Seção 6.2, ele permite que provedores que implementam apenas o OIDC, sem as extensões adicionais exigidas pelo OID Fed, se integrem à federação com esforço mínimo, por meio de um processo de registro gerenciado pelo SGB.

Esse *proxy* atua com duas faces. Para a federação, ele é uma LE que suporta o protocolo OID Fed e para o provedor integrado ao BAITA, ele atua como uma entidade OIDC, recebendo e repassando as requisições de autenticação entre a federação o provedor. Esse duplo papel é o que permite que provedores sem suporte ao OID Fed participem do barramennto de forma transparente.

Internamente, o Proxy-OID-Fed é implementado sobre o SATOSA, um *proxy* de autenticação maduro e amplamente adotado em soluções de FID. A Figura 29 ilustra sua arquitetura interna, que segue o modelo de funcionamento do SATOSA, projetado em três partes: *frontends*, núcleo e *backends*.

Figura 29 – Arquitetura interna do *Proxy-OID-Fed*.



Fonte: o autor.

Os *frontends* são os pontos de entrada do *proxy*, no qual as requisições de autenticação são recebidas. Em outras palavras, um *frontend*, no SATOSA, é o componente que atua como IdP para a entidade que inicia o fluxo de autenticação. Nesse caso, o Proxy-OID-Fed é composto por duas implementações de *frontend*. O *Frontend OIDC* expõe uma face de OP OIDC, utilizada quando o *proxy* abstrai a federação para uma RP. Este, por sua vez, conecta-se ao *proxy* como se este fosse o seu OP. Já o *Frontend OID Fed* expõe uma face de OP vista pela federação, utilizada quando o *proxy* representa um OP perante os demais provedores do OID Fed.

O núcleo do SATOSA media os fluxos entre *frontends* e *backends* por meio de um conjunto de microsserviços. O `Roteador Frontend` $\rightarrow$ `Backend` mapeia cada requisição recebida ao *backend* apropriado, conforme o cenário de mediação. Já o microsserviço de `Exposição de ECs` serve as ECs em nome de cada provedor integrado, delegando a assinatura desses artefatos ao SGB, que detém as chaves de federação, via requisição HTTP a um *endpoint* de API do SGB. O `Seletor de PI de destino` encaminha o fluxo ao provedor correto, seja por meio do WAYF, seja diretamente, quando o destino já é conhecido. Por fim, o `Gerenciador de estado customizado` mantém o estado de cada fluxo isolado, por meio de *cookies* distintos por rota, evitando colisões quando múltiplos fluxos de autenticação ocorrem simultaneamente sobre o mesmo *proxy*.

Os *backends* são os pontos de saída e determinam o protocolo com que o *proxy* se comunica com o provedor *upstream*. Em outras palavras, um *backend* do SATOSA atua como um SP e encaminha a autenticação para um IdP. Seguindo a mesma lógica dos *frontends*, o Proxy-OID-Fed é composto por duas implementações de *backend*. O *Backend OID Fed* faz o *proxy* atuar como uma RP da federação, resolvendo TCs e alcançando os OPs confiáveis. Já o *Backend OIDC* faz o *proxy* atuar como uma RP OIDC perante ao OP OIDC integrado ao BAITA.

A combinação entre um *frontend* e um *backend* define o cenário de mediação. O *Frontend OIDC* com o *Backend OID Fed* abstrai a federação para uma RP, enquanto o *Frontend OID Fed* com o *Backend OIDC* expõe um OP à federação.

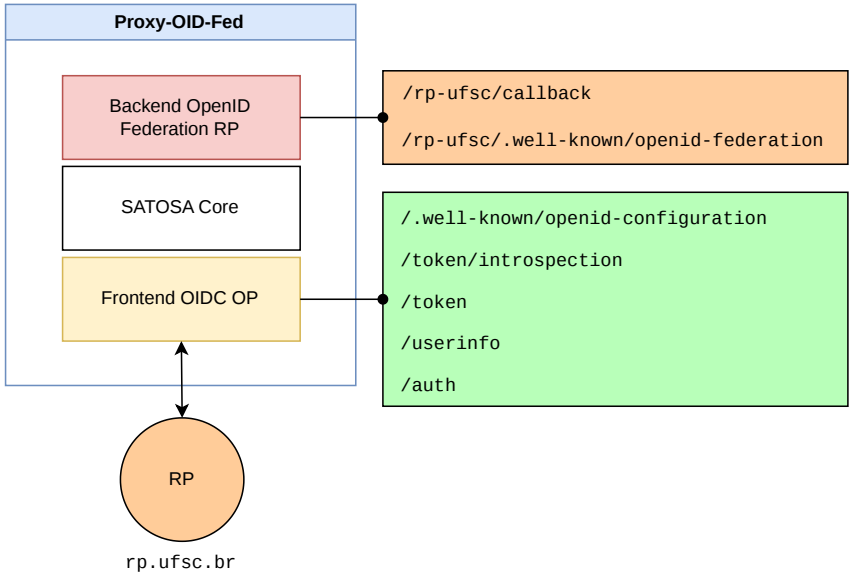
Como mostra a Figura 29, essas implementações de *frontends* e *backends* apoiam-se em serviços externos. O banco de dados *PostgreSQL*, compartilhado com o SGB, contém as *Entidades OID Fed* e a *Tabela de Registro de SPs/IdPs OIDC*. O *proxy* acessa essa base em modo somente leitura para obter as credenciais e os metadados dos provedores que representa. Isso é essencial para que o Proxy-OID-Fed seja configurado em tempo de execução, reconhecendo e aceitando comunicação com os provedores que representa, sem que precise ser reiniciado.

O *cache* Redis armazena as TCs já resolvidas, reduzindo a latência de resoluções subsequentes. O SGB, por fim, é responsável por registrar os subordinados e por assinar as ECs em nome do *proxy*, conforme detalhado na Seção 7.4.1.

Para cada provedor subordinado, o Proxy-OID-Fed expõe um conjunto de *endpoints* que o tornam operacional e visível na federação. O conjunto exato depende do papel do provedor. As Figuras 30 e 31 ilustram, respectivamente, os *endpoints* expostos quando o *proxy* abstrai a federação para uma RP e para um OP. Nos exemplos, adota-se o prefixo `rp-ufsc` para a RP

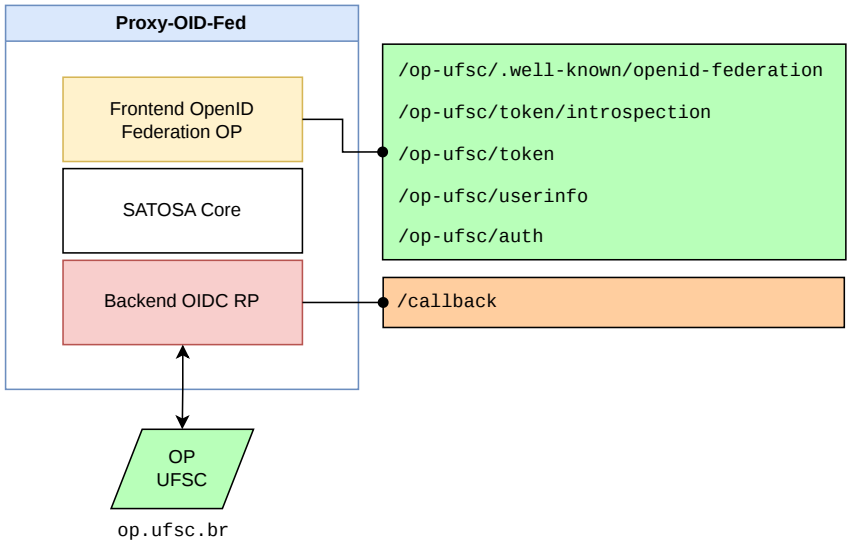
e `op-ufsc` para o OP, ilustrando como cada subordinado recebe um espaço de endereçamento próprio.

Figura 30 – *Endpoints* expostos pelo Proxy-OID-Fed abstraindo federação para uma RP.



Fonte: o autor.

Figura 31 – *Endpoints* expostos pelo Proxy-OID-Fed abstraindo a federação para um OP.



Fonte: o autor.

Quando o *proxy* abstrai a federação para uma RP, o *Frontend OIDC* expõe os *endpoints* OIDC padrão, como `/auth`, `/token`, `/userinfo`, `/token/introspection` e `/.well-known/openid-configuration`. O mantenedor da RP apenas configura o *proxy* como seu OP, sem qualquer alteração na sua aplicação. Esses *endpoints* podem ser compartilhados entre múltiplas RPs subordinadas porque cada requisição carrega o `client_id`, que o *proxy* utiliza para localizar o registro correspondente. Voltado à federação, o *Backend OID Fed* expõe *endpoints*

prefixados por RP registrada, como `/rp-ufsc/.well-known/openid-federation`, que publica a EC da RP, e `/rp-ufsc/callback`, onde o OP da federação entrega o código de autorização. Ao receber o código, o *proxy* o troca por *tokens* e os repassa à RP registrada, abstraindo dela o fluxo de autorização completo do OAuth 2.0 (Hardt, 2012) em um contexto federado.

Quando o *proxy* abstrai a federação para um OP, conforme a Figura 31, os papéis se invertem. O *Frontend OID Fed* apresenta-se à federação como o próprio OP, expondo *endpoints* prefixados por OP registrado, como `/op-ufsc/.well-known/openid-federation`, `/op-ufsc/auth`, `/op-ufsc/token`, `/op-ufsc/userinfo` e `/op-ufsc/token/introspection`. Voltado ao provedor registrado, o *Backend OIDC* expõe um único *endpoint* de *callback*, por onde recebe o retorno do OP real. Quando uma RP da federação faz uma requisição de autenticação ao OP exposto pelo *proxy*, este valida a TC da RP em nome do OP e repassa a requisição ao OP real utilizando as credenciais pré-registradas, liberando o subordinado dessas responsabilidades.

7.4.1 Registro e representação de um provedor

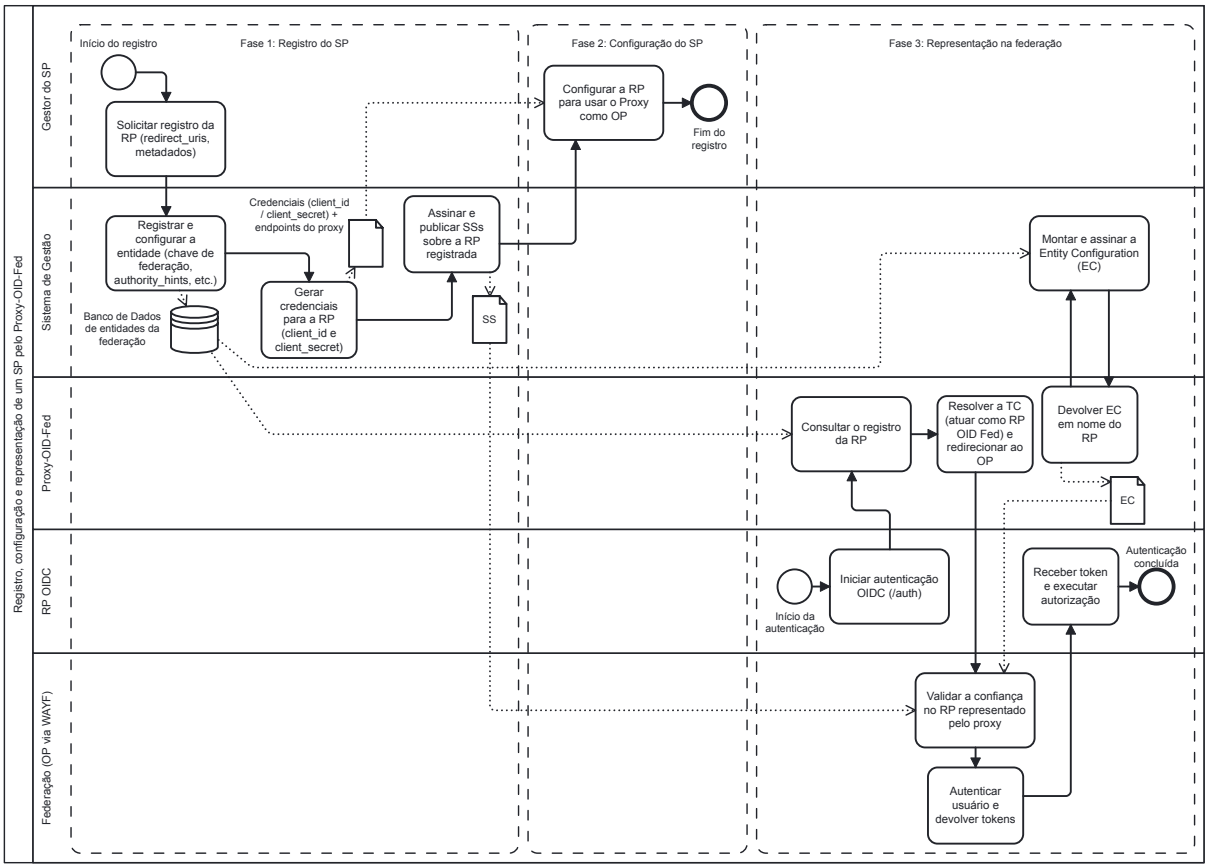
O ciclo de vida de um provedor integrado ao Proxy-OID-Fed compreende as fases de registro de configuração do provedor e de representação. A Figura 32 apresenta essas fases para um registro de uma RP OIDC.

A fase de registro é a interação entre o mantenedor do provedor e o SGB. O mantenedor submete, por meio da interface do sistema, os dados necessários como nome, informações de metadados e as URIs de redirecionamento. Essas interações são ilustradas em detalhes na Seção 8.1 e no Apêndice B. Na submissão do formulário, o SGB executa ações internas relacionadas com a criação da entidade na hierarquia da federação. O sistema automaticamente define configurações para esta entidade como a *claim* `authority_hints` e o par de chaves de federação para a entidade. Essas configurações são persistidas no banco de dados das entidades do barramento.

Além disso, o sistema gera automaticamente credenciais que devem ser configuradas no provedor, registrando-as na `Tabela de Registro de SPs/IdPs OIDC` e marcando a entidade como ativa. Em seguida, o sistema passa a assinar SSs para o novo subordinado, estabelecendo o elo de subordinação que permite a construção de TCs válidas. Ao final dessa fase, o SGB devolve ao mantenedor a confirmação do registro, acompanhada das credenciais e do *endpoint* do *proxy* que deverá ser configurado.

A fase de configuração ocorre inteiramente no domínio do mantenedor e não envolve o BAITA. De posse das credenciais e do *endpoint* do *proxy*, o mantenedor ajusta o cliente OIDC da sua aplicação para utilizar o Proxy-OID-Fed como seu OP. Na prática, configura o *endpoint* de descoberta (`/.well-known/openid-configuration`) apontando para o *proxy*, o `client_id` e o `client_secret` registrados. Essa é a única intervenção exigida do mantenedor na sua própria infraestrutura e equivale a apontar a aplicação para um novo OP. Toda a complexidade do OID Fed permanece encapsulada no barramento. Concluída a configuração, o provedor está

Figura 32 – Sequência de registro e configuração de um SP e sua subsequente representação na federação pelo Proxy-OID-Fed.



Fonte: o autor.

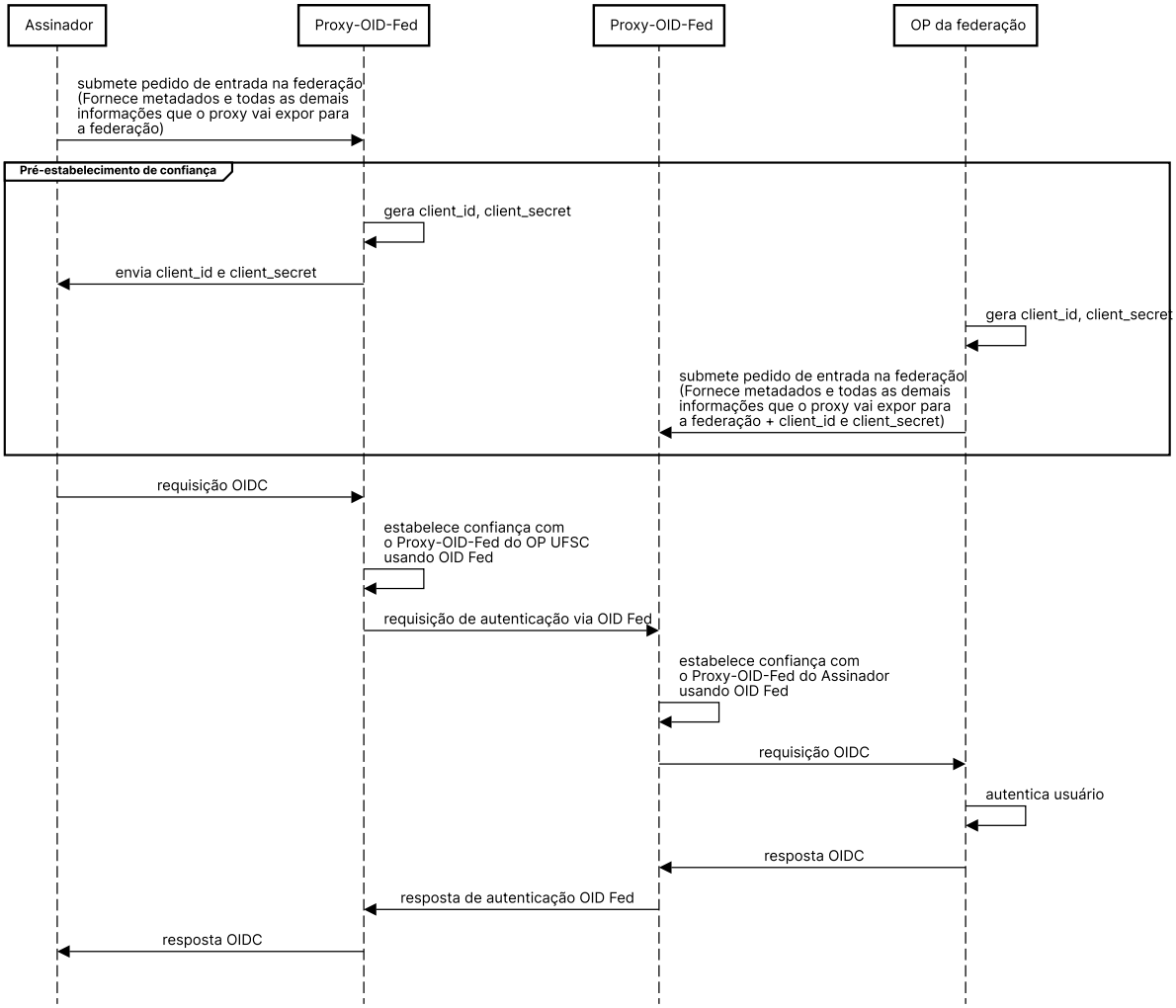
apto a operar na federação.

A fase de representação demonstra como o *proxy* passa a atuar em nome do provedor. Como o Proxy-OID-Fed compartilha o banco de dados com o SGB, o novo registro torna-se imediatamente visível para ele. Quando um usuário inicia uma autenticação a partir do SP recém-integrado, a aplicação direciona a requisição ao `Frontend OIDC` do *proxy*, identificando-se pelo `client_id`. O Proxy-OID-Fed consulta a base de registros, recupera as credenciais e os metadados correspondentes e assume o papel de RP da federação por meio do `Backend OID Fed`, resolvendo a TC e conduzindo o fluxo até um OP da CAFe. Quando uma entidade da federação precisa descobrir o provedor ou validar a confiança nele, requisita a EC ao *endpoint* prefixado exposto pelo *proxy*. Ao final, o *proxy* entrega à aplicação os *tokens* OIDC resultantes. Dessa forma, o provedor torna-se operacional e confiável na federação, embora continue a falar apenas OIDC.

O fluxo demonstrado no diagrama de sequência da Figura 32 é análogo para o caso de integração de um OP ao BAITA. Neste caso, as credenciais serão criadas pelo OP e informadas ao barramento no momento do registro. A Figura 33 ilustra de forma simplificada a fase de registro para ambos os casos e o fluxo de autenticação por meio do OID Fed. Tanto o PS `Assinador` quanto o OP `OP UFSC` não implementam OID Fed, porém implementam OIDC. Na

Figura 33, ambos integram-se à federação por meio do componente Proxy-OID-Fed.

Figura 33 – Fluxo de registro e posterior comunicação entre OPs e RPs OIDC por meio do Proxy-OID-Fed.



Fonte: o autor.

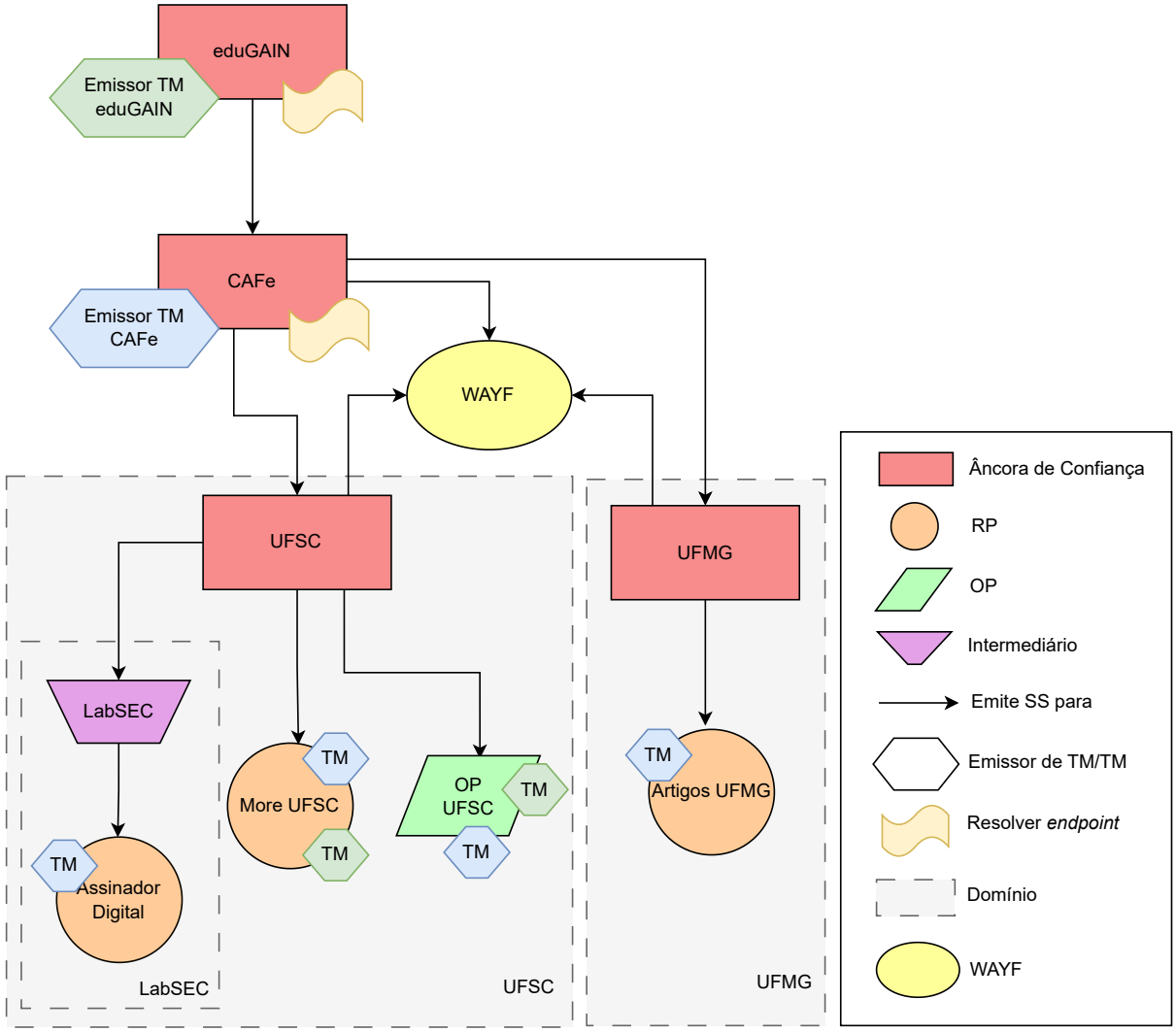
Após o registro, o fluxo de autenticação acontece normalmente, com os *proxies* abstraindo a validação da TC, que é uma operação obrigatória segundo o OID Fed.

7.5 WAYF

O WAYF é um componente visível ao usuário final e crítico para a interoperabilidade da federação. Sua função vai além do papel tradicional de um WAYF, listar os PIs disponíveis para que o usuário escolha sua instituição (Pöhn; Hommel, 2023). No BAITA, o WAYF atua como um *proxy* de identidade completo. Ele intercepta a comunicação entre RP e OP, normaliza os atributos da identidade do usuário e garante o nível de autenticação solicitado.

Em se tratando da topologia da federação, o WAYF é subordinado tanto da TA da CAFe quanto das TAs das instituições. A Figura 34 estende a hierarquia apresentada na Figura 23 para destacar o posicionamento do WAYF nessa topologia.

Figura 34 – Posicionamento do WAYF na topologia da federação.



Fonte: o autor.

Na Figura 34, a TA da CAFe e cada TA das instituições subordinadas à CAFe, `UFSC` e `UFMG`, emitem SSs para o WAYF. A subordinação às TAs das instituições permite que o WAYF seja confiável no escopo restrito do domínio da instituição (*i.e.*, acessível por aplicações e OPs que tem como TA a instituição).

O WAYF é implementado sobre o *Keycloak*, uma plataforma madura de gerenciamento de identidade e acesso amplamente adotada em federações acadêmicas. A Figura 35 ilustra a arquitetura interna, organizada em camadas, a exemplo dos demais componentes. A interface de descoberta de provedores é implementada utilizando a ferramenta *Keycloakify*[13] em *React*.

O núcleo de processamento é configurado para utilizar as extensões customizadas que foram implementadas por meio de *Service Provider Interface* (SPI). Como o *Keycloak* ainda não possui suporte ao OID Fed, um dos SPIs implementados atua na descoberta de IdPs do BAITA. Para isso, o WAYF se comunica com o SGB para requisitar as entidades acessíveis via TA da CAFe, filtra por tipo de entidade (`openid_provider`), e renderiza uma interface

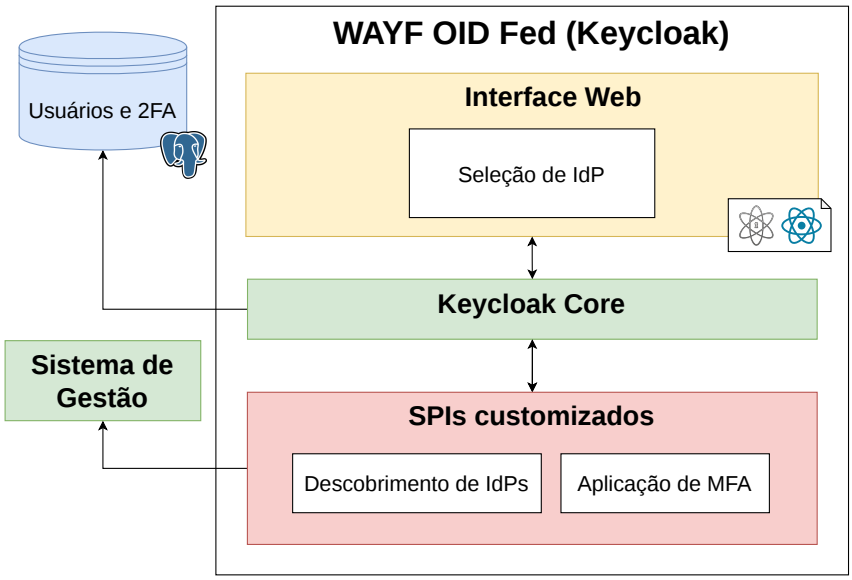
[13] https://www.keycloakify.dev/

interativa para seleção.

Outra extensão que faz parte do WAYF garante que o segundo fator de autenticação seja aplicado durante a autenticação do usuário. O WAYF, após receber a resposta de autenticação do OP *upstream*, verifica a *claim* que diz respeito ao método de autenticação utilizado, e se for o caso, solicita o segundo fator de autenticação para o usuário.

Outras extensões implementadas garantem a propagação, ao OP *upstream*, de parâmetros como escopos customizados, requerimentos de nível de autenticação e o `idp_hint`. Este último implementa o mecanismo de *IdP hinting* especificado pela AARC-G061 (AARC, 2021), cuja RP indica previamente qual PI deve ser utilizado, simplificando ou mesmo dispensando a etapa de descoberta para o usuário final. A especificação padroniza, no contexto das federações de pesquisa e educação, um parâmetro (`aarc_idp_hint`) que carrega o identificador do provedor desejado, o *EntityID*, no SAML2, ou o *issuer*, no OIDC.

Figura 35 – Arquitetura detalhada do WAYF.



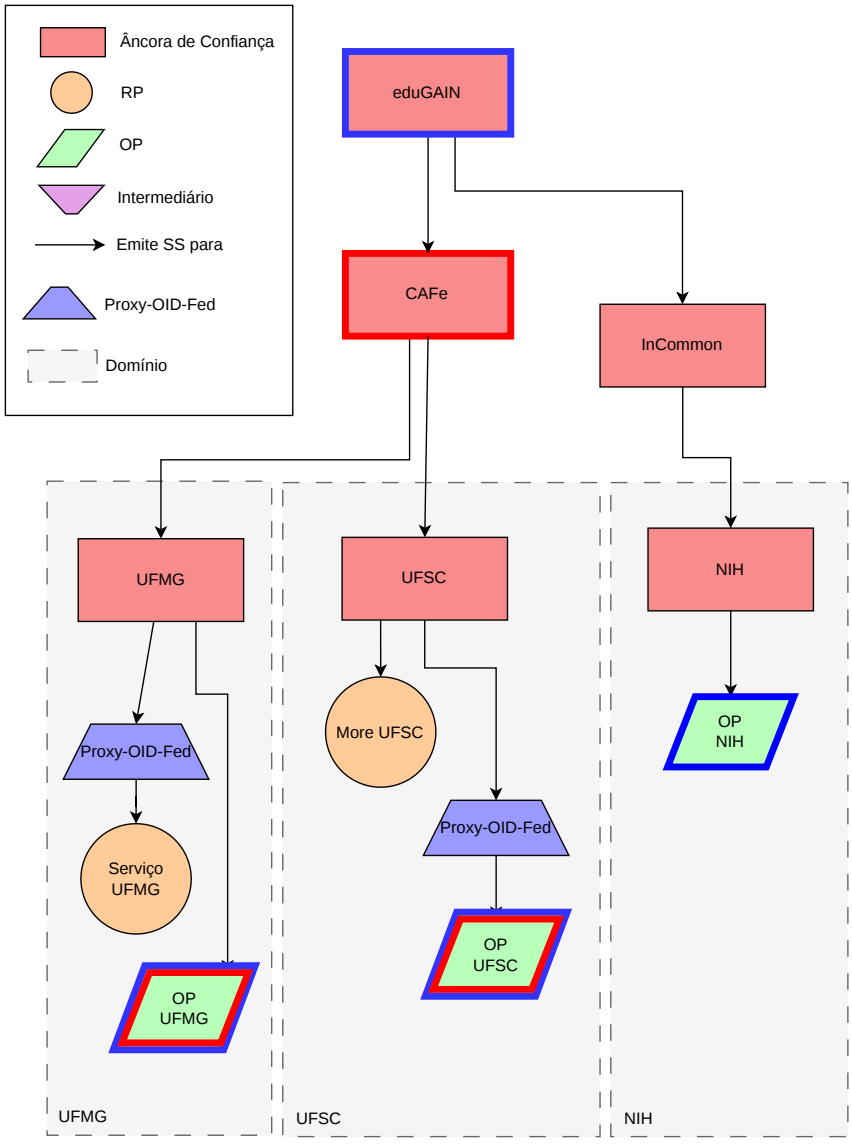
Fonte: o autor.

7.5.1 Descoberta de Provedores de Identidade

O algoritmo de descoberta de OPs do WAYF é acionado quando ele recebe uma requisição de autenticação de uma RP. O parâmetro `trust_anchor`, incluído na requisição, informa qual TA deve ser usada como ponto de partida da descoberta. Trata-se do parâmetro definido pelo OID Fed para a resolução de TCs empregado, por exemplo, no *endpoint* de resolução (`/federation_resolve_endpoint`), que identifica a TA na qual a cadeia de confiança deve ancorar-se. O valor, nesse caso, é o identificador da TA na federação. Caso não seja informado, o WAYF utiliza as TAs configuradas por padrão pelo administrador (geralmente, a TA da CAFe e a TA da eduGAIN).

O algoritmo realiza uma Busca em Largura (BFS, *Breadth-First Search*) na topologia da federação, consultando o *endpoint* de listagem de subordinados (`/federation_list_endpoint`) de cada entidade não-folha (Hedberg et al., 2026). Para cada entidade encontrada, o WAYF acessa sua EC via `/.well-known/openid-federation` e verifica se o campo `entity_type` dentro de `metadata` contém `openid_provider`. As entidades que satisfazem esse critério são adicionadas à lista de OPs disponíveis, que é então apresentada ao usuário. A Figura 36 exemplifica o resultado desse percurso sobre uma topologia ilustrativa.

Figura 36 – Resultado da busca em largura na federação executada pelo WAYF.



Fonte: o autor.

No exemplo da Figura 36, a busca parte das duas TAs configuradas por padrão: a CAFe e a eduGAIN. A partir da eduGAIN, âncora da interfederação, o algoritmo alcança não apenas a CAFe, mas também a federação norte-americana `InCommon` e, através dela, o domínio `NIH` e o seu provedor `OP NIH`. A partir da CAFe, a busca desce para os domínios `UFSC` e `UFMG`. Dentro de cada domínio, o algoritmo continua a expandir os IMs até alcançar as LEs (por vezes

representadas pelo Proxy-OID-Fed).

Ao inspecionar a EC de cada entidade, o WAYF coleta somente aquelas cujo entity_-type é openid_provider. No exemplo, OP UFSC, OP UFMG e OP NIH. As entidades do tipo RP, como More UFSC e Serviço UFMG, são visitadas durante a busca, mas descartadas da lista apresentada ao usuário. Nota-se ainda que a descoberta é indiferente à forma como o OP se conecta à hierarquia. Tanto os OPs subordinados diretamente a uma TA de uma instituição, como o OP UFMG, quanto os expostos por um Proxy-OID-Fed, como o OP UFSC, são encontrados da mesma maneira.

A busca pode ser configurada para considerar apenas OPs que possuam o TM da CAFe, garantindo que apenas provedores reconhecidos pelo operador da federação sejam listados. O metadado keywords, definido nas ECs dos OPs, também pode ser utilizado para filtrar os resultados segundo o perfil da RP que solicitou a autenticação (Hedberg et al., 2026).

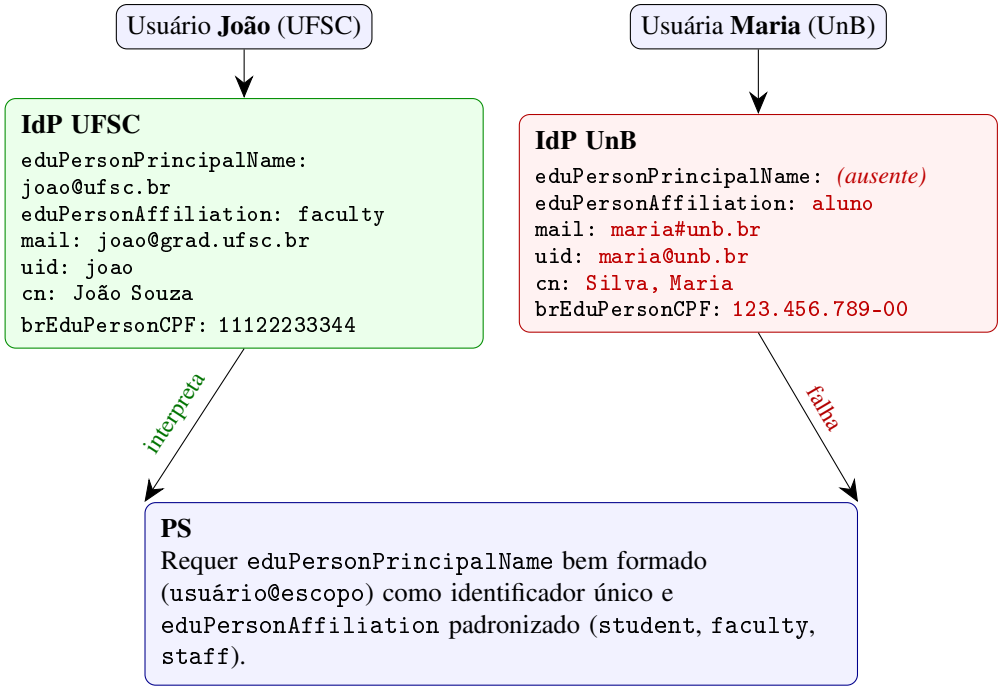
7.5.2 Normalização de Atributos

A interoperabilidade semântica é um dos problemas mais persistentes em federações de grande escala. Na CAFe atual, mesmo com a política existente que obriga a integração de IdPs *Shibboleth* (RNP, 2021), a autenticação resulta em atributos com valores semanticamente incorretos ou publicados em campos inadequados dependendo do IdP consultado. Isso gera falhas de autenticação e dificulta o desenvolvimento de PSs que precisam interpretar corretamente os atributos do usuário. A Figura 37 ilustra um exemplo concreto desse problema. Nela, dois usuários, cada um autenticado pelo IdP da sua instituição, com o mesmo conjunto de atributos. O IdP UFSC os publica corretamente, enquanto o IdP UnB introduz erros sintáticos e semânticos, de modo que o PS interpreta o primeiro, mas falha com o segundo.

Após receber a resposta de autenticação do OP escolhido pelo usuário, o WAYF aplica um mapeamento de atributos antes de repassar a resposta à RP. O mapeamento é baseado nos esquemas padronizados adotados pela CAFe:

- ***eduPerson***: esquema de atributos de identidade definido pela REFEDS para uso em federações acadêmicas internacionais (Refeds, 2020). Define atributos como eduPerson-PrincipalName, identificador único do usuário no escopo da instituição, e eduPersonAffiliation, vínculo do usuário com a instituição (professor, estudante, servidor);
- ***inetOrgPerson***: esquema padrão para representação de pessoas em diretórios LDAP (Smith, 2000), utilizado para atributos básicos como mail, cn e sn;
- ***brEduPerson***: extensão brasileira ao *eduPerson* definida pela RNP (Cafe, 2020), que inclui atributos específicos do contexto nacional como o brEduPersonCPF e o brEduPersonUniqueId.

Figura 37 – Problema de interoperabilidade sintática e semântica de atributos.



Fonte: o autor.

Como primeira implementação, o WAYF reaproveita o mapeador de atributos já mantido pela RNP[14], invocando-o internamente para traduzir os atributos recebidos do OP *upstream*, em vez de expor essa responsabilidade a cada PS, como ocorre atualmente. O mapeamento, nesse caso, é configurável por instituição. O operador pode definir regras específicas para IdPs que utilizam representações fora do padrão. Essa centralização da normalização elimina a necessidade de cada PS implementar seu próprio mapeamento, o que é exigido atualmente pela API de mapeamento da RNP e gera atrito para os mantenedores dos serviços. O resultado da normalização é um conjunto de *claims* do OIDC padronizado, que a RP recebe independentemente do OP que autenticou o usuário.

Trata-se, contudo, de uma solução inicial. O mapeamento semântico robusto de atributos entre SAML2 e OIDC depende de padrões de representação ainda em consolidação no contexto das federações acadêmicas, tema aprofundado na Seção 9.1.

7.5.3 Garantia de Múltiplos Fatores de Autenticação

A garantia de MFA em uma federação de grande escala apresenta um desafio operacional significativo. Exigir que todos os PIs a implementem implica atualizar centenas de instâncias gerenciadas de forma independente, processo que tende a ser lento e desigual (Ometov et al., 2018). O WAYF resolve esse problema centralizando a execução do segundo fator de autenticação, quando necessário.

[14] https://hmg.api.mapeador.rnp.br/

Após receber a resposta de autenticação do OP, o WAYF verifica a *claim* `acr` (*Authentication Context Class Reference*) (Sakimura et al., 2023a) para determinar o nível de garantia com que o usuário foi autenticado. O BAITA deve adotar o *REFEDS Assurance Framework* 2.0 (REFEDS, 2023) como referência para a definição e comunicação desses níveis, mantendo compatibilidade com o padrão internacional para federações acadêmicas. Apesar disso, esse padrão ainda carece de funcionalidades para conseguir representar com clareza os fatores de autenticação utilizados no processo de autenticação. Essa adoção não foi implementada no escopo deste trabalho e é apontada na Seção 9.1.

Se o nível de garantia retornado pelo OP for inferior ao exigido pela RP, por exemplo, se a RP solicita autenticação de múltiplos fatores, mas o OP realizou apenas autenticação por senha (identificada pela *claim* `amr`), o WAYF solicita o segundo fator diretamente ao usuário antes de devolver a resposta. Desse modo, a garantia de MFA é assegurada na federação independentemente da implementação individual de cada PI.

Cabe ressaltar que a realização plena dessa funcionalidade depende de discussões que extrapolam o escopo deste trabalho, notadamente o armazenamento centralizado de dados dos usuários e sobre a representação padronizada dos fatores e métodos de autenticação no *schema* OIDC adotado no contexto das federações acadêmicas. Esses pontos são aprofundados na Seção 9.1.

A Figura 38 ilustra o papel do WAYF ao apresentar um fluxo de autenticação completo, no qual a RP `Artigos UFMG` solicita a autenticação de um usuário cujo OP de origem é o `OP UFSC`, exposto na federação pelo `Proxy-OID-Fed`.

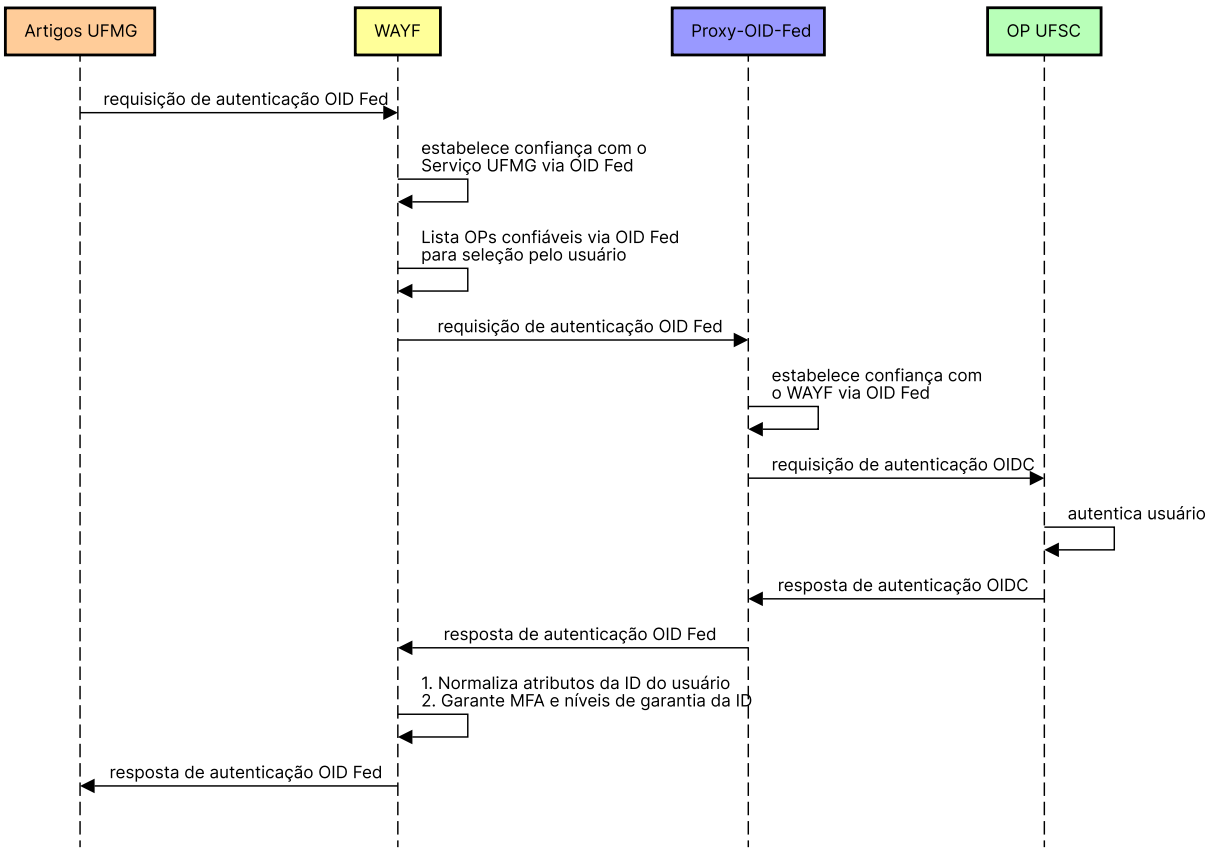
O fluxo inicia quando a RP envia uma requisição de autenticação OID Fed ao `WAYF`. Antes de prosseguir, o `WAYF` estabelece a confiança com a RP, resolvendo a sua TC no OID Fed, e então executa o algoritmo de descoberta descrito na Seção 7.5.1, apresentando ao usuário a lista de OPs confiáveis para que ele selecione sua instituição de origem. Selecionado o `OP UFSC`, o `WAYF`, agora no papel de RP, encaminha uma nova requisição de autenticação OID Fed ao `Proxy-OID-Fed`, que representa esse OP na federação. O `Proxy-OID-Fed`, por sua vez, estabelece a confiança com o `WAYF` resolvendo a sua TC e traduz a requisição para o OIDC convencional, repassando-a ao `OP UFSC`, que autentica o usuário.

A resposta percorre o caminho inverso. O `OP UFSC` devolve uma resposta de autenticação OIDC ao `Proxy-OID-Fed`, que a converte de volta para o OID Fed e a entrega ao `WAYF`. Nesse ponto, antes de devolver o resultado à RP, o `WAYF` executa as funções de normalização dos atributos da identidade do usuário, conforme a Seção 7.5.2, e assegura o nível de garantia e o MFA exigidos, conforme a Seção 7.5.3. Por fim, o `WAYF` entrega à `Artigos UFMG` a resposta de autenticação OID Fed já normalizada.

7.6 *PROXY*-SAML

À semelhança do Proxy-OID-Fed, o Proxy-SAML é implementado sobre o SATOSA. Ao configurar o conjunto *frontend*/*backend* de formas opostas, o mesmo componente atende

Figura 38 – Fluxo de autenticação passando pelo WAYF.



Fonte: o autor.

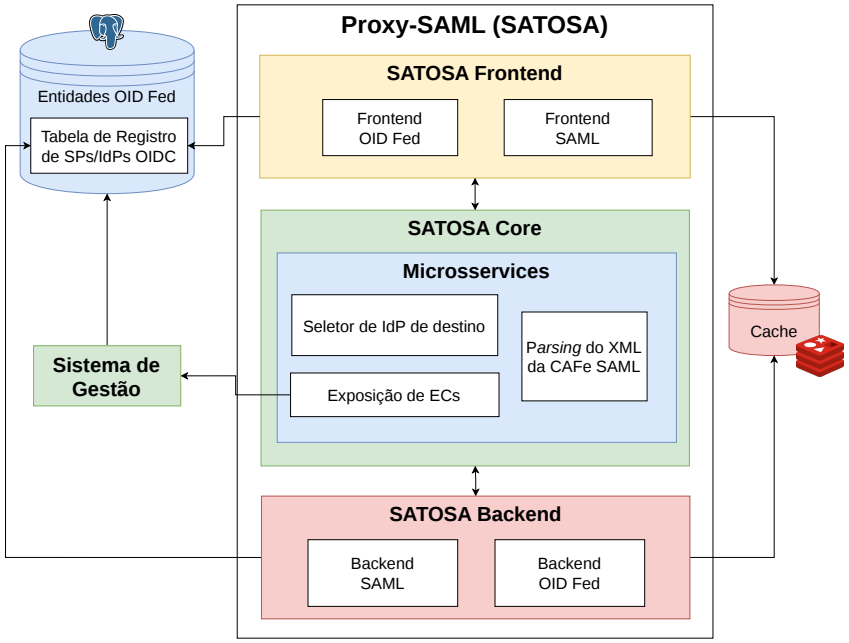
tanto ao cenário de RPs modernas acessando PIs SAML2 quanto o inverso. Conforme apresentado na Seção 6.4, a estratégia de compatibilidade é organizada em três níveis progressivos, cujos fluxos são detalhados na presente Seção.

Por utilizar o SATOSA como base, a arquitetura do Proxy-SAML é análoga à apresentada na Seção 7.4, dela diferindo pelos *plugins* de protocolos habilitados e por um microsserviço adicional. A Figura 39 ilustra essa arquitetura.

Diferentemente do `Proxy-OID-Fed`, que opera apenas com *plugins* OIDC e OID Fed, o Proxy-SAML habilita simultaneamente *plugins* SAML2 e OID Fed em ambas as faces. O *frontend* dispõe de um *Frontend OID Fed* e de um *Frontend SAML*, enquanto o de *backend* dispõe de um *Backend SAML* e de um *Backend OID Fed*. A escolha do conjunto *frontend*/*backend* determina o nível de compatibilidade exercido, conforme detalhado na Seção 7.6.1. No núcleo, além dos microsserviços de `Exposição de ECs` e `Seletor de PI de destino`, também presentes no `Proxy-OID-Fed`, o Proxy-SAML inclui um microsserviço de `Parsing do XML da CAFe SAML2`, responsável por descobrir dinamicamente os provedores legados declarados no arquivo de metadados da CAFe.

Esse microsserviço cumpre dois papéis, um para cada nível de compatibilidade. Os PIs SAML2 são extraídos para serem expostos como OPs na nova federação, viabilizando o nível 2 de suporte ao SAML2. Analogamente, os PSs SAML2 são extraídos para que o *Frontend*

Figura 39 – Arquitetura do Proxy-SAML.



Fonte: o autor.

SAML do *proxy* possa confiar neles e atendê-los como um PI, viabilizando o nível 3. Assim como o `Proxy-OID-Fed`, o componente compartilha o banco de dados e o *cache* Redis com o SGB, que assina os artefatos de federação em seu nome.

Para que os PIs legados sejam reconhecidos como provedores na federação OID Fed, o Proxy-SAML emprega o microsserviço de *parsing* mencionado, que lê periodicamente o arquivo XML de metadados da federação CAFe SAML2[15] e, para cada PI SAML2 encontrado, expõe um OP correspondente na federação OID Fed, com uma EC e um SS emitidos em seu nome. Isso torna os PIs legados visíveis e acessíveis às RPs modernas sem qualquer alteração nos servidores *Shibboleth* das instituições. Os PSs SAML2, em contrapartida, não são expostos como entidades OID Fed. Eles permanecem como provedores SAML2 e, no nível 3, apenas precisam reconhecer o *frontend* SAML2 apresentado pelo *proxy*.

Diferentemente do Proxy-OID-Fed, cujos PIs e PSs são registrados pela interface do SGB, esse módulo não aceita registros manuais. Seus subordinados são determinados exclusivamente pelo conteúdo do arquivo de metadados, sendo atualizados automaticamente quando novos PIs são adicionados à federação SAML2.

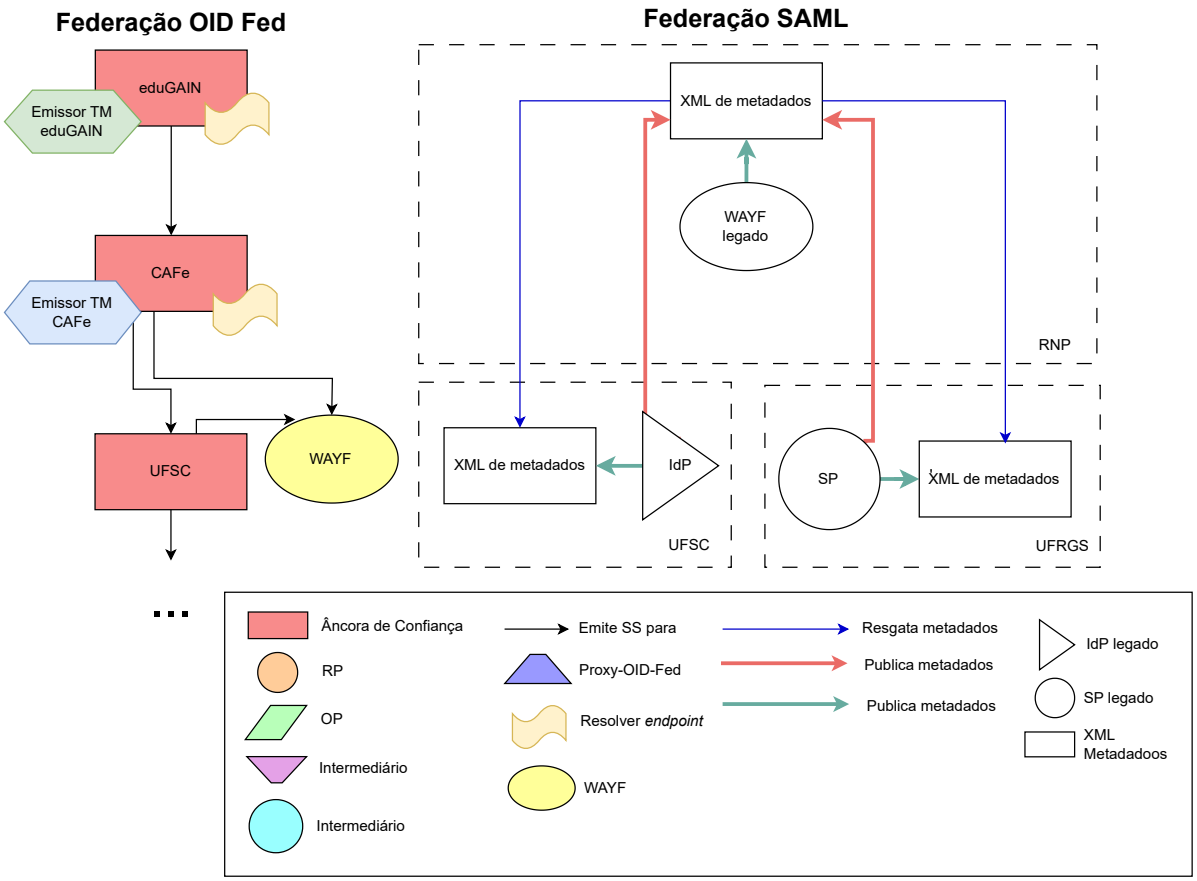
7.6.1 Níveis de Compatibilidade com o Legado

No estado inicial (nível 1), as duas federações operam de forma independente e sem interação. PSs SAML2 acessam PIs *Shibboleth* na federação SAML2, enquanto PSs OID Fed comunicam-se com PIs que implementam OID Fed na nova federação. Não há tradução de mensagens nem necessidade do Proxy-SAML. A Figura 40 ilustra esse cenário, que corresponde ao

[15] https://ds.cafe.rnp.br/metadata/cafe-metadata.xml

início da implantação do barramento. Nela, nota-se que as duas federações coexistem, mas não há comunicação entre elas.

Figura 40 – Representação da topologia com suporte ao legado no nível 1.



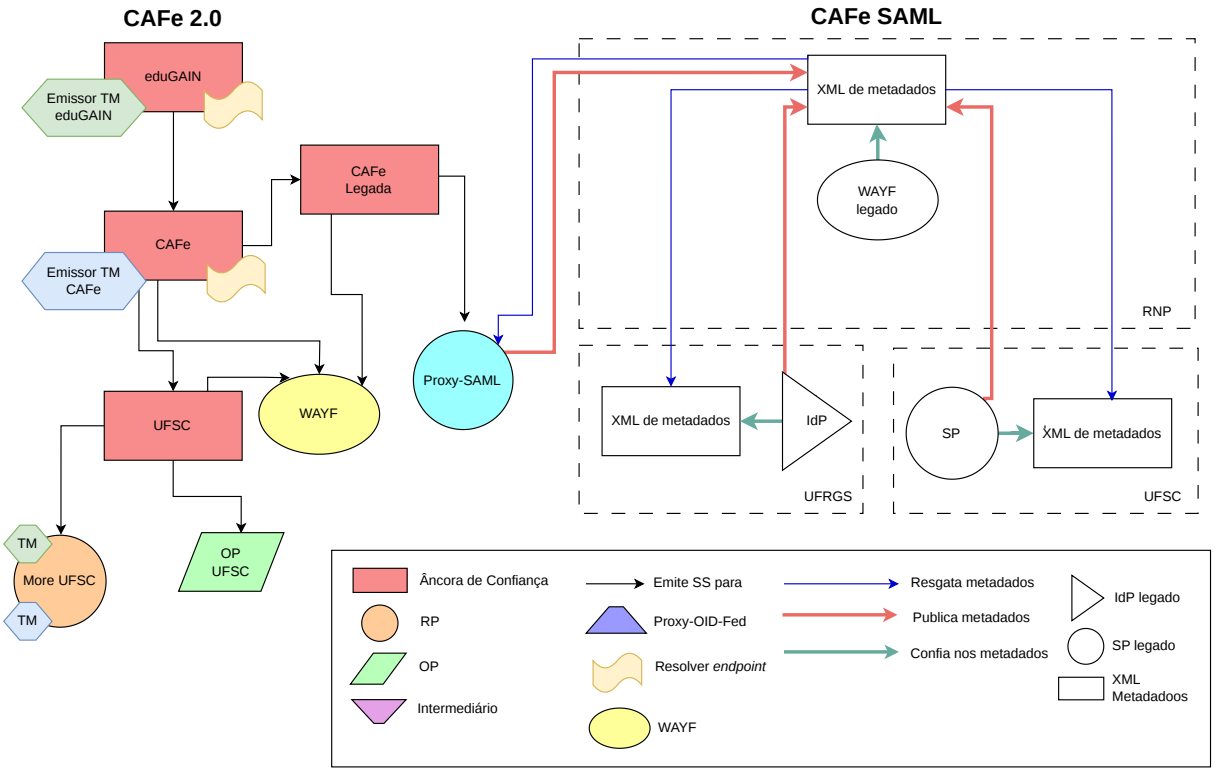
Fonte: o autor.

O nível 2 proporciona aos PSs da nova federação delegar a autenticação de usuários a PIs legados. Nesse caso, o Proxy-SAML é configurado com um *frontend* OID Fed e um *backend* SAML2. Na federação OID Fed é registrado como um OP e, na federação SAML2, como um PS. Por meio do módulo de *parsing* descrito acima, cada PI SAML2 do arquivo de metadados da CAFe é exposto como um OP na nova federação. A Figura 41 mostra o posicionamento do *proxy* entre as duas federações.

O fluxo de autenticação correspondente ao nível 2 de compatibilidade é apresentado na Figura 42. Nela, uma RP OID Fed (`More UFSC`) envia uma requisição de autenticação ao WAYF. Importante notar que `More UFSC` também poderia ser um SP OIDC integrado ao barramento, e, neste caso, o Proxy-OID-Fed executaria a requisição de autenticação OID Fed. Por questões de simplificação, isso está abstraído na Figura.

O `WAYF` lista os OPs disponíveis. Essa listagem inclui os PIs legados expostos pelo Proxy-SAML. Após o usuário selecionar um PI legado, o WAYF encaminha a requisição ao Proxy-SAML, que a traduz para uma `SAMLRequest` e a envia ao PI SAML2 correspondente (neste caso, o `IdP UFRGS`). A `SAMLResponse` retornada é convertida de volta para o formato

Figura 41 – Representação da topologia com suporte ao legado nos níveis 2 e 3.



Fonte: o autor.

OIDC e entregue ao WAYF, que normaliza os atributos antes de repassá-los à RP.

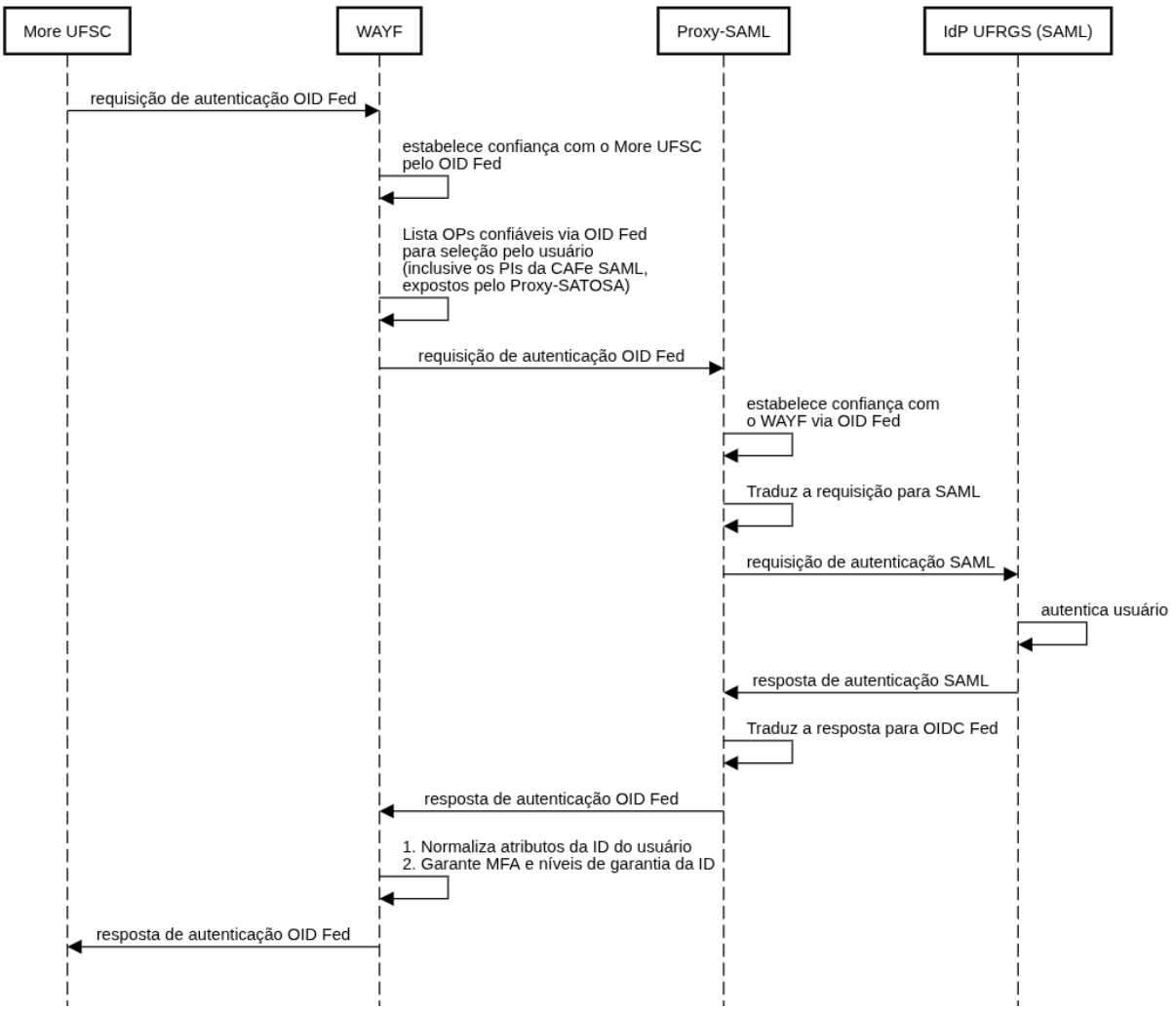
A Figura 43 detalha os *endpoints* expostos pelo *proxy*, considerando o nível 2 de compatibilidade. O `Frontend OID Fed` expõe, sob um prefixo próprio conforme IdP SAML2 (`/op-ufsc-saml`), os *endpoints* que fazem esse IdP ser visto como um OP na federação OID Fed.

Os *endpoints* de exposição da EC (`/op-ufsc-saml/.well-known/openid-federation`), de autenticação (`/op-ufsc-saml/auth`), *token* (`/op-ufsc-saml/token`), e *userinfo* (`/op-ufsc-saml/userinfo`) são expostos pelo Proxy-SATOSA. Cada PI legado descoberto no arquivo de metadados recebe o seu próprio conjunto prefixado, de modo a tornar-se uma entidade distinta e endereçável na federação. Isso possibilita que o WAYF, ao realizar a descoberta de OPs, liste os IdPs SAML2 de forma transparente. Voltado ao PI SAML2, o `Backend SAML` do Proxy-SAML expõe o *endpoint* de consumo de asserções (`/acs/post`), no qual recebe a `SAMLResponse` emitida pelo PI após a autenticação do usuário.

No nível 3 de suporte, o cenário inverso é atendido pela configuração oposta do mesmo Proxy-SAML, agora com um *frontend* SAML2 e um *backend* OID Fed. Nesse modo, o *proxy* atua como PI na federação SAML2 e como RP na nova federação, valendo-se da mesma topologia de federação ilustrada na Figura 41. Assim, PSs legados podem autenticar usuários em OPs integrados ao BAITA de forma transparente.

A Figura 44 detalha o fluxo de autenticação deste nível. O processo inicia quando um PS SAML2 legado envia uma `SAMLRequest` ao *Frontend SAML* do Proxy-SAML, que se

Figura 42 – Fluxo de autenticação com PI legado no nível 2 de compatibilidade.

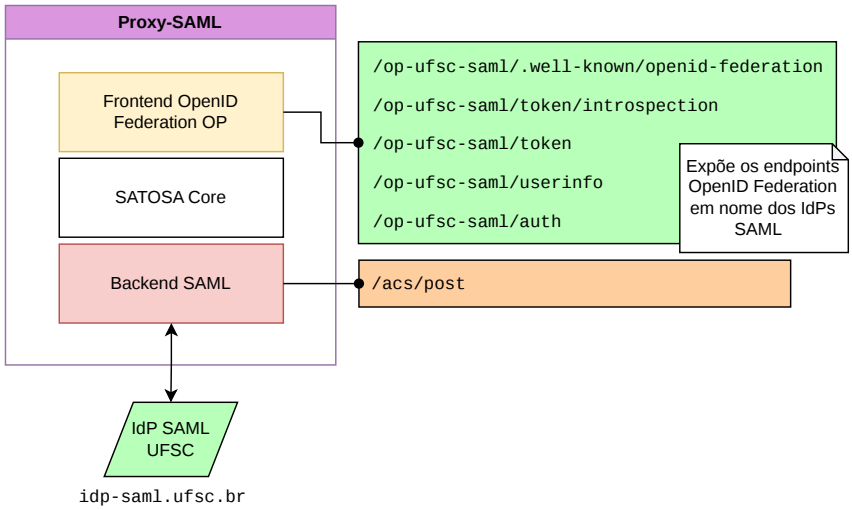


Fonte: o autor.

apresenta a ele como o seu PI. O *proxy* traduz essa requisição em uma requisição de autenticação OID Fed e, por meio do seu *Backend OID Fed*, atuando como RP, encaminha-a ao WAYF. Ao recebê-la, o WAYF estabelece a confiança com o Proxy-SAML resolvendo a sua TC no OID Fed e, em seguida, apresenta ao usuário a lista de OPs confiáveis para seleção de sua instituição de origem. Como explicado anteriormente, essa lista inclui também os PIs da CAFe SAML2 expostos pelo próprio Proxy-SAML no nível 2, de modo que os dois níveis de compatibilidade se compõem de forma transparente.

Após a seleção do provedor de origem, o WAYF encaminha a requisição de autenticação OID Fed ao OP escolhido (nesse caso, um OP OID Fed). Esse OP, por sua vez, estabelece a confiança com o WAYF via OID Fed, autentica o usuário e retorna a resposta de autenticação ao WAYF. Antes de propagar o resultado, o WAYF desempenha o seu papel de *proxy* de identidade, descrito na Seção 7.5. Ele normaliza os atributos da identidade do usuário segundo os esquemas adotados pela CAFe e garante o nível de garantia e o MFA exigidos. A resposta normalizada é então devolvida ao Proxy-SAML, que a traduz para uma `SAMLResponse` e a entrega

Figura 43 – *Endpoints* expostos pelo Proxy-SAML no nível 2 de compatibilidade.



Fonte: o autor.

ao PS legado. Do ponto de vista do PS, todo o processo é indistinguível de uma autenticação SAML2 convencional com um PI *Shibboleth*, sem que ele perceba que a identidade foi, de fato, fornecida por um OP moderno da nova federação.

No que tange aos *endpoints*, o nível 3 inverte a configuração do nível 2. O *Frontend SAML* assume o papel de PI, expondo os *endpoints* de *Single Sign-On* do SAML2, de redirecionamento *HTTP-Redirect* (`/sso/redirect`) e para o retorno da resposta de autenticação *HTTP-POST* (`/sso/post`). Além disso, expõe o *endpoint* de metadados SAML2 (`/metadata`), pelos quais o PS legado inicia a autenticação.

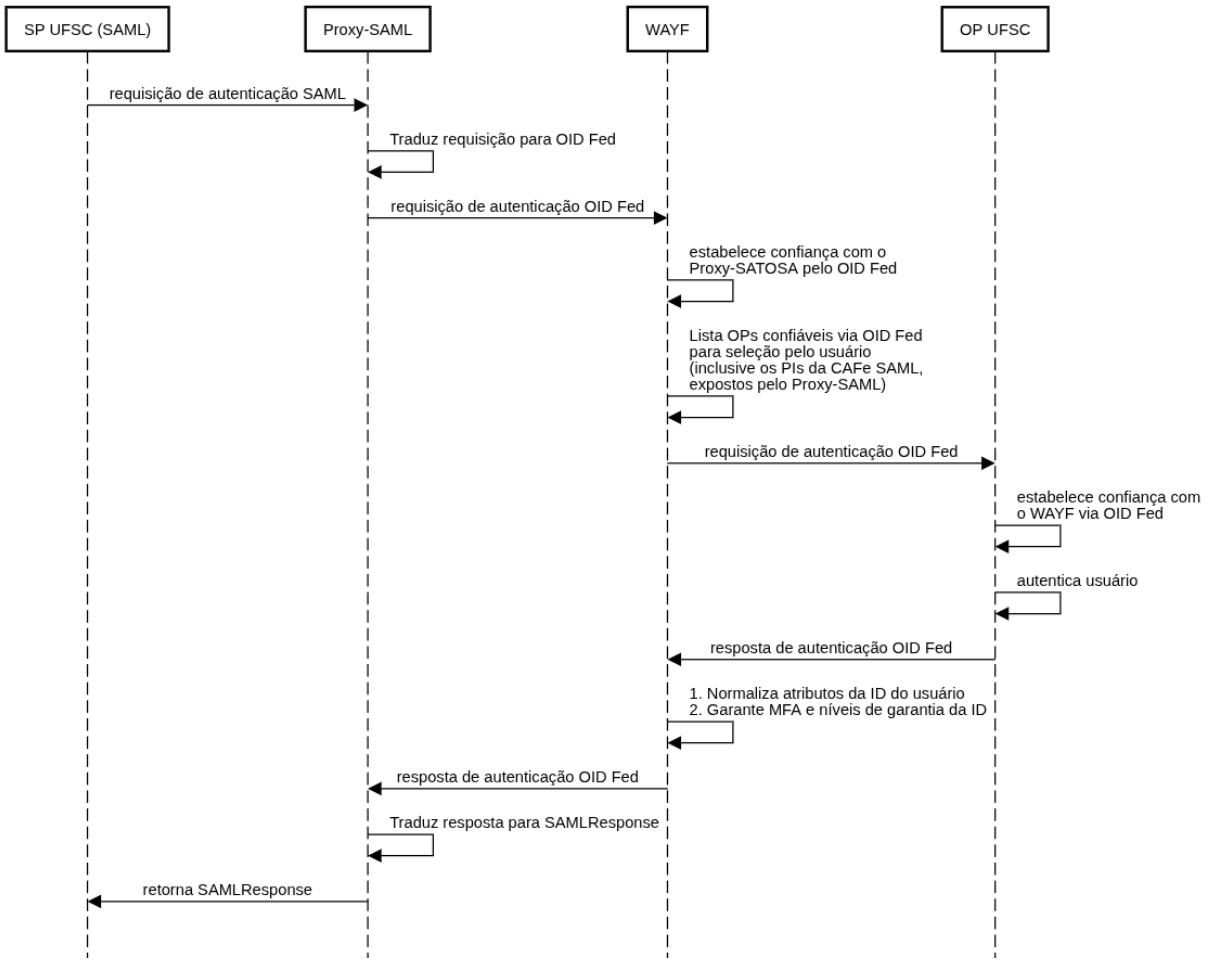
Diferentemente do nível 2, esses *endpoints* não são prefixados. O *proxy* expõe uma única face de PI SAML2 para todos os PSs legados, que se identificam pelo seu *entity ID* SAML2 no momento da requisição. Voltado à nova federação, o *Backend OID Fed* atua como uma RP, expondo o seu *endpoint* de *callback* (`/callback`), onde recebe o código de autorização, e a sua própria EC (`/.well-known/openid-federation`). A Figura 45 detalha esses *endpoints*.

Em ambos os níveis 2 e 3, o Proxy-SAML aplica um módulo de mapeamento de atributos que traduz os atributos SAML2, expressos no formato `urn:oid:` ou como *friendly names* para as *claims* OIDC correspondentes e vice-versa, seguindo as semânticas definidas pelos esquemas *eduPerson* (Refeds, 2020) e *brEduPerson* (Cafe, 2020). Essa normalização garante que a identidade do usuário seja interpretada corretamente independentemente do protocolo de origem.

7.7 PORTAL DE BENEFÍCIOS

O PoB é o componente do barramento voltado ao usuário final. Seu objetivo é aumentar a visibilidade dos serviços disponíveis na federação, resolvendo problemas identificados na

Figura 44 – Fluxo de autenticação com PS legado no nível 3 de compatibilidade.



Fonte: o autor.

prospecção do GT-BAITA. O PoB posiciona o BAITA como um ecossistema de descoberta e acesso a serviços acadêmicos. A Figura 46 apresenta o diagrama de casos de uso do PoB.

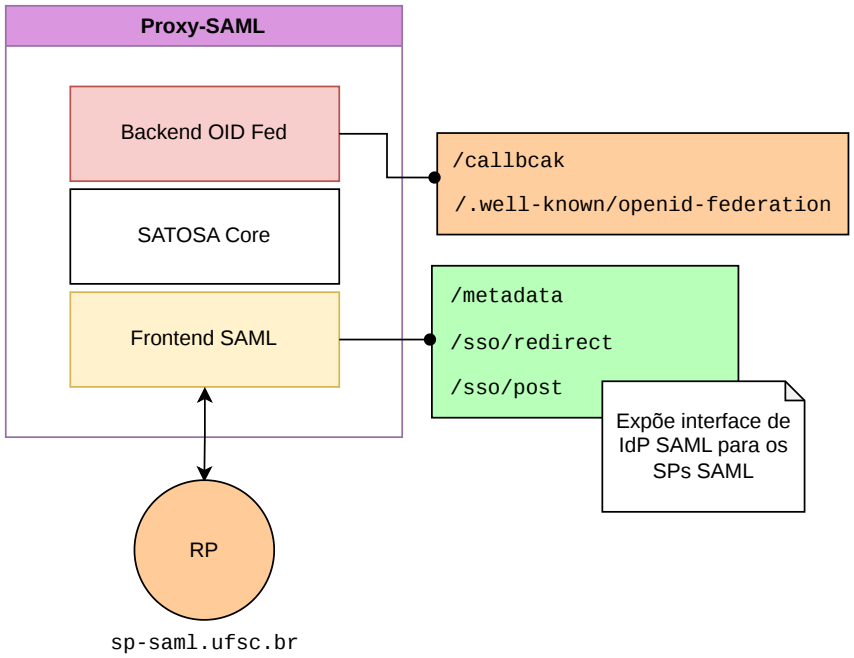
Conforme ilustrado, o usuário pode explorar o catálogo de serviços, navegando livremente pelos PSs e filtrar os serviços por instituição, restringindo a listagem ao escopo da sua organização de origem, por exemplo. Ambas as ações são públicas e não exigem autenticação. O usuário pode ainda acessar uma aplicação, sendo direcionado do portal a um PS do barramento.

Esse caso de uso inclui o de realizar *login* federado. Para utilizar o serviço, o usuário autentica-se por meio do seu PI de origem, em um fluxo que passa pelo WAYF. O *login* federado também pode ser realizado diretamente no portal, habilitando a experiência personalizada proposta na Seção 9.1, na qual o catálogo passa a refletir a afiliação do usuário.

Antes de detalhar a sua implementação, convém situar o PoB na topologia da federação. A Figura 47 mostra que o PoB é uma RP, subordinada direta da TA da CAFe. A partir dela, o PoB resolve TCs válidas até qualquer provedor público, como o `More UFSC` ou o `Serviço UFMG`.

O PoB é implementado como uma aplicação de duas camadas. Um *frontend React* e

Figura 45 – *Endpoints* expostos pelo Proxy-SAML no nível 3 de compatibilidade.



Fonte: o autor.

um *backend Flask* conectados por uma API. A Figura 48 ilustra essa arquitetura.

A interface *web* do PoB é construída com *React*, *TypeScript* e *Shadcn* e organiza-se em três páginas principais: a `HomePage`, que exibe o carrossel dos serviços mais acessados e as categorias temáticas; a `Página de Benefícios`, que oferece busca textual, filtragem por palavras-chave e ordenação (por popularidade ou data); e a `Página de detalhes do serviço`, que apresenta os detalhes completos de um serviço.

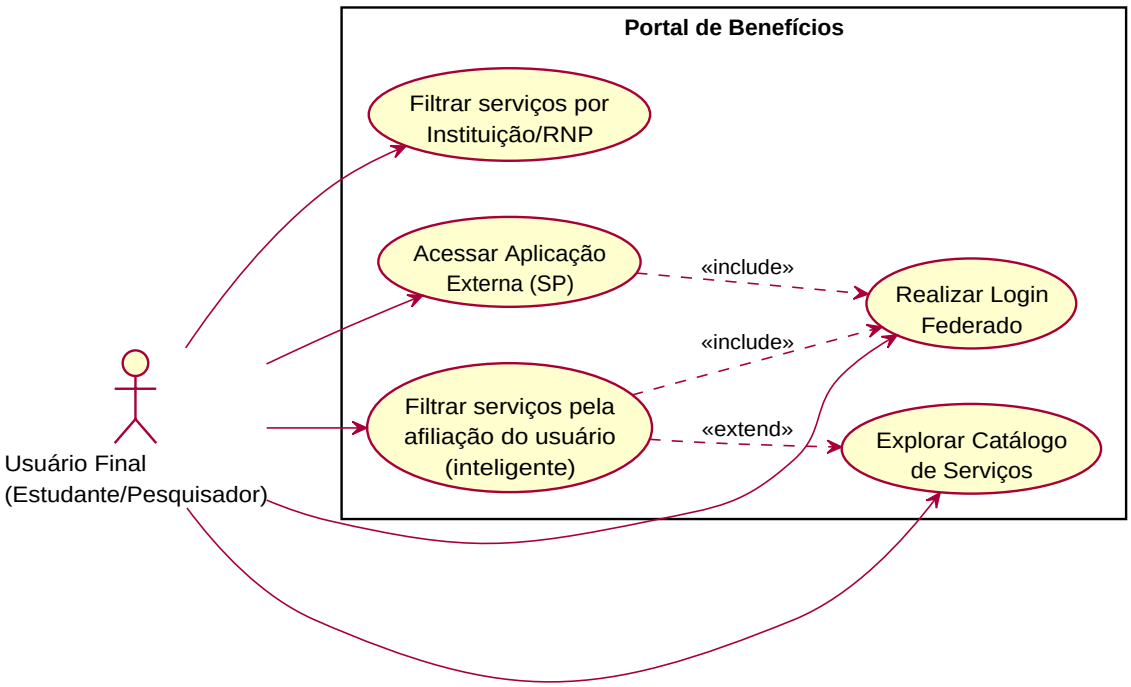
A camada de API, implementada em *Flask* e servida via *Gunicorn*, recebe as requisições da interface *web* e as encaminha aos motores do *backend*. Ela organiza-se em três grupos de rotas (`Autenticação`, `Busca e filtros` e `Observabilidade`), que espelham esses motores.

O motor de busca é o núcleo funcional do PoB e reúne duas responsabilidades. A primeira é manter um catálogo atualizado e dinâmico dos serviços do barramento. Para isso, executa uma descoberta *top-down*, análoga à realizada pelo WAYF (Seção 7.5.1), porém orientado a RPs. Ele percorre a topologia da federação em uma BFS, a partir das TAs configuradas, e seleciona as entidades cujo metadado `entity_type` é `openid_relying_party`.

Para compor cada item do catálogo, o motor de busca reúne metadados de fontes complementares. Do próprio algoritmo de descoberta, obtém as *claims* publicadas nas ECs das RPs, como `display_name`, `logo_uri`, `policy_uri`, `information_uri` e `keywords` (Hedberg et al., 2026). Para RPs subordinadas ao `Proxy-OID-Fed`, esses metadados são informados no ato do registro e incluídos nas ECs geradas em seu nome. Do SGB, recupera informações adicionais cadastradas pelo responsável pelo serviço, como imagens ilustrativas.

Outra funcionalidade desse motor é realizar *busca e filtros* sobre o catálogo resultante,

Figura 46 – Casos de uso do PoB.



Fonte: o autor.

abrangendo a pesquisa textual, a filtragem por palavras-chave e a paginação. Para reduzir a latência, os resultados são mantidos em um *cache Redis*, evitando refazer a varredura na hierarquia da federação a cada requisição.

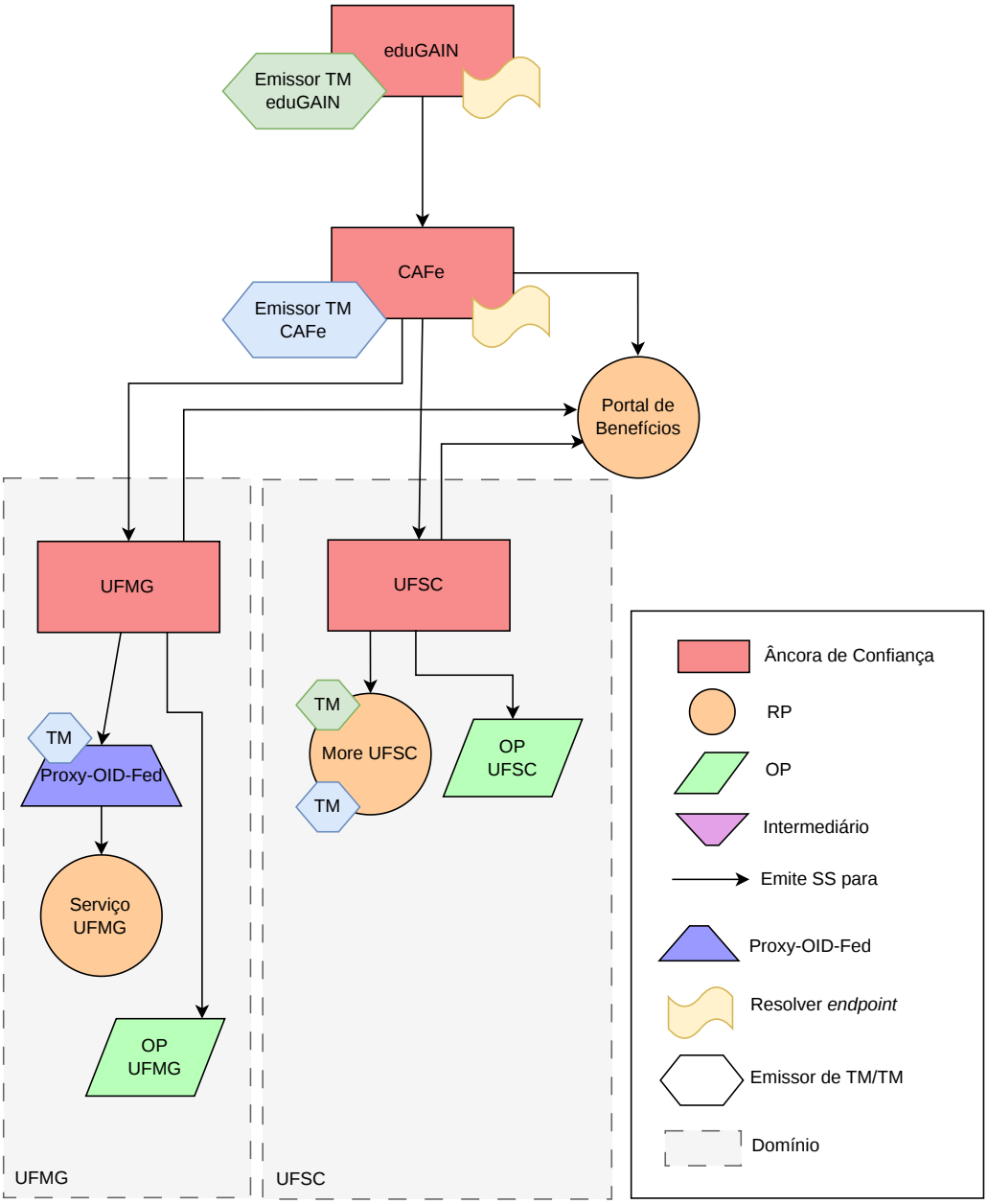
O motor autenticação implementa a autenticação do usuário final no portal e é a base da personalização proposta na Seção 9.1. Compreende a `gestão de usuários` e o `controle de sessão`, cujas sessões ativas são mantidas no mesmo *cache Redis*. Como detalhado naquela Seção, essa funcionalidade ainda não foi implementada, mas está prevista na arquitetura para habilitar a filtragem de serviços de acordo com a afiliação do usuário. Essa implementação depende da funcionalidade de gestão de subfederações, ainda não suportada pelo BAITA.

O motor de observabilidade é responsável pela `análise de acesso aos serviços`. As métricas relacionadas aos serviços catalogados pelo PoB podem servir como substrato para ações estratégicas dos responsáveis do serviço, para definir ações de evolução do serviço e medir o seu impacto na comunidade acadêmica.

O estado atual da implementação, como prova de conceito, metrifica o número de acesso ao serviço (*i.e.*, quantas vezes o usuário acessou o serviço pela interface do PoB). Cada clique em um serviço incrementa essa métrica, que é etiquetada pelo `entity_id` do SP e exportada para a `stack de observabilidade`. O *ranking* dos serviços mais acessados, exibido no carrossel da `HomePage`, é calculado em tempo real consultando o *Prometheus*.

Quanto à persistência, o catálogo não é armazenado de forma permanente, sendo reconstruído sob demanda a partir da descoberta na federação. A descoberta é executada pelo

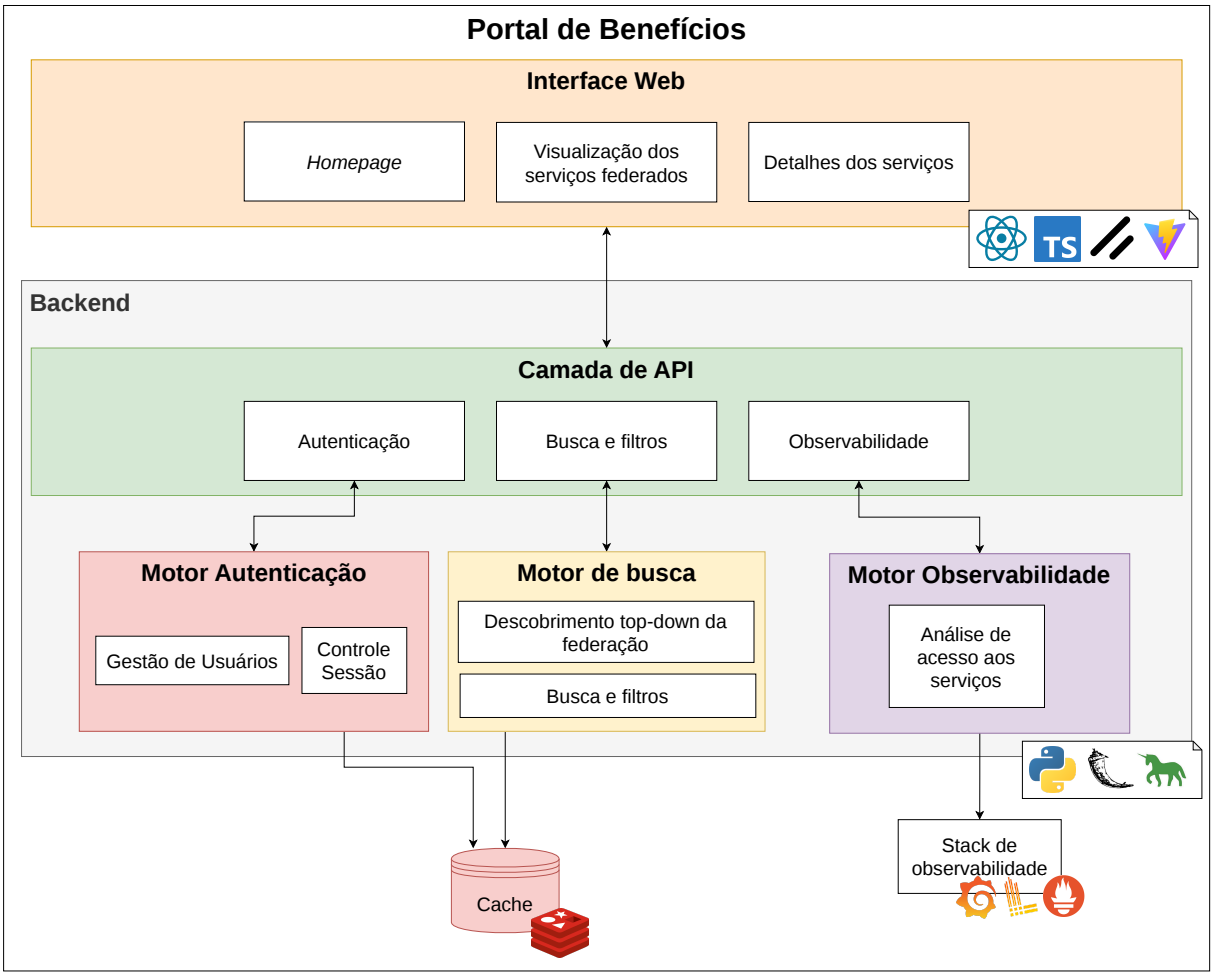
Figura 47 – Posicionamento do PoB na topologia da federação.



Fonte: o autor.

SGB e consultada pelo PoB por meio de requisição de API HTTP. Os dois sistemas externos utilizados pelo *backend* do PoB, ambos compartilhados com o restante do barramento e ilustrados na Figura 48, são (i) o *cache Redis*, que guarda os resultados da descoberta na federação e as sessões de usuário e (ii) a `stack de observabilidade`, que concentra as métricas de uso. A atual implementação do PoB é *stateless*. Uma evolução para um PoB *stateful*, que persista o histórico e as preferências do usuário para sustentar um sistema de recomendação de serviços, é tratada como trabalho futuro na Seção 9.1.

Figura 48 – Arquitetura interna do *Portal de Benefícios*.



Fonte: o autor.

8 RESULTADOS E DISCUSSÕES

O presente capítulo apresenta os resultados obtidos ao longo do desenvolvimento do BAITA, e discute as decisões arquiteturais e de implementação, bem como os *trade-offs* delas decorrentes. O principal resultado é um protótipo funcional do barramento, que materializa os componentes propostos. Foram implementados o SGB, o PoB, o `Proxy-OID-Fed`, o `Proxy-SAML` e o WAYF, além de uma camada de observabilidade. Nesse protótipo, o fluxo completo de autenticação de uma RP OIDC através do WAYF até um PI OIDC, ambos intermediados pelo `Proxy-OID-Fed` foi exercitado de ponta a ponta.

O barramento encontra-se implantado em um ambiente de homologação na infraestrutura de *testbeds*[1] da RNP, acessível publicamente. Esse ambiente permite a validação contínua das funcionalidades em condições próximas às de produção e serve de base empírica para as discussões deste capítulo.

Cabe ressaltar que o BAITA é uma iniciativa em evolução. Parte das funcionalidades projetadas na Seção 7 permanece em estágio de projeto, como a gestão de subfederações, o filtro inteligente do PoB e o suporte completo ao nível 3 de compatibilidade com o SAML2, conforme registrado entre os trabalhos futuros na Seção 9.1. Algumas delas, encontram-se em fase inicial de implementação.

8.1 SISTEMA DE GESTÃO DO BARRAMENTO

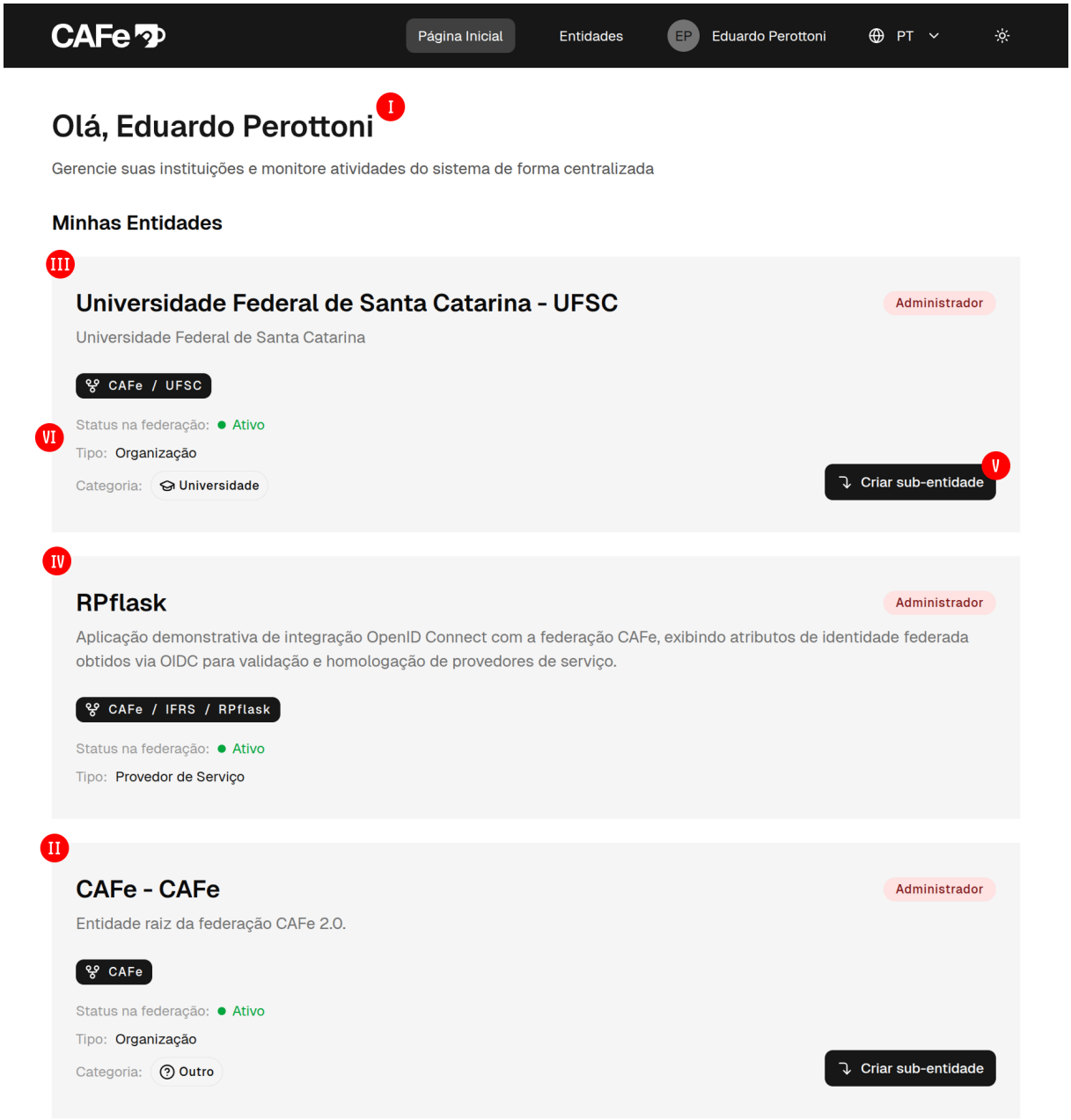
O SGB está acessível pelo ambiente de homologação[2]. É por meio dele que os operadores da RNP e os administradores das instituições gerenciam o ciclo de vida das entidades do barramento.

Após a autenticação do usuário no SGB, o mesmo é direcionado à página inicial, apresentada na Figura 49. Essa tela lista as entidades relacionadas com o usuário, apontando seu nível de permissão sobre essas entidades (`Administrador`, `Editor` ou `Leitor`). A partir desta tela, o usuário acessa diretamente os detalhes da entidade, a criação de novos provedores ou a visualização da topologia.

No exemplo da Figura 49, o usuário autenticado é `Eduardo Perottoni` (I), que possui o papel de `Administrador` sobre três entidades distintas: a primeira é a `CAFe`, entidade raiz da federação `CAFe 2.0` (II); a segunda é a `Universidade Federal de Santa Catarina (UFSC)`, uma organização da categoria *Universidade* (III); a terceira é a `RPflask`, uma RP de homologação subordinado ao `IFRS` (IV). Para as duas organizações, `CAFe` e `UFSC`, a interface oferece o atalho `Criar sub-entidade` (V), pois organizações podem agregar subordinados. Já para a `RPflask`, uma entidade folha, esse atalho não é exibido. Cada *card* apresenta ainda a posição da entidade na hierarquia da federação, o seu *status* e a sua categoria (VI).

[1] https://rnpmais.rnp.br/testbeds
[2] https://admin.baita.testbeds.rnp.br

Figura 49 – Página inicial do SGB.



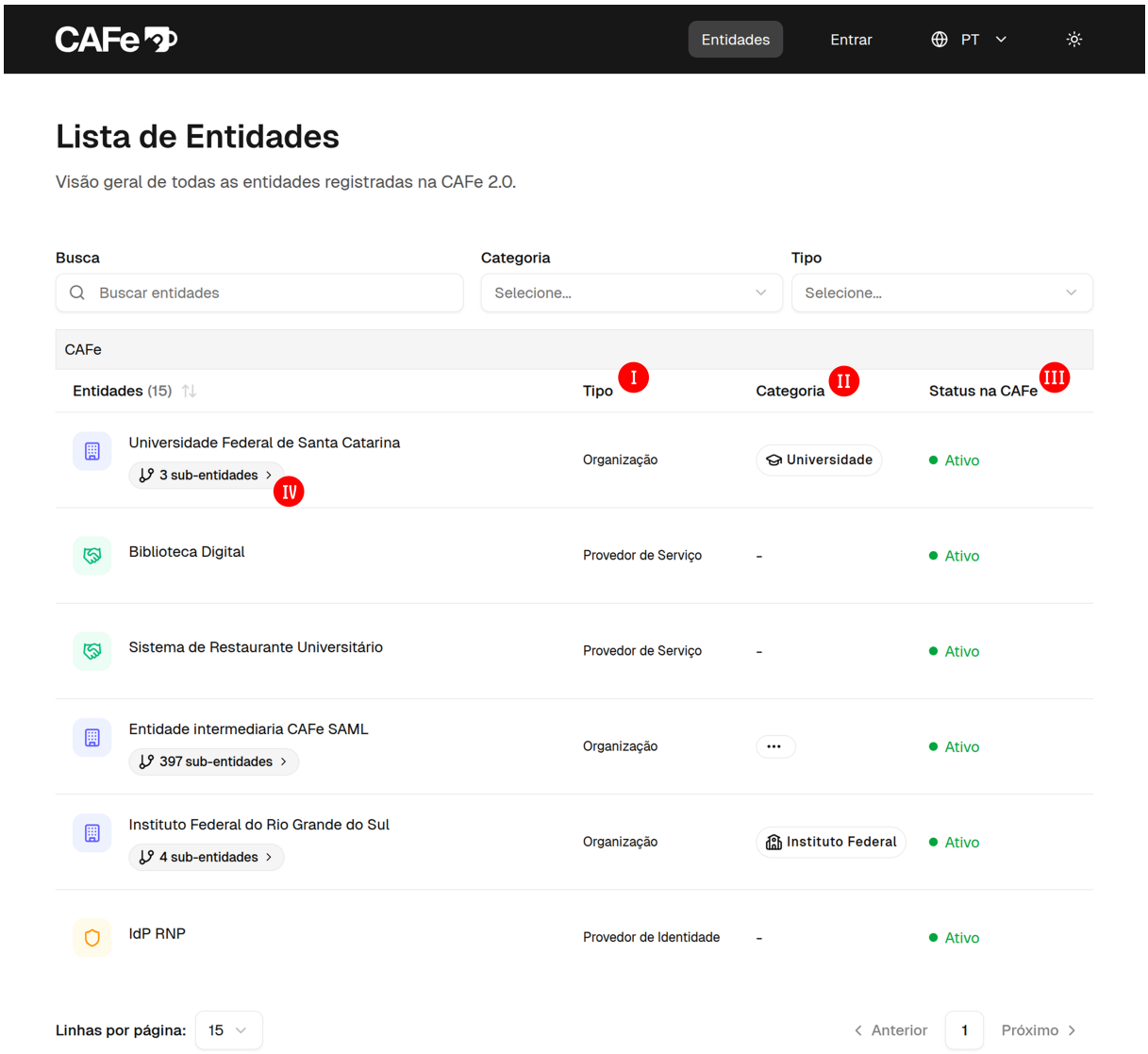
Fonte: o autor.

A tela central do SGB é a Lista de Entidades, apresentada na Figura 50. Ela exibe a topologia das entidades registradas na `CAFe 2.0` com o seu tipo (`Organização`, `Provedor de Serviço` ou `Provedor de Identidade`) (I), a sua categoria (II), o seu *status* na CAFe (III), os administradores responsáveis e o número de subentidades (IV). A tabela de entidades permite navegação na hierarquia da federação, ao clicar no botão que apresenta a contagem de subentidades.

Uma barra de filtros, no topo, permite buscar e restringir a listagem por categoria, tipo e administrador. Como se observa na Figura 50, as organizações, a exemplo da `Universidade`

Federal de Santa Catarina, agregam subentidades, ao passo que os provedores aparecem como entradas individuais. Destaca-se ainda a Entidade intermediária CAFe SAML, que concentra as centenas de provedores legados expostos pelo Proxy-SAML.

Figura 50 – Tela de Lista de Entidades do SGB.



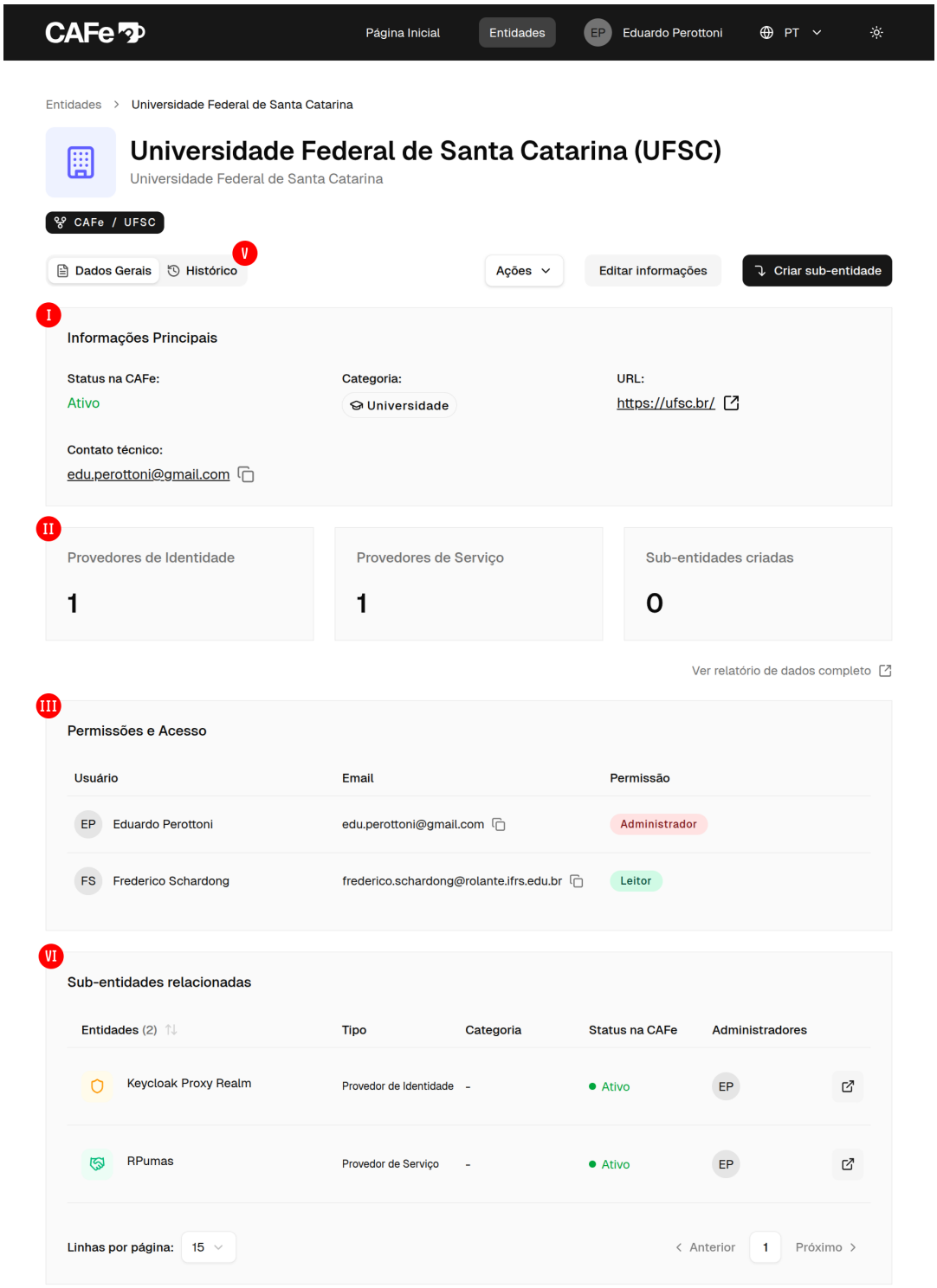
Fonte: o autor.

Cada entidade possui uma página de detalhes, ilustrada na Figura 51 para a organização Universidade Federal de Santa Catarina (UFSC). A página reúne as informações principais (I) da entidade, *status* na CAFe, categoria, URL e contato técnico. Um resumo quantitativo dos provedores e subentidades sob a sua árvore (II), a seção de Permissões e Acesso (III) e a lista de Subentidades relacionadas (IV).

No exemplo, a UFSC é uma organização ativa da categoria *Universidade*, com um provedor de identidade e um provedor de serviço sob a sua responsabilidade. A seção de Permissões e Acesso indica que Eduardo Perottoni é Administrador da organização, ao passo que Frederico Schardong possui apenas permissão de Leitor. A lista de Subentidades

`relacionadas`, por sua vez, apresenta os provedores subordinados a partir dos quais o administrador pode navegar para cada um. É também nessa tela que o administrador gerencia quem pode operar a entidade e cria novas subentidades, por meio do atalho `Criar sub-entidade`.

Figura 51 – Página de detalhes de uma organização (`UFSC`) no SGB.



Fonte: o autor.

A página de detalhes contém a aba `Histórico` da entidade (V), apresentada na Fi-

gura 52, que registra a trilha de auditoria de todas as alterações realizadas sobre a entidade. Cada evento exibe a data e a hora, o tipo de ação, como `Edição de metadados`, `Criação de Usuário` ou `Exclusão de Usuário`, uma descrição da alteração e o usuário responsável, acompanhado do seu papel. Os eventos de edição podem ser expandidos para revelar os detalhes da mudança.

No exemplo, o histórico da `UFSC` registra duas edições dos metadados da entidade (I e II) e a movimentação do usuário `frederico.schardong` entre os grupos de permissão, adicionado ao grupo de administradores e removido do grupo leitores (III e IV). Todas as modificações foram atribuídas a `Eduardo Perottoni`. Esse registro confere rastreabilidade e responsabilização às operações de gestão, requisito essencial em uma federação de larga escala.

O SGB disponibiliza ainda uma tela de configurações de perfil, apresentada na Figura 53 e acessível a partir do menu do usuário no cabeçalho da interface. Nela, o usuário edita os seus dados básicos (I) e gerencia os seus métodos de autenticação na seção de Segurança (II). São suportados três fatores: senha, *passkey* (FIDO2/WebAuthn) e *OTP* via *QR Code* (TOTP), cada qual com o seu estado atual e a ação correspondente para defini-lo ou redefini-lo. No exemplo, o usuário `Eduardo Perottoni` mantém a sua senha, alterada há cinco dias, e ainda não configurou a *passkey* nem o *OTP*.

A implementação materializa o modelo de administração descentralizada discutido na Seção 7.3. O operador de uma organização superior delega a gestão de uma sub-entidade a um administrador responsável, que se cadastra no SGB e passa a gerir as suas próprias entidades, sem intervenção do operador. O Apêndice A percorre esse processo de ponta a ponta, desde a delegação do domínio até a ativação de um OP OIDC no BAITA. De maneira análoga, o cadastro de um PS OIDC no barramento, incluindo a sua publicação automática no PoB, é ilustrado, passo a passo, no Apêndice B.

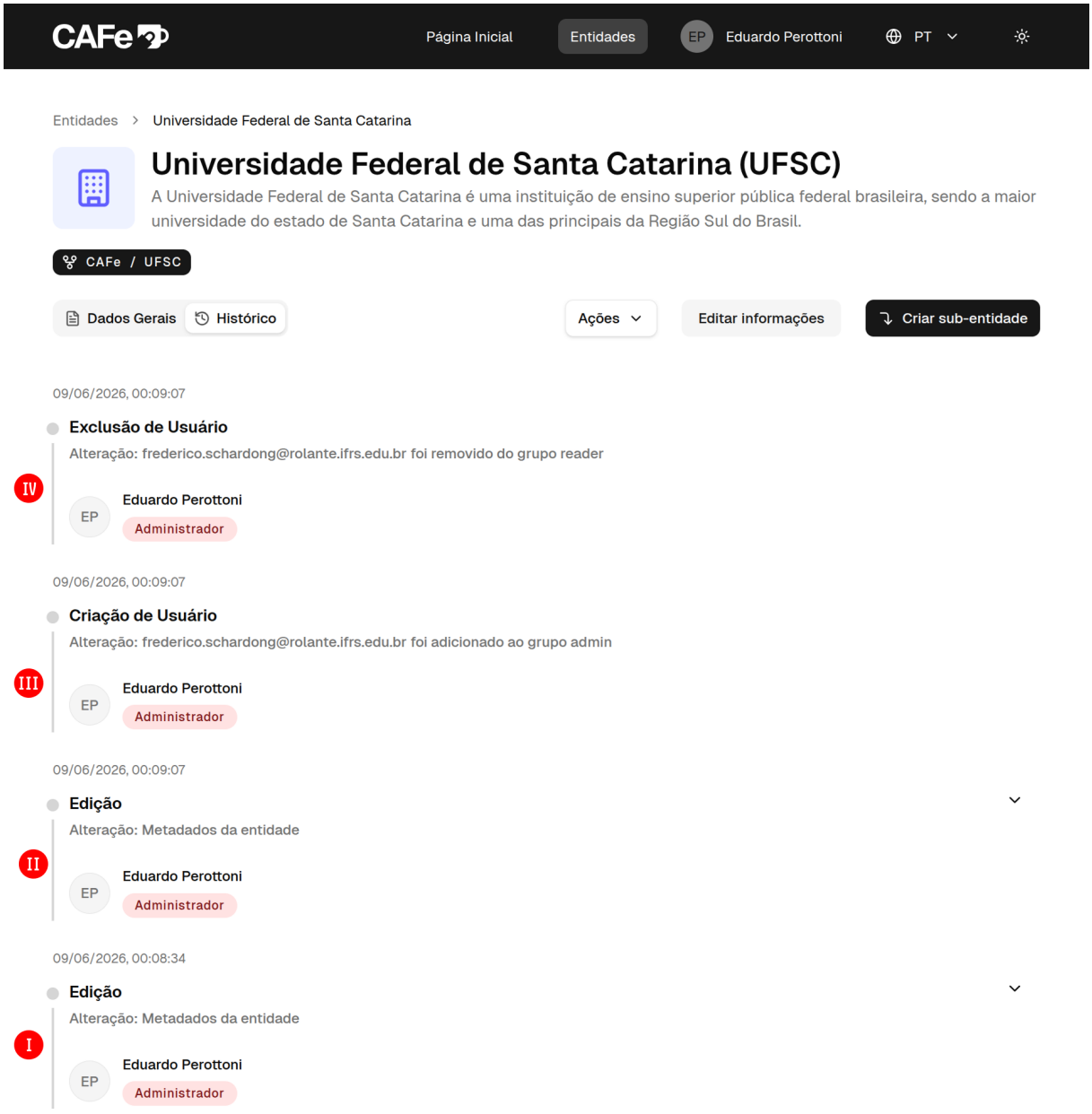
O protótipo do SGB não cobre a totalidade da arquitetura projetada no Capítulo 7. A lacuna mais significativa é a gestão de subfederações (Seção 7.2.4). A separação entre entidades públicas e privadas por meio de TAs distintas não foi implementada, de modo que todas as entidades registradas são, na prática, públicas.

Em se tratando da Seção 7.6.1, o suporte ao legado SAML2 restringe-se ao nível 2 de compatibilidade. O `Proxy-SAML` implementado lê o arquivo de metadados da CAFe e expõe cada PI SAML2 como um OP na nova federação, atendendo a RPs modernas que autenticam usuários em PIs legados.

De forma semelhante, as políticas de metadados a serem impostas pela TA da CAFe ainda não foram instanciadas, aguardando a consolidação do Perfil Técnico da eduGAIN. A evolução dessas lacunas é tratada entre os trabalhos futuros (Seção 9.1).

As funcionalidades apresentadas evidenciam a centralização da gestão da federação na infraestrutura da RNP. O SGB não apenas concentra o registro e o ciclo de vida das entidades hospedadas, mas também custodia as suas chaves de federação e assina, em nome das entidades, os artefatos do OID Fed. Vale lembrar que o protocolo exige que cada entidade mantenha chaves criptográficas próprias para a federação, distintas das empregadas na assinatura

Figura 52 – Aba de *Histórico* de uma entidade no SGB.

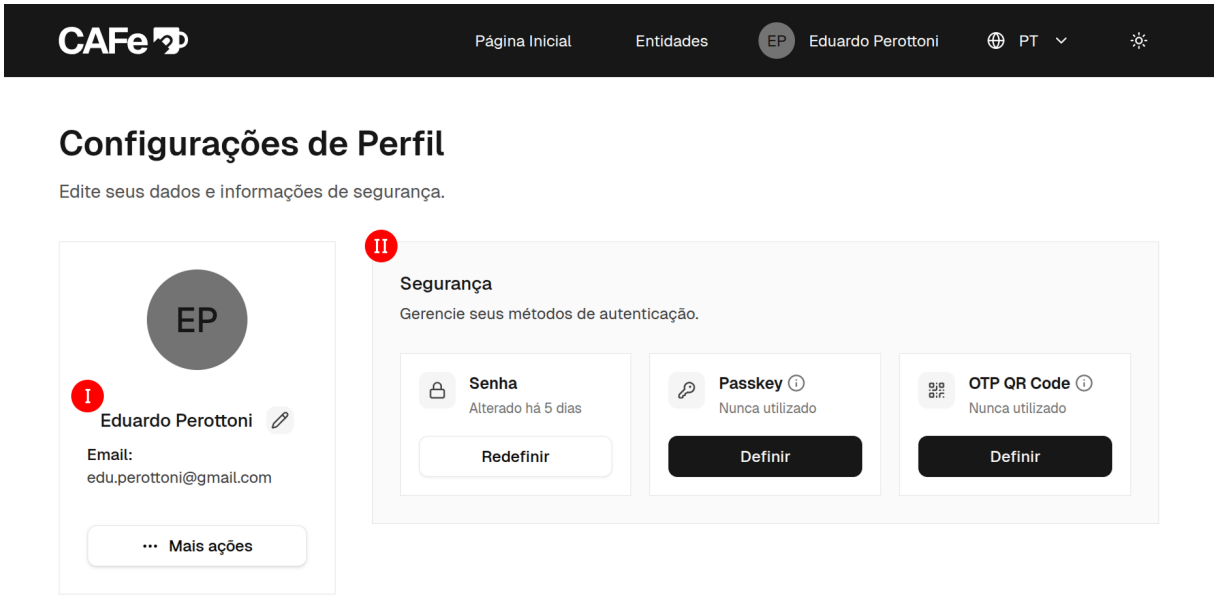


Fonte: o autor.

de *tokens* OIDC (Hedberg et al., 2026), que, no modelo hospedado, o SGB gera, custodia e rotaciona automaticamente em nome dos provedores. Essa concentração tem duas implicações relevantes. Do ponto de vista da disponibilidade, o SGB torna-se um ponto único de falha. A sua indisponibilidade compromete a gestão e a renovação dos artefatos de todas as entidades por ele administradas. Do ponto de vista da segurança, o comprometimento do SGB ou das chaves que ele custodia impactaria todos os provedores hospedados, e não apenas um deles.

Em contrapartida, é justamente essa centralização que materializa o principal valor do barramento, a drástica redução do atrito para a adesão das instituições. Como ilustram os Apêndices A e B, integrar um provedor à federação resume-se a preencher alguns campos de

Figura 53 – Tela de configurações de perfil e segurança do usuário no SGB.



Fonte: o autor.

formulários guiados, em interface *web*, sem interações complexas. A instituição não precisa implementar as extensões do OID Fed, gerar e rotacionar chaves de federação e nem operar qualquer infraestrutura adicional que o faça. O SGB, em conjunto com os *proxies*, encarrega-se de tudo isso em seu nome. Esse resultado responde à carência de mão de obra técnica qualificada nas instituições para operar a infraestrutura de IDF, uma das dores mais recorrentes identificadas na fase de prospecção do GT-BAITA.

Configura-se, assim, um *trade-off* entre facilidade de adoção de um lado, e descentralização e soberania de outro. O BAITA opta deliberadamente pela premissa de redução de atrito, mas sem tornar a centralização compulsória. As instituições que demandem controle total sobre as suas chaves e a sua hierarquia podem implementar o OID Fed nativamente e ancorar a sua própria sub-árvore à federação, dispensando o *proxy*. Além disso, os riscos inerentes à centralização podem ser mitigados por medidas operacionais, como a execução em infraestrutura de alta disponibilidade, com redundância, e a proteção e a rotação periódica das chaves custodiadas.

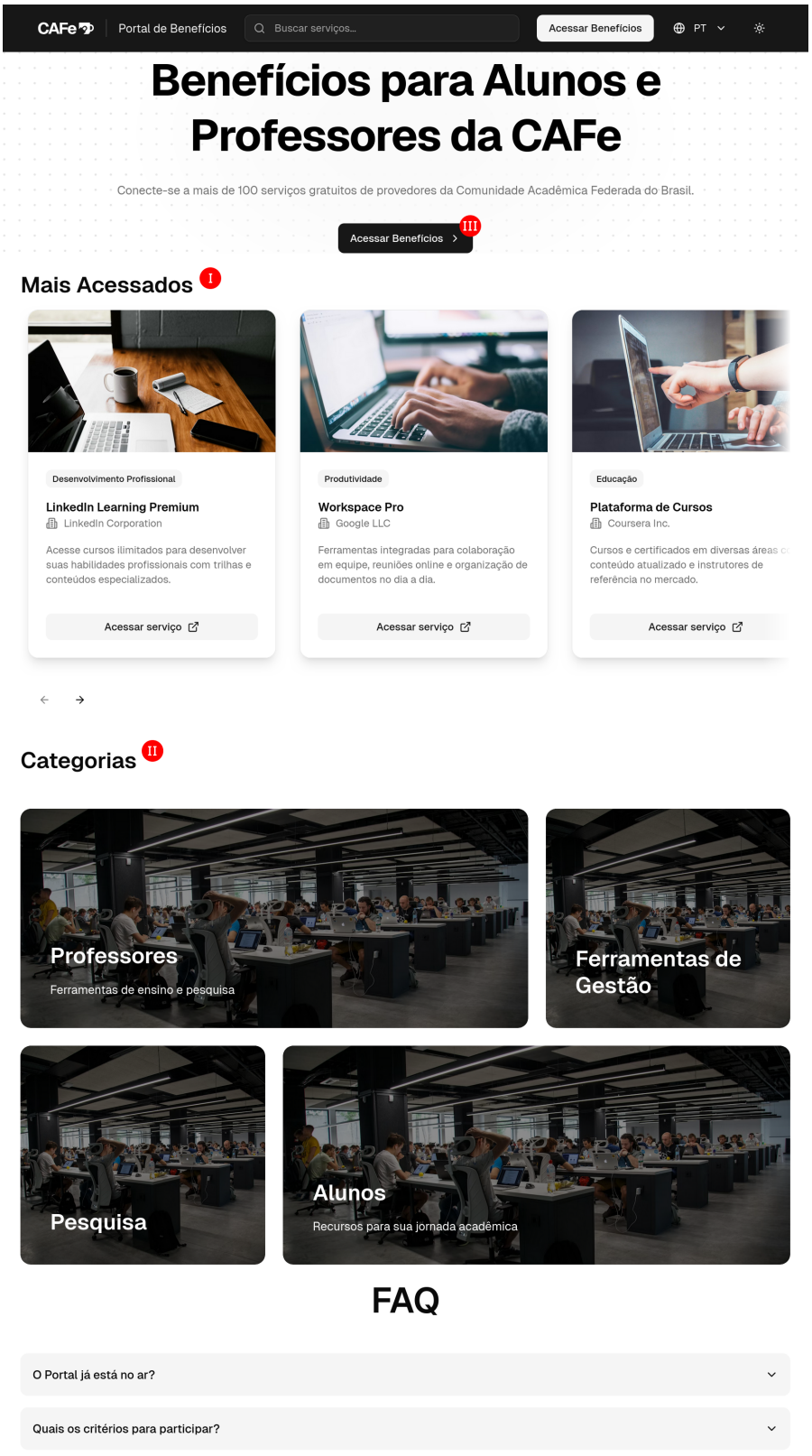
8.2 PORTAL DE BENEFÍCIOS

Como explicado, o PoB[3] é a porta de entrada do usuário final no ecossistema, funcionando como um catálogo de descoberta dos serviços disponíveis na federação. A página inicial do portal, apresentada na Figura 54, contém um carrossel dos serviços mais acessados (I) e um

[3] O PoB está acessível, no ambiente de homologação, em https://baita.testbeds.rnp.br ou https://servicos.baita.testbeds.rnp.br

conjunto com algumas das categorias temáticas que organizam o catálogo (II). O *ranking* exibido no carrossel é calculado em tempo real a partir da métrica de cliques coletada pela camada de observabilidade, como citado nas Seção 7.7, refletindo o uso real da comunidade.

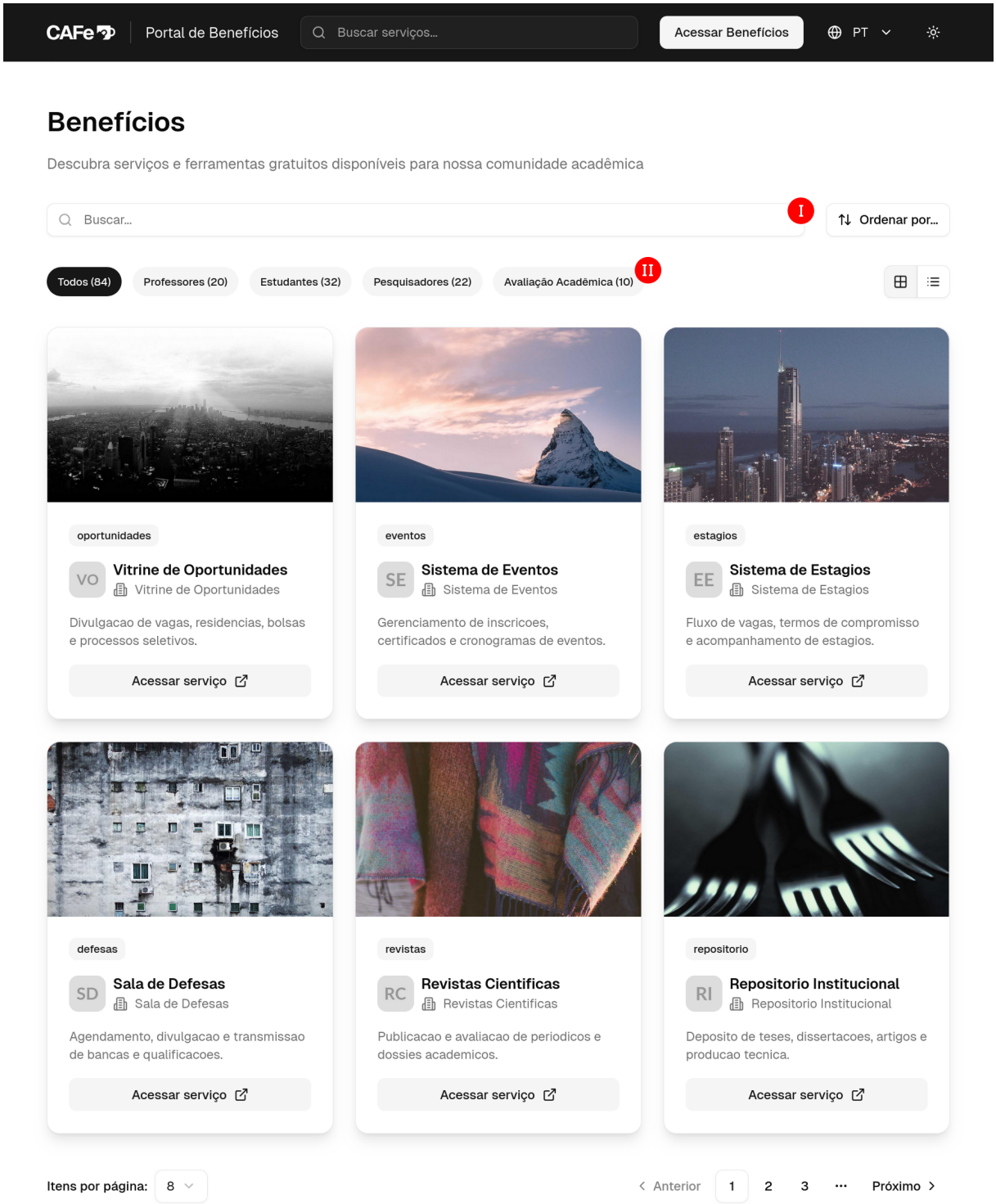
Figura 54 – Página inicial do PoB.



Fonte: o autor.

A partir dela, o usuário acessa a `Página de Benefícios` (III), apresentada na Figura 55. Esta página apresenta o catálogo completo como uma grade de *cards*, cada um identificando um serviço pelo seu nome, categoria e descrição resumida. Uma barra de busca textual (I) e filtros por palavra-chave (II) permitem refinar a listagem, enquanto a ordenação por nome ou por data de inclusão auxilia a navegação.

Figura 55 – Página de Benefícios do PoB, com busca e filtros sobre o catálogo.



Fonte: o autor.

O catálogo não é mantido de forma estática. Como consequência, todo PS registrado no barramento passa a ser automaticamente visitado pelo algoritmo de busca e exibido no portal, sem qualquer cadastro manual adicional. O Apêndice B percorre esse encadeamento de ponta a ponta, do registro de um serviço no SGB até a sua publicação no PoB.

A página de detalhes de um serviço, apresentada na Figura 56 para o serviço `RPflask`, uma aplicação para testes do barramento, reúne a descrição completa (I), as imagens, a categoria (II), as políticas de uso e os termos de serviço, a data de inclusão e o contato de suporte (III), uma lista de serviços relacionados (IV) e o botão `Acessar serviço`. Para os serviços de acesso restrito, geralmente esse botão desencadeia a autenticação federada, que passa pelo WAYF. No exemplo, o `RPflask`, é apresentado na categoria `Ensino & Aprendizagem`, tendo o *Ambiente Virtual de Aprendizagem* entre os serviços relacionados sugeridos ao usuário.

O PoB opera sobre a federação com suas TAs públicas. A autenticação do usuário final no portal e a consequente personalização do catálogo por afiliação (funcionalidades explicadas na Seção 9.1) não foram implementadas, pois o componente ainda não dispõe de uma camada de autenticação. Pela mesma razão, funcionalidades dependentes de estado, como um sistema de recomendação de serviços, também não foram desenvolvidas, permanecendo como trabalho futuro.

8.3 FLUXO DE AUTENTICAÇÃO COM WAYF E *PROXIES*

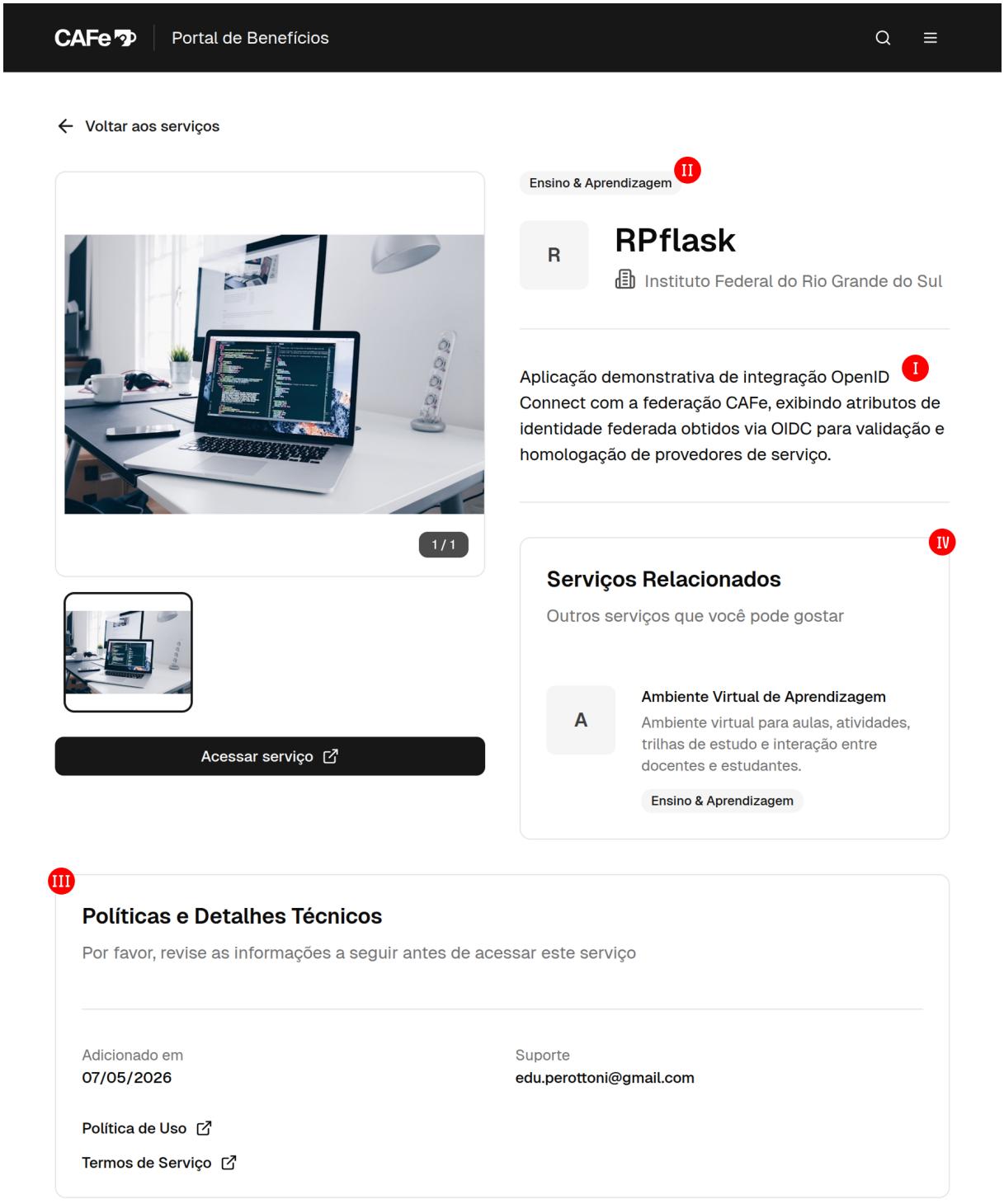
Para validar o fluxo de autenticação federada de ponta a ponta, apresenta-se o acesso à `RPflask`, aplicação apresentada na Seção 8.2. O fluxo é executado considerando dois IdPs distintos, ambos mediados pelo WAYF. O primeiro é um provedor SAML2, pertencente à *Chimarrão* (a federação SAML2 de homologação da CAFe) com a comunicação intermediada pelo Proxy-SAML. O segundo é um OP OIDC de teste registrado no barramento, o `KPR` (*Keycloak Proxy Realm*) com a comunicação intermediada pelo `Proxy-OID-Fed`. O ponto de partida, comum a ambos os caminhos, é a página inicial da `RPflask`, exibida na Figura 57.

Após o usuário acionar o botão de `login`, a `RPflask`, por ser uma RP integrada ao `Proxy-OID-Fed`, delega a este o papel de seu OP. O `Proxy-OID-Fed` redireciona o usuário ao WAYF, que apresenta a tela de descoberta de instituição. Nessa tela, os OPs OIDC registrados e os provedores SAML2 legados convivem de forma homogênea, conforme previsto na Seção 7.5.1.

A Figura 58 mostra a busca pela `UFSC` e a sua seleção sob a entidade intermediária `CAFe SAML`, enquanto a Figura 59 mostra a busca pelo OP `KPR`, integrado previamente ao BAITA. Para o usuário, não há distinção entre um provedor SAML2 e um OIDC intermediado pelo Proxy-OID-Fed.

No primeiro caminho, ao selecionar o provedor SAML2 legado, o usuário é conduzido à página de *login* institucional da `UFSC` (Figura 60). Nesse ponto, o `Proxy-SAML` atua como um provedor de serviço SAML2 perante o IdP SAML2, conduz a autenticação no formato SAML2 e traduz a asserção resultante em um *token* OIDC. Concluída a autenticação, a `RPflask` exibe

Figura 56 – Página de detalhes de um serviço no PoB.

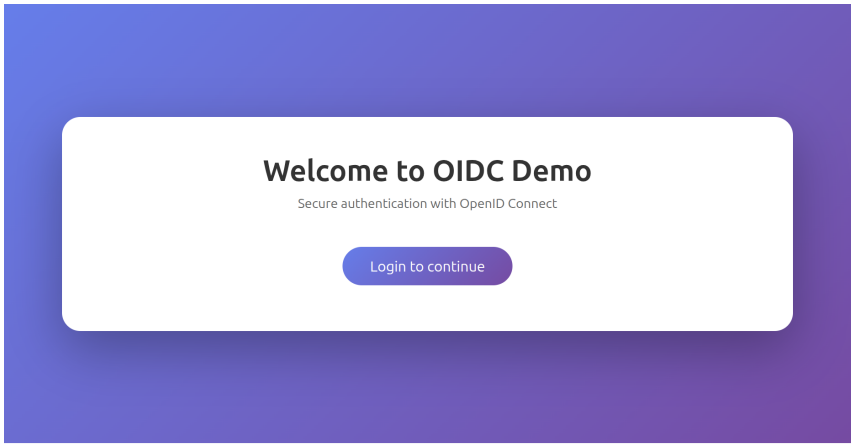


Fonte: o autor.

os atributos recebidos (Figura 61). Nota-se que atributos próprios dos *schemas* utilizados nas federações SAML2, como `eduPersonPrincipalName` e `eduperson_affiliation`, foram entregues à aplicação já como *claims* OIDC, comprovando o nível 2 de compatibilidade proposto.

No segundo caminho, ao selecionar o OP registrado `KPR`, o usuário é conduzido à sua página de *login* (Figura 62). Nesse caso, o Proxy-OID-Fed executa um fluxo OIDC perante ao

Figura 57 – Página inicial da `RPflask`.



Fonte: o autor.

OP previamente registrado no barramento. Ao final, a `RPflask` exibe os atributos do usuário autenticado (Figura 63), desta vez incluindo *claims* personalizadas do provedor, como `cargo` e `departamento`.

Os dois caminhos validam, na prática, a proposta de interoperabilidade do BAITA. Em ambos, o WAYF mediou a descoberta e a normalização, e a `RPflask` recebeu os atributos do usuário de forma uniforme, sem conhecer o protocolo ou a tecnologia do provedor de origem. O Proxy-SAML permitiu compatibilidade à federação *Chimarrão* e o Proxy-OID-Fed expôs a RP e intermediou o OP OIDC registrado, comprovando a interoperabilidade entre os dois protocolos de federação.

O papel central do WAYF no fluxo de autenticação implica *trade-offs* relevantes que devem ser gerenciados com cuidado. Do ponto de vista da privacidade, o WAYF processa dados de identidade de todos os usuários que se autenticam via barramento, sendo capaz, em princípio, de observar quem se autentica em quais serviços. Essa centralização, porém, é a contrapartida do benefício de redução de atrito na implementação da garantia de MFA e da normalização semântica dos atributos de identidade. Essa decisão de implementação desonera os gestores dos provedores da responsabilidade de implementação dessas funcionalidades.

O risco de concentração de dados é reconhecido e abordado pela comunidade que padronizou o modelo de *proxy*, e pode ser atenuado por medidas de minimização. As diretrizes de agregação de atributos da AARC estabelecem práticas concretas nesse sentido, recomendando que os agregadores expirem os registros armazenados conforme a informação de validade recebida e, na ausência de uma data de expiração explícita, segundo as diretrizes de proteção de dados da organização ou em até três meses após uma atualização (AARC, 2017b). Há, ademais, precedente operacional de *proxy* que opera com exposição mínima. O serviço *InAcademia*, também implementado sobre o *SATOSA*, atua como um *proxy* de aplicação de políticas que não revela atributos do usuário aos serviços conectados, exceto a informação de que ele pertence a uma instituição acadêmica (IdentityPython, 2019). Esses precedentes corroboram a viabilidade da abordagem de minimização preconizada para o WAYF.

Figura 58 – Descoberta no WAYF: seleção de um provedor SAML2.



Fonte: o autor.

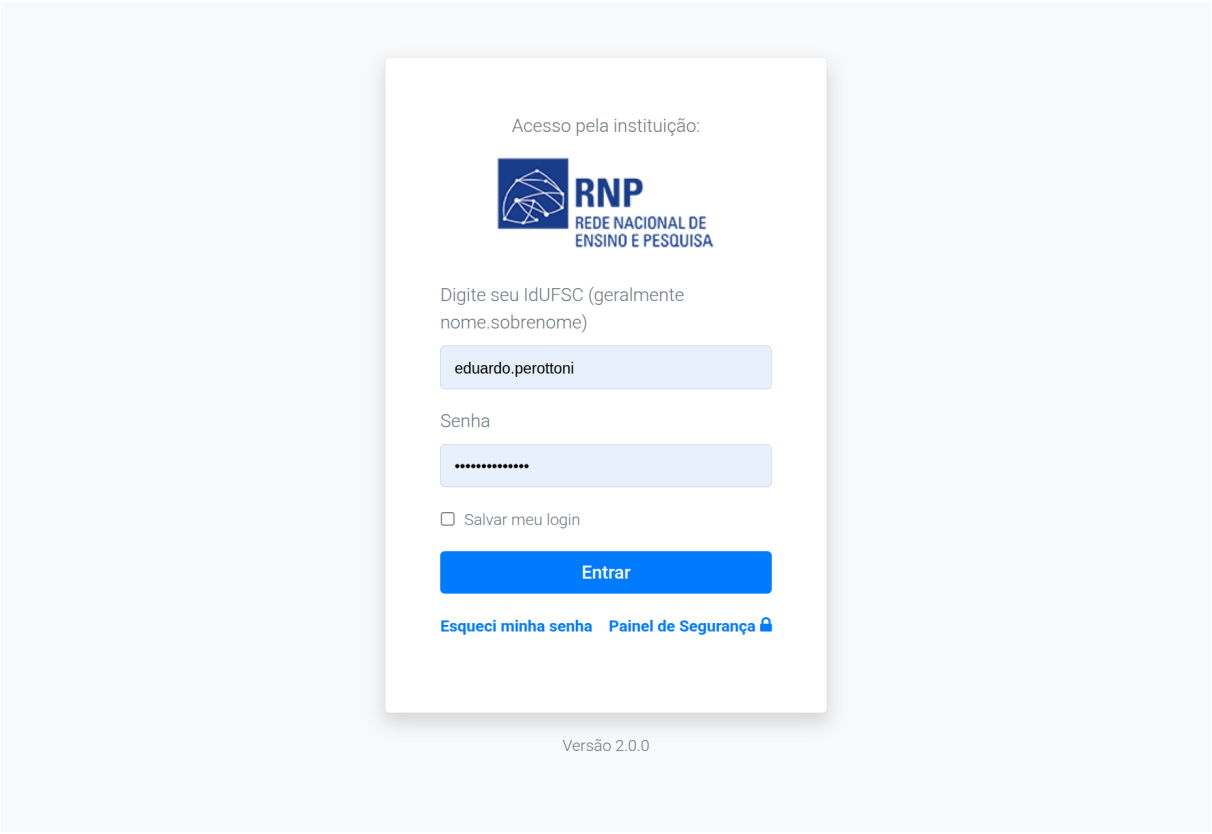
Do ponto de vista da disponibilidade, o acesso a qualquer serviço federado depende do WAYF. O componente deve, portanto, ser executado em infraestrutura de alta disponibilidade, com redundância ativa e monitoramento contínuo. O uso de múltiplas instâncias com balanceamento de carga é recomendado para mitigar o risco de ponto único de falha.

Figura 59 – Descoberta no WAYF: seleção do OP OIDC registrado `KPR`.



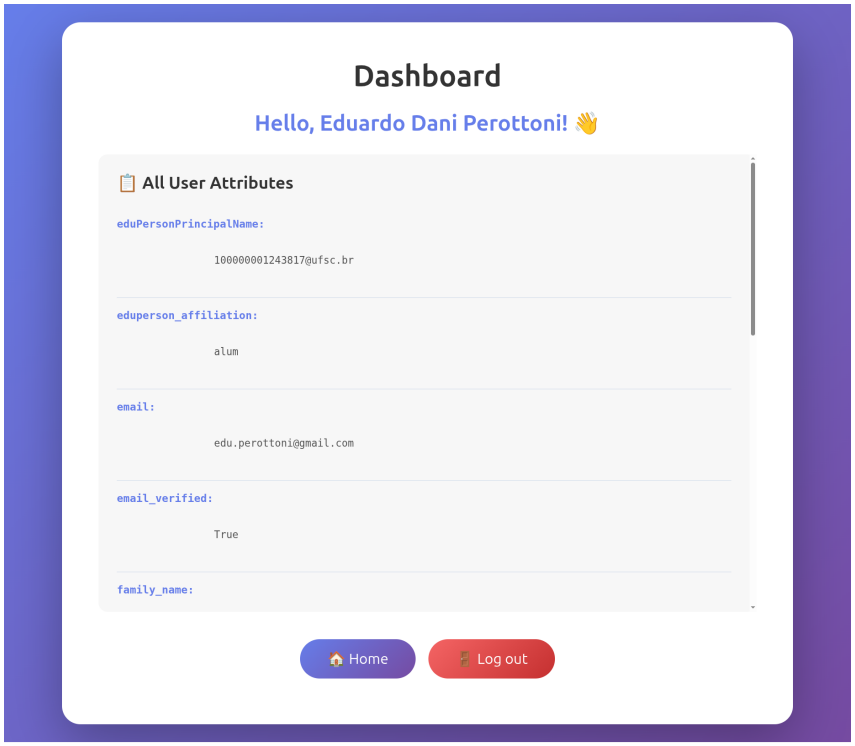
Fonte: o autor.

Figura 60 – Página de *login* do provedor SAML2 da `UFSC`.



Fonte: o autor.

Figura 61 – Exibição dos atributos recebidos do IdP SAML2 pelo `RPflask`.



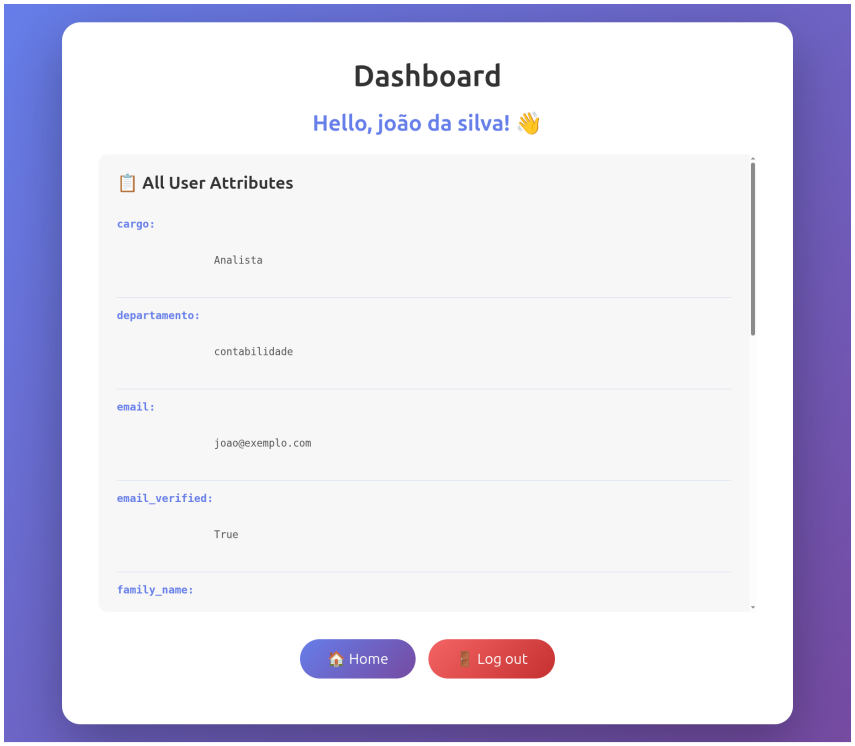
Fonte: o autor.

Figura 62 – Página de *login* do OP OIDC de teste `KPR`.



Fonte: o autor.

Figura 63 – Exibição dos atributos recebidos do OP OIDC pelo `RPflask`.



Fonte: o autor.

9 CONCLUSÃO

O presente trabalho propôs e avaliou o BAITA, um barramento de interoperabilidade acadêmico fundamentado no protocolo *OpenID Federation 1.0*, com o intuito de integrar provedores de identidade e de serviço no contexto da CAFe. Retomando o objetivo geral definido no Capítulo 2, o trabalho não apenas propôs essa arquitetura, mas a instanciou em um protótipo funcional, implantado em ambiente de homologação na infraestrutura de *testbeds* da RNP, comprovando a viabilidade do uso do *OpenID Federation 1.0* para esse fim.

O percurso até esse resultado cumpriu os objetivos específicos traçados. A revisão das especificações de IDF e o estudo do *OpenID Federation 1.0* permitiram identificar quais dos seus mecanismos eram pertinentes ao contexto da CAFe e fundamentaram as decisões de projeto. O estudo das implementações existentes do protocolo e de ferramentas correlatas, como o *SATOSA*, viabilizou o seu reaproveitamento na prova de conceito, evitando reimplementações desnecessárias. Sobre esses fundamentos, o Capítulo 7 apresentou a arquitetura proposta, na qual as dores identificada na prospecção do GT-BAITA foram abordadas pelos componentes propostos.

O Capítulo 8 demonstrou a materialização dessa arquitetura. O SGB concretizou o modelo de administração descentralizada, permitindo que uma instituição seja integrada à federação pelo simples preenchimento de formulários guiados, sem implementar o protocolo, gerar chaves ou operar qualquer infraestrutura própria. A implementação do PoB consolidou um portal dinâmico de serviços, alimentado automaticamente a cada novo provedor registrado. O resultado chave, porém, foi a validação do fluxo de autenticação de ponta a ponta (Seção 8.3), comprovando na prática a interoperabilidade pretendida inicialmente.

Conclui-se que o *OpenID Federation 1.0* é não apenas aplicável, mas vantajoso como base para a evolução da CAFe, por oferecer um caminho de migração que preserva o legado SAML2 enquanto habilita as funcionalidades modernas do OIDC. O protótipo desenvolvido sustenta essa tese e estabelece uma base concreta sobre a qual as funcionalidades ainda em estágio de proposta podem ser construídas, conforme se delineia a seguir.

9.1 TRABALHOS FUTUROS

Diversas funcionalidades projetadas permanecem como trabalho futuro. No âmbito do protocolo, destacam-se as políticas de metadados, ainda não instanciadas pela TA da CAFe e cuja representação semântica verificável entre os mundos SAML2 e OIDC constitui um desafio em aberto. Outro ponto é o nível 3 de compatibilidade com o legado, implementação em andamento que demanda o registro de um IdP atuando como *proxy* na federação SAML2. Esse registro encontra-se em fase de discussão interna na RNP. O processo de adesão da CAFe e da Chimarrão pressupõe operacionalmente o uso do *Shibboleth*, uma vez que seus requisitos, roteiros de instalação, automação de adesão e o suporte técnico oferecido contemplam apenas esse *software* (RNP, 2026a; RNP, 2026b). Essa restrição dificulta a integração de um IdP construído

sobre outra tecnologia.

No WAYF, a centralização da garantia de MFA ainda depende de soluções para a representação padronizada dos níveis de garantia, em consolidação na comunidade, como o *REFEDS Assurance Framework* (REFEDS, 2023). De forma relacionada, a normalização semântica de atributos, hoje baseada no mapeador da RNP, esbarra na ausência de uma representação canônica dos atributos acadêmicos em *claims* OIDC. Essa lacuna, já resolvida no SAML2 pelos esquemas *eduPerson* (Refeds, 2020) e *brEduPerson* (Cafe, 2020), vem sendo tratada por iniciativas como o grupo *OIDCre* da REFEDS (REFEDS OIDCre Working Group, 2024b; Basney, 2024; REFEDS OIDCre Working Group, 2024a) e o *Research and Education Working Group* da *OpenID Foundation* (OpenID Foundation, 2024). Na mesma direção, a revisão de 2025 da Arquitetura de Referência da AARC, a AARC-BPA-2025 (AARC, 2025), propõe um perfil de atributos para a representação consistente de informações de identidade e autorização, alinhado à adoção do OID Fed. A adoção dessas iniciativas tornaria a normalização verificável e interoperável entre federações.

No que tange a gestão da federação, o gerenciamento de subfederações com o isolamento de escopos de confiança por meio de TAs públicas e privadas foi projetada e validada conceitualmente, mas não implementada. Complementarmente, decorre do *trade-off* de centralização o suporte à participação nativa de entidades e sub-árvores no OID Fed. Permitir que instituições que demandem soberania ancorem à TA da CAFe a sua própria entidade OID Fed, recebendo do SGB apenas a declaração de subordinação, sem delegar a custódia das suas chaves e demais artefatos.

Quanto ao PoB, uma frente é a sua transformação em um portal inteligente, que personalize o catálogo conforme a afiliação do usuário. Isso exige que o portal autentique o usuário, tornando-se ele próprio uma RP na federação, e que a gestão de subfederações esteja em operação, de modo que a visibilidade dos serviços respeite o mesmo escopo de confiança que a hierarquia de TAs impõe na camada de protocolo. Como evolução complementar, a transição do PoB para um componente que armazena estado, como informações sobre as preferências dos usuários, permitiria oferecer recomendações personalizadas, combinando o histórico do usuário com as métricas de uso já coletadas.

9.2 CONTRIBUIÇÕES EM ESPECIFICAÇÕES TÉCNICAS INTERNACIONAIS

Em paralelo ao desenvolvimento do BAITA e ao estudo das especificações que o fundamentam, o autor deste trabalho participou de discussões técnicas no âmbito da *OpenID Foundation*, a entidade responsável pela padronização dos protocolos da família do OIDC. Essa participação resultou no reconhecimento formal em duas especificações publicadas, cujos textos citam o autor na seção de Agradecimentos.

A primeira é o próprio *OpenID Federation 1.0* (Hedberg et al., 2026), protocolo sobre o qual o BAITA é fundamentado, publicado em fevereiro de 2026. Neste protocolo, o autor

contribuiu por meio de *issues*[1] e de *pull requests*[2]. A segunda é o *OpenID Connect Relying Party Metadata Choices 1.0*[3] (Jones et al., 2026), publicada em março de 2026, que define mecanismos para que uma RP expresse preferências e restrições sobre os metadados que aceita de um OP, promovendo maior interoperabilidade semântica em fluxos OIDC.

[1] *Issues* abertas pelo autor no repositório oficial do OID Fed (https://github.com/openid/federation): #176 (*Access control of entities in the federation*), #233 (*Trust Chain Resolution for calls to Introspection Endpoint by a Federated Protected Resource*), #235 (*Application of constraints on Trust Chain resolution*), #241 (*Better organization on Entity Statement chapter*) e #290 (*TM status response with more details*).

[2] *Pull requests* submetidos pelo autor no repositório oficial do OID Fed (https://github.com/openid/federation): #234 (*Fixes syntax errors and figure 5*), #240 (*Application of constraints on Trust Chain resolution*) e #289 (*Clarification about Explicit Registration Responses and Entity Statements*).

[3] Contribuição do autor a esta especificação, registrada no repositório oficial (https://github.com/openid/rp-metadata-choices): *pull request* #9 (*fix: metadata parameter name*).

REFERÊNCIAS

AARC. **AARC Blueprint Architectures: Deliverable DJRA1.2**. [S.l.], 2017. Disponível em: https://aarc-community.org/wp-content/uploads/2017/05/DJRA1.2-AARC-Blueprint-Architectures-1.pdf. Acesso em: 2 jun. 2026.

AARC. **Guidelines on Attribute Aggregation: Deliverable JRA1.4B**. [S.l.], 2017. Disponível em: https://aarc-community.org/wp-content/uploads/2017/03/AARC-JRA1.4B.pdf. Acesso em: 2 jun. 2026.

AARC. **AARC Blueprint Architecture 2019: Guideline AARC-G045**. [S.l.], 2019. DOI: 10.5281/zenodo.3672784. Disponível em: https://aarc-community.org/guidelines/aarc-g045/. Acesso em: 14 jun. 2026.

AARC. **A Specification for IdP Hinting: Guideline AARC-G061**. [S.l.], 2021. DOI: 10.5281/zenodo.4596667. Disponível em: https://aarc-community.org/guidelines/aarc-g061/. Acesso em: 15 jun. 2026.

AARC. **Blueprint Architectures Initial Revision (AARC-BPA-2025): Deliverable D1.1**. [S.l.], 2025. Disponível em: https://wiki.geant.org/download/attachments/738885707/D1.1%20Blueprint%20Architectures%20Initial%20revision.pdf. Acesso em: 15 jun. 2026.

ACOnet Identity Federation. **IDP 5 Attribute Release**. 2023. Disponível em: https://wiki.univie.ac.at/display/federation/IDP+5+Attribute+release. Acesso em: 2 jun. 2026.

Agenzia per l'Italia Digitale (AgID). **SPID - Sistema Pubblico di Identità Digitale**. 2024. Disponível em: https://www.spid.gov.it/.

ALDOSARY, M.; ALQAHTANI, N. A survey on federated identity management systems limitation and solutions. **International Journal of Network Security & Its Applications (IJNSA) Vol**, v. 13, 2021.

ALLEN, C. The path to self-sovereign identity. **Life with Alacrity**, 2016.

ALMENÁREZ, F. et al. Towards dynamic trust establishment for identity federation. In: **Proceedings of the 2009 Euro American conference on telematics and information systems: new opportunities to increase digital citizenship**. [S.l.: s.n.], 2009. p. 1–4.

BASEER, S.; CHARUMATHI, K. Multi-factor authentication: A user experience study. **Available at SSRN 4840295**, 2024.

BASNEY, J. **Conveying eduPerson and SCHAC Attributes as JSON Web Token (JWT) Claims**. 2024. Internet-Draft `draft-refeds-oidcre-eduperson-jwt-claims`. Disponível em: https://refeds-oidcre.github.io/eduperson-jwt-claims/draft-refeds-oidcre-eduperson-jwt-claims.html.

BELOTTO, L.; ROCHA, L.; RIBEIRO, R. **Declaração de Práticas de Registro de Metadados**. 2020. Disponível em: https://svn.cafe.rnp.br/repos/CAFe/termos_politicas_CAFe/mrps.pdf.

BHARGAV-SPANTZEL, A. et al. User centricity: a taxonomy and open issues. In: **Proceedings of the second ACM workshop on Digital identity management**. [S.l.: s.n.], 2006. p. 1–10.

BHARGAV-SPANTZEL, A.; SQUICCIARINI, A. C.; BERTINO, E. Establishing and protecting digital identity in federation systems. In: **Proceedings of the 2005 Workshop on Digital Identity Management**. New York, NY, USA: Association for Computing Machinery, 2005. (DIM '05), p. 11–19. ISBN 1595932321. Disponível em: https://doi.org/10.1145/1102486.1102489.

BLANK, S. Why the lean start-up changes everything. **Harvard Business Review**, v. 91, n. 5, p. 63–72, maio 2013. Disponível em: https://hbr.org/2013/05/why-the-lean-start-up-changes-everything.

BLANK, S.; DORF, B. **The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company**. Pescadero, CA: K&S Ranch, 2012.

CABARCOS, P. A. et al. Enabling saml for dynamic identity federation management. In: SPRINGER. **Joint IFIP Wireless and Mobile Networking Conference**. [S.l.], 2009. p. 173–184.

CAFE. **Especificação de uso dos atributos CAFe**. 2020. Disponível em: https://svn.cafe.rnp.br/repos/CAFe/termos_politicas_CAFe/brEduPersonV2.pdf.

CAMP, J. Digital identity. **IEEE Technology and society Magazine**, IEEE, v. 23, n. 3, p. 34–41, 2004.

CILogon. **CILogon FAQ**. 2025. Disponível em: https://www.cilogon.org/faq. Acesso em: 2 jun. 2026.

CLAUSS, S.; KÖHNTOPP, M. Identity management and its support of multilateral security. **Computer Networks**, Elsevier, v. 37, n. 2, p. 205–219, 2001.

CONNECT2ID. **OpenID Federation and the trust chain explained**. 2024. Disponível em: https://connect2id.com/learn/openid-federation.

CONSORTIUM, S. **The Shibboleth Project**. 2025. Disponível em: https://www.shibboleth.net/about-us/the-shibboleth-project/.

DAGSER. **Especificação de uso dos atributos CAFe**. 2011. Disponível em: https://svn.cafe.rnp.br/repos/CAFe/termos_politicas_CAFe/CAFe-PoliticaDeUsoProvedoresDeIdentidade.pdf.

DALMAZO, L. M. T. et al. Serviço GIdLab: Impulsionando a pesquisa experimental em gestão de identidade. In: **Anais do XV Computer on the Beach**. Balneário Camboriú, SC, Brasil: Universidade do Vale do Itajaí (UNIVALI), 2024.

DAS, S. et al. Evaluating user perception of multi-factor authentication: A systematic review. **arXiv preprint arXiv:1908.05901**, 2019.

DENNISS, W.; BRADLEY, J. **OAuth 2.0 for Native Apps**. RFC Editor, 2017. RFC 8252. (Request for Comments, 8252). Disponível em: https://www.rfc-editor.org/info/rfc8252.

DIJK, N. van. **Pilot Components**. GÉANT, 2024. Accessed on June 22, 2025. Disponível em: https://wiki.geant.org/display/GWP5/Pilot+Components.

DIJK, N. van; VAGHETTI, D. **eduGAIN OID Federation Pilot**. 2024. Disponível em: https://indico.cern.ch/event/1438628/contributions/6190755/attachments/2982511/5255619/eduGAIN%20OpenID%20Fedederation%20Pilot%20-%20FIM4R.pdf.

DREEF, C.; VAGHETTI, D.; KREMERS, M. **Infoshare OpenID Federation Pilot (eduGAIN / GN5-2 T&I)**. 2025. Disponível em: https://events.geant.org/event/1910/.

DRIESSEN, V. **A successful Git branching model**. 2010. https://nvie.com/posts/a-successful-git-branching-model/. Acesso em: 1 jun. 2026.

eduGAIN. **Federation topologies for eduGAIN**. 2025. Acesso em: 16 jun. 2026. Disponível em: https://wiki.geant.org/spaces/eduGAIN/pages/1379696652/Federation+topologies+for+eduGAIN.

European Union. **Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024**. 2024. Disponível em: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1183&qid=1722335279175.

FERDOUS, M. S.; POET, R. Dynamic identity federation using security assertion markup language (saml). In: SPRINGER. **IFIP Working Conference on Policies and Research in Identity Management**. [S.l.], 2013. p. 131–146.

FLANAGAN, H. **Is There a Future for REFEDS and R&E Federations?** 2024. Disponível em: https://refeds.org/a/2984.

FLETCHER, G. **OpenID Connect Native SSO for Mobile Apps 1.0**. [S.l.], 2025. Accessed: 2025-06-12. Disponível em: https://openid.net/specs/openid-connect-native-sso-1_0.html.

GÉANT. **Enabling the EOSC AAI Federation with eduGAIN and MyAccessID**. 2025. Disponível em: https://connect.geant.org/2025/06/04/enabling-the-eosc-aai-federation-with-edugain-and-myaccessid. Acesso em: 2 jun. 2026.

HALL, S. **A identidade cultural na pós-modernidade**. 11. ed. Rio de Janeiro: DP&A, 2006.

HANSEN, M.; SCHWARTZ, A.; COOPER, A. Privacy and identity management. **IEEE Security & Privacy**, IEEE, v. 6, n. 2, p. 38–45, 2008.

HARDT, D. **The OAuth 2.0 Authorization Framework**. RFC Editor, 2012. RFC 6749. (Request for Comments, 6749). Disponível em: https://www.rfc-editor.org/info/rfc6749.

HARRIS, N.; VAGHETTI, D. **eduGAIN Technical Profiles Working Group**. 2025. Accessed on 2025-06-22. Disponível em: https://wiki.geant.org/display/eduGAIN/eduGAIN+Technical+Profiles+Working+Group.

HEDBERG, R. et al. **OpenID Federation 1.0**. 2026. Disponível em: https://openid.net/specs/openid-federation-1_0.html.

HOWES, T. A.; SMITH, M. C.; GOOD, G. S. **Understanding and Deploying LDAP Directory Services**. 2. ed. Boston, MA, USA: Addison-Wesley Professional, 2003.

HUGHES, J.; MALER, E. Security assertion markup language (saml) v2. 0 technical overview. **OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08**, Citeseer, v. 13, p. 12, 2005.

IAM4NFDI. **NFDI-AAI Architecture**. 2024. Disponível em: https://cvs.data.kit.edu/nfdi-aai-doc/architecture. Acesso em: 2 jun. 2026.

IdentityPython. **SATOSA Wiki**. 2019. Disponível em: https://github.com/IdentityPython/SATOSA/wiki. Acesso em: 2 jun. 2026.

InCommon Federation. **InCommon Federation**. 2024. Acesso em: 16 maio 2025. Disponível em: https://www.incommon.org/federation/.

Internet2. **Internet2**. 2025. Disponível em: https://internet2.edu/.

ITALIA, D. **SPID CIE OIDC Documentation**. 2024. Disponível em: https://italia.github.io/spid-cie-oidc-docs/en/.

ITALIA, D. **La Federazione delle Identità**. Developers Italia, 2025. Accessed on June 22, 2025. Disponível em: https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/la_federazione_delle_identita.html.

JONES, M.; BRADLEY, J. **Trust Establishment with OpenID Federation**. 2024. PDF presentation. Internet Identity Workshop (IIW), Mountain View, CA. Disponível em: https://self-issued.info/presentations/OpenID_Federation_18-Apr-24.pdf.

JONES, M. B. **Enabling Scalable Multi-lateral Federations with OpenID Connect**. 2020. Presentation at Identiverse 2020 Virtual. Accessed: 2025-06-11. Disponível em: https://self-issued.info/presentations/Identiverse_2020-Enabling_Scalable_Multi-lateral_Federations_with_OpenID_Connect-Michael_Jones.pdf.

JONES, M. B. **OpenID Connect Federation 1.0: Enabling Trust at Scale**. 2020. Presentation at the OpenID Summit Tokyo 2020. Accessed: 2025-06-11. Disponível em: https://self-issued.info/presentations/OpenID_Connect_Federation_Japan_24-Jan-20.pdf.

JONES, M. B.; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. RFC Editor, 2015. RFC 7519. (Request for Comments, 7519). Disponível em: https://www.rfc-editor.org/info/rfc7519.

JONES, M. B. et al. **OpenID Connect Relying Party Metadata Choices 1.0**. 2026. Disponível em: https://openid.net/specs/openid-connect-rp-metadata-choices-1_0-final.html.

JONES, M. B.; SAKIMURA, N.; BRADLEY, J. **OAuth 2.0 Authorization Server Metadata**. RFC Editor, 2018. RFC 8414. (Request for Comments, 8414). Disponível em: https://www.rfc-editor.org/info/rfc8414.

JØSANG, A.; POPE, S. User centric identity management. In: CITESEER. **AusCERT Asia Pacific information technology security conference**. [S.l.], 2005. v. 22, p. 2005.

KAWASAKI, T. **OpenID Federation 1.0**. 2023. Disponível em: https://darutk.medium.com/oidc-federation-c2840622dc8f.

KHASKHELI, G. M.; SHERBAZ, M.; SHAIKH, U. R. A comparative usability study of single-factor and two-factor authentication. **Tropical Scientific Journal**, v. 1, n. 1, p. 17–27, 2022.

KYLAU, U. et al. Trust requirements in identity federation topologies. In: **2009 International Conference on Advanced Information Networking and Applications**. [S.l.: s.n.], 2009. p. 137–145.

LODDERSTEDT, T. et al. **OAuth 2.0 Security Best Current Practice**. [S.l.], 2023. Disponível em: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics.

LODDERSTEDT, T.; MCGLOIN, M.; HUNT, P. **OAuth 2.0 Threat Model and Security Considerations**. [S.l.], 2013. Disponível em: https://www.rfc-editor.org/info/rfc6819.

LODDERSTEDT, T.; YASUDA, K.; LOOKER, T. **OpenID for Verifiable Credential Issuance**. 2024. Disponível em: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html.

LOPEZ, J.; OPPLIGER, R.; PERNUL, G. Authentication and authorization infrastructures (aais): a comparative survey. **Computers & Security**, v. 23, n. 7, p. 578–590, 2004. ISSN 0167-4048. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167404804001828.

MARCO, G. D. **Work begins on unified SPID and CIE OpenID Connect SDKs**. Developers Italia, 2022. Accessed on June 22, 2025. Disponível em: https://developers.italia.it/en/news/2022/03/15/work-is-starting-SDK-SPID-CIE.

MARCO, G. D. et al. **OpenID Federation Extended Subordinate Listing 1.0 - draft 02**. [S.l.], 2025. Disponível em: https://openid.net/specs/openid-federation-extended-listing-1_0.html.

MARCO, G. D. et al. **OpenID Federation Wallet Architectures 1.0**. 2024. Disponível em: https://openid.net/specs/openid-federation-wallet-1_0.html.

MARCO, G. D. et al. **OpenID Federation 1.0: Shaping The Advanced Infrastructure of Trust**. 2024. Presented at the 18th FIM4R Workshop at TIIME 2024, May 20–24, 2024, Maribor, Slovenia. Disponível em: https://self-issued.info/presentations/OpenID_Federation_1.0_for_18th_FIM4R_Workshop_at_TIIME_2024.pdf.

MORABITO, G. **Integrazione di OAuth 2.0 e OpenID Connect nell'infrastruttura pan-Europea di identità digitale**. Dissertação (Tesi di Laurea Magistrale (Master's Thesis)) — Politecnico di Torino, 2023. Accessed on June 22, 2025. Disponível em: https://webthesis.biblio.polito.it/secure/26640/1/tesi.pdf.

MÜHLE, A. et al. A survey on essential components of a self-sovereign identity. **Computer Science Review**, Elsevier, v. 30, p. 80–86, 2018.

OASIS. **SAML Core Specification Version 2.0**. 2005. Disponível em: https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

OASIS. **Web Services Federation Language (WS-Federation) v1.2**. 2009. Disponível em: https://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.pdf.

OMETOV, A. et al. Multi-factor authentication: A survey. **Cryptography**, v. 2, n. 1, 2018. ISSN 2410-387X. Disponível em: https://www.mdpi.com/2410-387X/2/1/1.

OpenID Foundation. **Research and Education (R&E) Working Group**. 2024. Disponível em: https://openid.net/wg/rande/.

OSTERWALDER, A.; PIGNEUR, Y. **Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers**. Hoboken, NJ: John Wiley & Sons, 2010.

OWASP Foundation. **Credential Stuffing - OWASP**. 2024. Acesso em: 18 maio 2025. Disponível em: https://owasp.org/www-community/attacks/Credential_stuffing.

PEROTTONI, E. D. et al. **Whitepaper do MVP – Fase 1: GT-BAITA**. [S.l.], 2026. Disponível em: https://drive.google.com/file/d/1mDitzv85Je-2R6l42aDfHs9gJzvJOWDb/view?usp=sharing.

PFITZMANN, A.; HANSEN, M. **A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management**. [S.l.], 2010. V0.34. Disponível em: https://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.

PöHN, D.; HOMMEL, W. New directions and challenges within identity and access management. **IEEE Communications Standards Magazine**, v. 7, n. 2, p. 84–90, 2023.

RADHA, V.; REDDY, D. H. A survey on single sign-on techniques. **Procedia Technology**, v. 4, p. 134–139, 2012. ISSN 2212-0173. 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012. Disponível em: https://www.sciencedirect.com/science/article/pii/S2212017312002988.

REFEDS. **Federation 2.0 – Arrival of the IdP-SP Proxy**. 2019. REFEDS. Disponível em: https://refeds.org/a/2128. Acesso em: 2 jun. 2026.

REFEDS. **eduPerson**. 2020. Disponível em: https://wiki.refeds.org/display/STAN/eduPerson.

REFEDS. **REFEDS Assurance Framework 2.0**. 2023. https://refeds.org/assurance. Acessado em: 30 de junho de 2025. Disponível em: https://refeds.org/assurance.

REFEDS OIDCre Working Group. **Best Current Practice for Mapping SAML 2.0 Attributes to OpenID Connect Claims in Research and Education**. 2024. Disponível em: https://github.com/surfnet-niels/refeds-oidcre-saml-oidc-mapping.

REFEDS OIDCre Working Group. **OIDCre — OpenID Connect for Research and Education Working Group**. 2024. Disponível em: https://wiki.refeds.org/display/GROUPS/OIDCre.

RNP. **Inclusão de metadados na Federação Chimarrão**. 2020. Disponível em: https://ajuda.rnp.br/cafe/processo-de-adesao/adesao-idp/inclusao-de-metadados-na-federacao-chimarrao.

RNP. **Politica da Federação CAFe**. 2021. Disponível em: https://svn.cafe.rnp.br/repos/CAFe/termos_politicas_CAFe/Politica%20da%20Federacao%20CAFe.pdf.

RNP. **Comunidade Acadêmica Federada (CAFe)**. 2024. Disponível em: https://rnpmais.rnp.br/comunidade-academica-federada-cafe.

RNP. **Requisitos de Uso — CAFe**. 2024. Disponível em: https://ajuda.rnp.br/cafe/requisitos-de-uso. Acesso em: 2 jun. 2026.

RNP. **Lista de Clientes**. 2025. Disponível em: https://ajuda.rnp.br/cafe/lista-clientes.

RNP. **Requisitos de Uso: Provedor de Identidade da CAFe**. 2026. Disponível em: https://ajuda.rnp.br/cafe/idp-cafe/requisitos-de-uso.

RNP. **Roteiro para instalação de um IdP Shibboleth**. 2026. Disponível em: https://ajuda.rnp.br/cafe/processo-de-adesao/adesao-idp/roteiro-para-instalacao-de-um-idp.

SAKIMURA, N.; BRADLEY, J.; AGARWAL, N. **Proof Key for Code Exchange by OAuth Public Clients**. RFC Editor, 2015. RFC 7636. (Request for Comments, 7636). Disponível em: https://www.rfc-editor.org/info/rfc7636.

SAKIMURA, N.; BRADLEY, J.; JONES, M. **OpenID Connect Dynamic Client Registration 1.0 incorporating errata set 2**. [S.l.], 2023. Accessed on 2024-06-27. Disponível em: https://openid.net/specs/openid-connect-registration-1_0.html.

SAKIMURA, N. et al. **OpenID Connect Core 1.0**. 2023. Disponível em: https://openid.net/specs/openid-connect-core-1_0.html.

SAKIMURA, N. et al. **OpenID Connect Discovery 1.0 incorporating errata set 2**. 2023. Accessed on 2026-07-09. Disponível em: https://openid.net/specs/openid-connect-discovery-1_0.html.

SCHARDONG, F.; CUSTÓDIO, R. Self-sovereign identity: a systematic review, mapping and taxonomy. **Sensors**, MDPI, v. 22, n. 15, p. 5641, 2022.

SCHARDONG, F. et al. Pioneering the future of electronic identity: from post-quantum cryptography to fiduciary principles. 2024.

SERMERSHEIM, J. **Lightweight Directory Access Protocol (LDAP): The Protocol**. RFC Editor, 2006. RFC 4511. (Request for Comments, 4511). Disponível em: https://www.rfc-editor.org/info/rfc4511.

SHAPIRO, M. Structure and encapsulation in distributed systems: the proxy principle. In: **Proceedings of the 6th International Conference on Distributed Computing Systems (ICDCS)**. Cambridge, MA, USA: IEEE, 1986. p. 198–204.

Shibboleth. **Shibboleth Project Wiki**. 2025. Disponível em: https://shibboleth.atlassian.net/wiki.

Shibboleth Consortium. **Attribute Filter Configuration — Identity Provider 4**. 2021. Disponível em: https://shibboleth.atlassian.net/wiki/spaces/IDP4/pages/1265631865/AttributeFilterConfiguration. Acesso em: 2 jun. 2026.

SMITH, M. C. **Definition of the inetOrgPerson LDAP Object Class**. RFC Editor, 2000. RFC 2798. (Request for Comments, 2798). Disponível em: https://www.rfc-editor.org/info/rfc2798.

SOLTANI, R.; NGUYEN, U. T.; AN, A. A survey of self-sovereign identity ecosystem. **Security and Communication Networks**, Wiley Online Library, v. 2021, n. 1, p. 8873429, 2021.

SPORNY, M. et al. **Verifiable Credentials Data Model v2.0**. 2025. https://www.w3.org/TR/vc-data-model-2.0. W3C Recommendation. Disponível em: https://www.w3.org/TR/vc-data-model-2.0.

SULESKI, T. et al. A review of multi-factor authentication in the internet of healthcare things. **Digital Health**, SAGE Publications Sage UK: London, England, v. 9, p. 20552076231177144, 2023.

SWITCHAAI. **Simple Demo**. 2025. Disponível em: https://help.switch.ch/aai/demo/simple/.

TERBU, O. et al. **OpenID for Verifiable Presentations**. 2024. Disponível em: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html.

UK federation. **Participation in eduGAIN**. s.d. https://www.ukfederation.org.uk/content/Documents/EduGAINParticipation. Operado pela Jisc. Acesso em: 10 jul. 2026.

VAGHETTI, D. **eduGAIN: OpenID Federation Pilot**. 2024. Apresentação em Workshop no GARR Workshop 24. Disponível em: https://www.garr.it/en/news-events/publications/slide-ws24/7172-sessione-6-0611-04-vaghetti-0611/file.

VAGHETTI, D. **eduGAIN OpenID Federation POC**. 2024. Disponível em: https://indico.cern.ch/event/1325302/contributions/5753884/attachments/2790679/4866600/eduGAIN%20OpenID%20Federation%20POC%20-%20FIM4R%20Copenhagen%20TIIME.pdf.

WANGHAM, M. S. et al. Gerenciamento de identidades federadas. **MINICURSOS X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais**, p. 1–52, 2010.

WIKI, S. **FlowsAndConfig**. 2019. Disponível em: https://shibboleth.atlassian.net/wiki/spaces/CONCEPT/pages/928645290/FlowsAndConfig.

WINDLEY, P. **Digital Identity: Unmasking Identity Management Architecture (IMA)**. O'Reilly Media, 2005. ISBN 9780596553944. Disponível em: https://books.google.com.bz/books?id=o8mHSbDHgPsC.

Apêndices

APÊNDICE A – FLUXO DE INTEGRAÇÃO DE UM PROVEDOR DE IDENTIDADE OIDC

Este apêndice ilustra, por meio de capturas de tela do SGB, o fluxo completo de integração de uma organização e de seu OP OIDC ao BAITA. Como exemplo, utiliza-se a integração da Sociedade Brasileira de Computação (SBC). O processo divide-se em duas fases. Primeiro, a criação da organização e o *onboarding* do seu administrador. Depois, o registro do PI propriamente dito.

A.1 CRIAÇÃO DA ORGANIZAÇÃO E DELEGAÇÃO

O processo inicia com o operador da CAFe criando a organização e delegando a sua administração a um gestor da instituição. Na Figura 64, o operador da CAFe preenche os dados iniciais da SBC (nome oficial, sigla e categoria) e, na seção de `Permissões e Acesso`, adiciona o *e-mail* do gestor (`gestor.ti.sbc@gmail.com`) como `Administrador` da organização.

O gestor designado recebe um convite por *e-mail* e, ao aceitá-lo, é direcionado ao formulário de cadastro do SGB (Figura 65), no qual informa os seus dados pessoais. Em seguida, o sistema exige a configuração de um segundo fator de autenticação, por meio do registro de uma *passkey* ou da leitura de um *QR Code* por um aplicativo autenticador (*OTP*), conforme a Figura 66. Nesse exemplo, o usuário optou pelo aplicativo autenticador e o configurou (Figura 67). Para finalizar o registro no SGB, o usuário deve definir uma senha. Esse passo é ilustrado na Figura 68.

Concluído o cadastro, o gestor acessa pela primeira vez a sua página inicial (Figura 69) e completa as informações da organização, preenchendo a URL do seu site oficial e uma descrição institucional, além de revisar o nome, a sigla e a categoria definidos no ato da criação (Figura 70). Após a confirmação do preenchimento (Figura 71), a `SBC` passa a constar na `Lista de Entidades` global, como uma organização ativa.

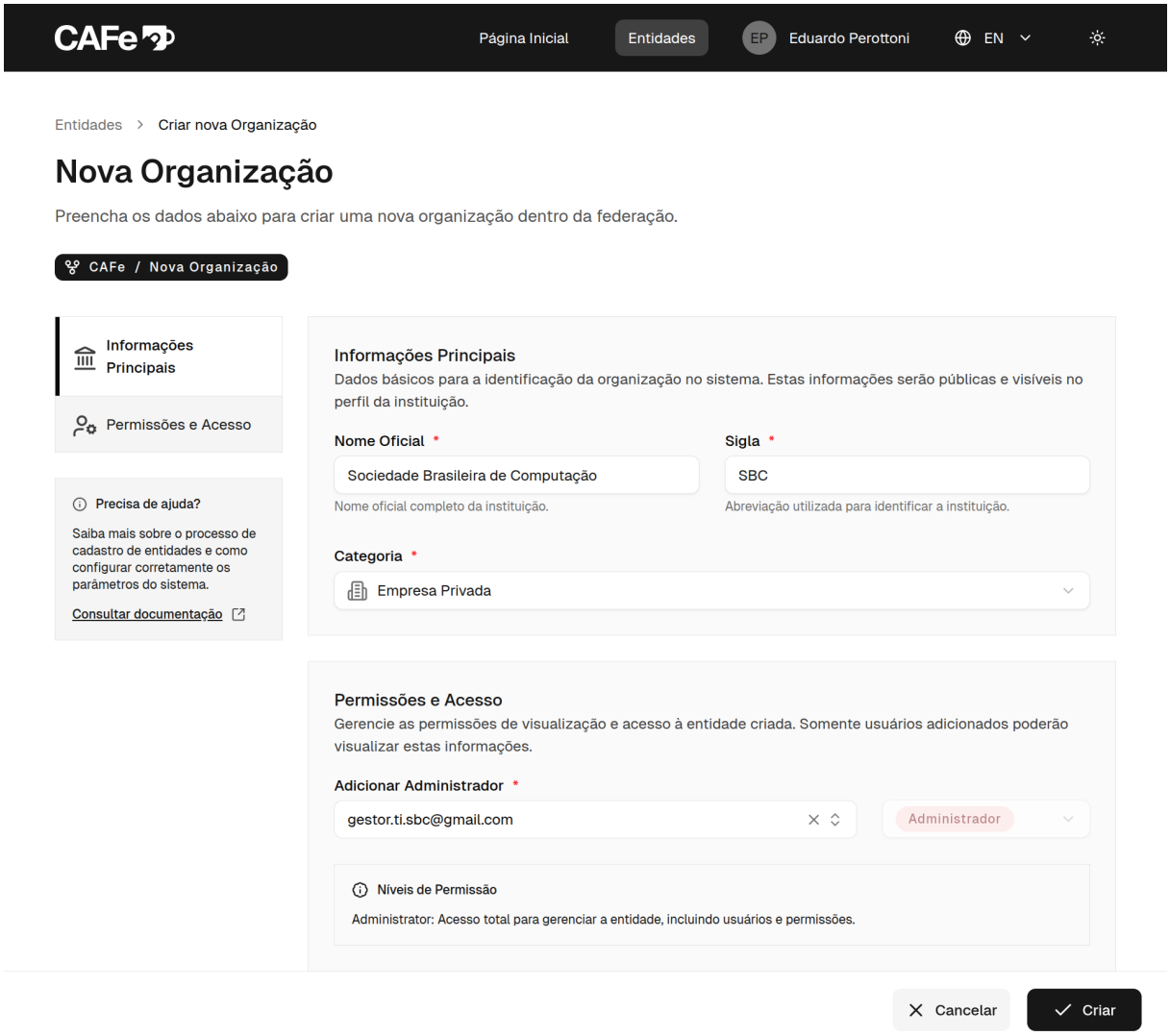
A.2 REGISTRO DO PROVEDOR DE IDENTIDADE

Habilitado a gerir a sua organização, o gestor inicia então a criação do PI da instituição. Na página de detalhes da `SBC`, ele aciona o menu `Criar sub-entidade`, e seleciona a criação de um `Novo Provedor de Identidade` (Figura 72).

O gestor preenche o formulário de criação do PI com os dados do provedor (Figura 73). Nesse caso, a delegação é feita para o próprio gestor. Imediatamente após a submissão, o PI é criado com *status* `Pendente`, visível tanto na página inicial (Figura 74) quanto na sua página de detalhes, indicando que a sua configuração técnica ainda não foi concluída.

O gestor então precisa configurar o OP. Portanto, na página de edição, informa o `discovery endpoint` do OP OIDC e as credenciais (`client_id` e `client_secret`) geradas pelo OP. Opcionalmente, pode-se informar os *endpoints* a serem utilizados pelo Proxy-OID-Fed in-

Figura 64 – Criação e delegação da organização SBC ao gestor da instituição.



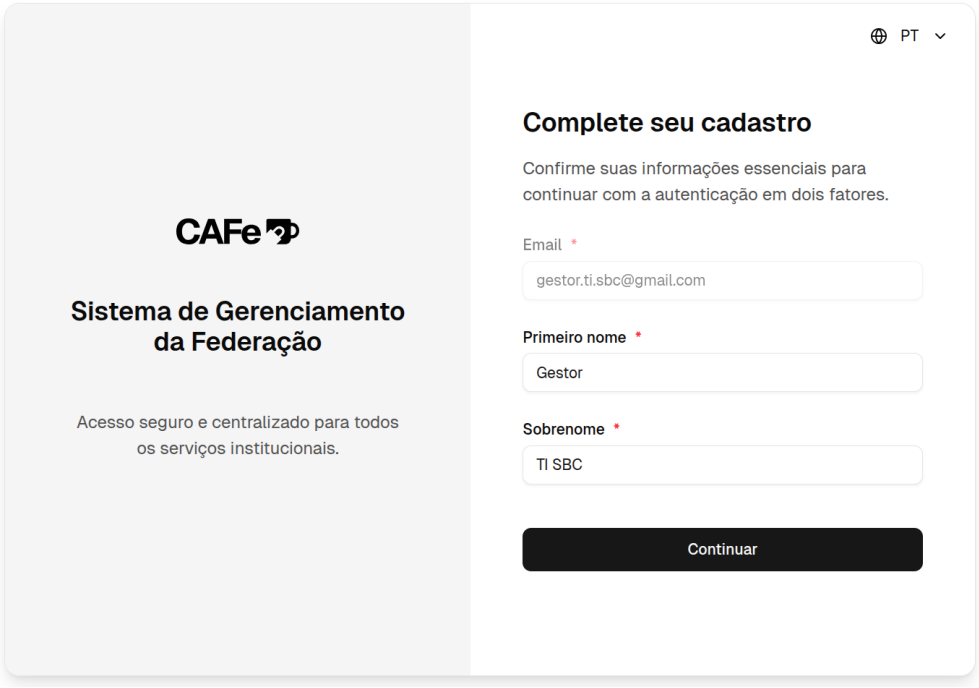
Fonte: o autor.

dividualmente na seção de `metadados`. A interface que possibilita essa configuração é mostrada na Figura 75. Após a submissão do formulário, a `redirect URL` é gerada pelo Proxy-OID-Fed e informada na página de detalhes do OP.

A página do provedor já ativo (Figura 76) reúne esses dados na seção `Configurações técnicas`. Nota-se que essas credenciais materializam a intermediação descrita na Seção 7.4, por meio dos quais o Proxy-OID-Fed passa a expor o PI na federação em nome da instituição.

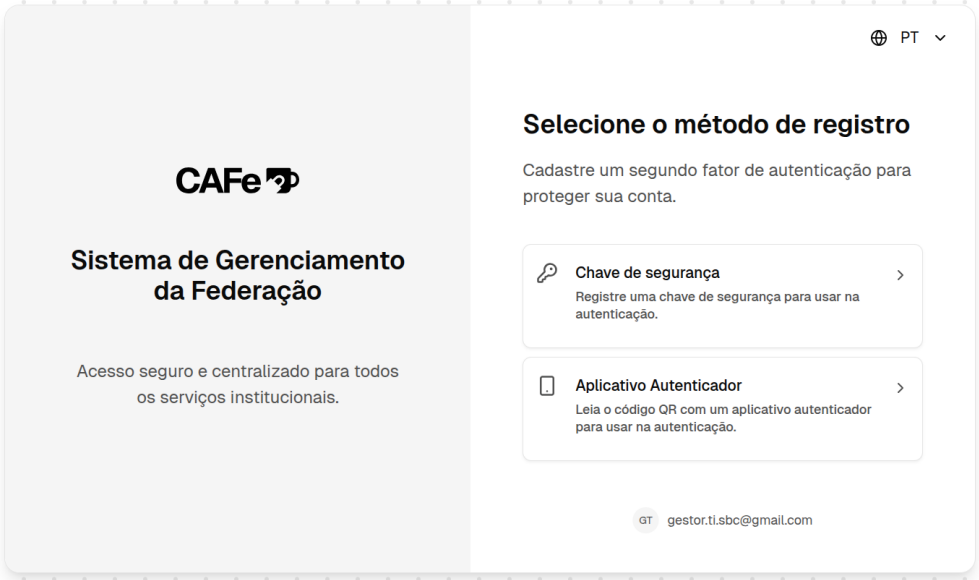
Uma vez ativo, o PI reflete-se na página inicial do gestor (Figura 77) e na Lista de Entidades, tanto na visão global quanto no contexto da `SBC`, como seu subordinado (Figura 78). A partir desse momento, o PI OIDC está plenamente operacional no barramento, descobrível e utilizável nos fluxos de autenticação da federação, sem que a instituição tenha precisado implementar qualquer extensão do OID Fed.

Figura 65 – Cadastro do gestor no SGB.



Fonte: o autor.

Figura 66 – Configuração do segundo fator de autenticação (*passkey* ou *OTP*).



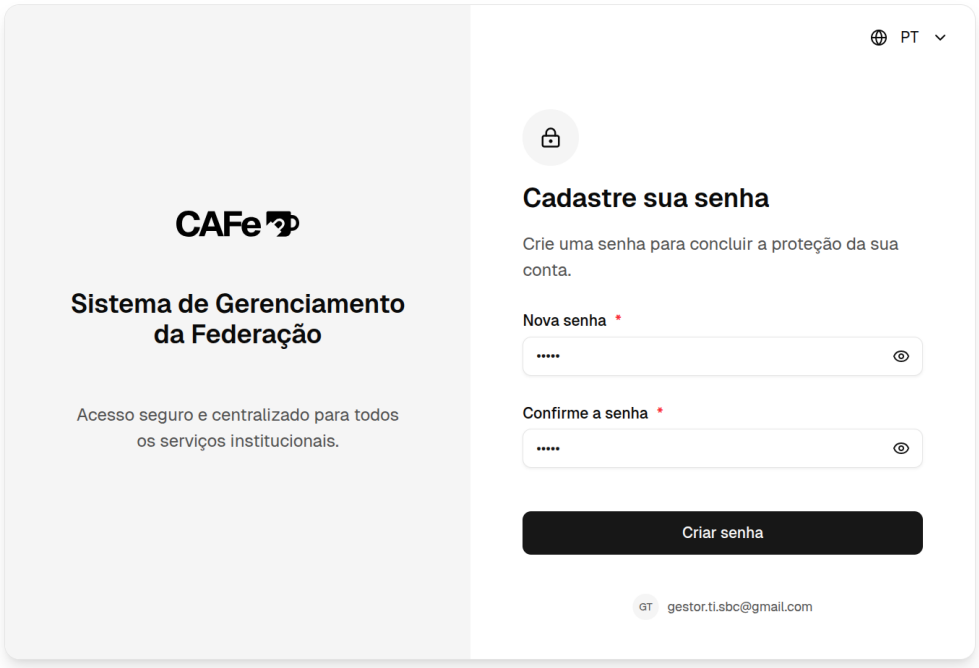
Fonte: o autor.

Figura 67 – Configuração do aplicativo autenticador.



Fonte: o autor.

Figura 68 – Configuração do aplicativo autenticador.



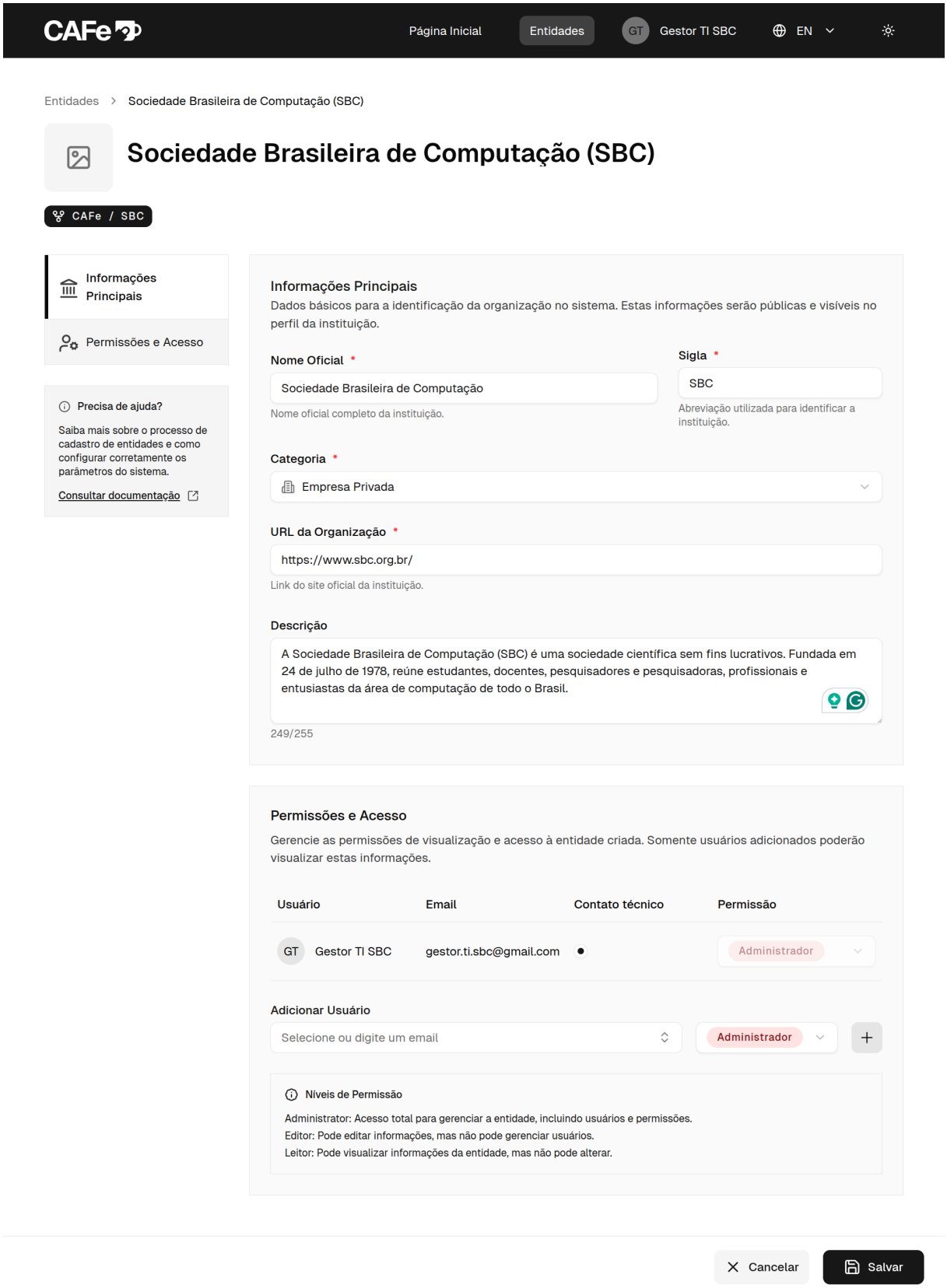
Fonte: o autor.

Figura 69 – Primeiro acesso do gestor à sua página inicial.



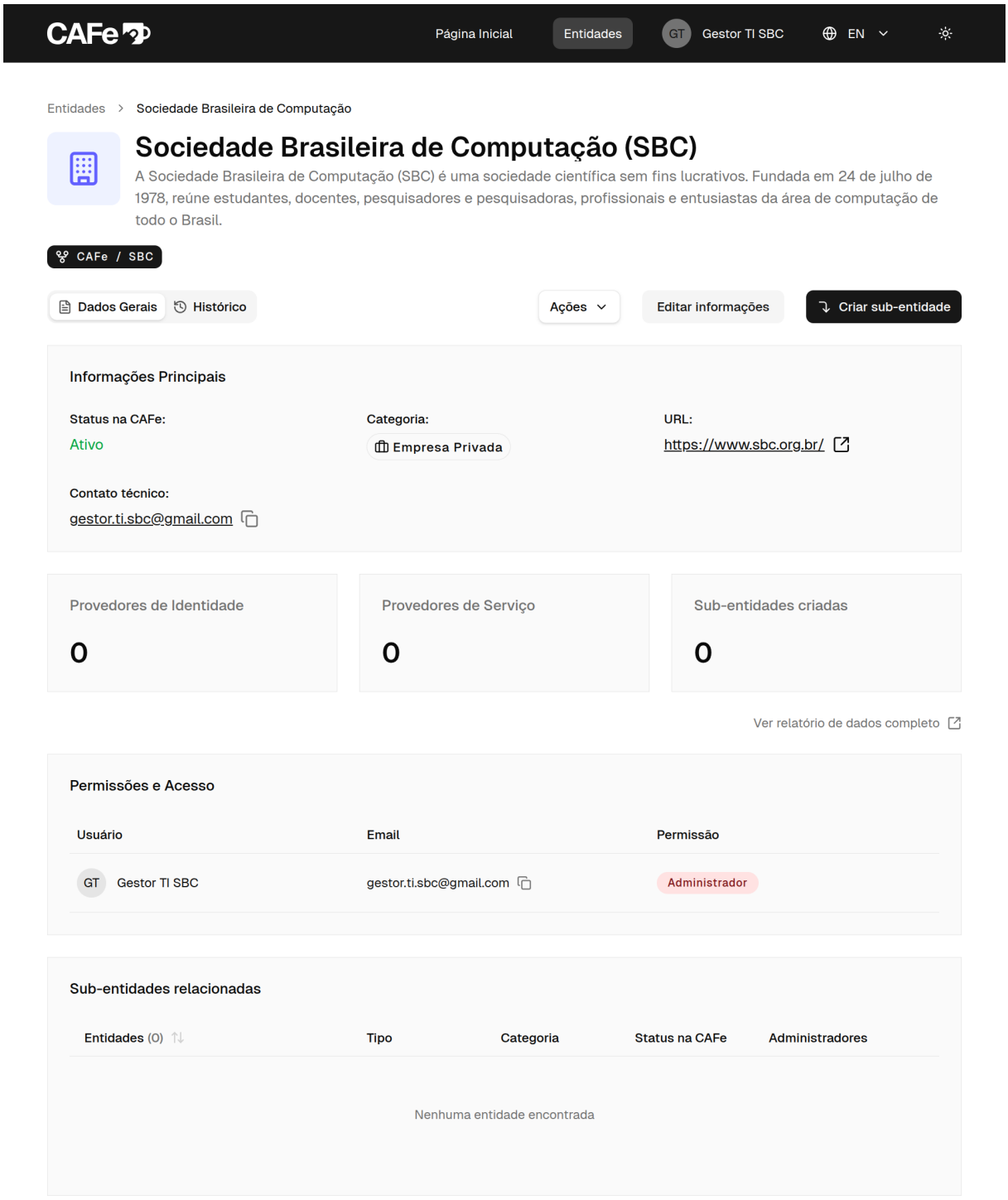
Fonte: o autor.

Figura 70 – Preenchimento das informações da organização SBC.



Fonte: o autor.

Figura 71 – Confirmação após o preenchimento do formulário da organização.



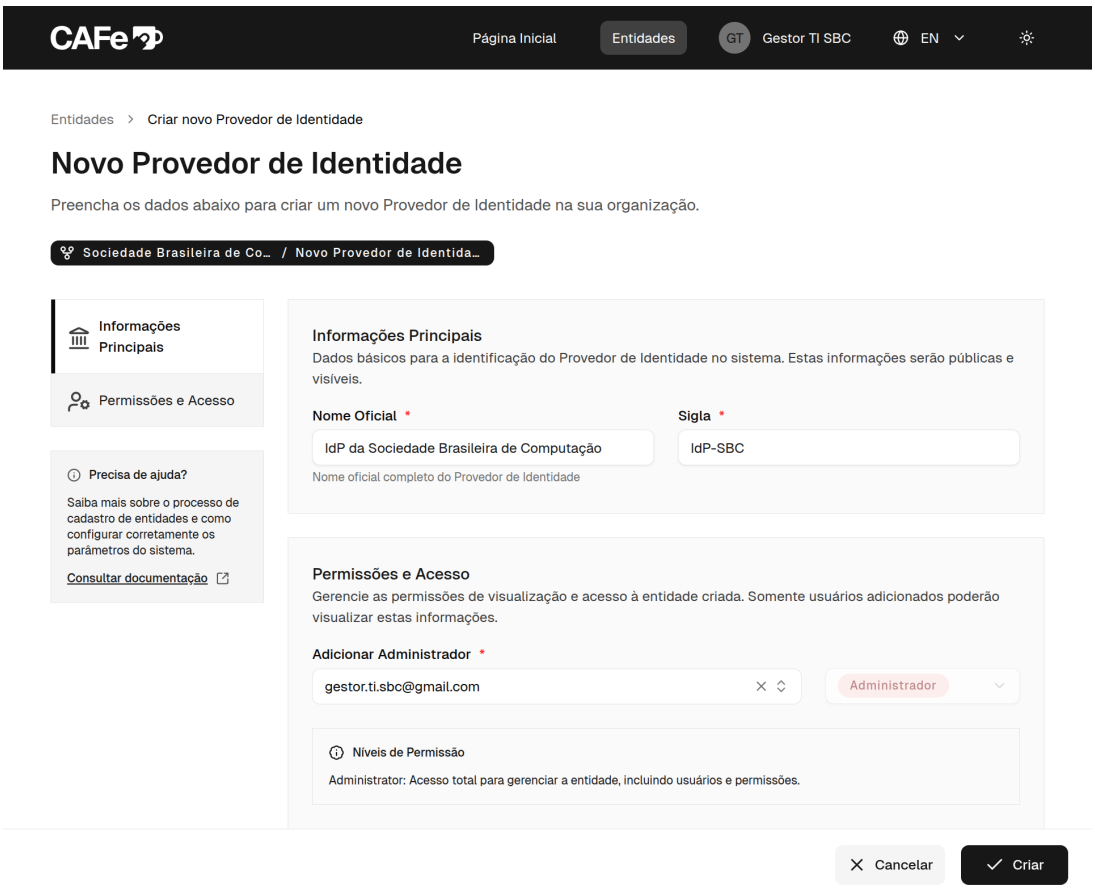
Fonte: o autor.

Figura 72 – Início da criação de um OP subordinado à SBC.



Fonte: o autor.

Figura 73 – Formulário de criação do PI da SBC.



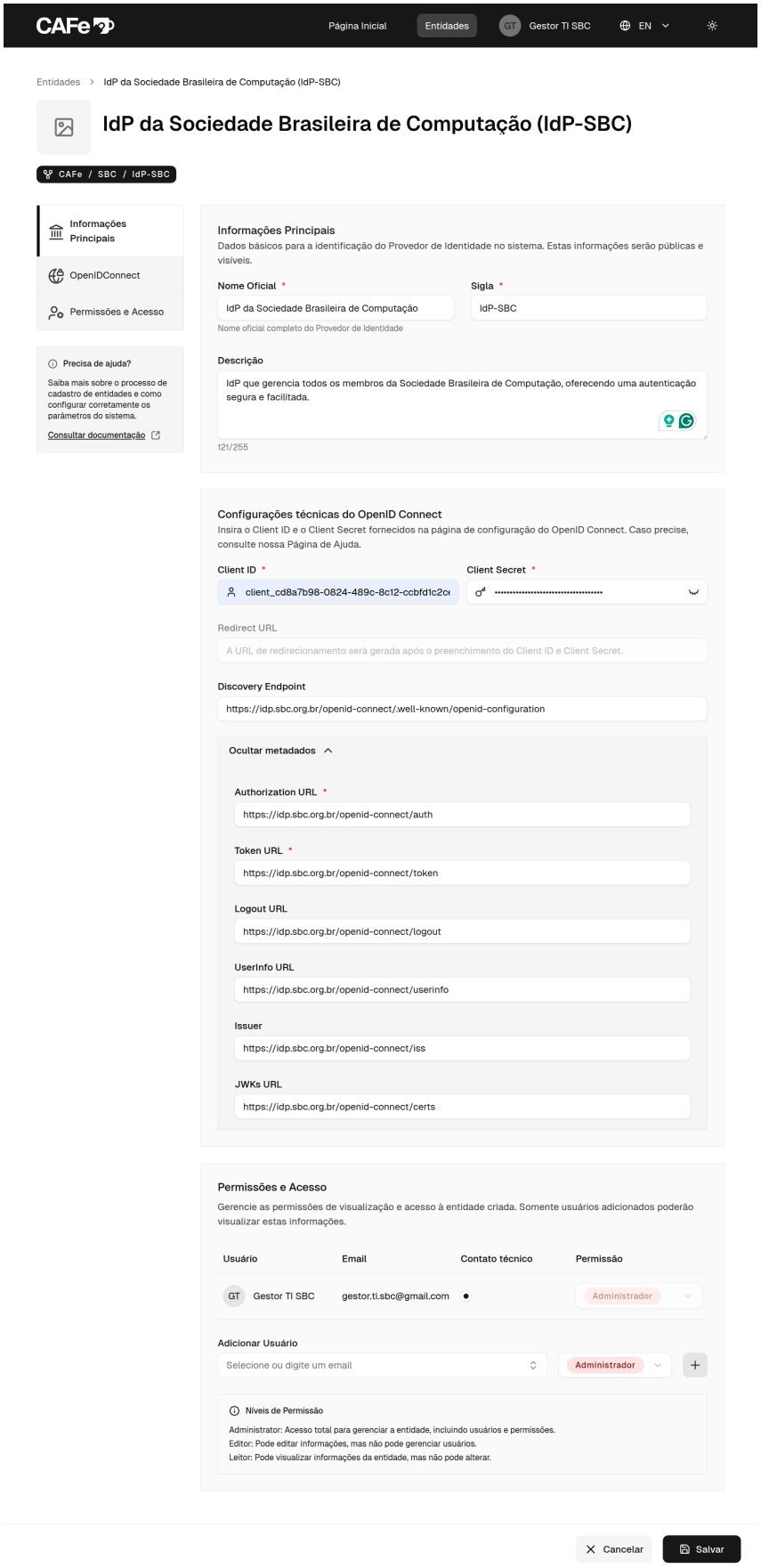
Fonte: o autor.

Figura 74 – Página inicial indicando o novo PI com status *Pendente*.



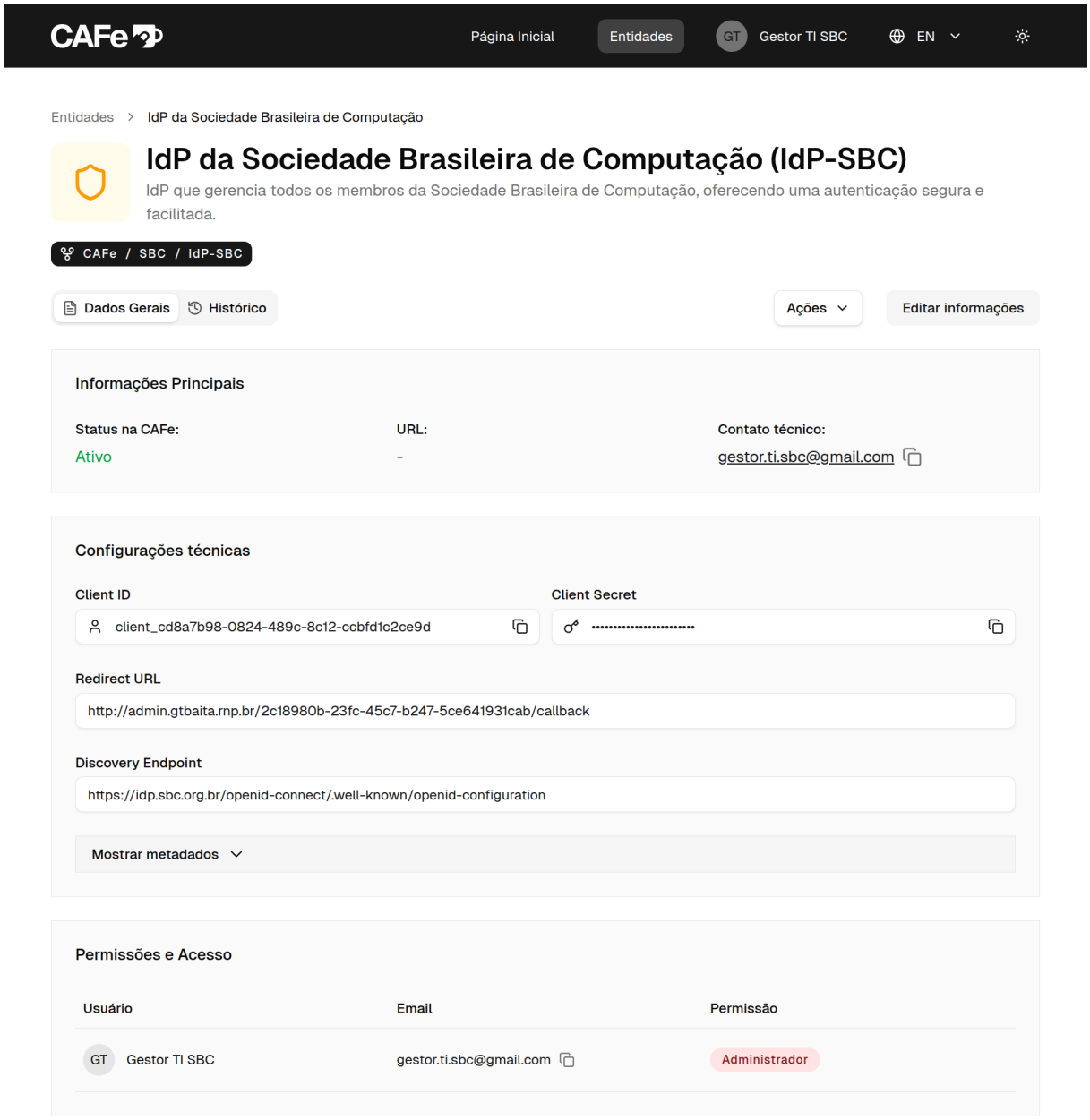
Fonte: o autor.

Figura 75 – Preenchimento da configuração técnica do PI.



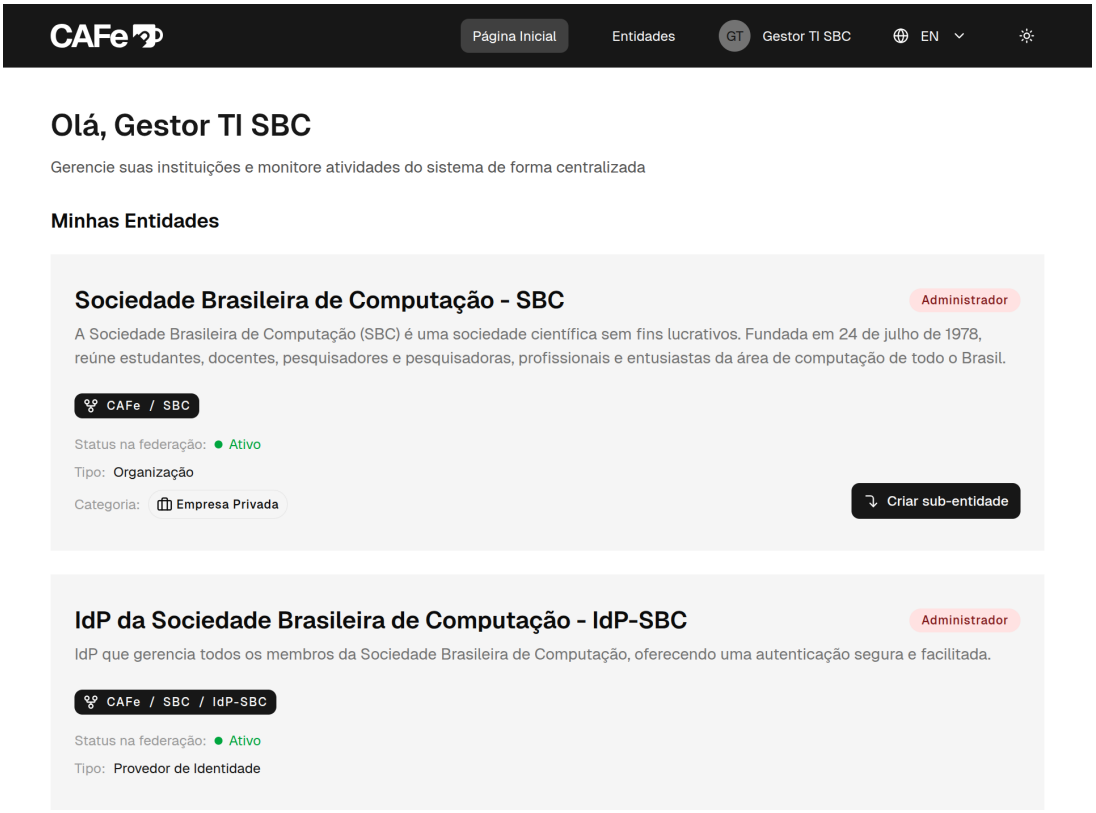
Fonte: o autor.

Figura 76 – Página de detalhes do PI Ativo.



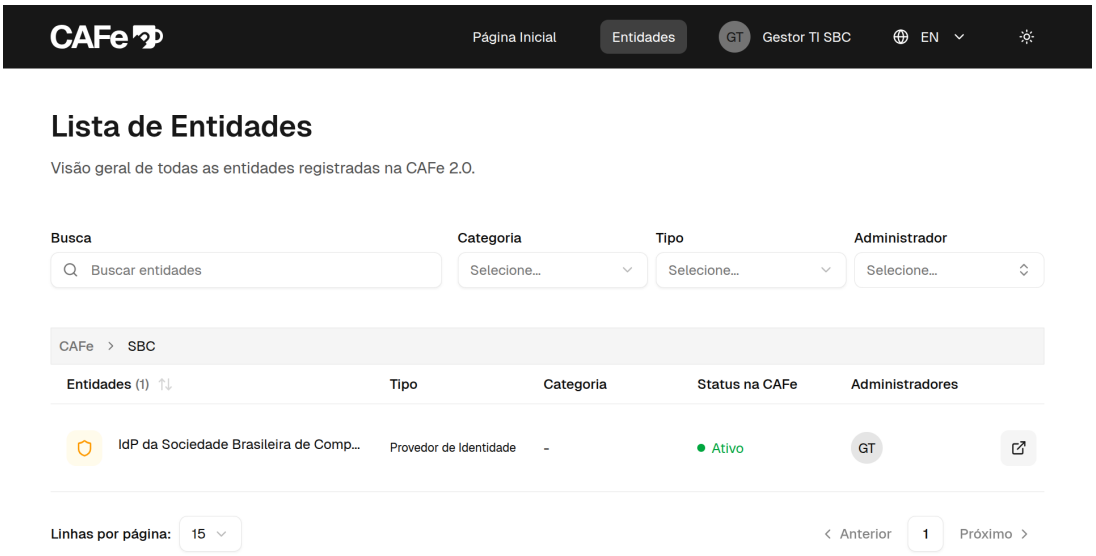
Fonte: o autor.

Figura 77 – Página inicial do gestor após a ativação do PI.



Fonte: o autor.

Figura 78 – O PI listado como subordinado à SBC na Lista de Entidades.



Fonte: o autor.

APÊNDICE B – FLUXO DE INTEGRAÇÃO DE UM PROVEDOR DE SERVIÇO OIDC

Este apêndice ilustra, por meio de capturas de tela do SGB, o fluxo de cadastro de um PS OIDC no BAITA. Como exemplo, registra-se o serviço `More UFSC` sob a organização `Universidade Federal de Santa Catarina (UFSC)`, que se supõe já existente na federação. Diferentemente do fluxo de um PI (Apêndice A), este se encerra com a publicação automática do serviço no PoB.

O cadastro parte de uma organização já existente. Na página de detalhes da `UFSC`, o administrador aciona o menu `Criar sub-entidade` e seleciona `Novo Provedor de Serviço` (Figura 79).

Figura 79 – Início da criação de um PS subordinado à `UFSC`.



Fonte: o autor.

Em seguida, o administrador preenche o formulário de cadastro do serviço (Figura 80). Além do nome oficial, esse formulário reúne os metadados que servirão para a construção da EC da entidade e, consequentemente, para o catálogo do PoB. Informa-se uma descrição, as palavras-chave para categorização, a URL da aplicação, links informativos, uma URL de *health check* para o monitoramento de disponibilidade, as políticas de uso e os termos de serviço, bem como o *upload* de imagens ilustrativas do serviço.

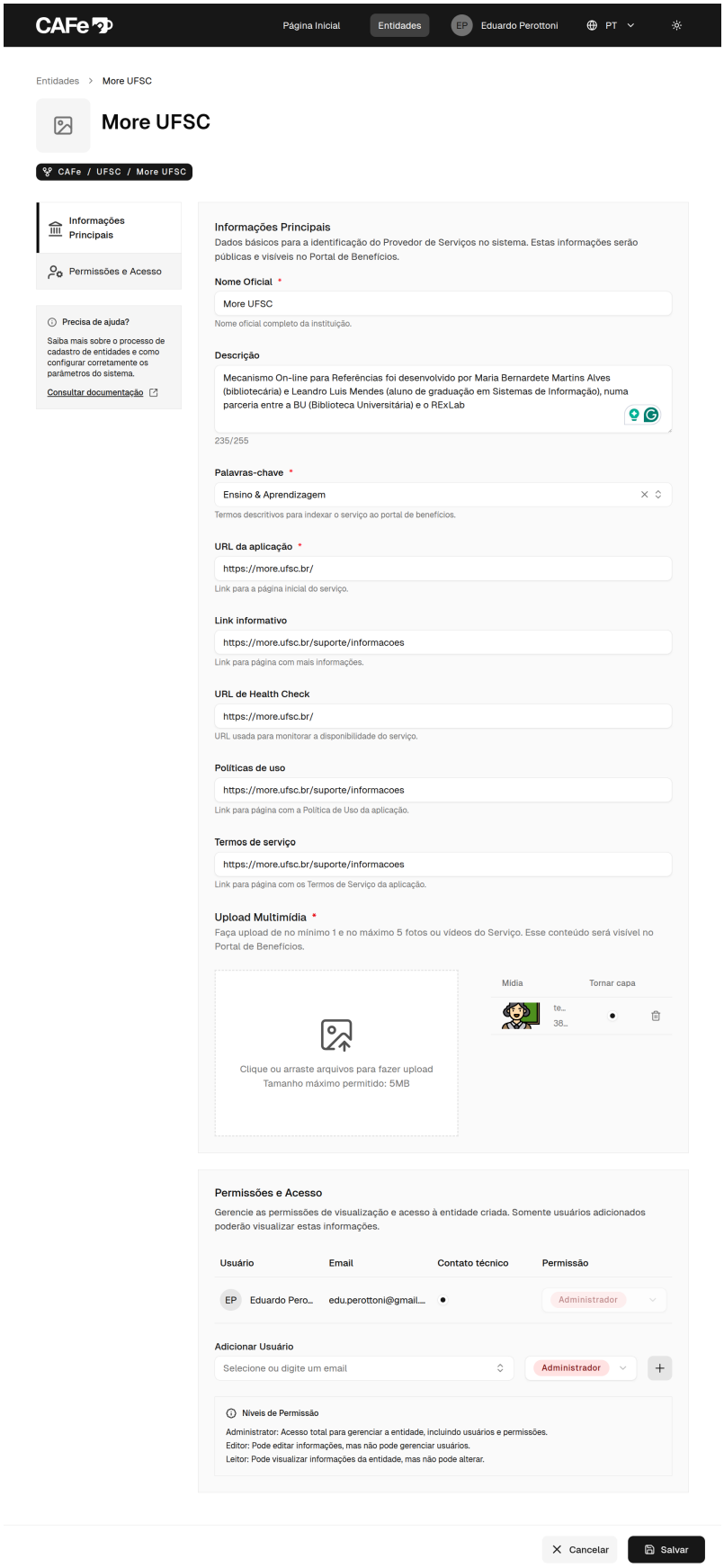
Concluído o preenchimento, o sistema conduz a `Configuração de Acesso`. Essa etapa tem por objetivo executar as configurações técnicas referentes à comunicação entre o serviço e o BAITA para autenticar um usuário. Primeiro, o administrador define se o serviço é *público* (acessível a qualquer pessoa, sem necessidade de autenticação) ou *restrito* (acessível apenas a usuários autenticados pela CAFe). No primeiro caso, não há necessidade de configurações técnicas, uma vez que não haverá comunicação entre a RP e o barramento para autenti-

cação de usuários. No segundo, configurações técnicas devem ser realizadas. Essas opções são ilustradas na Figura 81. Para um serviço restrito, o administrador informa a `redirect URI` da sua aplicação, *endpoint* no qual ela receberá as respostas de autenticação OIDC (Figura 82).

O barramento então gera e apresenta as credenciais que a aplicação deve utilizar para se comunicar com a federação (Figura 83). Nelas estão o *discovery endpoint*, que aponta para o `Proxy-OID-Fed`, o `client_id` e o `client_secret`. Diferentemente do fluxo de um PI, em que o barramento consome um provedor externo, aqui o próprio PS figura como cliente OIDC do Proxy-OID-Fed, o qual lhe expõe uma face de OP OIDC, conforme descrito na Seção 7.4.

Ao concluir, o serviço é cadastrado com sucesso e, sem qualquer ação adicional, recebe automaticamente uma página no PoB, cujo endereço é exibido para o gestor responsável (Figura 84). Por fim, a Figura 85 mostra essa página no PoB. O serviço `More UFSC` aparece catalogado, com a sua descrição, categoria, imagens, políticas e serviços relacionados, pronto para ser descoberto e acessado pelos usuários do barramento. Concretiza-se, assim, a dinamicidade do PoB (Seção 7.7).

Figura 80 – Formulário de cadastro do serviço `More UFSC`.



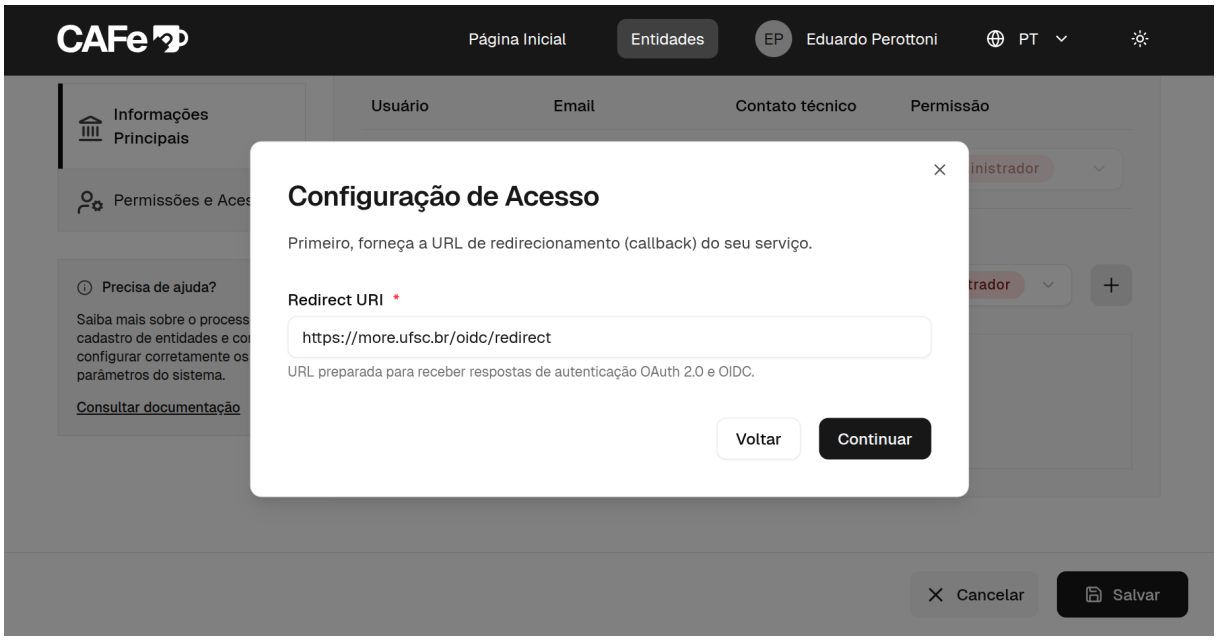
Fonte: o autor.

Figura 81 – Definição do tipo de acesso do serviço: público ou restrito.



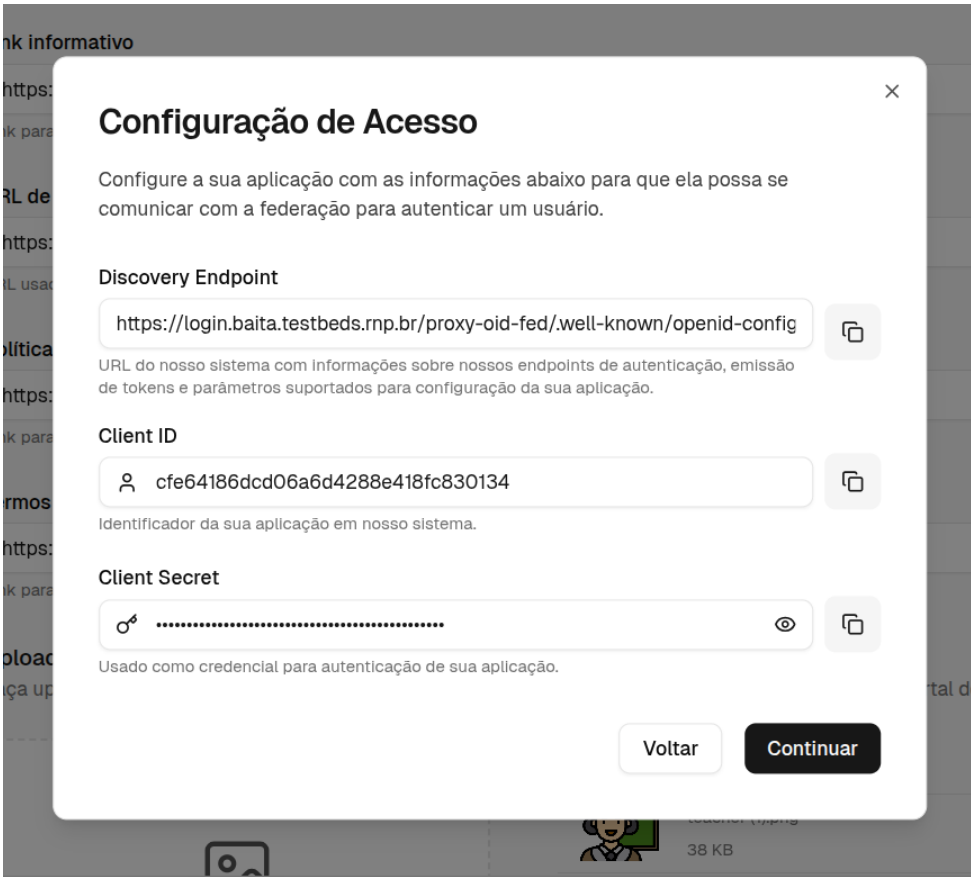
Fonte: o autor.

Figura 82 – Preenchimento da `redirect URI` do serviço.



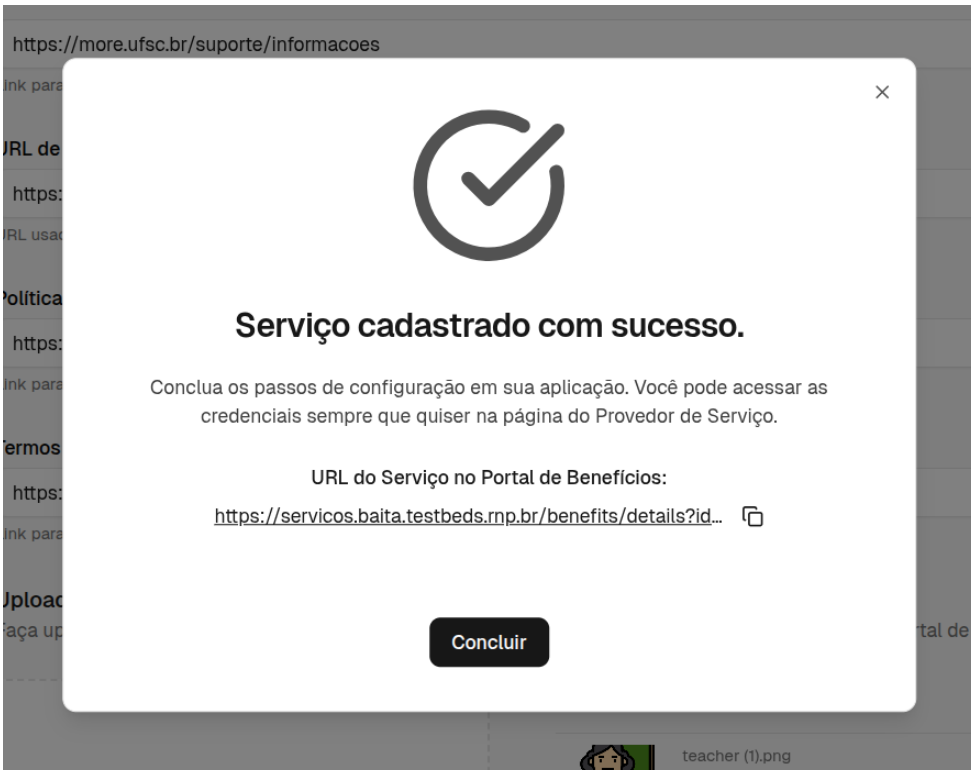
Fonte: o autor.

Figura 83 – Credenciais OIDC fornecidas ao PS.



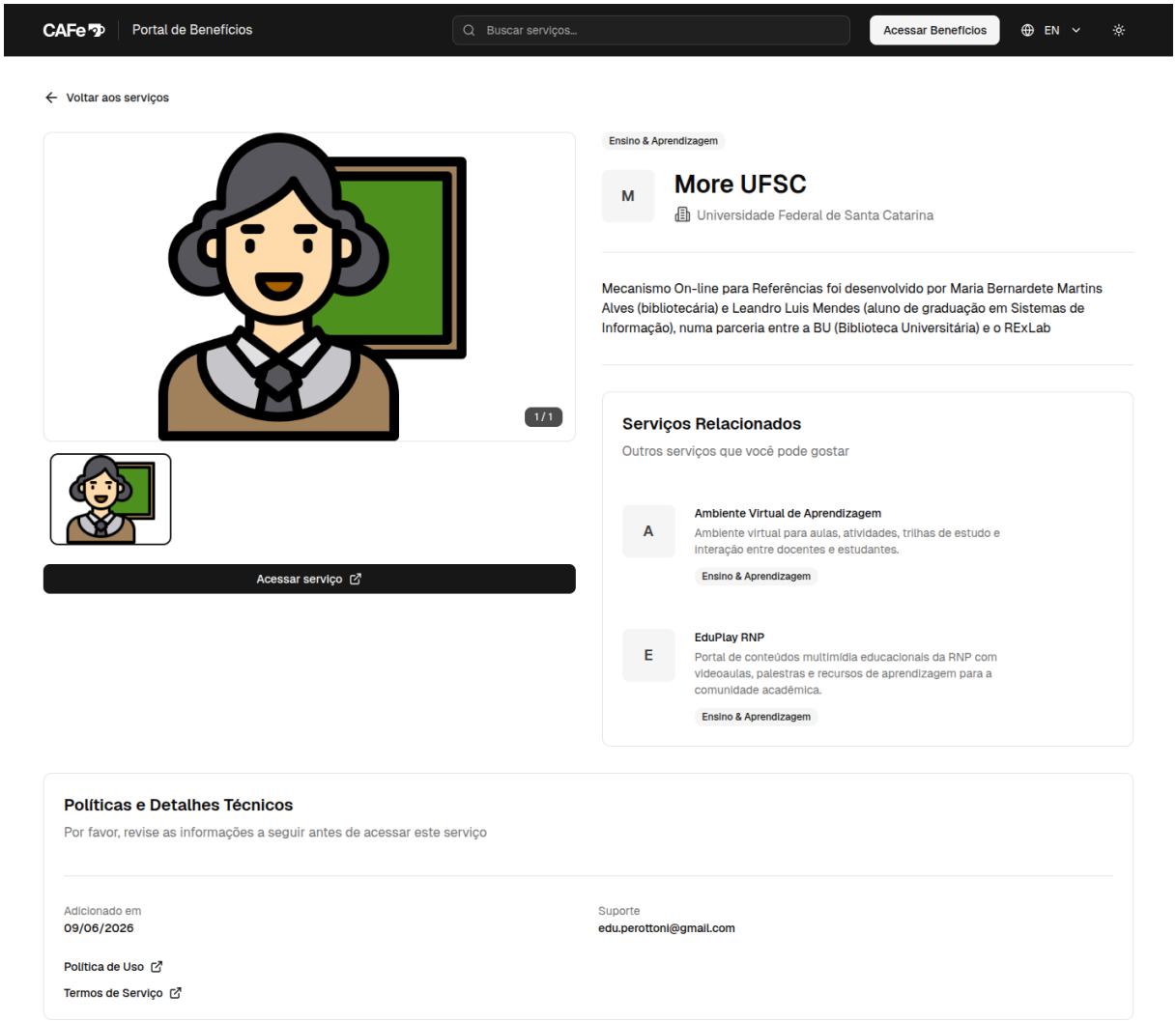
Fonte: o autor.

Figura 84 – Confirmação do cadastro do serviço e o seu endereço no PoB.



Fonte: o autor.

Figura 85 – O serviço `More UFSC` já catalogado no PoB.



Fonte: o autor.

APÊNDICE C – ARTIGO SOBRE O TCC NO FORMATO SBC

BAITA: Barramento de Interoperabilidade Acadêmico Baseado em *OpenID Federation 1.0* para a CAFe

Eduardo Dani Perottoni[1]

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

edu.perottoni@gmail.com

Abstract. *International digital identity federations are migrating from the Security Assertion Markup Language (SAML) 2.0 protocol to the new OpenID Federation 1.0. In this scenario, the Brazilian Federated Academic Community (CAFe) faces significant challenges due to its dependence on SAML, which imposes a complex integration process and operates over a centralized structure. This work proposes and evaluates BAITA, an academic interoperability bus based on OpenID Federation 1.0. A functional prototype comprising a management system, a benefits portal, translation proxies and an enhanced discovery service was implemented and deployed on RNP's testbed environment, validating end-to-end federated authentication with both OIDC and SAML providers.*

Resumo. *As federações de identidade digital internacionais migram do protocolo Security Assertion Markup Language (SAML) 2.0 para o novo OpenID Federation 1.0. Nesse cenário, a Comunidade Acadêmica Federada (CAFe) enfrenta desafios por sua dependência do SAML, que impõe uma integração complexa e opera sobre uma estrutura centralizada. Este trabalho propõe e avalia o BAITA, um barramento de interoperabilidade acadêmico baseado no OpenID Federation 1.0. Um protótipo funcional, composto por um sistema de gestão, um portal de benefícios, proxies de tradução e um serviço de descoberta aprimorado, foi implementado e implantado nos testbeds da RNP, validando a autenticação federada de ponta a ponta com provedores OIDC e SAML.*

1. Introdução

Identidade digital pode ser definida como um conjunto de atributos que descrevem unicamente uma entidade e a sua relação com as demais [Windley 2005]. Quando múltiplas organizações estabelecem uma relação de confiança e passam a reconhecer as mesmas credenciais para conceder acesso a serviços, tem-se uma Federação de Identidade Digital (FID) [Allen 2016]. Essas federações são especialmente relevantes no ambiente educacional, onde a troca de dados entre instituições é frequente.

No contexto brasileiro, a Comunidade Acadêmica Federada (CAFe), serviço oferecido pela Rede Nacional de Ensino e Pesquisa (RNP), conta com mais de 320 instituições conectadas e figura entre as cinco maiores federações do mundo [RNP 2024]. Assim como outras grandes federações, a CAFe utiliza o protocolo SAML 2.0 para o estabelecimento de confiança entre as suas entidades. Embora ainda amplamente adotado, esse protocolo teve o seu desenvolvimento estagnado e é apontado pelo *Research and Education FEDerations group* (REFEDS) como inadequado às novas necessidades

de gestão de identidade [Flanagan 2024]. Além das restrições do SAML, a CAFe impõe políticas rígidas de operação, como a exigência de que os Provedores de Identidade (PIs) sejam aplicações *Shibboleth* e um processo de aprovação longo e manual para a integração de Provedores de Serviço (PSs). Esse contexto resulta em complexidade na integração, hierarquia rígida e baixa adoção: no momento da escrita deste texto, a CAFe conta com 330 PIs, mas apenas 48 PSs registrados.

Para responder a esses desafios, a *OpenID Foundation* desenvolveu o *OpenID Federation 1.0* (OID Fed), um novo protocolo de federação que constrói a confiança de forma hierárquica e dinâmica, dispensando o provisionamento manual de metadados [Hedberg et al. 2026]. Este trabalho propõe e avalia o Barramento de Interoperabilidade Acadêmico (BAITA), cuja arquitetura é construída sobre o OID Fed e, além de modernizar a infraestrutura da CAFe, soluciona problemas específicos do ecossistema de instituições brasileiras. O objetivo geral é propor esse barramento, contemplando um serviço de descoberta de provedores, um sistema de gestão do barramento , um portal de descoberta de serviços e *proxies* de tradução de protocolos, e instanciá-lo em um protótipo que comprove a viabilidade do uso do OID Fed no contexto da CAFe.

O restante deste artigo está organizado como segue: A Seção 2 apresenta a fundamentação teórica sobre identidade federada, SAML, OIDC e o OID Fed; A Seção 3 compara os protocolos SAML e OID Fed; A Seção 4 discute os trabalhos relacionados de migração para o OID Fed; A Seção 5 descreve a metodologia e os desafios que orientaram o trabalho; A Seção 6 detalha a arquitetura do BAITA e os seus componentes; A Seção 7 apresenta e discute os resultados obtidos com o protótipo e, por fim, a Seção 8 conclui o trabalho e aponta perspectivas futuras.

2. Fundamentação Teórica

2.1. Identidade Digital, Federação, SAML e Shibboleth

A gestão de identidades no mundo digital evoluiu por meio de sucessivos modelos [Wangham et al. 2010]. No modelo isolado, cada serviço mantém e autentica as próprias credenciais, o que sobrecarrega o usuário com múltiplas senhas e amplia a superfície de ataque. O modelo centralizado introduziu o *Single Sign-On* (SSO), no qual um único PI autentica o usuário para diversos serviços de um mesmo domínio [Jøsang and Pope 2005]. A federação estende esse conceito para além das fronteiras de um domínio. Múltiplas organizações estabelecem acordos de confiança de modo que uma identidade emitida em uma instituição seja reconhecida pelas demais [Aldosary and Alqahtani 2021].

O modelo de identidade federada envolve três papéis principais: o usuário; o Provedor de Identidade (PI), responsável por autenticar o usuário e emitir asserções sobre a sua identidade; e o Provedor de Serviço (PS), que consome essas asserções para conceder acesso a um recurso. O *Security Assertion Markup Language* (SAML) 2.0 é a especificação que padroniza a troca dessas asserções entre PI e PS [OASIS 2005]. Uma asserção SAML é um documento XML, assinado digitalmente, que transporta declarações de autenticação e atributos do usuário entre domínios de segurança distintos. A especificação define ainda os *bindings* (formas de transporte das mensagens, tipicamente sobre HTTP) e os perfis que combinam esses elementos para atender a casos de uso como o SSO na *web*.

Na CAFe, os PIs são operados por meio do *Shibboleth*, um conjunto de *software* de código aberto que implementa o SAML e é amplamente adotado por federações acadêmicas para prover autenticação federada e SSO [Consortium 2025]. A confiança na CAFe é gerenciada de forma centralizada. O operador da federação coleta os metadados de cada participante, consolida-os em um único e extenso arquivo XML e o distribui periodicamente a todos os membros. Uma entidade só é reconhecida pelas demais após constar nesse arquivo, o que torna o processo de integração dependente de um ciclo manual de aprovação e de propagação de metadados. Esse modelo, somado à exigência de operar o *Shibboleth*, cria uma barreira técnica relevante, sobretudo para instituições que já dispõem de provedores baseados em protocolos mais modernos e precisam manter uma segunda infraestrutura exclusivamente para a federação.

2.2. OAuth 2.0 e OpenID Connect

O OAuth 2.0 é um *framework* de autorização que permite a uma aplicação obter acesso limitado a recursos protegidos de um usuário sem que este compartilhe as suas credenciais [Hardt 2012]. O OpenID Connect (OIDC) é uma camada de identidade construída sobre o OAuth 2.0, estendendo-o com funcionalidades de autenticação e permitindo que aplicações-cliente verifiquem a identidade do usuário final [Sakimura et al. 2023]. A principal extensão do OIDC é o *ID Token*, um *token* de segurança representado como um *JSON Web Token* (JWT) [Jones et al. 2015] que carrega *claims* sobre o evento de autenticação. Diferentemente do SAML, o OIDC utiliza formatos leves como JSON e é amplamente adotado em aplicações *web* e *mobile* contemporâneas.

2.3. OpenID Federation 1.0

O OID Fed descreve métodos para que a confiança entre duas entidades seja estabelecida por meio da confiança de ambas em uma terceira parte, chamada *Trust Anchor* (TA) [Hedberg et al. 2026]. O protocolo define três papéis: a TA, âncora de confiança que emite declarações sobre as entidades imediatamente subordinadas; a *Intermediate Entity* (IM), entidade que se situa entre a TA e um provedor na hierarquia e também emite declarações sobre os seus subordinados; e a *Leaf Entity* (LE), entidade folha que desempenha um papel no protocolo de autenticação, como um PI ou um PS.

O componente básico do protocolo é a Declaração de Entidade (*Entity Statement*), um JWT assinado que contém as informações necessárias para que a entidade participe da federação. Toda entidade emite uma declaração autoassinada sobre si mesma, a Configuração da Entidade (*Entity Configuration*, EC), enquanto as entidades superiores publicam uma Declaração de Subordinado (*Subordinate Statement*, SS) para cada entidade abaixo delas. A confiança entre uma LE e uma TA é estabelecida dinamicamente por meio da validação de uma Cadeia de Confiança (*Trust Chain*, TC). Esta, é dada por uma sequência de declarações assinadas que parte da EC do sujeito, percorre os SSs emitidos pelas entidades intermediárias e termina na TA confiada pela entidade que executa a validação. Esse modelo dispensa a distribuição de um arquivo central de metadados e permite o registro descentralizado de provedores.

Além do estabelecimento de confiança, o OID Fed oferece mecanismos de governança. Por meio das Políticas de Metadados, as entidades superiores podem impor restrições aos metadados dos seus subordinados. Por exemplo, exigir determinados algoritmos de assinatura ou tipos de concessão OAuth 2.0. Essas políticas sãoaplicadas de

cima para baixo ao longo da cadeia de confiança. Os *Trust Marks* (TMs), por sua vez, são declarações que atestam a conformidade de uma entidade a um perfil específico, sendo usados, por exemplo, para indicar o pertencimento de uma entidade a uma federação específica [Connect2id 2024].

3. SAML 2.0 *versus* OpenID Federation

Embora ambos os protocolos enderecem a federação de identidade, eles adotam abordagens distintas em arquitetura, gerenciamento de confiança e flexibilidade. A diferença mais fundamental está no modelo de confiança. No SAML, a confiança é centralizada em um arquivo de metadados mantido pelo operador e distribuído a todos os membros; uma entidade só é reconhecida se constar nesse arquivo [Jones 2020]. Esse modelo é considerado frágil e pouco escalável. Por outro lado, o OID Fed constrói a confiança de forma hierárquica e dinâmica. O operador precisa registrar apenas as organizações diretamente subordinadas, e a confiança entre duas entidades quaisquer é resolvida em tempo real pela validação da cadeia de confiança.

O registro de provedores no SAML é centralizado e pode ser demorado, pois depende da inclusão da entidade no arquivo central e da propagação desse arquivo a todos os participantes. No OID Fed, o registro é dinâmico: quando um PS inicia uma requisição de autenticação, o PI pode construir e validar a cadeia de confiança em tempo real, permitindo o registro imediato. Há ainda diferenças no formato das mensagens. O SAML baseia-se em XML e em protocolos como o SOAP, enquanto o OID Fed adota JSON e APIs no estilo REST, mais leves e alinhadas às práticas de desenvolvimento atuais. Outras divergências estão na validação de metadados, mais granular no OID Fed graças às políticas de metadados. Adicionalmente, no SAML, a participação de uma mesma entidade em múltiplas federações é incomum e, em alguns casos, desaconselhada, como na *eduGAIN*, cujo modelo prevê que cada entidade seja registrada em uma única federação de origem [UK federation sd]. No OID Fed, o *design* não impõe desvantagens a essa participação. Por fim, o OID Fed, por ser construído sobre o OAuth 2.0, herda soluções robustas para aplicações *mobile* nativas e *Single Page Applications*, como o *Proof Key for Code Exchange* (PKCE) [Sakimura et al. 2015] e a especificação para aplicações nativas [Denniss and Bradley 2017], cenários não cobertos nativamente pelo SAML. A Tabela 1 sintetiza essas diferenças.

Tabela 1. Comparação entre a federação SAML e o *OpenID Federation*.

Característica	SAML/Shibboleth	OpenID Federation
Suporte a aplicações *mobile*	Limitado	Otimizado
Dinamicidade	Baixa	Alta
Interoperabilidade	Limitada	Alta
Integração	Complexa, manual	Automatizada, dinâmica
Escalabilidade	Limitada	Alta
Modelo de confiança	Bilateral, centralizado	Descentralizado, âncoras
Formato de mensagens	XML/SOAP	APIs REST/JSON
Adaptação a novas tecnologias	Lenta	Rápida

4. Trabalhos Relacionados

A migração do SAML para o OID Fed é uma tendência internacional, com iniciativas relevantes que oferecem modelos para a CAFe. O projeto diretamente análogo é o piloto de OID Fed da *eduGAIN*, conduzido pela GÉANT, rede pan-europeia que opera a interfederação acadêmica global [van Dijk and Vaghetti 2024]. Adicionalmente, um grupo de trabalho desenvolve um Perfil Tecnológico oficial da *eduGAIN* para o OID Fed [Harris and Vaghetti 2025]. A arquitetura do piloto prevê a *eduGAIN* como TA, as redes nacionais de educação e pesquisa como IMs e os provedores como LEs, empregando TMs para determinar o pertencimento à interfederação [van Dijk 2024]. O piloto, contudo, ainda não trata do suporte a múltiplos protocolos nem da coexistência com as federações SAML existentes [Dreef et al. 2025], lacunas que o BAITA preenche.

A mais significativa implementação do OID Fed em produção é o projeto SPID/CIE, do governo italiano, que unificou os seus dois esquemas nacionais de identidade digital sob o OID Fed [De Marco 2022, Agenzia per l'Italia Digitale (AgID) 2024]. O projeto disponibilizou *SDKs* de código aberto em diversas linguagens e propôs um IM que atua como *proxy* entre a federação e o PS, abstraindo as complexidades do OID Fed e centralizando a gestão de chaves, metadados e cadeias de confiança. Esse padrão de *proxy* inspira diretamente a estratégia de adoção adotada pelo BAITA. Por fim, um trabalho correlato aborda a interoperabilidade entre SAML e OIDC no contexto do *framework* europeu *eIDAS 2.0*, propondo um *proxy* de tradução que encapsula a complexidade da conversão de protocolos em um único componente [Morabito 2023]. Essa abordagem de ponte, de menor risco, é análoga à necessidade da CAFe de coexistir com o SAML durante a transição tecnológica entre protocolos.

5. Metodologia

O desenvolvimento do BAITA ocorreu no âmbito do Grupo de Trabalho GT-BAITA, fomentado pelo Programa de Serviços Avançados da RNP, e combinou uma fase de prospecção com o *design* iterativo da arquitetura e a implementação colaborativa do código-fonte. A prospecção seguiu a metodologia *Customer Discovery* [Blank and Dorf 2012], apoiada no *Business Model Canvas* [Osterwalder and Pigneur 2010], e consistiu em 61 entrevistas com *stakeholders* do ecossistema da CAFe, conduzidas entre julho e setembro de 2025. A análise dessas entrevistas revelou seis desafios centrais, sintetizados na Tabela 2. Esses desafios foram convertidos em premissas de *design* que orientaram a arquitetura proposta e são retomados, ao longo deste artigo, na descrição do componente concebido para endereçar cada um.

6. O Barramento de Interoperabilidade Acadêmico (BAITA)

6.1. Visão Geral da Arquitetura

O BAITA não consiste apenas na implementação de uma federação OID Fed, mas em um ecossistema que agrega facilidades de gestão e de descoberta de serviços. A Figura 1 apresenta os seus componentes em alto nível, organizados em três contextos: a infraestrutura da CAFe atualmente em produção, o próprio barramento e a instituição federada. A federação OID Fed, denominada *CAFe 2.0*, tem como núcleo o Sistema de Gestão do Barramento (SGB), sendo complementada pelo serviço de descoberta *Where Are You From* (WAYF), pelos *proxies* de interoperabilidade e pelo Portal de Benefícios (PoB).

Tabela 2. Desafios centrais identificados na fase de prospecção.

#	Desafio
1	Alto atrito na integração de provedores
2	Ausência de interoperabilidade semântica e de *Múltiplos Fatores de Autenticação* (MFA) consistente
3	Necessidade de coexistência com a federação legada SAML
4	Falta de controle de escopo dos provedores
5	Operação manual e custosa da federação
6	Invisibilidade dos serviços federados

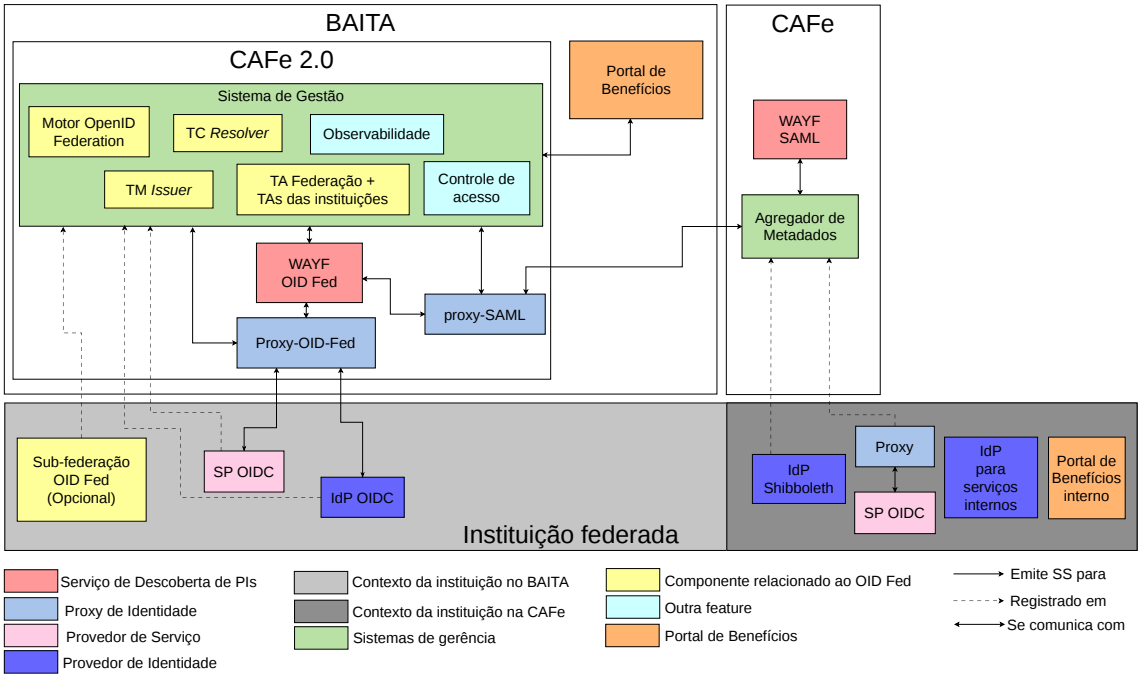


Figura 1. Componentes do BAITA em alto nível, organizados nos contextos da CAFe em produção, do barramento e da instituição federada.

6.2. Sistema de Gestão do Barramento (SGB)

O SGB é o componente por meio do qual os operadores da RNP e os administradores das instituições gerenciam o ciclo de vida das entidades do barramento. Ele materializa um modelo de administração descentralizada, cujo operador de uma organização superior delega a gestão de uma subentidade a um administrador responsável, que passa a gerir as suas próprias entidades sem intervenção externa. O SGB custodia as chaves de federação das entidades hospedadas e assina, em seu nome, os artefatos do OID Fed, gerando-os e rotacionando-os automaticamente. Essa abordagem reduz drasticamente o atrito de adesão. Integrar um provedor resume-se a preencher formulários guiados em interface *web*, sem que a instituição precise implementar as extensões do OID Fed ou operar infraestrutura adicional, endereçando o Desafio 1. O SGB oferece ainda trilha de auditoria, gestão de permissões e configuração de fatores de autenticação (senha, *passkey* FIDO2/WebAuthn e OTP/TOTP).

6.3. Descoberta e Normalização com o WAYF

O WAYF atua como intermediário entre PS e PI, permitindo que o usuário final selecione a sua instituição de origem. Embora o OID Fed resolva o estabelecimento de confiança, ele não aborda a interoperabilidade sintática e semântica dos atributos. Para isso, o WAYF é projetado para, além de descobrir o PI do usuário, atuar como normalizador de atributos e garantidor de MFA, endereçando o Desafio 2. Nessa tela de descoberta, os provedores OIDC registrados e os PIs SAML legados convivem de forma homogênea, sem que o usuário perceba distinção entre eles.

6.4. Proxies de Interoperabilidade e Fluxo de Integração

Dois *proxies* viabilizam a coexistência de tecnologias. O Proxy-OID-Fed permite que provedores baseados em OIDC participem da federação sem implementar as complexidades do OID Fed, expondo a RP e intermediando o OP registrado. O Proxy-SAML lê o arquivo de metadados da CAFe SAML e expõe cada PI SAML legado como um OP na nova federação, atendendo RPs modernas que autenticam usuários em PIs legados. Dessa forma, o barramento oferece suporte gradual ao legado SAML (Desafio 3), sem exigir alterações nos servidores *Shibboleth* das instituições.

A Figura 2 detalha, como um processo em notação BPMN, o fluxo de integração de um PS por meio do Proxy-OID-Fed, dividido em três fases. Na Fase 1 (Registro), o gestor do PS solicita o registro da RP, informando dados como as *redirect URIs*; o SGB registra e configura a entidade, gerando a chave de federação e os *authority hints*, e devolve as credenciais OIDC (`client_id` e `client_secret`) e os *endpoints* do *proxy*. Durante a Fase 2 (Configuração), o SGB assina e publica as declarações de subordinado sobre a RP registrada, e o gestor apenas configura a sua aplicação para usar o *proxy* como OP. Na Fase 3 (Representação), quando um usuário inicia a autenticação, o `Proxy-OID-Fed` consulta o registro da RP, monta e assina a sua Configuração da Entidade, resolve a cadeia de confiança atuando como uma RP OID Fed e redireciona o fluxo ao OP *upstream*. Ao final, o *proxy* entrega os *tokens* à aplicação. Assim, o provedor torna-se operacional e confiável na federação embora continue a falar apenas OIDC, reduzindo o atrito de adesão (Desafio 1).

6.5. Portal de Benefícios (PoB)

O PoB é a porta de entrada do usuário final no ecossistema, funcionando como um catálogo de descoberta dos serviços disponíveis na federação e endereçando o Desafio 6. Todo PS registrado no barramento é automaticamente descoberto por um algoritmo que percorre a topologia da federação e passa a ser exibido no portal, sem cadastro manual adicional. O catálogo oferece busca textual, filtros por categoria e um *ranking* de serviços mais acessados, calculado a partir de métricas coletadas por uma camada de observabilidade.

7. Resultados e Discussão

Um protótipo funcional do barramento foi implementado e implantado em ambiente de homologação na infraestrutura de *testbeds* da RNP[1], acessível publicamente. Foram materializados o SGB, o PoB, o Proxy-OID-Fed, o Proxy-SAML e o WAYF, além de uma

[1] Ambiente de homologação do BAITA: `https://baita.testbeds.rnp.br`.

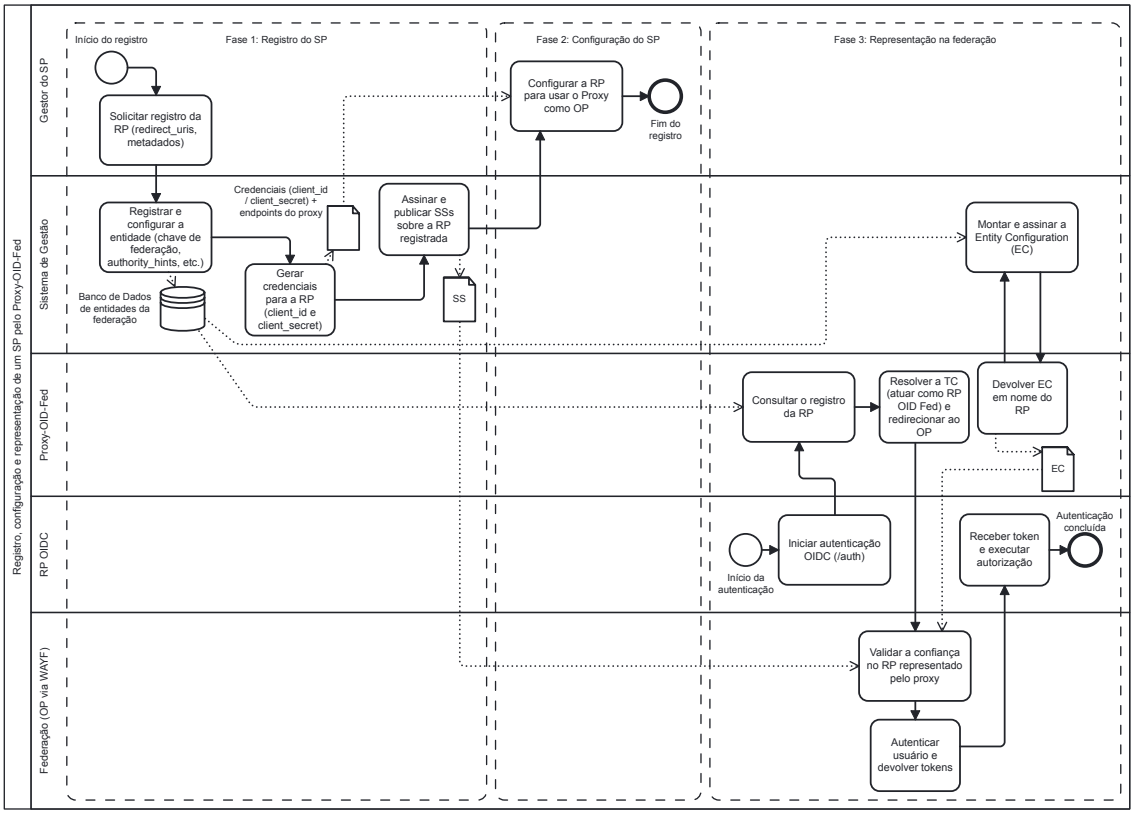


Figura 2. Fluxo de registro, configuração e representação de um PS pelo Proxy-OID-Fed, em notação BPMN.

camada de observabilidade. Os *proxies* foram construídos sobre o *SATOSA*, enquanto o WAYF reaproveita o *Keycloak* estendido por meio de *Service Provider Interfaces* customizadas.

Para avaliar a interoperabilidade de ponta a ponta, exercitou-se o fluxo de autenticação federada de uma aplicação de teste até dois PIs distintos, ambos mediados pelo WAYF: um provedor SAML legado, intermediado pelo Proxy-SAML, e um OP OIDC registrado no barramento, intermediado pelo Proxy-OID-Fed. Em ambos os caminhos, a aplicação recebeu os atributos do usuário de forma uniforme, sem conhecer o protocolo ou a tecnologia do provedor de origem. No caminho SAML, atributos próprios dos *schemas* utilizados nas federações SAML, como `eduPersonPrincipalName`, foram entregues à aplicação já como *claims* OIDC, comprovando a interoperabilidade entre os dois protocolos de federação.

Os resultados evidenciam um *trade-off* deliberado entre facilidade de adoção, de um lado, e descentralização e soberania, de outro. Ao custodiar as chaves e assinar os artefatos das entidades hospedadas, o SGB torna-se um ponto de concentração de confiança e um potencial ponto único de falha. Contudo, essa centralização não é compulsória. Instituições que demandem controle total podem implementar o OID Fed nativamente e ancorar a sua própria sub-árvore à federação, dispensando o *proxy*. De modo análogo, o papel central do WAYF, capaz de observar quem se autentica em quais serviços, impõe cuidados de privacidade, mitigáveis por medidas de minimização de dados, alinhadas às diretrizes de agregação de atributos da comunidade acadêmica.

Cabe ressaltar que o BAITA é uma iniciativa em evolução no momento de escrita deste texto. Parte das funcionalidades projetadas, como a gestão de subfederações (Desafio 4) e o suporte completo ao SAML, permanece em estágio de projeto, registrada entre os trabalhos futuros.

8. Conclusão

Este trabalho propôs e avaliou o BAITA, um barramento de interoperabilidade acadêmico fundamentado no *OpenID Federation 1.0*, com o intuito de integrar provedores de identidade e de serviço no contexto da CAFe. Mais do que propor a arquitetura, o trabalho a instanciou em um protótipo funcional, implantado nos *testbeds* da RNP, comprovando a viabilidade do uso do protocolo para esse fim e a interoperabilidade entre os mundos OIDC e SAML. Como trabalhos futuros, destaca-se a gestão de subfederações, a instanciação das políticas de metadados pela TA da CAFe e o suporte completo ao SAML, preparando a CAFe para inovações como a Identidade Digital Descentralizada.

Referências

Agenzia per l'Italia Digitale (AgID) (2024). Spid - sistema pubblico di identità digitale.

Aldosary, M. and Alqahtani, N. (2021). A survey on federated identity management systems limitation and solutions. *International Journal of Network Security & Its Applications (IJNSA) Vol*, 13.

Allen, C. (2016). The path to self-sovereign identity. *Life with Alacrity*.

Blank, S. and Dorf, B. (2012). *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company*. K&S Ranch, Pescadero, CA.

Connect2id (2024). OpenID Federation and the trust chain explained.

Consortium, S. (2025). The shibboleth project.

De Marco, G. (2022). Work begins on unified spid and cie openid connect sdks. Accessed on June 22, 2025.

Denniss, W. and Bradley, J. (2017). OAuth 2.0 for Native Apps. RFC 8252.

Dreef, C., Vaghetti, D., and Kremers, M. (2025). Infoshare openid federation pilot (edugain / gn5-2 t&i).

Flanagan, H. (2024). Is there a future for refeds and r&e federations?

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

Harris, N. and Vaghetti, D. (2025). edugain technical profiles working group. Accessed on 2025-06-22.

Hedberg, R., Jones, M. B., Solberg, A. Å., Bradley, J., De Marco, G., and Dzhuvinov, V. (2026). OpenID Federation 1.0.

Jones, M. B. (2020). Enabling scalable multi-lateral federations with OpenID Connect. Presentation at Identiverse 2020 Virtual. Accessed: 2025-06-11.

Jones, M. B., Bradley, J., and Sakimura, N. (2015). JSON Web Token (JWT). RFC 7519.

Jøsang, A. and Pope, S. (2005). User centric identity management. In *AusCERT Asia Pacific information technology security conference*, volume 22, page 2005. Citeseer.

Morabito, G. (2023). Integrazione di oauth 2.0 e openid connect nell'infrastruttura pan-europea di identità digitale. Tesi di laurea magistrale (master's thesis), Politecnico di Torino. Accessed on June 22, 2025.

OASIS (2005). Saml core specification version 2.0.

Osterwalder, A. and Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons, Hoboken, NJ.

RNP (2024). Comunidade acadêmica federada (cafe).

Sakimura, N., Bradley, J., and Agarwal, N. (2015). Proof Key for Code Exchange by OAuth Public Clients. RFC 7636.

Sakimura, N., Bradley, J., Jones, M. B., de Medeiros, B., and Mortimore, C. (2023). Openid connect core 1.0.

UK federation (s.d.). Participation in eduGAIN. `https://www.ukfederation.org.uk/content/Documents/EduGAINParticipation`. Operado pela Jisc. Acesso em: 10 jul. 2026.

van Dijk, N. (2024). Pilot components. Accessed on June 22, 2025.

van Dijk, N. and Vaghetti, D. (2024). edugain oid federation pilot.

Wangham, M. S., de Mello, E. R., da Silva Böger, D., Guerios, M., and da Silva Fraga, J. (2010). Gerenciamento de identidades federadas. *MINICURSOS X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 1–52.

Windley, P. (2005). *Digital Identity: Unmasking Identity Management Architecture (IMA)*. O'Reilly Media.