[PT] Open Finance Brasil Financial-grade API Security Profile 1.0 Implementers Draft 3

Prefácio

The normative version in English

A Estrutura Inicial do Open Finance Brasil (EIOFB) é responsável por criar padrões e especificações necessárias para atender aos requisitos e obrigações da Legislação do Open Finance do Brasil, conforme originalmente delineado pelo Banco Central do Brasil. É possível

que alguns dos elementos deste documento estejam sujeitos a direitos autorais ou patenteados. O EIOFB não se responsabiliza pela identificação de qualquer ou todos esses direitos.

O Financial-grade API 1.0 do Open Finance Brasil consiste nas seguintes partes:

- Open Finance Brasil Financial-grade API Security Profile 1.0¶
- Open Finance Brasil Dynamic Client Registration Profile 1.0

Estas partes são destinados a serem usados com RFC6749, RFC6750, RFC7636, OIDC, FAPI-1-Baseline e FAPI-1-Advanced

Introdução

A Financial-grade API do Open Finance Brasil é um perfil OAuth altamente seguro que visa fornecer diretrizes de implementação específicas para segurança e interoperabilidade que podem ser aplicadas a APIs na área de Open Finance do Brasil que requerem um nível de privacidade superior ao fornecido pelo padrão Financial-grade API Security Profile 1.0 - Part 2: Advanced. Entre outras melhorias, esta especificação aborda considerações de privacidade identificadas em FAPI-1-Advanced que são relevantes nas especificações do Open Finance Brasil, mas não foram, até agora, exigidas por outras jurisdições.

Embora seja possível codificar um provedor de OpenID e parte de confiança a partir dos primeiros princípios usando esta especificação, o público principal para esta especificação são as partes que já possuem uma implementação certificada do Financial-grade API Security Profile 1.0 - Part 2: Advanced e deseja obter a certificação para o programa Brasil Open Finance.

Convenções Notacionais

As palavras-chave "deve" (shall), "não deve" (shall not), "deveria" (should), "não deveria" (should not) e "pode" (may) presentes nesse documento devem ser interpretadas conforme as diretrizes descritas em ISO Directive Part 2 observando seguinte equivalência:

- "deve" ⇒ equivalente ao termo "shall" e expressa um requerimento definido no documento (nas traduções é similar ao termo "must", que pode denotar um requerimento externo ao documento);
- "não deve" ⇒ equivalente ao termo "shall not" e também expressa um requerimento definido no documento;
- "deveria" e "não deveria"⇒ equivalente ao termo "should" e "should not" e expressa uma recomendação
- "pode" ⇒ equivalente ao termo "may" indica uma permissão

Estas palavras-chave não são usadas como termos de dicionário, de modo que qualquer ocorrência deles deve ser interpretada como palavras-chave e não devem ser interpretados com seus significados de linguagem natural.

1. Escopo

Este documento especifica o método para os aplicativos

- obterem de maneira segura os tokens OAuth necessários para acesso a dados críticos de acordo com os requisitos do Open Finance Brasil;
- utilizarem o OpenID Connect para identificação do usuário do Open Finance; e
- utilizarem o OpenID Connect para afirmar a identidade do cliente.

Este documento é aplicável a todos os participantes do Open Finance no Brasil.

2. Referências normativas

Os seguintes documentos referenciados são indispensáveis para a adoção das especificações deste documento. Para referências datadas, apenas a edição citada se aplica. Para referências não datadas, deve-se aplicar a última edição do documento referenciado (incluindo quaisquer emendas).

ISODIR2 - ISO/IEC Directives Part 2

RFC6749 - The OAuth 2.0 Authorization Framework

RFC6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage

RFC7636 - Proof Key for Code Exchange by OAuth Public Clients

RFC6819 - OAuth 2.0 Threat Model and Security Considerations

RFC7515 - JSON Web Signature (JWS)

RFC7519 - JSON Web Token (JWT)

RFC7591 - OAuth 2.0 Dynamic Client Registration Protocol

RFC7592 - OAuth 2.0 Dynamic Client Registration Management Protocol

BCP195 - Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)

OIDC - OpenID Connect Core 1.0 incorporating errata set 1

FAPI-CIBA - Financial-grade API: Client Initiated Backchannel Authentication Profile

OIDD - OpenID Connect Discovery 1.0 incorporating errata set 1

OIDR - OpenID Connect Registration 1.0 incorporating errata set 1

RFC8705 - OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens

JARM - Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)

PAR - OAuth 2.0 Pushed Authorization Requests

JAR - OAuth 2.0 JWT Secured Authorization Request

FAPI-1-Baseline - Financial-grade API Security Profile 1.0 - Part 1: Baseline

FAPI-1-Advanced - Financial-grade API Security Profile 1.0 - Part 2: Advanced

FAPI-2-Baseline - Financial-grade API Security Profile 2.0 - Part 1: Baseline

LIWP - OIDF FAPI WG Lodging Intent Working Paper

LIWP - OIDF FAPI WG Lodging Intent Working Paper

OFB-FAPI-DCR - Open Finance Brasil Financial-grade API Dynamic Client Registration Profile 1.0

RFC4648 - The Base16, Base32, and Base64 Data Encodings

3. Termos e definições

Para efeitos deste documento, os termos definidos em RFC6749, RFC6750, RFC7636, OpenID Connect Core e ISO29100 se aplicam.

4. Símbolos e termos abreviados

- **API** - Application Programming Interface (Interface de programação de aplicativo)
- **CSRF** - Cross Site Request Forgery
- **DCR** - Registro de cliente dinâmico
- **EIOFB** - Estrutura Inicial do Open Finance Brasil
- **FAPI** - Financial-grade API
- **HTTP** - Protocolo de transferência de hipertexto
- **MFA** - Multi-Factor Authentication (Autenticação por Múltiplos Fatores)
- **OIDF** - OpenID Foundation
- **REST** - Representational State Transfer (Transferência de Estado Representacional)
- **TLS** - Transport Layer Security (Segurança da Camada de Transporte)

5. Profile de Segurança para o Open Finance Brasil

5.1. Introdução

O perfil de segurança do Open Finance Brasil especifica requisitos adicionais de segurança e de identificação para o acesso a API's com recursos críticos protegidas pelo OAuth 2.0 Authorization Framework, que consiste em RFC6749, RFC6750, RFC7636, FAPI-1-Baseline, FAPI-1-Advanced e outras especificações.

Este perfil descreve as capacidades e os recursos de segurança que devem ser oferecidos por servidores e clientes que são necessários para o Programa do Open Finance Brasil, definindo as medidas para mitigar ou endereçar:

- ataques que abordam considerações de privacidade identificadas na cláusula 9.1 de [FAPI-1 Advanced].
- o requisito de concessão de acesso granular a recursos, com vistas à minimização de dados;
- o requisito de informar sobre o contexto da autenticação do usuário (claim Authentication Context Request - acr) que foi realizada por um Provedor OpenID, com vistas a favorecer o adequado gerenciamento do risco decorrente do acesso do usuário;

5.2. Disposições de segurança do Open Finance Brasil

5.2.1. Introdução

O Open Finance Brasil tem um requisito para endereçar considerações de privacidade que foram identificadas, mas não abordadas na especificação final FAPI-1-Advanced, sem impor requisitos adicionais aos Servidores de Autorização que estão sendo propostos em FAPI-2-Baseline.

Este perfil descreve os requisitos de gerenciamento de clientes necessários para dar suporte ao ecossistema Open Finance Brasil mais amplo.

Como um perfil do OAuth 2.0 Authorization Framework, este documento exige o seguinte para o perfil de segurança do Open Finance Brasil.

5.2.2. Servidor de Autorização

O Servidor de Autorização deve suportar as disposições especificadas na cláusula 5.2.2 de Financial-grade API Security Profile 1.0 - Parte 2: Advanced.

Além disso, ele deve:

1. deve realizar a autenticação do cliente utilizando private_key_jwt;
2. deve exigir requisições do tipo "pushed authorization requests" PAR;
3. deve publicar metadados de descoberta (incluindo a do endpoint de autorização) por meio do documento de metadado especificado em OIDD e [RFC8414] (".well-known");

4. deve suportar os parâmetros claims como definido no item 5.5 do OpenID Connect Core;
5. deve suportar o atributo acr "urn:brasil:openbanking:loa2" como definido no item 5.2.2.3;
6. deveria suportar o atributo acr "urn:brasil:openbanking:loa3" como definido no item 5.2.2.3;
7. deve implementar o endpoint "userinfo" como definido no item 5.3 do OpenID Connect Core;
8. deve suportar o escopo parametrizável ("parameterized OAuth 2.0 resource scope") consent como definido no item 6.3.1 de OIDF FAPI WG Lodging Intent Pattern;
9. pode suportar Financial-grade API: Client Initiated Backchannel Authentication Profile;
10. (requisito temporariamente retirado);
11. deve suportar refresh tokens;
12. deve emitir access tokens com o tempo de expiração entre 300 (mínimo) e 900 (máximo) segundos;
13. deve sempre incluir a claim acr no id_token;
14. deve suportar os valores code e id_token para o atributo response_type;
15. não deve permitir o recurso de rotação de refresh tokens;
16. deve garantir que em caso de compartilhamento do Servidor de Autorização para outros serviços, além do Open Finance, não divulgue e/ou possibilite o uso de métodos não certificados no ambiente do Open Finance;
17. deve garantir que as configurações divulgadas aos demais participantes através do OpenID Discovery (indicado pelo arquivo de Well-Known cadastrado no Diretório) sejam restritos aos modos de operação aos quais a instituição se certificou;
    a. deve manter em suas configurações os métodos para os quais ainda haja clientes ativos;
    b. deve atualizar os cadastros que utilizem métodos não certificados, através de tratamento bilateral entre as instituições envolvidas;
18. deve recusar requisições, para o ambiente do Open Finance, que estejam fora dos modos de operação ao qual a instituição certificou seu Servidor de Autorização;
19. o tempo mínimo de expiração do request_uri deve ser de 60 segundos;
20. deve recusar requisições que não apresentem o cabeçalho x-fapi-interaction-id em endpoints de recursos protegidos;
21. deve exigir a utilização de Proof Key for Code Exchange (PKCE);
22. deve exigir a utilização do subject_type “public”;
23. deve exigir a utilização do response_mode “fragment”;
24. *Deve emitir refresh_tokens (JWT ou opaco) sem prazo de validade nos cenários onde o mesmo é necessário.*

5.2.2.1. Token de ID como assinatura separada

O Servidor de Autorização deve suportar as disposições especificadas na cláusula 5.2.2.1 de Financial-grade API Security Profile 1.0 - Parte 2: Advanced

1. Deve obrigatoriamente criptografar o id_token nos momentos de call-back e chamada do endpoint de token;
2. Para a criptografia do id_token deve ser utilizada chave disponível no JWKS informado no parâmetro jwks_uri, com o atributo “use”:”enc”, durante o registro do cliente, indicada através do cabeçalho kid do documento JWT;
3. O uso de outros cabeçalhos para indicação da chave utilizada, como x5u, x5c, jku ou jkw é vetado conforme definido na cláusula 2 OIDC.

5.2.2.2. Clarificações sobre a "sub" Claim

Este perfil usa a definição oficial encontrada em: Analise requisitos de criptografia ID_TOKEN. Isso significa que o sub é um identificador nunca transferido ou alterado para o usuário final dentro da emissora (detentora/transmissora).

5.2.2.3. Solicitando o "urn:brasil:openbanking:loa2" ou "urn:brasil:openbanking:loa3" Solicitação de contexto de autenticação

Esse perfil define "urn:brasil:openbanking:loa2" e "urn:brasil:openbanking:loa3" como novas classes de "Authentication Context Request" (ACR)

- LoA2: mecanismo de autenticação com a adoção de um único fator
- LoA3: mecanismo de autenticação com múltiplos fatores de autenticação

A seguinte orientação deve ser observada para o mecanismo de autenticação das APIs do Open Finance Brasil:

- De acordo com o Art. 17 da Resolução Conjunta nº 01, as instituições devem adotar procedimentos e controles para autenticação de cliente compatíveis com os aplicáveis ao acesso a seus canais de atendimento eletrônicos.
- Em observância à regulação em vigor, sugere-se que:
  - Para a autenticação do usuário em autorizações de acessos às APIs de compartilhamento de dados (Fase 2), os Authorization Servers deveriam adotar, no mínimo, método compatível com LoA2; e
  - Para a autenticação do usuário em autorizações de acessos às API's das fases subsequentes, os Authorization Servers deveriam adotar método de autenticação compatível com LoA3 ou superior.

Em todos os casos, a adoção de mecanismo de autenticação mais rigoroso (LoA3 ou superior) fica a critério da instituição transmissora ou detentora de conta, de acordo com sua avaliação de riscos e de forma compatível com os mecanismos habitualmente utilizados.

Esclarecimentos adicionais sobre fatores de autenticação

São fatores de autenticação:

- Aquilo que você conhece, como uma senha ou frase secreta
- Aquilo que você tem, como um token, smartcard ou dispositivo
- Aquilo que "você é", ou seja, autenticação condicionada a apresentação de uma característica física exclusivamente sua, como a validação por biometria

Para realizar autenticação por múltiplos fatores (MFA) é necessário que o usuário apresente, ao menos, dois diferentes fatores dos listados acima. Um mesmo fator usado mais de uma vez - por exemplo, a apresentação de suas senhas que ele conhece - não pode ser aceito como MFA.

5.2.2.4 Escopos obrigatórios no endpoint de descoberta (well-known)

O servidor de autorização, contidos em organizações que possuem a role DADOS, deve obrigatoriamente declarar os escopos abaixo em seu endpoint de descoberta (well-known), independentemente se a instituição forneça ou não os produtos referentes aos escopos abaixo listados:

- invoice-financings
- financings
- loans
- unarranged-accounts-overdraft
- bank-fixed-incomes
- credit-fixed-incomes
- variable-incomes
- treasure-titles
- funds
- exchanges

Os escopos não listados acima devem ser declarados caso a instituição forneça produtos referentes aos mesmos (ex: accounts, payments).

5.2.3. Cliente confidencial

Um cliente confidencial deve apoiar as disposições especificadas na cláusula 5.2.3 de Financial-grade API Security Profile 1.0 - Part 2: Advanced,

Além disso, o cliente confidencial

1. deve suportar objetos de solicitação encrypted;
2. deve suportar solicitações de autorização push (pushed authorization requests) PAR;

3. deve suportar o escopo de recurso OAuth 2.0 parametrizado consent conforme definido na cláusula 6.3.1 OIDF FAPI WG Lodging Intent Pattern;
4. deve suportar refresh tokens;
5. não deve incluir um valor específico na claim acr;
6. deve definir a claim acrcomo essential;
7. deve suportar todos os métodos de autenticação especificados no item 14 da seção 5.2.2 da Financial-grade API Security Profile 1.0 - Part 2: Advanced incluindo as diferentes combinações de métodos de encaminhamento dos Requests Objects (usando ou não PAR - item 11);
8. não deve permitir o recurso de rotação de refresh tokens;
9. deve enviar o cabeçalho x-fapi-interaction-id em endpoints FAP

6. Considerações de segurança

Os participantes devem apoiar todas as considerações de segurança especificadas na cláusula 8 Financial-grade API Security Profile 1.0 - Parte 2: Advanced e o Manual de Segurança de Banco Central do Brasil. O ICP Brasil emite certificados RSA x509 somente, portanto, para simplificar, a seção remove o suporte para algoritmos EC e exige que apenas algoritmos de criptografia recomendados pela IANA sejam usados.

6.1. Considerações sobre assinatura do conteúdo de mensagens (JWS)

Para garantir a integridade e o não-repúdio das informações tramitadas em API's sensíveis e que indicam essa necessidade na sua documentação, deve ser adotado a estrutura no padrão JWS definida na RFC7515 e que inclui:

- Cabeçalho (JSON Object Signing and Encryption - JOSE Header), onde se define o algoritmo utilizado e inclui informações sobre a chave pública ou certificado que podem ser utilizadas para validar a assinatura;
- Payload (JWS Payload): conteúdo propriamente dito e detalhado na especificação da API;
- Assinatura digital (JWS Signature): assinatura digital, realizada conforme parâmetros do cabeçalho.

1. Cada elemento acima deve ser codificado utilizando o padrão Base64url RFC4648 e, feito isso, os elementos devem ser concatenados com "." (método JWS Compact Serialization, conforme definido na RFC7515).
2. O payload das mensagens (requisição JWT e resposta JWT) assinadas devem incluir as seguintes claims presentes na RFC7519:

- aud (na requisição JWT): o Provedor do Recurso (p. ex. a instituição detentora da conta) deverá validar se o valor do campo aud coincide com o endpoint sendo acionado;
- aud (na resposta JWT): o cliente da API (p. e. instituição iniciadora) deverá validar se o valor do campo aud coincide com o seu próprio organisationId listado no diretório;
- iss (na requisição JWT e na resposta JWT): o receptor da mensagem deverá validar se o valor do campo iss coincide com o organisationId do emissor;
- jti (na requisição JWT e na resposta JWT): o valor do campo jti deverá ser preenchido com o UUID definido pela instituição de acordo com a [RFC4122] usando o versão 4;
- iat (na requisição JWT e na resposta JWT): o valor do campo iat deverá ser preenchido com horário da geração da mensagem e de acordo com o padrão estabelecido na RFC7519 para o formato NumericDate.
- cty (na requisição JWT e na resposta JWT): o valor do campo cty deverá ser preenchido caso as operações de assinatura ou criptografia aninhadas não sejam empregadas, o uso desse parâmetro de cabeçalho não é recomendado. No caso de assinatura aninhada ou criptografia ser empregada, este parâmetro de cabeçalho deve estar presente; neste caso, o valor deve ser "JWT", para indicar que um JWT aninhado é transportado neste JWT. Embora os nomes dos tipos de mídia não diferenciem maiúsculas de minúsculas, é recomendável que "JWT" sempre seja escrito com caracteres maiúsculos para compatibilidade com implementações herdadas.

1. O content-type HTTP das requisições e respostas com mensagens JWS deve ser definido como: "application/jwt".
2. No cabeçalho JOSE deve constar os seguintes atributos:

- alg - deve ser preenchido com o valor PS256";
- kid - deve ser obrigatoriamente preenchido com o valor do identificador da chave utilizado para a assinatura;
- Em caso de erro na validação da assinatura pelo Provedor do Recurso a API deve retornar mensagem de erro HTTP com status code 400 e a resposta deve incluir na propriedade code do objeto de resposta de erro especificado na API (ResponseError) a indicação da falha com o conteúdo BAD_SIGNATURE.
- Erros na validação da mensagem recebida pela aplicação cliente (p. ex. iniciador de pagamento) devem ser registrados e o Provedor do Recurso (p. ex. instituição detentora de conta) deve ser notificado.

1. O receptor deve validar a consistência da assinatura digital da mensagem JWS exclusivamente com base nas informações obtidas do diretório, ou seja, com base nas chaves publicadas no JWKS da instituição no diretório.

2. As assinaturas devem ser realizadas com uso do certificado digital de assinatura especificado no Padrão de Certificados Open Finance Brasil;
3. A claim iat deve ser numérica no formato Unix Time GMT+0 com tolerância de +/- 60 segundos;
4. A claim do jti deve ser única para um clientId dentro de um intervalo de tempo de 86.400 segundos (24h), não podendo ser reutilizada neste período. Em caso de reutilização, deverá ser retornado o código de erro HTTP 403. Demais casos seguir instrução da RFC 6749, item 5.2;

6.1.1. Considerações sobre algoritmos de assinatura

Para JWS, clientes e servidores de autorização

devem usar o algoritmo PS256;

6.1.2. Considerações de algoritmo de criptografia

Para JWE, clientes e servidores de autorização

devem usar RSA-OAEP com A256GCM

6.1.3. Considerações sobre o uso seguro do Transport Layer Security

Para TLS, endpoints do Servidor de Autenticação e endpoints do Servidor de Recursos usados diretamente pelo cliente:

1. devem suportar TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
2. devem suportar TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
3. As funcionalidades "TLS Session Resumption" e "TLS Renegotiation" devem ser desabilitadas

7. Considerações sobre compartilhamento de dados

7.1. Mecanismo de Autorização

7.1.1. Introdução

Os mecanismos existentes para gerenciar adequadamente o acesso aos recursos definidos em RFC6749 são insuficientes para atender aos requisitos de um ecossistema de compartilhamento de dados moderno. Aproveitar strings de escopo estático não fornece aos consumidores controle de granularidade suficiente para compartilhar com terceiros. O Open Finance Brasil optou por implementar uma API de consentimento como um recurso protegido OAuth 2.0 que pode ser usado para gerenciar o acesso granular aos recursos. A referência ao recurso de consentimento será transmitida como parte de um escopo de recurso dinâmico OAuth 2.0.

7.1.2. Definição de Escopo de Consentimento Dinâmico

Este perfil define o escopo dinâmico do OAuth 2.0 "consentimento" da seguinte maneira:

- string 'consent' ou 'recurring-consent'; e
- delimitador de dois pontos ":"; e
- Consent API REST Resource Id retornado por uma criação bem-sucedida de Open Finance Consent Resource;

Adicionalmente:

- Consent Resource Id deve incluir caracteres seguros para url;
- Consent Resource Id deve ser "namespaced";
- Consent Resource Id deve ter propriedades de um nonce Nonce;

7.1.3. Exemplo de escopo de consentimento dinâmico

consent:urn:bancoex:C1DD33123

7.2. Ciclo de vida da autorização

7.2.1. Introdução

O recurso de consentimento tem um ciclo de vida gerenciado separada e distintamente da estrutura de autorização OAuth 2.0. As transições de estado e comportamentos esperados e condições de erro esperados dos Recursos REST protegidos com este perfil são definidos nas especificações funcionais da API publicadas pelo Open Finance Brasil.

7.2.2. Servidor de autorização

Além dos requisitos descritos nas disposições de segurança do Open Finance Brasil, o Servidor de Autorização

1. deve apenas emitir refresh_tokens quando vinculados a um consentimento ativo e válido;
    a. Não deve emitir refresh_token quando o consentimento estiver no status "CONSUMED" (para fase 3);
    b. Deve emitir um access_token por meio de um grant_type do tipo client credentials no status "CONSUMED" (para fase 3).
2. só deve compartilhar o acesso aos recursos quando apresentado access_token vinculado a um consentimento ativo e com o status "AUTHORISED". Para tokens gerados com o scope: payments ou recurring-payments, *o* status do consentimento não será validado;
    a. No cenário de recebimento de token inválido, deve ser retornado status code 401.
3. deve revogar os refresh tokens e, quando aplicável, os access tokens quando o Consentimento (Consent Resource) relacionado for apagado;

4. deve garantir que os access tokens são emitidos com os scopes necessários para permitir acesso aos dados especificados em elemento Permission do Consentimento (Consent Resource Object) relacionado;
5. não deve rejeitar pedido de autorização com scopes além do necessário para permitir acesso a dados definidos em elemento Permission do Consentimento (Consent Resource Object) relacionado;
6. pode reduzir o escopo solicitado para um nível que seja suficiente para permitir o acesso aos dados definidos em elemento Permission do Consentimento (Consent Resource Object) relacionado;
7. deve manter registros sobre o histórico dos consentimento para permitir a adequada formação de trilhas de auditoria em conformidade com a regulação em vigor;
8. deve retornar falha na autenticação e o código de retorno _accessdenied no parâmetro erro (como especificado na seção 4.1.2.1 da RFC6749) caso o CPF do usuário autenticado não seja o mesmo indicado no elemento loggedUser do Consentimento (Consent Resource Object);
9. deve retornar falha na autenticação e o código de retorno _accessdenied no parâmetro erro (como especificado na seção 4.1.2.1 da RFC6749) caso o elemento businessEntity não tenha sido preenchido no Consentimento (Consent Resource Object) relacionado e o usuário tenha selecionado ou se autenticado por meio de credencial relacionada à conta do tipo Pessoa Jurídica (PJ);
10. deve condicionar a autenticação ou seleção de contas do tipo PJ à consistência entre o CNPJ relacionado à(s) conta(s) e o valor presente no elemento businessEntity do Consentimento (Consent Resource Object). Em caso de divergência deve retornar falha na autenticação e o código de retorno *access_denied* no parâmetro erro (como especificado na seção 4.1.2.1 da RFC6749);
11. deve emitir *refresh_token* com validade não inferior à validade do consentimento ao qual está relacionado, respeitado os demais critérios acima.

7.2.3. Cliente confidencial

Além dos requisitos descritos nas disposições de segurança do Open Finance Brasil, o Cliente Confidencial

1. deve, sempre que possível, revogar e cessar o uso de refresh e de access tokens vinculados a um consentimento (Consent Resource Object) que foi excluído;
2. deve excluir consentimentos (Consent Resource Objects) que estão expirados;

8. Reconhecimentos

Agradecemos a todos que estabeleceram as bases para o compartilhamento seguro e seguro de dados por meio da formação do Grupo de Trabalho FAPI da OpenID Foundation, o GT de Segurança do Open Finance Brasil e aos pioneiros que ficarão em seus ombros.

As seguintes pessoas contribuíram para este documento:

- Alexandre Siqueira (Mercado Pago)
- Joseph Heenan (Authlete)
- Marcos Rodrigues (Itaú)
- Mário Ginglass (BNDES)
- Nic Marcondes (Quanto)
- Ralph Bragg (Raidiam)

Appendix A. Avisos

ofereçam uma promessa de patente de não fazer valer certas reivindicações de patentes contra outros contribuidores e implementadores. A Estrutura Inicial do Open Finance Brasil convida qualquer parte interessada a trazer à sua atenção quaisquer direitos autorais, patentes, pedidos de patentes ou outros direitos de propriedade que possam abranger a tecnologia que possa ser necessária para praticar esta especificação.

Author's Address

OFBIS GT Security

Open Finance Brasil Initial Structure

Email: gt-seguranca@openfinancebrasil.org.br

URI: https://openfinancebrasil.org.br