Capítulo 1

Autenticação e Autorização: antigas demandas, novos desafios e tecnologias emergentes

Emerson Ribeiro de Mello*, Shirlei Aparecida de Chaves*, Carlos Eduardo da Silva†, Michelle Silva Wangham‡§, Andrey Brito¶ e Marco Aurélio Amaral Henriques‖

Abstract

Identity and access management integrates policies, business processes, and technologies to enable authentication and authorization of subjects before and during an online transaction. Technological developments, and social and regulatory demands, such as personal data protection regulations, constantly pose challenges for identity management. This chapter begins with a characterization of identity and access management models, which includes the decentralized identity model. It then presents some technologies and standards to meet new demands and challenges regarding security, privacy, usability, and user empowerment. It also characterizes software identity, the use of authorization and access control in web applications, ending with an overview of the topics covered.

Resumo

A gestão de identidade e de acesso integra políticas, processos de negócios e tecnologias para permitir a autenticação e autorização de sujeitos antes e durante uma transação online. Evoluções tecnológicas, demandas sociais e regulatórias, como as leis de proteção de dados pessoais, impõem constantemente desafios para a gestão de identidade. Este capítulo inicia-se com uma caracterização dos modelos de gestão de identidade e de acesso, o que inclui o modelo de identidade descentralizada, para depois apresentar

*Instituto Federal de Santa Catarina. Email: mello@ifsc.edu.br, shirlei.chaves@ifsc.edu.br
†Sheffield Hallam University, UK. Email: C.DaSilva@shu.ac.uk
‡Universidade do Vale do Itajaí. Email: michelle.wangham@rnp.br
§Rede Nacional de Ensino e Pesquisa
¶Universidade Federal de Campina Grande. Email: andrey@computacao.ufcg.edu.br
‖Universidade Estadual de Campinas. Email: maah@unicamp.br

algumas tecnologias e padrões para atender novas demandas e desafios relacionados à segurança, privacidade, usabilidade e empoderamento dos usuários. Também se caracteriza a identidade de software, utilização de autorização e controle de acesso em aplicações web, finalizando com um apanhado geral sobre os temas tratados.

1.1. Introdução

De acordo com a meta 16.9 dos Objetivos de Desenvolvimento Sustentável das Nações Unidas, os países signatários da Agenda 2030 devem fornecer identidade legal para todos, incluindo identidade digital. Cada pessoa tem o direito de participar plenamente na sua sociedade e de ser reconhecida como uma pessoa perante a lei. No entanto, cerca de um bilhão de pessoas em todo o mundo não têm meios de provar sua identidade, o que é essencial para protegerem os seus direitos e permitir acesso a serviços (GROUP, 2018).

Uma identidade digital é uma representação única de uma entidade que seja suficiente para identificar esta entidade em uma transação *online* (GRASSI; GARCIA; FENTON, 2020). Uma entidade é qualquer coisa existente no mundo real (uma pessoa, máquina, aplicação, objeto físico, empresa), sendo que esta pode possuir múltiplas identidades. A prova de identidade estabelece que uma entidade é quem ela afirma ser em um processo de autenticação digital (GRASSI; GARCIA; FENTON, 2020).

Conforme apresentado pelo relatório do Fórum Econômico Mundial intitulado *A Blueprint for Digital Identity* (FORUM, 2016), uma das lacunas do cenário de identidade digital é confundir autenticação com identidade. Soluções de autenticação utilizam processos de coleta de atributos e identificação do usuário (*onboarding*) preexistentes, baseados em modelos de identidade digital.

Segundo Allan (2020), a gestão de identidade e de acesso (*Identity And Access Management* – IAM) consiste em um conjunto de processos e tecnologias que visa permitir o relacionamento e confiança entre pessoas, serviços ou coisas (como no contexto de *Internet of Things* – IoT). A IAM visa garantir a identidade de uma entidade (usuário, dispositivo, software), garantir a qualidade das informações de uma identidade (identificadores, credenciais e atributos) e prover procedimentos de autenticação, autorização e auditoria (ITU, 2009).

Diante da transformação digital acelerada, decorrente da pandemia de Covid-19, da redefinição dos perímetros de segurança da informação nas instituições, das necessidades de usuários e de empresas por segurança, proteção de dados pessoais e usabilidade, a área de IAM se mostra relevante e desperta interesse da academia, do governo e das empresas. Embora os processos para implementar a IAM muitas vezes sejam complexos, estes são extremamente necessários, em especial, para lidar com ataques cada vez mais sofisticados e para implementação do modelo de confiança zero, uma estratégia de segurança que vem sendo muito recomendada (ROSE et al., 2020). Este modelo orienta a "nunca confiar e sempre verificar", ou seja, desconfiar "por padrão" e confiar "por exceção".

Segundo o relatório anual sobre violação de dados da IBM (2022), 83% das organizações sofreram mais de uma violação, sendo que o custo médio de uma violação de dados foi de 4,35 milhões dólares em 2022. O uso de credenciais roubadas ou comprometidas continua sendo o principal vetor de ataque em 19% das violações, tendo um custo médio

de 4,50 milhões de dólares. Essas violações de credenciais tiveram o ciclo de vida mais longo — 243 dias para identificar a violação e outros 84 dias para conter a violação. De acordo com relatório, organizações que fizeram uso de soluções de IAM, bem como a adoção de autenticação multifator, conseguiram reduzir os custos com a violação de dados em 224 mil dólares, 187 mil dólares, respectivamente.

De acordo com a pesquisa "Pesquisa Global de Identidade e Fraude 2021[1]"da Serasa (2021), 55% dos consumidores disseram que a segurança é o aspecto mais importante de sua experiência online e 33% disseram estar preocupados com roubo de identidade. A pesquisa constatou um aumento do conforto e preferência que os consumidores têm por métodos de segurança físicos e baseados em comportamento - ou invisíveis. Os consumidores, com base em sua segurança percebida, classificaram os seguintes métodos como os três mais seguros para autenticação:

- 74% dos consumidores disseram biometria física, que inclui principalmente reconhecimento facial e impressões digitais em dispositivos móveis;
- 72% dos consumidores disseram senhas de uso único (*One-Time Password* – OTP) enviados para dispositivos móveis;
- 66% dos consumidores disseram análise comportamental, que aproveita os comportamentos observados passivamente em navegadores e dispositivos móveis, sem fricção.

Os modelos de gestão de identidade sofreram evoluções constantes para adequarem-se aos novos serviços, modelos de negócio e tecnologias bem como às novas restrições impostas por meio de leis, como o Regulamento Geral de Proteção de Dados da União Europeia (*General Data Protection Regulation* – GDPR) (UNION, 2016) e a Lei Geral de Proteção de Dados Pessoais (LGPD) (BRASIL, 2018) no Brasil. Segundo Allen (2016), tal evolução pode ser dividida em quatro estágios: identidade centralizada, identidade federada, identidade centrada no usuário e identidade digital descentralizada.

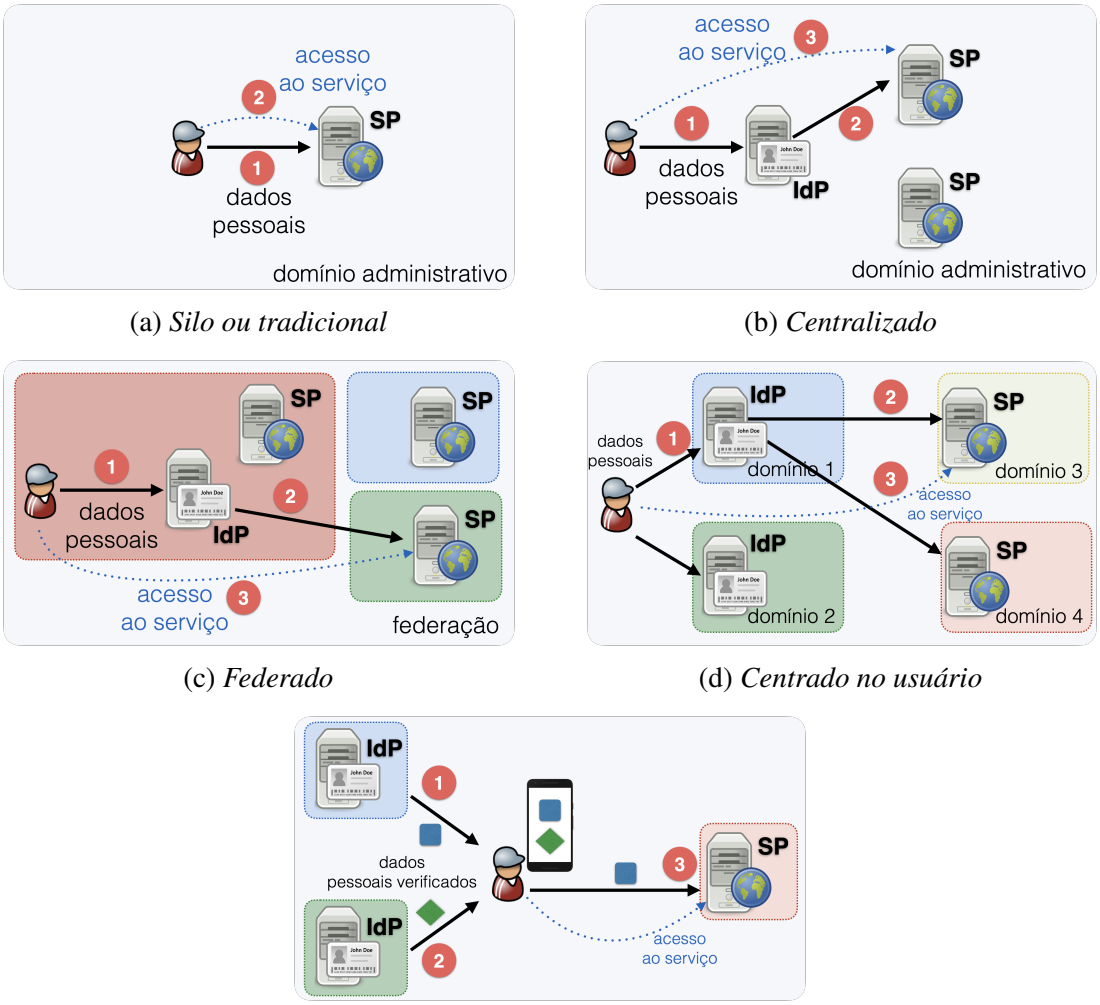
No minicurso apresentado no SBSeg de 2010 (WANGHAM; MELLO et al., 2010), discorreu-se sobre o gerenciamento de identidades federadas, apresentando os principais conceitos e tecnologias da época para implementação do modelo. Em 2013, o minicurso (WANGHAM; DOMENECH; MELLO, 2013) analisou infraestruturas de autenticação e autorização na IoT. No SBSeg 2019, o minicurso (NAKAMURA et al., 2019) foi específico para o modelo de identidade digital descentralizado, apresentando conceitos e implementação de alguns casos de uso. Por fim, em 2021, o minicurso de Falcão et al. (2021) abordou o tema de identidade de *software*.

O objetivo geral deste capítulo é apresentar as demandas históricas, novos desafios e tecnologias empregadas para autenticação, autorização e controle de acesso em sistemas distribuídos. Os principais problemas que permeiam a gestão de identidade e de acesso serão analisados, tais como: modelos de gestão de identidade, robustez do processo de autenticação, autenticação contínua e dinâmica, usabilidade e distribuição de responsabilidade, autenticação e autorização no cenário da Internet das coisas.

[1]A pesquisa teve como base 3 estudos produzidos entre junho de 2020 e janeiro de 2021. Foram entrevistados 9 mil consumidores e 2700 executivos de empresas de 10 países, incluindo o Brasil.

1.1.1. Modelos de gestão de identidade

Na literatura (WANGHAM; MELLO et al., 2010; ALLEN, 2016), os modelos são classificados como uma progressão de estágios. Apesar da literatura apresentar uma pequena divergência na nomenclatura para alguns destes modelos, Preukschat e Reed (2021) destacam que uma das principais diferenças entre os modelos é referente à forma como o usuário se relaciona com a organização ou serviço no qual ele estabelece a sua identidade digital. Esse relacionamento se refere ao quanto os dados da identidade do usuário estão efetivamente sob o seu controle e também ao quanto desses dados ele precisa compartilhar, direta ou indiretamente, para a utilização de um serviço. Os modelos de GId, normalmente, envolvem três atores, a saber: usuário que deseja acessar um recurso ou serviço; provedor de identidade (*Identity Provider* – IdP); e provedor de serviço (*Service Provider* – SP). O IdP é responsável pela autenticação e gerenciamento de informações do usuário. Os SPs, também conhecidos como terceiras partes confiáveis (*Relying Party* – RP), são entidades que fornecem serviços aos usuários e que delegam a autenticação destes aos IdPs.



(a) *Silo ou tradicional*

(b) *Centralizado*

(c) *Federado*

(d) *Centrado no usuário*

(e) *Descentralizado*

Figura 1.1: *Classificação dos modelos de gestão de identidade. Adaptado de (WANGHAM; MELLO et al., 2010)*

Na Figura 1.1 são representados os fluxos com dados pessoais do usuário (seta de linha contínua) e de acesso ao serviços (seta de linha pontilhada) nos diferentes modelos de gestão de identidade. O detalhamento sobre cada um destes modelos é apresentado logo a seguir.

O primeiro estágio consiste no modelo tradicional ou modelo baseado em silo (veja Figura 1.1a). Neste modelo, o provedor de serviço (*Service Provider* – SP) é o responsável por gerenciar seus usuários. Não existe o compartilhamento de identidades de usuários por diferentes serviços. Assim, o usuário precisará criar uma identidade digital específica para cada provedor de serviço com quem for interagir. Alguns autores também classificam este modelo como sendo centralizado.

O segundo estágio consiste no modelo centralizado (veja Figura 1.1b). O provedor de identidade (*Identity Provider* – IdP) é o único que possui controle administrativo sobre as identidades digitais dos usuários e compartilha os dados destes usuários com os demais provedores de serviços que obrigatoriamente estão dentro do mesmo domínio administrativo onde se encontra o IdP. Para o usuário tem-se a facilidade em não precisar agora criar uma identidade digital para cada SP com quem for interagir, contudo o IdP é quem de fato está controlando seus dados pessoais. O *Central Authentication Service* (CAS) *Protocol* (BRAMHALL et al., 2017) pode ser usado para implementar este modelo.

O modelo federado é considerado como o terceiro estágio (veja Figura 1.1c) e assemelha-se muito com o segundo estágio, contudo, o usuário poderá acessar também SPs que não fazem parte de sua instituição de origem, mas que fazem parte da federação na qual seu provedor de identidade também participa. O IdP continua controlando os dados do usuário e sempre participa da interação entre o usuário e o SP. O SAML (OPEN, 2022) pode ser usado para implementar este modelo, sendo este usado pela Comunidade Acadêmica Federada (CAFe) da RNP e por diversas outras federações acadêmicas no mundo.

Como quarto estágio tem-se o modelo de identidade centrada no usuário (veja Figura 1.1d), o qual apresenta esforços para que a experiência do usuário seja melhor e para que haja uma maior descentralização das informações e da confiança. Com este modelo começaram a surgir as ideias de que uma identidade digital deveria ficar totalmente sob o controle de seu dono. Entretanto, o foco maior está em duas frentes: consentimento do usuário, dando a este a visibilidade dos atributos que são compartilhados pelo IdP ao SP, e interoperabilidade, para facilitar a autenticação entre múltiplos provedores de serviços.

Neste estágio é possível dar ao usuário o controle total de sua identidade, mas ao custo de ele ter que criar seu próprio serviço de autenticação centrada no usuário, como a instalação de um serviço OpenID (OPENID, 2022), por exemplo. Como a maior parte dos usuários não têm condições de criar tal serviço, o caminho natural demonstrado na prática, é o uso de serviços deste tipo disponibilizados por grandes provedores já consolidados como Facebook, Google, Apple e a conta gov.br[2], por exemplo. O conceito *Bring you own identity* (BYOI), em tradução livre, "traga a sua própria identidade", foi cunhado como uma forma de uso propiciada por este modelo de gestão de identidade. Para implementar este modelo é comum o uso do OpenID Connect (OPENID, 2022), que apresenta uma

[2]https://www.gov.br/governodigital/pt-br/conta-gov-br/conta-gov-br/

camada de identidade sobre o protocolo OAuth2 (HARDT, 2012).

Dessa forma, os dados do usuário ainda continuam nas mãos dos provedores, que detêm o controle sobre os mesmos, e estes podem desabilitar um usuário a qualquer momento, mesmo sem apresentar justificativas para tal. Além disso, observa-se uma centralização ainda maior do processo de autenticação em alguns poucos e grandes provedores, o que nos remete a um dos problemas básicos dos primeiros estágios, que é a centralização dos dados no provedor de identidade, sem o controle do usuário.

O conceito de Identidade Digital Descentralizada (IDD), também chamada de Identidade Autossoberana, do inglês *Self-Sovereign Identity* (SSI), apresenta-se como o último estágio dos modelos de gestão de identidade (veja Figura 1.1e). Cabe destacar que o termo identidade autossoberana pode ser mal interpretado de forma a dar a entender que o indivíduo poderia emitir sua própria identidade. A sociedade está organizada em sistemas políticos que possuem estruturas governamentais com papéis bem definidos e cabem somente a estas a soberania para identificação de seus cidadãos. Desta forma, a identidade autossoberana propõe-se em dar ao usuário a soberania para administrar suas identidades digitais e não em criá-las (LÓPEZ, 2020).

Segundo (ALLEN, 2016), apesar do modelo centrado no usuário ter permitido que identidades centralizadas pudessem ser usadas como identidades federadas interoperáveis, e respeitando o consentimento do usuário, ainda era necessário um modelo no qual o usuário estivesse no centro do processo autenticação, cabendo somente a ele ditar as regras de uso sobre seus dados pessoais e que não houvesse qualquer intermediação do IdP no acesso ao SP. No modelo descentralizado, o próprio usuário é o responsável por manter suas identidades digitais, por exemplo em um aplicativo de carteira digital em seu telefone inteligente, cujos atributos são atestados criptograficamente por seus emissores (e.g. Secretaria de Segurança Pública) e poderão ter sua integridade e autenticidade verificada pelos provedores de serviço.

Os identificadores descentralizados (*Decentralized Identifiers* – DIDs) (W3C, 2022a) e as credenciais verificáveis (*Verifiable Credentials* – VC) (W3C, 2022b) são dois padrões da W3C que estão sendo considerados como pilares para soluções de gestão de identidade descentralizada. Ainda existem diversas tecnologias e *frameworks*, abertos e proprietários, sendo que muitos fazem uso de livro razão distribuído (e.g. *blockchain*) e o Hyperledger Indy[3] é um que tem despertado bastante interesse da comunidade.

Identificadores descentralizados

Segundo (W3C, 2022a), pessoas e organizações usam identificadores únicos globais, nos mais variados contextos. Por exemplo, pessoas usam endereço de email, nomes de usuários em redes sociais, número de telefone, CPF etc. Como identificadores para produtos ou serviços tem-se o número serial, URI (NOTTINGHAM, 2020), UUID (LEACH; MEALLING; SALZ, 2005) etc. Porém, os identificadores amplamente usados por pessoas, nas interações com serviços na Internet, não estão de fato sob seu controle, uma vez que são emitidos e controlados por autoridades externas. Por exemplo, o endereço de email de

[3]https://www.hyperledger.org/use/hyperledger-indy

uma pessoa pode ser excluído de uma organização assim que esta deixar de fazer parte do quadro de funcionários. Tais identificadores também podem relevar informações pessoais mais do que é necessário na interação com um serviço, ou ainda, podem ser replicados de forma fraudulenta, podendo assim resultar no roubo da identidade de uma pessoa em um serviço.

Os identificadores descentralizados (DIDs) (W3C, 2022a) consistem em identificadores únicos globais, que podem referenciar qualquer tipo de entidade (e.g. pessoa, dispositivo, organização, software etc.), e que possibilitam aos detentores (e.g. pessoas, organizações etc.) serem seus controladores. Desta forma, não existe aqui a dependência de uma autoridade externa ou mesmo centralizada para emissão, gestão ou manutenção de DID. O DID, portanto, consiste em um identificador que apresenta *simultaneamente* as quatro propriedades a seguir:

- **Descentralizado** – não há necessidade de uma autoridade central para emiti-lo;
- **Persistente** – não há necessidade em ter uma organização mantenedora para que continue a existir;
- **Resolvível** – pode ser usado no processo de resolução para recuperar metadados associados a este (chamados de documentos DID);
- **Criptograficamente verificável** – é possível comprovar criptograficamente seu controle e posse.

Um DID consiste de uma URI – *Uniform Resource Identifier* – que é uma cadeia de caracteres segmentada em três partes (veja Figura 1.2): 1) identificador do esquema (*did:*); 2) identificador do método DID; e 3) identificador único determinado pelo método DID. O método DID é definido em uma especificação própria, e externa à especificação do próprio DID, na qual são detalhadas como DID e documentos DID são criados, atualizados, resolvidos e desativados.

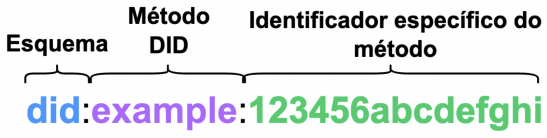


Figura 1.2: *Exemplo de um DID. Adaptado de W3C (2022a)*

O DID pode ser usado na resolução de metadados que estejam associados a este, sendo estes metadados chamados de documentos DID. Nestes documentos estão contidas informações de interesse do controlador, bem como métodos (tipicamente baseados em criptografia de chave pública) que permitem a verificação destas informações. A Figura 1.3 apresenta uma visão geral da arquitetura DID e o relacionamento entre seus componentes básicos. Um resumo de cada um dos componentes é apresentado a seguir, de acordo com W3C (2022a).

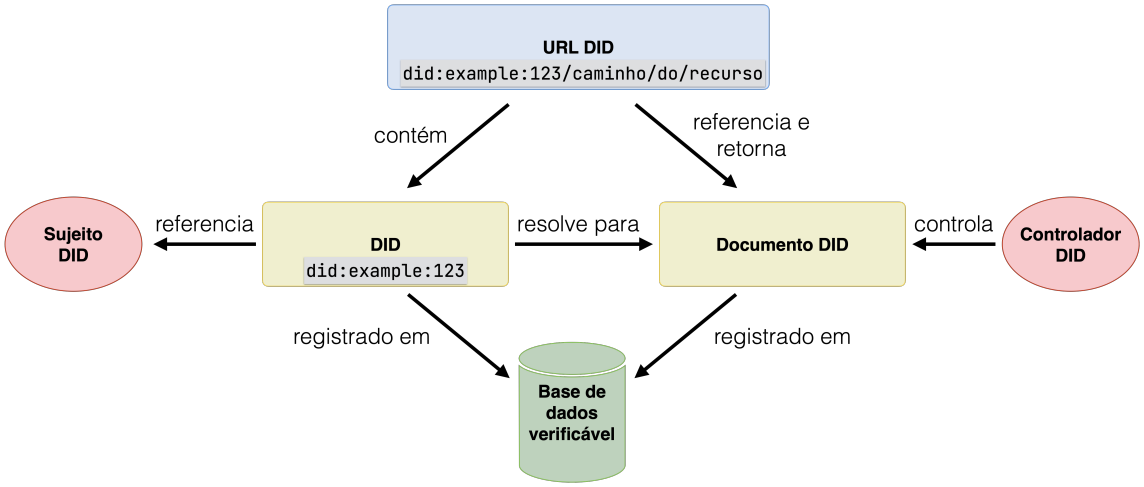


Figura 1.3: *Visão geral da arquitetura DID e relacionamentos entre seus componentes básicos. Adaptado de W3C (2022a)*

- **Sujeito DID** – é a entidade (pessoa, grupo, organização, coisa, conceito etc.) identificada pelo DID;
- **Documento DID** – contém informações associadas ao sujeito identificado pelo DID - como por exemplo, chaves públicas para verificações, serviços relevantes para interação com o sujeito DID etc.;
- **Controlador DID** – a entidade (pode ser o próprio sujeito, mas não necessariamente) com capacidade de fazer alterações no documento DID;
- **URL DID** – estende a sintaxe básica de um DID, incorporando outros componentes padrões de uma URI, como *path*, *query* e *fragment*;
- **Base de dados verificável (*Verifiable Data Registry*)** – solução subjacente de armazenamento que permite a criação, verificação, atualização e desativação de DID e documentos DID (e.g. livro razão distribuído, redes P2P, sistema de arquivos distribuídos etc.).

Credenciais verificáveis

As Credenciais Verificáveis (VCs) (W3C, 2022b) são o ícone mais visível da infraestrutura que se convencionou chamar de Identidade Autossoberana e os DIDs permitem criar identificadores que possuam os atributos desejados de um identificador global único. As VCs foram propostas para permitir expressar credenciais digitais de forma similar às credenciais físicas, como documentos de identidade, carteira de habilitação etc. (veja Figura 1.4), de modo que sejam criptograficamente seguras, respeitem a privacidade e que possam ser interpretadas por máquinas.

O modelo de dados das VCs prevê que estas podem ser representadas, bem como o material criptográfico associado, como documentos JSON-LD (W3C, 2020) ou na forma de *tokens* JWT (JONES; BRADLEY; SAKIMURA, 2020) (veja Listagem 1.1).

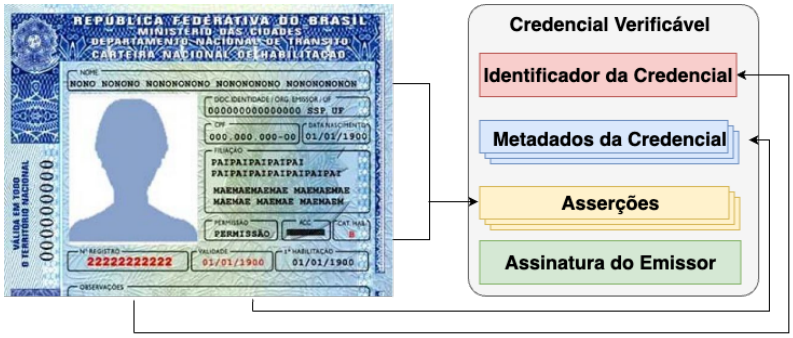


Figura 1.4: *Mapeamento de conceitos de Credenciais Verificáveis para o equivalente de credencial física. Adaptado de Preukschat e Reed (2021)*

```
{
    "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "id": "http://example.edu/credentials/1872",
    "type": [
        "VerifiableCredential",
        "AlumniCredential"
    ],
    "issuer": "https://example.edu/issuers/565049",
    "issuanceDate": "2010-01-01T19:23:24Z",
    "credentialSubject": {
        "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
        "alumniOf": {
            "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
            "name": [
                {
                    "value": "Example University",
                    "lang": "en"
                },
                {
                    "value": "Exemplo de Universidade",
                    "lang": "pt_BR"
                }
            ]
        }
    },
    "proof": {
        "type": "RsaSignature2018",
        "created": "2017-06-18T21:19:10Z",
        "proofPurpose": "assertionMethod",
        "verificationMethod": "https://example.edu/issuers/565049#key-1",
        "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
    sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
    X16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
    PAYuNzVBAh4vGHSrQyHUdBBPM"
    }
}
```

Listagem 1.1: Exemplo de uma VC representada com JSON-LD. Fonte: (W3C, 2022b)

Quando comparado com o gerenciamento de identidades federadas, a arquitetura de VCs apresenta uma terminologia similar (veja Figura 1.5). Podemos relacionar no modelo de VCs o Verificador (*Verifier*) como sendo equivalente ao SP, o Emissor (*Issuer*) como sendo equivalente ao IdP e o usuário como sendo o Detentor (*holder*). Porém, as

semelhanças em termos de modelo de confiança e comunicação não podem ser analisadas como equivalentes, pois são fundamentalmente diferentes.

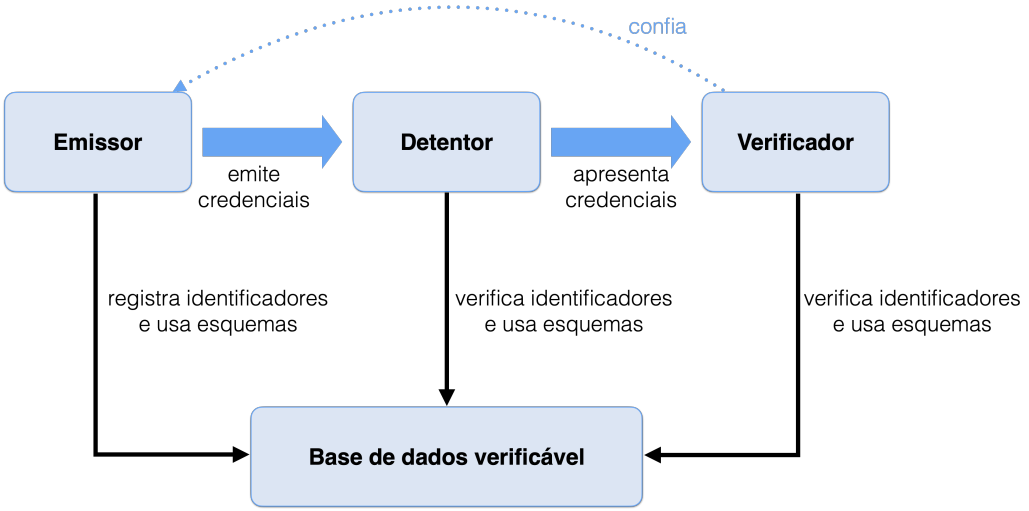


Figura 1.5: *Papéis e fluxos de comunicação com VCs. Adaptado de W3C (2022b)*

Na Figura 1.5 é possível notar que o usuário, detentor da credencial, está no centro da comunicação. Quando o Detentor apresenta sua credencial ao Verificador, a exemplo do que acontece no mundo físico quando se apresenta um documento a uma parte interessada, não há uma comunicação com o emissor. A recuperação da informação necessária para validar a credencial é feita diretamente pelo Verificador, acessando a base de dados verificável. Isto é, o Emissor não toma conhecimento para qual serviço ou em que situação o Detentor está apresentando sua credencial. Neste caso, tanto o Detentor, como o Emissor e o Verificador confiam na base de dados verificável.

O Verificador confia no Emissor, de acordo com suas próprias regras de confiança. Por exemplo, um conselho profissional pode confiar em uma universidade como autoridade para emissão de uma VC que ateste que uma pessoa obteve um título necessário para exercer determinada profissão. A universidade deve disponibilizar publicamente (por exemplo, divulgando seu DID) o material criptográfico necessário para que a VC emitida possa ser verificada. Cabe ao Verificador ir até à base de dados verificável para obter a informação necessária para verificar se a VC apresentada pelo Detentor está íntegra e é autêntica, ou seja, emitida por um Emissor no qual confia.

O Detentor é o responsável por manter suas VCs e pode, por exemplo, fazer uso de *softwares* como carteiras digitais instaladas em dispositivos que possui, como um telefone inteligente. Segundo Grüner et al. (2020), o modelo de gestão de identidade descentralizado possui desafios para que possam ser amplamente utilizados, sendo o armazenamento de credenciais pelo usuário um destes. No modelo descentralizado, os dados do usuário ficam armazenados sob sua responsabilidade, em sua carteira digital, ao contrário do modelo centralizado e federado, em que esses dados são armazenados pelo IdP em bases de dados próprias.

1.1.2. Autorização e controle de acesso

Quando se discute políticas e mecanismos para controle de acesso é comum utilizar uma abstração chamada de modelo de controle de acesso. Tal abstração permite descrever as características principais dos mecanismos de controle de acesso. Eles constituem um tipo de linguagem universal que permite que usuários, fornecedores e desenvolvedores interajam entre si, fornecendo um entendimento comum para o desenho e implementação de mecanismos de controle de acesso.

Existem diversos modelos de controle de acesso e, neste capítulo, iremos discutir os dois principais modelos que podem ser encontrados na maioria das ferramentas disponíveis. Antes de apresentar os modelos é necessário a definição de algumas terminologias que serão utilizadas.

- **Sujeito**: Também conhecido como principal, o sujeito é uma representação de uma entidade que deseja acessar um sistema. Normalmente é um usuário que deseja realizar alguma operação mas também pode ser um processo, outro sistema ou uma coisa;
- **Objeto**: Uma representação de um recurso que é acessado por um sujeito por meio de uma operação e protegido pelo mecanismo de controle de acesso;
- **Operação**: Representa uma ação realizada pelo sujeito no objeto;
- **Permissão** (privilégio): é a autorização para realizar uma ação em um sistema. É normalmente associada com o par operação-objeto.

A matriz de controle de acesso constitui a abstração mais básica quando se desenha e analisa políticas de controle de acesso. Ela pode ser interpretada como uma tabela onde cada linha representa um sujeito, e cada coluna representa um objeto no sistema protegido. Cada célula da tabela contém o conjunto de direitos de acesso que cada sujeito possui para o respectivo objeto.

Uma matriz de controle de acesso pode ser representada de várias maneiras, por exemplo, agrupando as permissões por linhas onde as mesmas são associadas a cada sujeito (também conhecido como lista de capacidades ou controle de acesso baseado em identidade), ou através do agrupamento por colunas, associadas aos objetos. Este último é também conhecido como listas de controle de acesso (*Access Control List - ACL*) e é a estratégia normalmente utilizada para o controle de permissões em sistemas de arquivos. Entretanto, a implementação de matrizes de controle de acesso através de listas de capacidades e ACLs são soluções não indicadas para sistemas *web* e distribuídos. Tais soluções não são adequadas para ambientes de larga escala, que possuem um grande número de usuários e objetos protegidos.

No modelo de controle de acesso baseado em papéis (*Role-Based Access Control - RBAC*) as decisões de acesso são baseadas na definição de papéis que os usuários assumem dentro de uma organização (FERRAIOLO; KUHN, 1996). Neste modelo existe uma ligação indireta entre o usuário e a permissão de acesso. Em uma política RBAC um conjunto de permissões é associado a um determinado papel. Por outro lado, usuários são

alocados a esses papéis, fazendo com que cada usuário tenha direito às permissões de seu papel. Deste modo, a administração das políticas de controle de acesso pode acontecer de maneira independente à administração dos usuários do sistema. Por exemplo, uma determinada permissão pode ser adicionada a um papel existente, autorizando seu uso por todos os usuários com esse papel, sem a necessidade de alterar os dados de todos os usuários afetados.

O modelo RBAC foi padronizado em 2004 pelo NIST (ANSI, 2004) e conta com um conjunto de extensões para prover suporte a diferentes casos de uso para políticas de controle de acesso. As extensões principais são o modelo hierárquico e as separações de responsabilidades estática e dinâmica.

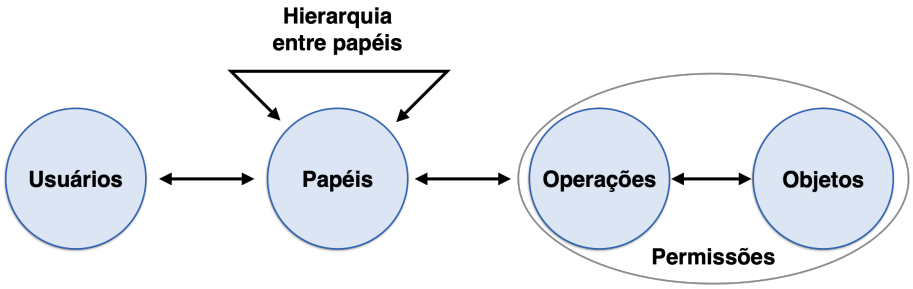


Figura 1.6: *Modelo RBAC hierárquico. Adaptado de (ANSI, 2004)*

O modelo RBAC hierárquico (apresentado na Figura 1.6) adiciona o conceito de hierarquias entre os papéis, de modo que um papel pode herdar as permissões associadas a outro papel enquanto adiciona outras permissões. Com isso, se consegue capturar mais facilmente a estrutura organizacional de uma instituição. Neste caso, existe responsabilidades e privilégios em comum entre dois ou mais papéis, mas um destes papéis possui outras permissões que não estão disponíveis para os outros. Por exemplo, um papel funcionário pode ter permissão para leitura de uma informação enquanto um papel diretor herda esta permissão e adiciona uma operação de escrita.

Outras extensões preveem a inclusão de restrições nas políticas de controle de acesso. Tais restrições são associadas ao conceito de separação de responsabilidades (*Separation of Duties - SOD*). No modelo RBAC um usuário pode estar associado a um ou mais papéis que são ativados durante uma sessão de uso do sistema (por exemplo, após a autenticação do usuário). Alguns sistemas permitem que usuário troquem seu papel ativo, enquanto que outras implementações fazem com que usuários tenham todos seus papéis ativos ao mesmo tempo. A restrição de SOD diz respeito a um conjunto de papéis que não podem ser ativados simultaneamente por um usuário.

Na SOD estática, a restrição acontece através de regras que são definidas junto à política de controle de acesso, por exemplo, quais papéis não podem ser alocados para um mesmo usuário simultaneamente, ou restrições baseadas no objeto sendo acessado. Por exemplo, o RBAC pode garantir que os usuários não possam ser membros da função de compra e da função de aprovação simultaneamente. A SOD estática garante que a mesma pessoa não possa comprar e aprovar a compra.

Na SOD dinâmica a restrição é normalmente associada aos papéis que podem estar

ativos ao mesmo tempo no sistema. Tal restrição pode ser associada a um usuário, que não pode ativar dois papéis diferentes em uma mesma sessão, ou à quantidade de usuários com um determinado papel que podem estar ativo simultaneamente. Neste caso, o sistema permite que a mesma pessoa esteja na função de compras e na função de aprovação, mas essas estariam proibidas de aprovar suas próprias compras. Elas só poderiam aprovar as compras de terceiros.

O modelo RBAC simplifica a definição de políticas de controle de acesso de acordo com as funções de um usuário em uma organização. Uma vez que uma política de controle de acesso foi definida, com seus respectivos papéis, regras, hierarquias e restrições, as tarefas relacionadas com a administração dos mecanismos de controle de acesso se resumem a adicionar ou remover usuários de determinados papéis.

Em 2012, o NIST publicou uma extensão ao modelo RBAC, denominado RBAC *Policy-Enhanced* (RPE) (ANSI, 2012), incorporando algumas funcionalidades que não constavam no modelo original mas que foram identificadas ao longo dos anos após sua publicação. A especificação RPE define e incorpora suporte a um conjunto de regras para os dois tipos de restrições: estática e dinâmica. Para restrições SOD estática, a especificação define regras que permitem restringir a alocação de diversos papéis a um mesmo usuário, diversas permissões a um mesmo papel, ou até mesmo quais usuários não podem ser alocados a determinados papéis. Para restrições SOD dinâmica, temos a adição de regras que consideram informações que são fornecidas por elementos externos ao do mecanismo de controle de acesso. Dentre as novas regras encontram suporte para restrições que influenciam quando um determinado papel pode ser ativado considerando, por exemplo, o horário do dia, a localização geográfica ou um determinado valor de um atributo.

Entretanto, o modelo RBAC apresenta algumas limitações, que acabaram motivando o desenvolvimento de outros modelos. Dentre essas limitações podemos mencionar a dificuldade associada com a tarefa conhecida como "engenharia de papéis", ou seja, a definição de todos os papéis e suas respectivas permissões em uma organização. Existe um conflito entre a facilidade de administração e a definição de políticas de segurança "mais fortes". Este último exige a definição de papéis e permissões mais granulares, o que resulta em uma maior quantidade de papéis que precisam ser alocados aos usuários, dificultando a administração das políticas de controle de acesso.

Uma evolução do modelo RBAC é o controle de acesso baseado em atributos (*Attribute-Based Access Control - ABAC*) (HU et al., 2014). Definido pelo NIST em 2014, no modelo ABAC as decisões de acesso são tomadas sobre um conjunto de regras baseados nos valores de atributos do sujeito que está requisitando acesso, do objeto sendo acessado, da operação sendo realizada e de condições ambientais tais como, hora do dia, dia da semana, localização geográfica ou qualquer outro atributo disponível no sistema. Neste caso, uma política de controle de acesso é composta por um conjunto de regras *booleanas* com os atributos e seus valores. O ABAC baseia-se nos modelos mencionados anteriormente, no sentido que cada um dos elementos considerados pelos modelos anteriores podem ser vistos como um atributo, mas os estendem ao permitir o uso de outros atributos. O modelo ABAC permite a implementação do modelo RBAC ao se utilizar somente o atributo papel nas regras de uma política de controle de acesso.

Não considerado estritamente como um modelo de controle de acesso, mas uma prática que vem ganhando adeptos, devido ao seu uso pelo Google, é o conceito de autorização de três fatores[4] (*Three-Factor Authorization - 3FA*) (ADKINS et al., 2020). A ideia geral é que algumas decisões de acesso exijam uma autorização explícita por um ser humano (que atua como autorizador) após sua avaliação pelos mecanismos de controle de acesso. Tal autorização deve ser realizada usando um sistema (ou dispositivo) diferente do que o utilizado para se realizar a operação. Por exemplo, uma prática adotada pela Google para algumas tarefas mais críticas é o uso de 3FA exigindo uma autorização explícita do usuário através de seu smartphone. A mesma estratégia é utilizada para usuários comuns se autenticando em sua conta Google, onde é enviada uma notificação, em um aplicativo no telefone inteligente do usuário o qual já esteja autenticado junto ao Google, questionando se o mesmo autoriza a realização da operação de *login* em um outro dispositivo.

Infraestrutura de autorização e controle de acesso

Com o crescente uso de regras de controle de acesso mais complexas, sua implementação em código torna-se mais arriscada. Isso tem contribuído para a adoção de infraestruturas de controle de acesso, nas quais as decisões de acesso são realizadas por um componente separado do código que implementa a lógica de negócio da aplicação (HU et al., 2014; ADKINS et al., 2020). Tais serviços podem ser encontrados, por exemplo, nas plataformas de nuvem dos grandes provedores e são conhecidos como *frameworks* de autorização. Por exemplo, as plataformas de nuvem da Google[5] e da Amazon[6] oferecem um serviço de *Identity & Access Management* (IAM).

Esses serviços implementam o conceito de mecanismo de autorização, que engloba um conjunto de funções lógicas bem definidas. Sua função principal é receber uma requisição de acesso do sujeito para executar uma determinada operação em um objeto, verificar se tal operação é permitida de acordo com as regras definidas na política de controle de acesso e retornar um resultado positivo ou negativo que é então seguido pelo sistema.

Essas funções são capturadas em uma arquitetura de referência (HU et al., 2014) composta por quatro componentes principais: *Policy Enforcement Point* (PEP), *Policy Decision Point* (PDP), *Policy Information Point* (PIP), e *Policy Administration Point* (PAP). Na Figura 1.7 é apresentada a arquitetura de referência com esses componentes que serão detalhados a seguir.

O PDP é responsável pela tomada de decisão sobre uma determinada requisição utilizando como base uma política de controle de acesso junto com os atributos disponíveis (do sujeito, do objeto, da operação e do ambiente). O PEP atua como um porteiro, interceptando as requisições de acesso de um sujeito que são então enviadas ao PDP, e cumprindo as decisões recebidas como respostas do PDP. O PIP é responsável por obter as informações que o PDP necessita para tomar suas decisões, podendo utilizar uma ou

[4]Não confundir com a autenticação de dois fatores (*Two-factor Authentication* – 2FA), que é comentado na Subseção 1.2.2.

[5]https://cloud.google.com/iam

[6]https://aws.amazon.com/iam/

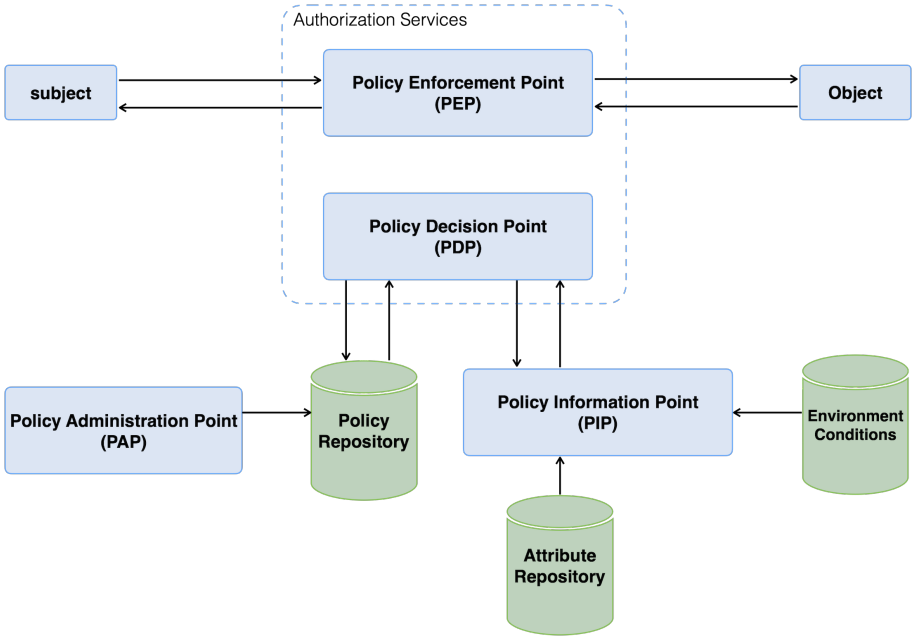


Figura 1.7: *Arquitetura de referência para mecanismos de controle de acesso. Fonte: (HU et al., 2014)*

mais fontes de atributos dentro da organização. O PAP corresponde a uma interface para a administração de políticas, que são então armazenadas em um repositório. Todos os componentes podem ser implementados de maneira centralizada ou distribuída, podendo estar logicamente e fisicamente separados uns dos outros ou agrupados em um único componente.

Tecnologias

Dentre as tecnologias consideradas como referências para autorização e controle de acesso a que mais se destaca é a XACML (RISSANEN, 2017). A *eXtensible Access Control Markup Language* é um padrão definido pela OASIS e compreende a definição de uma linguagem para criação de políticas de controle de acesso em formato interoperável, uma arquitetura e um modelo para o processamento de políticas e tomada de decisão. XACML define uma linguagem em XML para a especificação de políticas ABAC, e um protocolo para realizar requisições sobre decisões de controle de acesso (trocas de mensagens entre PEP e PDP), sendo implementado por diversas infraestruturas de autorização. Atualmente, em sua versão 3.0, a linguagem XACML define também um conjunto de extensões, incluindo um perfil para representação de políticas em formato JSON, perfil para definições de políticas RBAC, assim como um perfil para seu uso em arquiteturas RESTful.

O OAuth2 (HARDT, 2012) é um padrão para delegação de acessos em sistemas distribuídos. Sua principal funcionalidade é permitir que uma aplicação tenha acesso a recursos hospedados em outras aplicações em nome de um usuário. É considerado pela indústria o padrão *de facto* para autorização, sendo utilizado pelos grandes provedores de serviço da Internet (Amazon, Google, Facebook, Microsoft, e Twitter) para o compartilha-

mento de informações de contas de usuários para provedores de serviços de terceiros (veja o modelo de gestão de identidade centrado no usuário apresentado na Subseção 1.1.1).

A especificação OAuth2 provê suporte a diversos tipos de clientes (por exemplo, aplicações rodando em navegadores *web*, aplicações *web* executando em um servidor, aplicações em dispositivos móveis etc.) e define um conjunto de papéis, tipos de autorização e fluxos de autorização permitindo uma padronização e compartilhamento de informações entre organizações diferentes. Os fluxos definem as mensagens trocadas entre participantes, enquanto que os papéis identificam as responsabilidades dos participantes dentro do fluxo. Eles includem a identificação de proprietário de recurso (*Resource owner*) que delega o acesso a um cliente através de um servidor de autorização, o qual emite um *token* de acesso. O cliente então apresenta este *token* de acesso ao servidor do recurso que valida o *token* antes de liberar acesso ao recurso para o cliente.

O OAuth2 e o XACML desempenham funções distintas e podem ser facilmente utilizados em conjunto. Enquanto o OAuth2 pode ser utilizado para a delegação de acesso, políticas XACML podem ser utilizadas para decidir se um determinado acesso deve ser permitido ou não.

1.1.3. Privacidade e usabilidade

Do ponto de vista da engenharia de software, a usabilidade é considerada como um requisito de qualidade. Em IEEE... (1998), a usabilidade é definida como o quão fácil um usuário é capaz de aprender a operar, fornecer as entradas e interpretar as saídas de um sistema ou produto. Jøsang, Zomai e Suriadi (2007) definem um conjunto de princípios sobre usabilidade da segurança, classificando tais princípios como ações e conclusões. Eles consideram que o usuário deve entender quais ações de segurança precisa tomar e deve ter conhecimento e prática suficientes para tomar as ações corretas; além disso, a carga física e cognitiva para tomada dessas ações, mesmo que repetitivas, deve ser tolerável. As conclusões de segurança têm relação com a observação feita pelo usuário e, a partir desta, ele deve obter evidências sobre a segurança do sistema. Assim, em termos de usabilidade, para o usuário deve ser possível derivar a conclusão de segurança a partir da informação provida.

Segundo Schaar (2010), a maioria das pessoas possui pouca afinidade com tecnologia da informação e assim, não estaria na melhor posição para tomar decisões que impactam na proteção de seus dados pessoais e de outras pessoas. O conceito de privacidade desde a concepção (*Privacy by Design*), cunhado por Ann Cavoukian na década de 90 (CAVOUKIAN, 2009), considera que a privacidade dos usuários deve estar em foco desde a concepção do sistema ou produto e mantida até sua execução. Assim, considera que os sistemas ou produtos devam ser projetados de forma a minimizar a quantidade de informação pessoal tratada, além de fazer uso de mecanismos de segurança para garantir a proteção dos dados pessoais de seus usuários. Tais conceitos também estão presentes na Lei Geral de Proteção de Dados Pessoais (LGPD) (BRASIL, 2018, art. 46, §2) e na GPDR Europeia (UNION, 2016).

Como apresentado na Seção 1.2, a linha que separa o abuso do uso legítimo dos *cookies* pelos *sites* na Internet é tênue. A leis de proteção de dados pessoais exigem em muitas situações o consentimento do usuário antes que seus dados possam ser tratados. O

uso de *cookies* nos *sites* geralmente entram nessa situação, pois normalmente se enquadram nas situações em que o consentimento é exigido, haja visto que muitos são usados para propaganda e análise de comportamento e, portanto, precisam apresentar um termo de consentimento ao usuário. Para que o correto funcionamento do *site* não seja limitado ou completamente impedido, existe o conceito de *cookies* estritamente necessários, que se aplica ao que é essencialmente necessário armazenar para prover o serviço ao usuário, como por exemplo *cookies* para *login* em áreas protegidas ou *cookies* que mantém estado de carrinhos de compras[7]. Ainda assim, o usuário precisa ser avisado sobre o uso dos *cookies*.

O termo de consentimento também é apresentado aos usuários em soluções de autenticação que seguem o modelo de gestão de identidade federado (veja Subseção 1.1.1), como as federações acadêmicas, baseadas no protocolo SAML, ou que seguem o modelo centrado no usuário, como as aplicações que fazem uso do *login* social (contas Google, Facebook, Apple etc.), baseadas no protocolo OpenID Connect, que podem ainda questionar quais atributos o usuário deseja compartilhar com o provedor de serviço.

Os mecanismos usados atualmente pelos sistemas na *web*, para garantir a aderência às leis de proteção de dados, impactam diretamente na usabilidade destes sistemas. Em muitos casos eles podem até nem serem de fato aderentes à legislação, pois tornam os usuários habituados a consentirem por padrão, sem que o consentimento seja de fato livremente dado, específico, informado e sem ambiguidade, conforme dita a legislação. Por exemplo, os usuários são frequentemente inundados com *banners* de consentimento para o uso de *cookies*, com diferentes interfaces, apresentando-se como uma barreira entre o usuário e o serviço que ele deseja acessar. Em Utz et al. (2019) é apresentado o resultado de um estudo com mais 80.000 usuários sobre os formulários de consentimento para uso de *cookies* no qual foi notado que a maioria dos usuários está inclinada a sempre aceitar todos os *cookies* a ter que escolher individualmente quais *cookies* deseja permitir. Utz et al. (2019) consideram ainda que não basta apenar exigir o consentimento de usuário, também é necessário providenciar um guia de forma a padronizar uma interface para tal.

1.2. Demandas, desafios e tecnologias

Para que aplicações possam continuar operando de maneira adequada, é necessária a adequação aos diversos novos desafios, como demandas sociais e regulatórias de proteção de dados pessoais, novas formas de integração de serviço e diversidade de dispositivos utilizados pelo usuário, incluindo dispositivos que atuam em nome do usuário. Nesta seção são apresentadas as demandas históricas na área de gestão de identidade e de acesso, os novos desafios e as novas tecnologias e padrões para atender tais demandas, considerando o legado histórico de sistemas e a afinidade dos usuários com a tecnologia.

1.2.1. Autenticação federada, privacidade e novos mecanismos dos navegadores *web*

Muitas tecnologias e padrões usados para permitir autenticação e autorização federada ou centrada no usuário em aplicações *web* (e.g. SAML, OpenID Connect), apesar de terem sido desenvolvidas de maneira independente, foram projetadas para usufruir de funcionalidades de propósito geral presentes nos padrões *web*, como redirecionamento de

[7]https://gdpr.eu/cookies/

URL, parâmetros de URL, *iframe* e *cookies*. Nesta seção é feita uma breve apresentação sobre tais funcionalidades para depois apresentar os principais desafios para autenticação e autorização federada diante de alterações que já estão sendo implementadas pelos navegadores *web*, em particular o redirecionamento e restrições sobre *cookies*.

Na especificação do HTML5 (LAWSON et al., 2021), é dito que um contexto de navegação consiste em um ambiente no qual objetos do tipo `Document` são apresentados para o usuário. Assim, quando se abre uma nova janela ou aba em um navegador *web*, tem-se ali um contexto de navegação. O elemento HTML `iframe` representa um contexto de navegação aninhado, permitindo assim que uma janela do navegador *web* seja dividida em segmentos, cada qual podendo exibir um objeto `Document` diferente, sendo que cada documento pode ainda vir de diferentes servidores *web*. Toda vez que acessamos um *site* e este apresenta propagandas, oriundas do serviço *Google AdSense*, estamos presenciando o uso do elemento `iframe`. Tal elemento pode ser usado também para permitir a autenticação federada sem fricção, assunto que será apresentado logo mais nesta seção.

Os redirecionamentos de URL estão previstos na especificação HTTP (FIELDING; NOTTINGHAM; RESCHKE, 2022) para indicar ao agente do usuário (normalmente um navegador *web*) que este deve tomar alguma ação para que possa atender o pedido. O agente do usuário é informado por meio de um código de resposta da classe `3XX`. Por exemplo, para indicar que o recurso desejado (e.g. página HTML) está disponível em uma outra URL, é feito uso dos códigos `301`, movido permanentemente, ou `307`, redirecionamento temporário.

O formato de uma URL é definido por scheme://host:port/path?queryString#fragment. O elemento `path` consiste em segmentos de texto, delimitados pelo caractere `/`, que identificam um recurso na *web*. O elemento *query string* consiste em uma lista de parâmetros que é separada do recurso desejado pelo caractere `?`. Cada parâmetro consiste em um par (`nome=valor`) e são delimitados entre si pelo caractere `&`. Por exemplo: *https//www.exemplo.com/page?chave1=val1&chave2=val2&chave3=val3*.

A lista de parâmetros pode ser usada para alterar o comportamento de uma aplicação *web*. Por exemplo, o agente do usuário ao acessar a URL *https://www.exemplo.com/produtos?ordem=maior-valor*, indica a aplicação *web* que deseja receber a lista de produtos ordenada pelo maior valor. Como também pode ser usada em uma campanha de *marketing*, para saber de onde um usuário veio para chegar em uma determinada página. Por exemplo, a lista de parâmetros da URL *https://www.exemplo.com/?utm_source=twitter&utm_medium=tweet&utm_campaign=summer-sale* indica que o usuário chegou em `exemplo.com` a partir de um *tweet*, sobre uma liquidação de verão, publicado na rede social Twitter.

Os exemplos apresentados acima são considerados casos de propósito geral dos padrões *web*. Casos de uso específico, como o fluxo de autenticação federada, combinam o redirecionamento de URL com lista de parâmetros para permitir que seus usuários naveguem entre provedores de serviço e provedores de identidade. Por exemplo, quando um usuário acessa o serviço de periódicos da CAPES, ofertado por um provedor de

serviço na federação CAFe[8], este é redirecionado ao seu provedor de identidade, para que se autentique e, após isto, volta a ser redirecionado ao serviço da CAPES. O caminho de redirecionamentos é ditado pela lista de parâmetros, como apresentado nesta URL exemplo: *https://www.periodicos.capes.gov.br/Shibboleth.sso/Login?target=https://www.periodicos.capes.gov.br/secure&entityID=https://shibboleth.ifsc.edu.br/idp/shibboleth*.

Um *cookie* HTTP (BARTH, 2011) consiste de um conjunto de pares (*nome=valor*) e surgiu com o intuito de permitir ao agente do usuário (e.g. navegador *web*) manter o estado da aplicação na interação com um servidor HTTP, uma vez que o protocolo HTTP não mantém estado entre pedidos subsequentes. Por meio do campo *Set-Cookie* no cabeçalho HTTP, um servidor pode enviar um conjuntos de pares (nome=valor) e metadados associados para o navegador *web* do usuário, cabendo a este último decidir se armazena localmente ou simplesmente ignora as informações recebidas.

Atualmente, os *cookies* HTTP são usados devido a três principais motivos: gerenciamento de sessão – informações que geram facilidades na interação e só ficam persistidas durante a interação do usuário com a aplicação; personalização – preferências do usuário, como o idioma do conteúdo, e que devem ficar persistidas mesmo após o usuário finalizar a execução do navegador *web*; rastreamento – registros e análises do comportamento do usuário, podendo tais informações serem usadas com o intuito de *marketing* direcionado.

Os *cookies* podem ser ainda classificados como primários (*first-party*) ou de terceiros (*third-party*). Os *cookies* primários são criados e gerenciados pelo servidor HTTP, responsável pelo domínio *web* (*site*) que o usuário acessou diretamente. Os *cookies* de terceiros são criados por empresas detentoras de outros domínios *web*, cujo conteúdo é acessado indiretamente pelo usuário. Por exemplo, o usuário acessa o *site example.com*, cuja página HTML contém um *banner* de propaganda carregado a partir do domínio *example.net*. Esse segundo *site* solicita a persistência de um *cookie*, no caso, cria-se aqui um *cookie* de terceiro. Assim, quando este navegador *web* visitar outros *sites*, que também incluam recursos de *example.net*, este último conseguirá rastrear essa navegação (veja Figura 1.8).

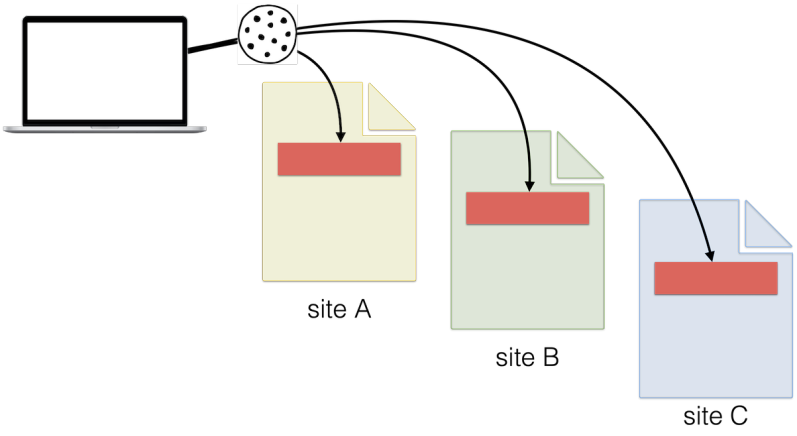


Figura 1.8: *Cookie de terceiro presente em diferentes sites que o usuário navega. Adaptado de Merewood (2020)*

[8]https://www.rnp.br/servicos/cafe

Ao criar um *cookie*, por meio do campo *Set-Cookie* no cabeçalho HTTP, o servidor pode definir algumas propriedades, como a *SameSite* (MEREWOOD, 2020), que no momento da escrita deste capítulo está em voga pelos principais navegadores *web*, porém ainda está como um *draft* IETF[9]. Esta propriedade permite que um *cookie* não possa ser enviado com requisições entre *sites*. Tem como objetivo proteger o usuário contra ataques de requisição forjada entre *sites* (*Cross-Site Request Forgery* – CSRF) e de *cookies* que objetivam fazer o rastreamento do usuário.

O redirecionamento de URL, parâmetros na URL e *cookies* são primitivas básicas das plataformas *web* e o uso legítimo destes possibilitaram uma melhor experiência de uso e a disponibilização de recursos mais ricos aos usuários. Porém, como são primitivas que podem ser usadas para uma grande variedade de finalidades, ao longo dos anos foram ocorrendo abusos no uso desses mecanismos, especialmente com propósitos de rastreamento e perfilamento de usuários, para propaganda direcionada, por exemplo. Isso têm causado um aumento de pressão social e especialmente regulatória por mais privacidade e proteção de dados.

Os desenvolvedores de navegadores *web* acabam intervindo e tomando medidas para mitigar a situação. Inclusive, tomam para si muitas vezes, conforme pontuado por Geradin, Katsifis e Karanikioti (2020), o papel de um "regulador de privacidade de fato", ao assumirem a interpretação dessas regulamentações e tomarem decisões de implementação para atendê-las, como a recente decisão dos principais desenvolvedores dos navegadores *web* de tornar por padrão o bloqueio total de *cookies* de terceiros.

A Google criou a iniciativa chamada *Privacy SandBox*[10] que tem por objetivo propor novas tecnologias que permitam proteger a privacidade dos usuários na *web* ao mesmo tempo que dá ferramentas para que desenvolvedores e empresas possam conduzir seus negócios na *web*. A empresa está buscando alternativas aos *cookies* de terceiros, iniciando com a proposta chamada *Federated Learning of Cohorts* (FloC) que posteriormente foi substituída pela proposta chamada *Topics*[11].

Em 2017, a empresa Apple implementou o *Intelligent Tracking Prevention* (ITP) (WILANDER, 2019) em seu navegador *web* Safari que teve como foco *cookies* de terceiros, porém, atualmente, a solução é mais ampla e bloqueia alternativas de contorno encontradas pelas empresas de anúncios, como os *link decoration*. A Mozilla implementou o *Enhanced Tracking Protection* (ETP) em seu navegador *web* Firefox, que bloqueia rastreadores de mídia social, *cookies* de rastreamento *cross-site*, *fingerprinters*, mineradores de criptomoedas, rastreadores de conteúdo e mais recentemente, remove parâmetros de URLs, a partir de uma lista conhecida, conforme configuração definida pelo usuário.

A decoração de *links* (*link decoration*) consiste na prática em usar parâmetros de URL para rastrear o usuário e assim contornar as restrições impostas com o bloqueio de *cookies* de terceiros. Os parâmetros podem ser adicionados na URL de maneira estática ou dinâmica, usando rotinas em Javascript, quando o usuário clica em um determinado elemento da página, como um anúncio.

[9]https://datatracker.ietf.org/doc/html/draft-west-first-party-cookies-07
[10]https://privacysandbox.com/
[11]https://blog.google/products/chrome/get-know-new-topics-api-privacy-sandbox/

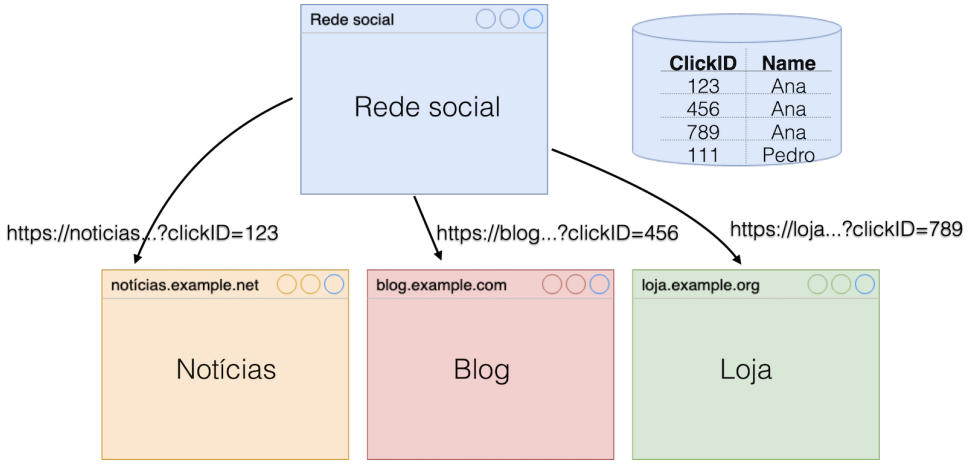


Figura 1.9: *Exemplo de link decoration com finalidade de rastreamento. Adaptado de Wilander (2019)*

No exemplo apresentado na Figura 1.9, a rede social adiciona o parâmetro *clickID* em todos os *links* externos ao seu domínio e o valor associado a este parâmetro tem ligação direta com um usuário real desta rede social. Quando o usuário clica no *link* para a ir ao site *loja.example.org*, a rede social faz uso de rotinas Javascript, já embutidas no *site* da Loja por algum outro motivo que seja interessante para este site, para extrair o parâmetro *clickID* da URL e o persiste como *cookie* primário do site da loja. O mesmo ocorre quando o usuário acessa os demais *links* a partir da rede social. Assim, toda vez que o usuário retornar à loja, a rotina Javascript da rede social lê o *cookie* primário e o encaminha para seus servidores, possibilitando assim rastrear o usuário.

Tem-se aqui mais um caso de abuso das primitivas da *web* e alguns navegadores *web*, como Safari e Firefox, estão criando mecanismos para coibir tal prática, identificando e removendo da URL parâmetros que são reconhecidos como rastreadores. Alguns sites, como o Facebook, começaram a cifrar toda URL de forma que estes mecanismos de bloqueio não consigam distinguir a parte útil da URL dos parâmetros que foram colocados unicamente com o propósito de rastreamento.

Iniciativas da comunidade de gestão de identidade

Redirecionamento e parâmetros de URL são essenciais para o funcionamento dos principais protocolos usados na autenticação federada. O grande desafio então é como manter o funcionamento das federações de identidade, com as políticas de privacidade de navegadores cada vez mais restritivas, respeitando a privacidade do usuário e melhorando sua usabilidade.

Embora o objetivo da arquitetura de identidade federada não seja rastrear o usuário, ela possibilita que o usuário possa ser rastreado pelo IdP, o qual sempre é contatado antes que consiga acessar os recursos providos pelo SP. Assim, o IdP toma conhecimento dos serviços que o usuário acessa, sem que isso seja de fato uma necessidade. Os SPs também podem fazer conluio para diretamente ou probabilisticamente ligarem seus usuários e assim fazer a prática de enriquecimento de perfil, adicionando mais dados do que realmente precisam sobre seus usuários.

Iniciativas de debates na comunidade de GId têm sido realizadas na forma de eventos, como por exemplo o *Federated Identity and Browser Workshop*, apoiado pela REFEDs (*Research and Education FEDerations group*) e realizado em 2021. A partir desse *workshop* houve uma articulação para criar um grupo comunitário da W3C sobre o tema. Atualmente este grupo está trabalhando na *Federated Credential Management*[12] que consiste em uma API para permitir que usuários façam login em *sites web* com suas contas federadas e ainda assim tendo a sua privacidade preservada.

O projeto SeamlessAccess (SEAMLESSACCESS, 2021) é uma iniciativa conjunta de quatro entidades ligadas à educação e publicação de pesquisa científica, a GÉANT[13], Internet2[14], NISO[15] (*National Information Standards Organization*) e STM[16] (*International Association of STM Publishers*), com o objetivo de resolver os atuais problemas de usabilidade das federações acadêmicas baseadas em asserções SAML, para permitir uma experiência de autenticação única (*Single Sign-On* – SSO) verdadeira e transparente.

Nessa proposta, o serviço de descoberta de IdP (*Discovery Service* – DS) continua a existir, porém os SPs podem escolher um dos três modos como o DS seria apresentado aos seus usuários: modo limitado, semelhante ao que se tem em alguns serviços na federação CAFe, ou seja, com o redirecionamento HTTP para o DS bem evidente ($SP \rightarrow DS \rightarrow IdP \rightarrow SP$); modo padrão, no qual o DS aparece embarcado e integrado com a página *web* do próprio SP e o redirecionamento ao IdP só aconteceria no primeiro acesso do usuário a um serviço federado durante aquela sessão do navegador *web* ($SP \rightarrow IdP \rightarrow SP$); modo avançado, com comportamento semelhante ao modo padrão, porém, permite ao SP indicar a lista de IdP que confia.

Alguns provedores de serviços presentes na federação CAFe, como o da Escola Superior de Redes, possuem o DS embarcado em sua própria página, permitindo que o usuário siga o fluxo $SP \rightarrow IdP \rightarrow SP$. Porém, se o usuário acessar outros provedores de serviços, durante a mesma sessão no navegador *web*, este ainda terá o passo adicional em indicar seu IdP novamente, em um DS embarcado ou dedicado, antes que consiga acessar o recurso desejado.

Com o projeto SeamlessAccess, a escolha do IdP do usuário é persistida no recurso de armazenamento local do navegador, o *localStorage*, e este permite que o usuário possa acessar diferentes provedores de serviços sem que tenha a necessidade de indicar o seu IdP em cada acesso. Ao contrário dos *cookies*, a informação armazenada no *localStorage* não é encaminhada em cada requisição e somente o domínio que a escreveu é quem tem permissão de lê-la. Assim, a solução proposta pelo SeamlessAccess continua efetiva mesmo diante das restrições de bloqueio de *cookies* de terceiros por parte dos navegadores *web*.

[12]https://fedidcg.github.io/FedCM
[13]https://geant.org/
[14]urlhttps://internet2.edu/
[15]https://niso.org/
[16]https://stm-assoc.org/

1.2.2. Robustez do processo de autenticação

Segundo NIST (2017a), o processo de autenticação digital busca garantir que um determinado sujeito possui controle sobre um ou mais autenticadores (e.g. senha, chave privada etc.) que estejam associados à sua identidade digital. Os sistemas de autenticação estão fundamentados sobre três fatores de autenticação:

- Aquilo que você sabe – senha, número de identificação pessoal (*Personal Identification Number* – PIN) etc;
- Aquilo que você possui – carteira de identificação, *token* criptográfico etc;
- Aquilo que você é – biometria do sujeito.

O primeiro uso do par *(username, password)* em sistemas computacionais é atribuído a Fernando Corbató na década de 60 (YADRON, 2014), quando buscava um meio para permitir o compartilhamento de computadores do tipo *Compatible Time-Sharing System* (CTSS) por vários usuários, de forma que cada usuário tivesse uma área privativa onde seus arquivos não poderiam ser vistos pelos demais usuários.

Apesar das fragilidades, o par *(username, password)*, ainda é amplamente empregado em processos de autenticação digital de sujeitos (DASGUPTA; ROY; NAG, 2017). Assim, sistemas de autenticação que dependem exclusivamente deste par precisam fazer uso de outros mecanismos de segurança para lidar com usuários que fazem uso de estratégias ruins para escolha de senha, que geram facilidade para ataques de força bruta, com possíveis *malwares* no dispositivo do usuário, que visam a captura da senha, ou ainda, com técnicas de engenharia social como o *phishing*, que também visam a captura da senha. Em Mello e Chaves (2020), notou-se que alguns provedores de serviços comerciais optaram por senha de uso único (*One-Time Password* – OTP) que é enviada para o email do sujeito durante o processo de autenticação. Assim, assumem que somente o usuário teria acesso à sua caixa postal, não precisam implementar processos para recuperar senhas e evitam a escolha de senhas frágeis por parte de seus usuários.

Fatores de autenticação do tipo “aquilo que você possui” ou “aquilo que você é” não estão suscetíveis aos mesmos ataques que fatores do tipo “aquilo que você sabe” estão. Por exemplo, um atacante remoto, que faz uso de força bruta para descoberta de senha, seria capaz de fazer o comprometimento em massa de contas de usuários de um serviço, considerando que não existam outros mecanismos de segurança para detecção e mitigação do ataque. Porém, se a autenticação do usuário também exigir algo que esteja fisicamente perto do usuário (e.g. um *token* criptográfico, sua impressão digital), tal ataque não teria sucesso.

A autenticação multifator (*Multi-Factor Authentication* – MFA), que em alguns casos também é chamada de autenticação com dois fatores (*Two Factor Authentication* – 2FA), visa aumentar a robustez do processo de autenticação por meio da combinação de mais de um fator de autenticação. Parte-se do pressuposto que aumenta muito o grau de dificuldade para que o atacante consiga comprometer mais de um fator.

Apesar do conceito de autenticação com dois fatores não ser algo novo, foi somente há pouco tempo que de fato ganhou popularidade. Atualmente, os principais provedores

de serviços comerciais na Internet exigem ou permitem que seus usuários façam uso da autenticação com dois fatores. Como primeiro fator é comum o uso de senhas escolhidas pelo próprio usuário (aquilo que você sabe) e como segundo fator as senhas de uso único (OTP), podendo estas serem enviadas por email, SMS, ligação telefônica ou ainda gerenciadas por meio de aplicativos (e.g. Google Authenticator ou Authy) em dispositivos móveis (veja Figura 1.10). Neste caso, assume-se que o sujeito está em posse do dispositivo do qual conseguirá obter a senha que deverá ser apresentada no processo de autenticação.

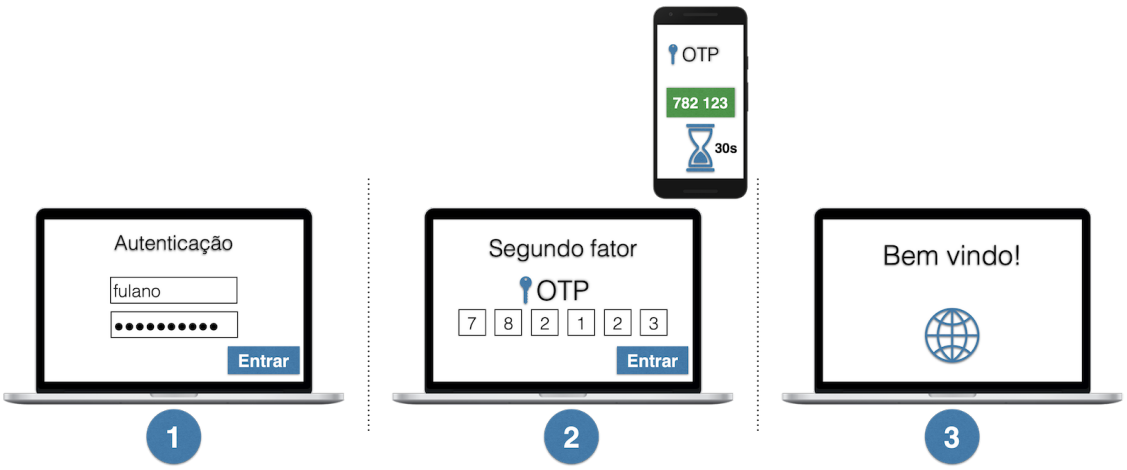


Figura 1.10: *Autenticação com dois fatores, tendo senha de uso único como segundo fator*

Segundo NIST (2017a), o uso de dois fatores de autenticação é o suficiente para um sistema de autenticação atingir o maior nível de confiança, na escala de um a três, definido em (NIST, 2017b), desde que os fatores usados sejam resistentes ao ataque de *phishing*. O uso de outras informações, como a localização geográfica ou o identificador do dispositivo usado pelo sujeito, ajudam na análise de risco para determinar se o sujeito é de fato detentor dos autenticadores, porém tais informações não são consideradas fatores de autenticação.

As senhas de uso único, enviadas por meio do serviço telefônico (ligação ou SMS) são consideradas de uso restrito pelo NIST, uma vez que existe um risco sobre a personificação da estação móvel do usuário. Assim, considera-se alinhado ao primeiro nível de confiança (o mais baixo nível), um sistema de autenticação que faz uso de uma senha como primeiro fator e senhas de uso único como segundo fator, desde que se faça uso de *tokens* físicos dedicados ou aplicativos TOTP (M'RAIHI et al., 2011) em telefones inteligentes. Apesar das senhas de uso único como segundo fator aumentarem a robustez do processo de autenticação, soluções baseadas no envio desta senha por email, ligação telefônica ou SMS ainda estão sujeitas a ataques de engenharia social, onde agentes maliciosos iniciam o processo de autenticação e induzem a vítima a fornecer o código OTP recebido.

O uso de aplicativos dedicados para geração do código OTP podem tornar este tipo de ataque mais difícil, pois exige uma consciência muito maior da vítima, podendo esta perceber que trata-se de um ataque. Porém, estes aplicativos apresentam um ponto negativo com relação a usabilidade, pois para cada autenticação iniciada por um usuário

correto, em um serviço correto, o usuário precisará procurar, abrir o aplicativo e transcrever o código que vê, dentro da janela de 30 segundos, no formulário de autenticação. A usabilidade fica ainda mais prejudicada se o usuário estiver usando seu telefone inteligente para acessar o serviço desejado e neste dispositivo estiver o aplicativo gerador de código OTP (REESE et al., 2019). Por fim, usuários de aplicações *web* que fazem uso deste tipo de solução ainda estão suscetíveis a ataque de *phishing* combinado com o ataque do homem no meio (*man-in-the-middle*) e sequestro de sessão por meio de *cookies* (GRIMES, 2019; MICROSOFT, 2022).

Soluções como o *Google Prompt* ou *Duo Push* surgiram como uma solução para prover uma melhor experiência de uso, quando comparados com aplicativos gerenciadores de códigos OTP. Durante o processo de autenticação, depois do usuário fornecer o primeiro fator, uma notificação aparece no telefone inteligente do usuário e basta esse pressionar o botão `SIM` para confirmar que é ele quem está querendo autenticar-se. Tal solução está suscetível a ataques, batizados de *MFA prompt bombing* (GOODIN, 2022), que apostam na falta de atenção da vítima para as notificações que aparecem em seu dispositivo.

A partir de 2015, a *Fast IDentity Online Alliance* (FIDO) publicou um conjunto de especificações abertas com o intuito de permitir que a autenticação de usuários na *web* seja simples e robusta. As especificações estão fundamentadas sobre criptografia de chave pública, autenticadores baseados em *hardware* seguro para armazenamento do material criptográfico, como *Secure Element* (SE), *Trusted Execution Environment* (TEE) e *Trusted Platform Module* (TPM). No caso é criado um par de chaves criptográficas, que não pode ser extraído do dispositivo FIDO, para cada *site* ou aplicativo (chave fica vinculada à URI do *site* ou aplicativo) que o usuário for se autenticar. Tem-se ainda a facilidade para o usuário evitar *phishing*, pois este não é responsável por confirmar se um *site* é realmente quem ele afirma ser, sendo esta uma responsabilidade do dispositivo FIDO.

Foram publicados os padrões *Universal Second Factor* (U2F) (SRINIVAS et al., 2017), *Universal Authentication Framework* (UAF) (MACHANI et al., 2020) e o *Client to Authenticator Protocol* (CTAP) (BRADLEY et al., 2021), sendo este último um complemento à especificação *Web Authentication* (WebAuthn) (HODGES et al., 2021) da W3C, que algumas vezes também é referenciada como FIDO2. Atualmente a recomendação é que provedores de serviços façam uso do WebAuthn e CTAP e não usem mais UAF e U2F, porém tais padrões serão explicados na sequência dada sua grande relevância histórica e para que o leitor entenda as implicações de cada alternativa que surgiu na evolução dos padrões FIDO.

A especificação U2F teve como foco dispositivos físicos (*tokens* criptográficos) que pudessem ser usados exclusivamente como segundo fator de autenticação, uma vez que o dispositivo em si não possui qualquer tipo de mecanismo de autenticação local que impeça seu uso por qualquer pessoa. De acordo com a especificação, os dispositivos U2F devem possuir um mecanismo físico, normalmente um botão capacitivo, para confirmar que o usuário está presente e participando de forma ativa de um pedido de autenticação (veja Figura 1.11). O motivo para isto é que, como os dispositivos U2F podem estar constantemente conectados na porta USB do computador do usuário, um aplicativo malicioso neste computador poderia iniciar um pedido de autenticação e usar o dispositivo sem que o usuário soubesse.

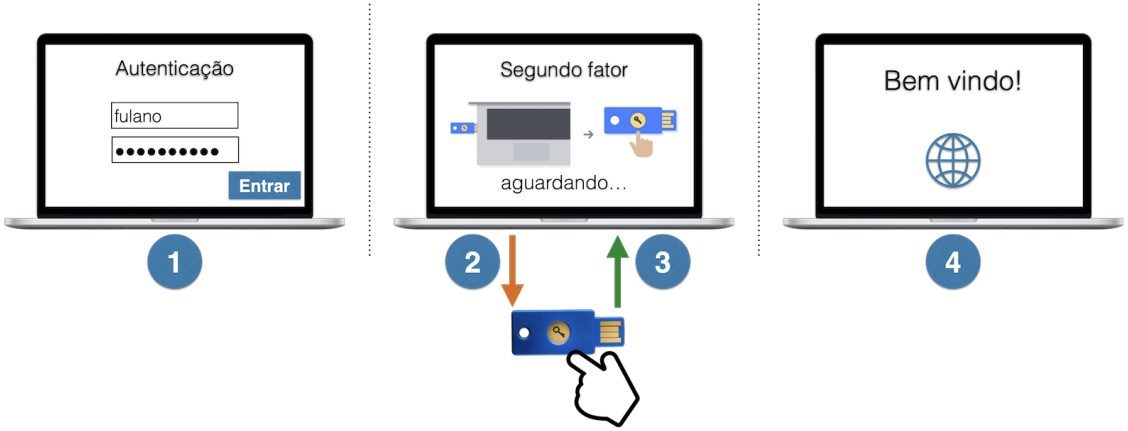


Figura 1.11: *Autenticação com dois fatores, tendo dispositivo U2F como segundo fator*

Dispositivos U2F podem possuir interface USB (A ou C), Bluetooth BLE ou *Near Field Communication* (NFC), permitindo assim que possam ser usados em computadores, telefones inteligentes ou *tablets*. Na Figura 1.12 são apresentados alguns exemplares de produtos que implementam a especificação U2F e também FIDO2. Todos os produtos implementam interface de comunicação NFC, sendo que os produtos da Google e da Yubiko possuem interface de comunicação USB C e da Solokey interface USB A.

Figura 1.12: *Chaves FIDO2 Google Titan, Yubiko Yubikey e Solokey*

A especificação UAF teve como foco proporcionar uma experiência de autenticação totalmente sem senha, isto é, um dispositivo que esteja de acordo com a UAF pode ser usado como o único fator de autenticação. No caso, este dispositivo (i.e computador ou telefone inteligente) precisa possuir um hardware de seguro (e.g. SE, TEE, TPM) para armazenar o material criptográfico e uma forma de autenticação local no dispositivo, normalmente por meio de algum leitor biométrico, para que seja destravada a chave privada para uso em um processo de autenticação. Dispositivos UAF estão aderentes ao terceiro nível de confiança de sistemas de autenticação definido em (NIST, 2017b).

Os padrões UAF e U2F são complementares e, apesar do U2F ter tido uma maior adoção no mercado, o número de provedores de serviços que fizeram uso deste ainda era pequeno. A especificação WebAuthn conduzida pela W3C em conjunto com a FIDO Alliance surgiu como uma solução para permitir a ampla adoção dos padrões FIDO, uma vez que a especificação padroniza uma API para os navegadores *web*. Atualmente, os principais navegadores *web*, de dispositivos móveis ou de computadores, possuem suporte

a especificação WebAuthn[17] e um grande número de provedores de serviços comerciais (e.g. Dropbox, GitHub, Google, iCloud etc.) já permitem o uso do WebAuthn como um fator de autenticação.

A especificação WebAuthn permite dispositivos como chaves de segurança USB (chamados de autenticadores externos) ou dispositivos como um telefone inteligente ou computador (chamados autenticadores de plataforma) que possuam um hardware seguro (e.g. TEE) mais um leitor que permita autenticação biométrica (e.g. leitor de impressão digital, reconhecimento facial etc.). Os dispositivos WebAuthn podem ser usados como o único fator de autenticação, quando for um telefone ou computador, ou como segundo fator de autenticação, quando for uma chave USB que não permite realizar a autenticação local do sujeito. Assim, autenticação baseada nos padrões FIDO conseguem atingir o mais alto nível de confiança (nível 3) descrito em (NIST, 2017b).

Dispositivos WebAuthn (de plataforma ou externos), na fase de registro (e.g. criação da conta do usuário em um provedor de serviço) sempre irão gerar um novo par de chaves criptográficas exclusivo para aquela conta de usuário e para aquele provedor de serviço. Se o dispositivo WebAuthn tivesse uma única chave privada e não gerasse uma nova por provedor de serviço, os provedores poderiam formar um conluio para tentar rastrear as atividades deste usuário, sendo que a chave privada atuaria como um identificador único e universal deste usuário.

O fato de sempre gerar um novo par de chaves na fase de registro, permite que dispositivos WebAuthn possam ser compartilhados, por exemplo, por todos os membros de uma família, sem possibilitar que um usuário consiga usar as chaves criptográficas do outro usuário. Assim, um mesmo dispositivo WebAuthn pode ser usado por um sujeito em diferentes provedores de serviço, por diferentes sujeitos em um mesmo provedor de serviço ou por diferentes sujeitos em diferentes provedores de serviços.

Se o dispositivo WebAuthn tiver capacidade de armazenamento, a chave privada pode ser armazenada nele, porém no caso de chaves USB como aquelas apresentadas na Figura 1.12, informações usadas para gerar a chave privada (mas não a própria chave) são encapsuladas e armazenadas no provedor de serviço. Durante a fase de autenticação, essas informações são encaminhadas pelo provedor de serviço ao computador do usuário e este as encaminha à chave USB, a qual será capaz de derivar a chave privada a partir destas informações, para então empregá-la no processo de autenticação.

Em NIST (2017b), é apresentado um conjunto de eventos que podem ocorrer durante do ciclo de vida de um autenticador (e.g. uma senha ou um dispositivo WebAuthn) que afeta diretamente o uso deste autenticador. Os eventos incluem a associação do autenticador à conta do sujeito no provedor de serviço, a perda, o roubo, a duplicação não autorizada, a expiração e a revogação.

Se o autenticador for uma senha e seu detentor vier a perdê-la, este poderia recorrer ao processo de redefinição de senha, que geralmente é composto por três passos: pedido de recuperação; verificação se o pedido não foi feito por um *bot*; redefinição de senha. O último passo pode fazer uso de perguntas de segurança previamente cadastradas pelo sujeito ou o envio de um código temporário para o email do sujeito o qual lhe permitirá

[17] https://webauthn.me/browser-support

redefinir a senha (MAQBALI; MITCHELL, 2018).

Se o autenticador perdido for do tipo "aquilo que você possui", como um *token* criptográfico, uma chave USB WebAuthn ou a semente usada para geração das senhas de uso único (OTP), o processo de redefinição não pode ser semelhante ao de redefinir uma senha, uma vez que tal processo não é tão robusto quanto o próprio processo de autenticação, tornando-se assim um possível alvo de ataques.

Alguns aplicativos gerenciadores de senhas de uso único, como o Authy[18], permitem que o usuário faça uma cópia de segurança das sementes que possui em seu telefone inteligente na nuvem, ou mesmo sincronizar as sementes por múltiplos dispositivos. A cópia de segurança é protegida unicamente por uma senha, que se for perdida, não poderá ser recuperada e, por consequência, tornará a cópia de segurança permanentemente indisponível.

Segundo o relatório de 2021 da Gartner (PHILLIPS, 2021), o WebAuthn venceu o pico das expectativas infladas e situa-se no vale da desilusão, uma vez que sua adoção vem avançando e conquistando um grande número de clientes. Porém, as chaves USB WebAuthn ainda enfrentam alguns desafios, do ponto de vista dos usuários finais. As chaves estão disponíveis para venda em poucos mercados, restringindo-se principalmente aos Estados Unidos da América e alguns países na Europa. As chaves custam em média US$ 35, assim observa-se uma adoção maior por empresas, que adquirem para seus funcionários, e um menor interesse quando a pessoa precisa adquirir com recursos próprios.

Além da logística e do custo monetário, as chaves USB geram uma dificuldade extra para os usuários caso estes venham a perdê-la. Ciente que uma mesma chave USB pode ser usada em diferentes provedores de serviços, a perda deste dispositivo pode deixar o usuário sem conseguir acessar um grande número de serviços, caso este não tenha configurado outros fatores de autenticação como opção de *backup*. Problema semelhante ocorre quando se usa autenticadores de plataforma, embarcados em telefones ou computadores. Apesar de ser mais difícil perder tais dispositivos, a troca destes, por exemplo quando o sujeito adquire um novo, é algo mais comum. Assim, este sujeito ficaria impossibilitado de passar pelo processo de autenticação com este novo dispositivo e para evitar isto, e não depender de fatores mais frágeis, precisaria ter também uma chave USB WebAuthn (autenticação externo) associada à sua conta.

O WebAuthn é um fator de autenticação robusto, resistente a *phishing*, simples de usar, mas a impossibilidade de recuperar as chaves privadas de dispositivo perdido faz com que os usuários continuem dependentes de outros fatores. Para o usuário comum, a autenticação é algo que deve simplesmente funcionar sem que necessite adquirir dispositivos adicionais ou lidar com inconveniências.

Em 2022, a FIDO Alliance e a W3C propuseram uma nova versão do nível 3 da especificação WebAuthn, chamada de *Multi-device FIDO Credentials* - para a qual algumas empresas estão usando o termo chaves de acesso (*passkeys*) (FIDO, 2022). Na proposta, considera-se que as chaves de acesso (*passkeys*) serão capazes de substituir as senhas (*password*) até mesmo em cenários que exigem um alto nível de segurança e confiança nos autenticadores.

[18]https://authy.com/

A proposta busca avançar em dois cenários principais: permitir que telefones inteligentes possam ser usados como autenticadores externos (como uma chave *Bluetooth*) e propor alterações na implementação dos autenticadores de plataforma (computadores e telefones) para permitir que as credenciais FIDO (chaves privadas) possam ser sincronizadas por múltiplos dispositivos, permitindo que o usuário as transfira facilmente para um novo computador ou telefone recém adquiridos.

A experiência do usuário com as chaves de acesso será semelhante a experiência com gerenciadores de senhas, que podem ter sua base sincronizada por diversos dispositivos, são protegidos por uma única senha principal e que permitem o preenchimento automático do formulário de autenticação. Desta forma, os usuários conseguirão acessar suas chaves de acesso (*passkeys*) nos seus diferentes dispositivos, mesmo em novos dispositivos, sem a necessidade de passar pelo processo de registro de credenciais para cada *site* ou aplicativo que já o tenha feito anteriormente.

Se o usuário possuir dispositivos de um mesmo fabricante, por exemplo, um telefone Android e um laptop Chromebook, então haverá a sincronização automática das chaves de acesso por meio de sua Conta Google. Porém, também é possível o cenário no qual o usuário possui um Chromebook e um telefone com o sistema iOS, ou seja, de diferentes fabricantes. Por exemplo, o usuário possui uma chave de acesso criada para o site *example.com* quando ele o acessou por meio de seu telefone com iOS. Quando este usuário for acessar pela primeira vez o site *example.com* por meio de seu computador (que também possui um autenticador de plataforma), o navegador *web* apresentará um diálogo questionando se deseja usar seu telefone como chave *Bluetooth*. Se sim, este usuário fará autenticação local em seu telefone (e.g. usando o reconhecimento facial) e, autenticando com sucesso, o site *example.com* pode associar uma nova chave de acesso específica para este computador. Desta forma, nos próximos acessos por meio desse computador não será mais necessário fazer uso do telefone inteligente como uma chave *Bluetooth*.

A segurança e o sincronismo das chaves de acesso de um usuário depende diretamente do sistema operacional subjacente do autenticador (plataforma) para suas contas *online* (e.g. Conta Google) e do método de segurança para restabelecer o acesso às chaves de acesso quando todos os dispositivos foram perdidos (FIDO, 2022). Apple, Google e Microsoft firmaram compromisso para acelerar a adoção das chaves de acesso e é esperado que os primeiros serviços, e adequações nos sistemas operacionais, estejam disponíveis ao longo do ano de 2023 (APPLE, 2022).

Por fim, a proposta também possibilita o cenário onde um provedor de serviço, por questões regulatórias ou de segurança, deseja realizar passos adicionais para ter certeza sobre o usuário, quando este apresentar uma chave de acesso por meio de novo dispositivo. Neste caso, a proposta permitirá que o provedor de serviço crie uma chave criptográfica adicional vinculada ao dispositivo. Assim, nos pedidos de autenticação posteriores, esta chave do dispositivo é também usada e ajudará ao provedor de serviço ter certeza de que o usuário está fazendo uso de um dispositivo já conhecido e não de um novo dispositivo.

Sistemas Adaptativos de Autenticação

Sistemas adaptativos de autenticação[19] estão aptos a modificarem dinamicamente seu comportamento para escolha do(s) melhor(es) mecanismo(s) em resposta a fatores contextuais, tais como localização, proximidade de dispositivos e outros atributos (CABARCOS; KRUPITZER; BECKER, 2019). Tornar adaptativa a autenticação nos provedores de identidade permite que esses ofereçam diversidade de mecanismos e de fatores de autenticação, monitoramento em tempo real (de modo a validar o usuário durante a sessão estabelecida – autenticação contínua) e extensibilidade para novas tecnologias e fatores de autenticação, sem que seja necessário a mudança completa da arquitetura do sistema.

Um sistema adaptativo compreende duas partes: um conjunto de recursos gerenciados e a lógica de adaptação. Mapeando para o domínio de autenticação, os recursos gerenciados são os autenticadores (disponíveis nos dispositivos e em aplicativos do usuário) e a lógica de adaptação é a camada de *software* encarregada de orquestrar seu uso de acordo com a situação detectada. Além disso, a lógica de adaptação pode ser executada no mesmo dispositivo da aplicação que deseja utilizar os autenticadores, ou em outro dispositivo, possibilitando diferentes casos de uso. Um exemplo de autenticação adaptativa com lógica no dispositivo ocorre quando um *smartphone* que detecta quando o usuário está em casa e desativa a proteção por senha (modificação automática de comportamento) até que ele se mude para um local diferente (HAYASHI et al., 2013). Outro exemplo, quando a lógica de adaptação é distribuída, é um sistema no qual um usuário se autentica com o leitor de impressão digital do *smartphone* para acessar seu *laptop*, quando os dois dispositivos estão próximos. A autenticação dinâmica, além de aumentar o nível robustez do processo de autenticação do ponto de vista de segurança computacional, também pode melhorar a usabilidade do sistema, visto que pode se adaptar para promover baixa fricção do usuário ao se autenticar.

De acordo com Dasgupta, Roy e Nag (2017), a autenticação contínua, também conhecida como autenticação ativa, foi introduzida em 2012 para abordar novas formas de validar a identidade dos usuários, em vez de usar apenas senhas tradicionais. O foco estava principalmente na biometria de comportamento baseada em *software* que capturava os dados da sessão para determinar se o usuário legítimo estava usando o sistema em um determinado momento. Segundo o Programa de Autenticação Ativa da DARPA (GUIDORIZZI, 2013), uma pessoa pode ser autenticada em intervalos regulares por:

- Aspectos físicos – impressão digital, geometria facial etc.;
- Interação com o sistema – padrão de pressionamento de tecla, padrão de digitação e movimento do mouse etc.;
- Contexto existente do usuário – análise semântica estrutural, como o usuário constrói sentenças, forense de autoria etc.; ou uso de dados de suas experiências, sendo linguística computacional ou como o usuário usa a língua.

[19]Na literatura, termos semanticamente similares são: autenticação ciente e baseada em contexto, autenticação ciente e baseada em risco.

A inclusão de diversos conjuntos de autenticadores melhora a flexibilidade do sistema adaptativo, favorecendo sua aplicabilidade a diferentes cenários. Alguns sistemas se concentram em melhorar a usabilidade por meio da adaptação, para a qual a biometria comportamental é uma boa candidata, uma vez que o usuário pode ser implicitamente "sentido" sem exigir interação explícita (RYU et al., 2021). A autenticação implícita possibilita que o usuário se autentique em algum serviço sem precisar que o mesmo coloque suas credenciais manualmente ou requer pouco envolvimento ativo do usuário (JAKOBSSON et al., 2009), o que proporciona uma grande aceitação (CABARCOS; KRUPITZER; BECKER, 2019).

As abordagens de autenticação implícita do usuário são ótimas candidatas para realizar a autenticação contínua. Por exemplo, as atividades na *web* de um usuário podem ser continuamente verificadas quanto a irregularidades em seu fluxo de trabalho normal e padrões de interação da interface do usuário. Os aplicativos de *smartphone* podem verificar regularmente a bio-assinatura ou o padrão de comportamento baseado em localização do usuário para detectar um impostor. A autenticação implícita e contínua pode ser utilizada para aumentar a qualidade do processo de autenticação, dado que tanto características comportamentais (JAKOBSSON et al., 2009) quanto contextuais (WU et al., 2019) podem ser utilizadas com a finalidade de detectar se o usuário conectado ainda é o usuário inicialmente autenticado. A autenticação implícita pode ainda ser usada para melhorar a experiência do usuário quando se faz uso de um segundo fator de autenticação.

Outro exemplo de abordagem adaptativa é a autenticação baseada em risco (*Risk-Based Authentication* - RBA), no qual um sistema captura e armazena diferentes tipos de informações (e.g. dados do usuário, do dispositivo, endereço IP, geolocalização, metadados do navegador, tipo de operação a ser executada no sistema etc.), a partir disso calcula uma pontuação e gera uma classificação de risco (e.g. baixo, médio ou alto), e então decide o nível apropriado de segurança e o mecanismo ou fator de autenticação a ser utilizado (NIST, 2017b). Muitas soluções RBA usam aprendizado de máquina. Os algoritmos dessas ferramentas monitoram e aprendem o comportamento do usuário ao longo do tempo para criar um perfil preciso dos padrões de login de um determinado usuário. Estas podem monitorar em tempo real dispositivos, horários típicos de login do usuário ou locais de trabalho habituais para identificar anomalias nos padrões de autenticação do mesmo. Eles verificam endereços IP e reputações de rede, além de dados de ameaças para essas redes no caminho da autenticação (como redes comprometidas).

Na Figura 1.13 é ilustrado um exemplo de RBA, no qual, se o risco for considerado baixo (por exemplo, dispositivo comum, local e horário usuais), exige-se somente uma autenticação baseada no par (usuário e senha). Em um risco médio (por exemplo, dispositivo desconhecido em local e em horário usuais), o serviço RBA solicita ao usuário informações adicionais (por exemplo, verificação de endereço de e-mail). Se a pontuação de risco for considerada alta (por exemplo, dispositivo desconhecido em local irreal e em horário incomum), uma nova requisição com um segundo fator pode ser solicitada ao usuário. No caso de uma pontuação de risco alto e operação a ser executada for crítica, o serviço poderá bloquear a tentativa de acesso (WIEFLING; DÜRMUTH; LO IACONO, 2020).

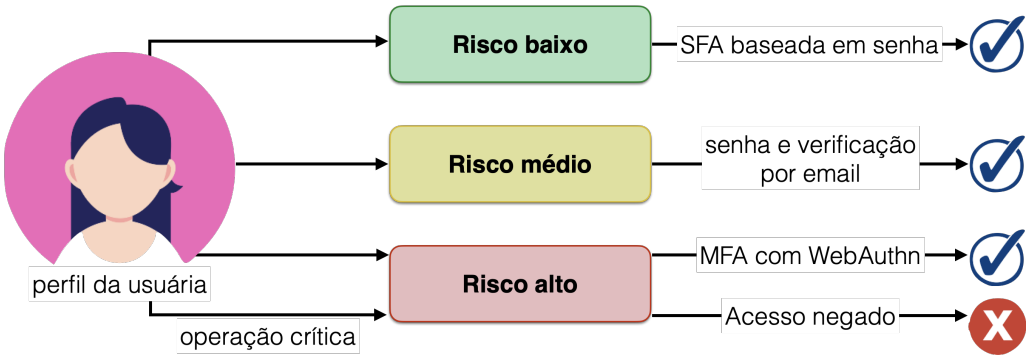


Figura 1.13: *Exemplo de Sistema de Autenticação Baseado em Riscos*

Algumas plataformas de identidade comerciais, como por exemplo a OKTA[20], a Azure Active Directory[21] e OneLogin[22], possibilitam a criação de políticas de acesso contextual que avaliam fatores de risco, como dispositivo, rede, localização, viagem, IP e outros contextos, em cada etapa do processo de autenticação. Em seguida, analisa o nível de risco com as configurações de autenticação apropriadas, como solicitar MFA ou usar autenticação sem senha para um acesso de baixo risco.

O *framework Shibboleth*, que implementa o padrão SAML e o modelo de identidades federadas, permite aos IdPs implementarem novos fluxos de autenticação, além daqueles já existentes por padrão (e.g. senha, certificado X.509, endereço IP). Esse *framework* possui ainda um tipo especial chamado *MFA Flow*, o qual fornece uma maneira programável de combinar diferentes tipos de fluxos de autenticação, bem como orquestrar sequências de execução destes fluxos para criarem cenários adaptativos de autenticação, inclusive baseados em riscos, que levem em consideração o contexto e o comportamento do usuário. Um cenário possível para a implementação de autenticação dinâmica em IdPs *Shibboleth* é utilizar autenticação implícita baseada em contexto (endereço IP, geolocalização e *user agent*) como segundo fator de autenticação. O IdP autentica o usuário com usuário e senha e armazena as informações de endereço IP, *user agent* e geolocalização. Para isso, o provedor de serviço (SP) executa, em segundo plano, um *software* responsável por solicitar periodicamente (p.ex., a cada 60 minutos) a autenticação implícita do usuário. Este SP pode ser, por exemplo, um aplicação de registro de presença de alunos. Caso esse processo de autenticação implícita falhe, o IdP redireciona o usuário para uma autenticação explícita, essa podendo ser feita com OTP, WebAuthn, entre outros.

1.2.3. Cenário internacional de federações acadêmicas

Universidades, instituições de pesquisa e empresas estão gerando grandes quantidades de dados que precisam estar disponíveis por meio de ambientes colaborativos de pesquisa que vão além dos limites de uma única organização (BROEDER et al., 2012) e até mesmo de um único país (ATHERTON et al., 2022). O serviço *eduGAIN* possibilita que pesquisadores possam usar suas credenciais institucionais para acessar inúmeros provedores de serviços

[20]https://www.okta.com/products/adaptive-multi-factor-authentication/

[21]https://docs.microsoft.com/en-us/azure/active-directory/identity-protection/howto-identity-protection-configure-risk-policies

[22]https://www.onelogin.com/learn/what-why-adaptive-authentication

disponíveis na interfederação acadêmica.

O conceito de *e-science* pode ser definido como uma forma de colaboração global em determinadas áreas da ciência e a ciberinfraestrutura que a suporta (TAYLOR, 2001). Nessa infraestrutura baseada na Internet, recursos computacionais e processamento de alto desempenho são compartilhados nas instituições ou em provedores de serviços em nuvem. Como resultado deste cenário, tem-se um grupo sem fronteiras que atua como uma rede de pessoas e instituições conectadas que colaboram com o objetivo de resolver problemas complexos e fazer ciência. Em alguns trabalhos, este grupo é chamado de Organização Virtual (*Virtual Organizations* – VO).

A maioria dos membros das federações acadêmicas são universidades ou centros de pesquisa; no entanto, profissionais autônomos e empresas também podem colaborar com a pesquisa em *e-science* (ATHERTON, C. J. et al., 2018). Profissionais autônomos muita vezes utilizam do login social (e.g. Google, Github, Linkedin) , os quais empregam geralmente o protocolo OpenID Connect e não o SAML, comumente usado nas federações acadêmicas. Diante da diversidade de mecanismos de autenticação e autorização, é comum que pesquisadores precisem gerenciar diversas credenciais de acesso e usá-las em sistemas de controle de acesso de forma desarticulada, gerando assim uma dificuldade para colaborações em *e-science* (BASNEY et al., 2019).

De acordo com Christopher John Atherton et al. (2018), os sistemas de gestão de identidade federada não foram projetados para serem usados em ambientes abertos e dinâmicos como as OVs. Existem problemas a serem resolvidos quando um usuário precisa colaborar em um ambiente no qual mais de uma federação está envolvida. Esta colaboração não deve tratar apenas dos aspectos tecnológicos, mas também das políticas organizacionais.

Um dos principais desafios relacionados ao uso de identidades federadas em OVs são os diversos atributos exigidos pelos diferentes SPs. A decisão de acesso a um serviço de *e-science* em uma OV depende não apenas dos atributos definidos pelo IdP do usuário, mas também dos atributos definidos na própria OV (CHAGAS et al., 2019). Por exemplo, o atributo de associação que identifica um usuário de instituição é membro de uma OV específica e o atributo que identifica o seu papel na OV. A falta de um padrão amplamente suportado para expressar o pertencimento a uma OV nas federações acadêmicas e na eduGAIN estreita o potencial dos sistemas federados para pesquisa colaborativa.

O projeto, financiado pela União Europeia, sobre Autenticação e Autorização para Pesquisas Colaborativas (*Authentication and Authorisation for Research and Collaboration*[23] – AARC), foi lançado em 2015 e finalizado em dezembro de 2019, e teve como objetivo: (i) identificar os requisitos necessários em pesquisa colaborativa internacional, indo além das capacidades de acesso federado atuais; (ii) entregar um modelo de arquitetura (*AARC Blueprint Architecture*), (iii) definir um grupo de diretrizes e políticas para contribuir com a interoperabilidade no contexto de autenticação e autorização; bem como (iv) avaliar os benefícios de uma plataforma abrangente para comunidades de pesquisa, por meio de casos de uso da comunidade e pilotos de integração da infraestrutura.

A GÉANT foi uma das principais colaboradoras do projeto AARC e, com base

[23]https://aarc-project.eu/

na AARC BPA, desenvolveu o serviço eduTEAMS[24], o qual expande a eduGAIN. O eduTEAMS é oferecido como serviço somente para usuários e comunidades de pesquisa na Europa e permite que pesquisadores possam criar e gerenciar times virtuais, utilizando provedores de identidade da eduGAIN e outros provedores de identidades confiáveis. As comunidades de pesquisa podem gerenciar seus usuários, organizá-los em grupos, atribuir papéis a eles e gerenciar de forma centralizada os direitos de acesso aos recursos Serviços *Web* e nativos. eduTEAMS atende usuários vindos da indústria ou cientistas que não tenham acesso a eduGAIN, por meio de um *proxy* que integra provedores de *logins* sociais, provedor ORCID e outros provedores comerciais.

O projeto de código aberto CILogon[25], fundamentado sobre o *framework* Shibboleth[26], o COManage[27] e baseada na arquitetura AARC, consiste em uma plataforma para gestão de identidade e de acesso para permitir a pesquisa colaborativa. A plataforma provê suporte a diferentes tecnologias de autenticação, fluxos para registro de usuários, vinculação de identidade (p.ex. com ORCID) e gerenciamento de grupos (BASNEY et al., 2019).

Unity [28] é um serviço de autenticação que oferece suporte a identidades federadas (SAML, OpenID), juntamente com o gerenciamento de grupos de usuários, atributos e credenciais. Permite a integração via LDAP, OAuth2, SAML e PAM. Também é possível integrá-lo com o *middleware* de computação científica UNICORE. O Unity é um software de código aberto que pode ser instalado e operado pela OV (colaboração de pesquisa), em contraste com o modelo de software como serviço do CILogon e eduTEAMS.

Periodicamente, a REFEDS apoia atividades de grupos de trabalho com o intuito de promover o diálogo abordando problemas e objetivos específicos sobre interfederação acadêmica. Com o objetivo de identificar o valor da federação e definir recomendações para melhorias no futuro, a REFEDS criou o grupo de trabalho Federação 2.0. Este grupo seguiu um processo estruturado para reunir contribuições de uma ampla gama de fontes de informação e perspectivas individuais, a fim de revisar os estados passados e atuais e formular possíveis cenários futuros para a evolução das federações acadêmicas e da interfederação. Esses dados foram analisados e sintetizados no relatório "*Academic Interfederation into the 2030s*" para articular o valor da federação acadêmica, identificar mudanças potenciais que podem aumentar esse valor e recomendar ações que as federações podem tomar para aumentar seu valor ao longo do tempo (ATHERTON et al., 2022).

1.2.4. Empoderamento dos usuários

Na Subseção 1.2.1, foram apresentadas e discutidas diversas questões relativas ao fluxo de autenticação e autorização envolvendo os mecanismos de plataformas *web*, os quais estão em mudanças devido ao fato de o aspecto tecnológico e financeiro sobreporem aos interesses do indivíduo, incluindo direitos básicos como o direito à privacidade e proteção de dados.

[24]https://www.eduteams.org
[25]https://www.cilogon.org/
[26]https://www.shibboleth.net/
[27]https://www.incommon.org/software/comanage/
[28]https://unity-idm.eu/

O próprio conceito de privacidade e proteção de dados pessoais em diversos momentos é tomado como sinônimo de segurança da informação. Porém, embora não haja privacidade e proteção de dados sem segurança da informação, os conceitos de privacidade e proteção de dados são muito mais amplos. Barth, Ionita e Hartel (2022) destacam que contemporaneamente privacidade refere-se fundamentalmente à informação, além de que seu escopo está permanentemente ampliando devido aos avanços sociais e tecnológicos. Entre os avanços sociais estão diversas legislações sobre proteção de dados pessoais, como as leis LGPD (BRASIL, 2018) e RGPD (UNION, 2016).

O ponto comum nestas legislações é o objetivo de empoderar o usuário, destacando direitos como o da auto-determinação informativa, o qual garante ao usuário o controle sobre a emissão e utilização de dados pessoais. Para que esse direito possa ser exercido, o usuário precisa ser informado sobre quais dados serão coletados, por quais motivos, se serão compartilhados e com quem, por quanto tempo serão mantidos, como serão descartados etc. Assim, muitas vezes a porta de entrada ou o primeiro contato com o usuário, para a coleta desses dados, é pela funcionalidade de criação de conta de um sistema de GId.

Os modelos de GId apresentados em Subseção 1.1.1 possuem diversas características que os colocam em maior ou menor grau com foco em tecnologia e pouco em questões humano-políticas. O modelo baseado em silo coloca um fardo grande no usuário na questão de gerenciamento de inúmeras contas, além do fato de possibilitar que seus dados pessoais estejam espalhados por diversas bases, com mecanismos diversos de segurança, aumentando o risco de vazamento de seus dados.

Nos modelos federado e centrado no usuário, o usuário passa a ter um fardo muito menor no gerenciamento de suas credenciais de acesso, além de passar poucas vezes (número de IdPs com o qual pretende interagir) pelo processo de cadastro, onde precisa fornecer seus dados pessoais. Os termos de consentimento, apresentados ao usuário pelo IdP, quando este pretende acessar um provedor de serviço, dão ao usuário o poder para ver quais dados o provedor de serviço está solicitando ao IdP e se concorda em compartilhá-los. O fato do usuário criar uma conta em um IdP e poder utilizá-la em vários provedores de serviços passa uma impressão de portabilidade de dados, mas na verdade o gerenciamento e controle dos dados ainda é atribuição única do IdP.

O modelo descentralizado é o que almeja deixar o usuário em controle desses dados, porém de acordo com o estudo conduzido por Ostern e Cabinakova (2019), a maioria dos trabalhos na literatura discutem sobre a viabilidade técnica destes sistemas de gestão de identidade descentralizada, mas não levam em consideração os requisitos de usuários e não apresentam uma avaliação sobre a usabilidade destes sistemas de forma que possam verificar se os usuários possuem as habilidades necessárias para proteger sua privacidade. Tal tipo de preocupação é essencial, uma vez que neste modelo o usuário é responsável pelo gerenciamento de seus dados.

Embora existam outras tecnologias para a implantação de um sistema de gerenciamento de identidades descentralizadas, a tecnologia *blockchain* é a mais utilizada nas soluções que estão sendo estudadas e desenvolvidas. A tecnologia provê um livro-razão distribuído (*Distributed Legder Technology* – DLT), protegido por provas criptográficas, o qual facilita troca de informações de modo seguro e confiável entre partes que não confiam

entre si, sem a necessidade de uma autoridade central para validar as transações. No estudo Dunphy e Petitcolas (2018), com relação a soluções de GId baseadas em DLT, os autores concluíram que há uma grande suposição que os usuários estão aptos a fazer um uso efetivo e correto de chaves criptográficas, bem como compreendem intuitivamente as implicações em referenciar atributos de identidade em uma DLT.

Segundo Angulo et al. (2011), o conceito de privacidade online não é simples de compreender e muitas vezes é necessário assistir o usuário de uma maneira não intrusiva. Acquisti et al. (2017) corrobora essa questão, destacando que decisões sobre privacidade e segurança são especialmente complexas *online*, sendo um dos motivos o fato de que tecnologia e ameaças evoluem constantemente. Além disso, Acquisti et al. (2017) pontua também que raramente segurança e privacidade são os objetivos principais do usuário, tendo recursos limitados para avaliar todas as opções e consequências. Portanto, de modo geral o usuário está em uma posição assimétrica de recursos e poder e por esse motivo evangelistas e reguladores sobre privacidade e proteção de dados são unânimes ao requerer que as soluções levem isso em consideração em todo o ciclo de vida do sistema.

Resumindo, qualquer que seja o modelo de GId adotado, para que ele seja centrado no usuário, que efetivamente o empodere e que também esteja de acordo com as modernas regulamentações de proteção de dados, não basta que seja disponibilizado um longo texto de política de privacidade com um botão para o usuário clicar e indicar que *leu e concordou* com os termos.

1.2.5. Identidades de Software

Um dos novos desafios na área de autenticação está no provisionamento de identidades para serviços e componentes de software. Assim como os usuários humanos, componentes de software precisam acessar sistemas como bancos de dados ou APIs diversas. Para controlar estes acessos, um dos componentes básicos de segurança na concepção é o princípio do menor privilégio, que define que cada entidade deve ter acesso apenas às informações e aos recursos necessários para o seu propósito (CAVOUKIAN, 2009). Assim, já que atualmente é cada vez mais comum que componentes de software tenham responsabilidades específicas e mínimas (e.g., paradigma de micro-serviços), componentes devem ter identidades únicas.

Outra tendência que reforça a necessidade de identidades específicas para componentes de software é o modelo de confiança zero (ROSE et al., 2020). Este modelo define que todos os recursos de um sistema devem ter identidades e que a segurança da rede não deve ser orientada a perímetros considerados confiáveis, mas sim que cada serviço deve usar autenticação e autorização forte em todas as comunicações. Isto significa que mesmo componentes em uma mesma rede devem se autenticar em todas as comunicações. Além disso, uma arquitetura de confiança zero deve considerar o monitoramento de seus recursos e que o acesso seja determinado com base em políticas dinâmicas.

No entanto, diferentemente dos usuários humanos, os componentes de software podem ter ciclos de vida bastante dinâmicos. Em uma aplicação de grande porte, instâncias de componentes de software podem ser criadas e extintas em escalas de milhares por dia, orquestrados por sistemas como Kubernetes[29]. Consequentemente, não só as identidades

[29]https://kubernetes.io

para componentes de software precisam ser específicas e robustas como as identidades para usuários humanos, mas também precisam ser provisionadas de forma automatizada.

Atualmente, diversas alternativas estão disponíveis para a geração de identidades para componentes de software. Entre elas destacamos o Kubernetes, Google BeyondProd[30], e SPIFFE/SPIRE (FELDMAN et al., 2020). Estas identidades estão tipicamente embutidas em certificados X.509 ou tokens JWT. Os certificados X.509 são especialmente populares pois podem ser usados em diferentes tipos de aplicações e até de forma transparente para aplicações legadas, por exemplo, colocando um componente *proxy* para encapsular conexões entre aplicações legadas ou como terminadores de conexões TLS que usam tais certificados. Os *tokens* JWT, por outro lado, estão mais sujeitos a ataques de reutilização e não são suportados de forma transparente. No entanto ainda são populares quando incorporados já no processo de desenvolvimento da aplicação.

Kubernetes

Kubernetes é, atualmente, a ferramenta mais popular para orquestração de aplicações em ambientes de nuvem. Uma vez que a aplicação é especificada através de um manifesto escrito em YAML, o Kubernetes faz a criação dos recursos computacionais (volumes, contêineres, balanceadores de carga etc.) e monitora estes recursos para recriá-los em caso de falhas.

O Kubernetes tem uma API para geração de certificados de forma automatizada, conhecida como a *Certificate API*. Esta API pode ser usada em combinação com controladores Kubernetes para automatizar a emissão de certificados da seguinte maneira: (1) um usuário ou aplicação gera uma chave localmente e submete um pedido de assinatura de certificado (*Certificate Signing Request* – CSR) para esta API especificando uma entidade que assinaria o certificado; (2) um administrador pode aprovar manualmente este certificado ou pode haver um controlador associado àquela entidade de assinatura que assina o certificado; (3) quando o status do certificado estiver atualizado, o usuário ou aplicação pode resgatar o certificado assinado.

Este processo de geração de certificados é pouco utilizado por aplicações e mais utilizados para geração de certificados aprovados manualmente para acesso à API do Kubernetes ou para geração de certificados aprovados automaticamente que servirão de identidade para outros componentes do próprio Kubernetes (como o Kubelet). Uma alternativa mais utilizada para aplicações é a utilização de contas de serviço (*Service Accounts*). Seguindo esta estratégia, estas contas podem ser geradas pelo desenvolvedor ou operador, atribuídas a uma aplicação durante a escrita do manifesto e, então, se tornam acessíveis de maneira programática pelos componentes de software executando no Kubernetes. Tipicamente, todo componente executando em Kubernetes tem uma identidade de serviço padrão (denominada *default*). As identidades são montadas automaticamente para serem acessíveis pelo componente executando no *pod* e o componente pode usar esta identidade (através de um *token* JWT associado) para acessar funcionalidades da própria API. Para uso com autenticação, um serviço poderia usar a API de validação de tokens do próprio

[30]https://cloud.google.com/docs/security/beyondprod

Kubernetes (a *Token Review API*) para validar tokens apresentados pelos componentes clientes.

Tanto a solução com a *Certificate API* como a solução baseada em *Service Account* têm limitações consideráveis, como não considerar características específicas da aplicação (apenas a associação entre a *Service Account* e o *pod* no manifesto) e depender de lógica inserida na aplicação para resgatar, usar e verificar os *tokens*. Finalmente, elas seguem o modelo centralizado de gestão de identidades, apenas aplicações no mesmo *cluster* reconheceriam as identidades de software.

BeyondProd

BeyondProd é um modelo de segurança para serviços recomendado pela Google. Ele é baseado em princípios de confiança zero e é definido como uma extensão do modelo BeyondCorp (WARD; BEYER, 2014). O BeyondCorp define que a autenticação dos usuários deve ser baseada em um contexto e não apenas em credenciais ou da localização. Este contexto pode considerar características do acesso como o horário e a origem, assim como características do dispositivo que está sendo usado, como configurações de segurança e nível de atualização. Seguindo também a abordagem de confiança zero, BeyondCorp reforça que não deve haver confiança implícita (e.g., comunicações locais na rede) e, portanto, não depende de VPNs.

O BeyondProd trata os micro-serviços da forma que o BeyondCorp trata os usuários: cada serviço precisa ser autenticado com base não só em credenciais estáticas, mas também no contexto onde está sendo executado. Assim, o BeyondProd estende o modelo de segurança das identidades de software providas pelo Kubernetes e detalhadas acima.

A principal forma de implementação do BeyondProd é a utilização de conexões TLS (*Transport Layer Security*[31]), mutualmente autenticadas (ou ATLS, uma variante do TLS proposta pela Google para conexões RPC[32]). Cada componente de software deve receber uma identidade e usar esta identidade para autenticar mutualmente o serviço que está acessando e a si mesmo. A implementação destas conexões mutualmente autenticadas podem ser diretamente na aplicação ou utilizando *proxies* terminadores de TLS que encapsulam a conexão entre dois pontos dentro de uma conexão TLS mutualmente autenticada. Assim, as aplicações da ponta não precisam ter conhecimento das identidades. Um exemplo de *proxy* de código aberto é o Envoy[33]. O Envoy pode ser usado tanto isoladamente para encapsular conexões quanto como parte de malhas de serviço (gerenciada por sistemas como o Istio[34] sobre um *cluster* Kubernetes).

No caso da Google, as identidades de software que serão entregues para o *proxy* TLS são gerenciadas pelo Borg (VERMA et al., 2015), uma ferramenta interna da Google e que deu origem ao Kubernetes. Ao contrário do Kubernetes puro, que entrega identidades puramente com base em um mapeamento estático, o Borg pode usar mecanismos

[31]https://www.rfc-editor.org/rfc/rfc8446.html
[32]https://cloud.google.com/docs/security/encryption-in-transit/application-layer-transport-security
[33]https://www.envoyproxy.io/
[34]https://istio.io

como a Autorização Binária (BOB[35]). O BOB utiliza várias estratégias para garantir a confiabilidade do código, por exemplo, permitindo o condicionando do fornecimento de identidade a características como a cadeia de suprimento para os artefatos de software (usando o padrão SLSA[36]). Estas evidências de integridade devem estar embutidas em políticas de segurança e associadas a imagens assinadas dos contêineres que contém os componentes de software que receberão as identidades. Para serviços em execução na nuvem da Google, o serviço de Autorização Binária está disponível e é compatível com serviços de orquestração de aplicações como o Google Kubernetes Engine ou o Anthos Service Mesh.

SPIFFE/SPIRE

Finalmente, SPIFFE (*Secure Production Identity Framework For Everyone*) é um conjunto de padrões para identificar serviços[37]. Considerando a natureza heterogênea de componentes de software, o objetivo do SPIFFE é prover interoperabilidade para estas identidades, sendo agnóstica quanto às plataformas e tecnologias.

As identidades SPIFFE (SPIFFE ID) são implementadas como URIs (*Uniform Resource Identifiers*). Por exemplo, a identidade *spiffe://example.com/database* poderia ser associada a um servidor de banco de dados que está localizado no domínio administrativo `example.com`. O nome do componente do software (`database`) é definido no registro da identidade e pode ser tanto um nome ou uma hierarquia de nomes amigáveis para humanos (como `database/server` e `database/client`), como sequências opacas de caracteres (como um valor de *hash*).

Além do formato da identidade, outros componentes chave do SPIFFE são as definições dos formatos de identidades verificáveis, os SVIDs (*SPIFFE Verifiable IDs*), e a especificação da API para emitir ou resgatar SVIDs (conhecida como *Workload API*). Por exemplo, no momento desta escrita, dois formatos para identidades verificáveis estão definidos X.509 e JWT.

SPIFFE, assim como sua implementação de referência SPIRE (*SPIFFE Runtime Environment*), são projetos graduados da *Cloud Native Computing Foundation* (CNCF). Projetos graduados são considerados estáveis para uso em produção. O SPIRE fornece APIs que permitem o estabelecimento de confiança entre componentes de software através da atestação das propriedades destes componentes e a emissão de identidades verificáveis, os SVIDs, para os mesmos. Por exemplo, de posse de SVIDs tipo X.509, os componentes podem criar conexões TLS mutualmente autenticadas.

De forma semelhante ao BeyondCorp, ele permite que clientes e servidores legados possam integrar um ambiente de confiança zero através de *proxies* que intermedeiam todas as conexões. No caso do SPIRE (e de outras implementações que usam identidades SPIFFE, como o Istio), as identidades são providas por um processo de atestação que pode ser baseada em propriedades do ambiente ou da própria aplicação. Para o caso do

[35]https://cloud.google.com/docs/security/binary-authorization-for-borg
[36]https://slsa.dev/
[37]https://github.com/spiffe/spiffe

ambiente, por exemplo, uma identidade específica pode ser provida para o componente caso ele seja executado em uma máquina virtual que pertence a um determinado grupo de segurança em um provedor público de nuvem ou em uma máquina com verificação de integridade (ex., via um *Trusted Platform Module* – TPM). Caso ele esteja executando em outro ambiente, o mesmo componente receberia uma identidade diferente, refletindo um nível de confiança diferente.

Além da validação do ambiente, o provisionamento de identidade pode ser condicionado a propriedades da própria aplicação, como o *hash* da imagem do contêiner ou até o *hash* do binário do componente. Finalmente, identidades emitidas por um domínio (ex., `example.com`) podem ser reconhecidas em outro (ex., `example.org`) caso os seus servidores SPIRE sejam federados. Neste caso, os servidores SPIRE trocam informações sobre seus certificados e as aplicações, ao receber suas identidades, também recebem os conjuntos de certificados-raiz dos outros domínios. Assim, o SPIRE pode enquadrar-se tanto na categoria de modelo centralizado de gestão de identidade, quanto no modelo federado (veja Seção 1.1.1).

O SPIRE pode ser usado junto com outras soluções de código aberto. Em especial, integra-se com o orquestrador de serviços Kubernetes e com *proxies* terminadores de TLS como o Envoy e o Ghostunnel[38]. Alternativamente, as identidades podem ser recuperadas e manipuladas diretamente pelos componentes de software, usando SDKs disponíveis para diversas linguagens de programação[39].

1.2.6. Autorização e controle de acesso

Assim como na autenticação, a utilização de uma infraestrutura externa para autorização e controle de acesso, combinado com o modelo ABAC (veja Subseção 1.1.2), traz diversos benefícios (BAILEY; CHADWICK; LEMOS, 2014), como por exemplo, um controle de acesso de baixa granularidade e regras que se aplicam a diversos sistemas dentro de uma instituição. Tal modelo vem sendo utilizado em diversos domínios de aplicação e vem sendo adotado pelos grandes provedores de serviços de computação em nuvem.[40] Na sequência apresentamos como tais mecanismos podem ser utilizados para proteger aplicações *web* e de Internet das coisas (IoT). Discutiremos também uma preocupação envolvendo a mitigação de ameaças internas.

Utilização em aplicações *Web*

Atualmente a maioria das aplicações *web* são baseadas no estilo de arquitetura REST (FIELDING, 2000), onde uma aplicação cliente interage com um servidor por meio de uma API. Existem diversos *frameworks*, nas mais variadas linguagens de programação, que oferecem suporte para este estilo arquitetural, por exemplo, Java Spring, Python Django, JavaScript Node.JS etc. Estes *frameworks* simplificam o desenvolvimento de aplicações Web e implementam algum mecanismo de controle de acesso, normalmente baseado no

[38]https://github.com/ghostunnel/ghostunnel

[39]https://spiffe.io/docs/latest/deploying/libraries/

[40]Por exemplo, o serviço *Identity and Access Management* da Amazon e o *Azure Active Directory* da Microsoft.

modelo RBAC com algumas extensões específicas de sua respectiva tecnologia.

Frameworks como Spring, Django e Node.JS permitem ao programador inserir uma anotação no código fonte para estabelecer as permissões de acesso para as operações oferecidas pela API. Node.JS, em conjunto com o *framework* Express, oferecem o conceito de *middleware* onde regras de controle de acesso são encapsuladas em módulos JavaScript que são então adicionados como interceptadores para as diversas funções oferecidas através de uma API REST. Já o *framework* Django explora o conceito de decoradores onde as permissões são adicionadas às definições das funções junto ao código fonte, com suporte adicional a instruções *if* arbitrárias definidas pelo programador.

Tais abordagens foram amplamente estudadas na literatura e são reconhecidas hoje por apresentarem diversos problemas, tais como, a possibilidade de erros ao definir políticas mais complexas e a dificuldade em se alterar uma política de acesso. De fato, os benefícios no uso de uma infraestrutura de autorização externa à aplicação são bem conhecidos (HU et al., 2014), sendo, por exemplo, uma das recomendações da Google em seu guia para o design e desenvolvimento de aplicações mais robustas (ADKINS et al., 2020) e adotado atualmente pelos grandes provedores de nuvem.

As soluções disponíveis normalmente envolvem a utilização de um componente PEP (veja Subseção 1.1.2) para interceptar as chamadas à API. Por exemplo, Silva, Medeiros e Sampaio (2019) desenvolveram PEP4Django, um *middleware* para o *framework* Django que atua como PEP de acordo com o padrão XACML. A implementação com integração ao servidor de autorização *WSO2 Identity service* está disponível como código-aberto no GitHub[41].

Outras soluções exploram o padrão de *API gateway* para onde as requisições a uma determinada API são encaminhadas. Esta é a abordagem normalmente oferecida pelos provedores de serviço de nuvem. Por exemplo, o serviço IAM da Amazon oferece uma implementação do modelo ABAC que pode ser utilizado para proteger acessos a outros serviços de nuvem ou a APIs REST através do serviço API Gateway que atua como um PEP.

Lidando com ameaças internas

Embora os mecanismos de controle de acesso sejam bastante efetivos para proteção de recursos, eles não são suficientes para detectar ameaças internas. Uma ameaça é considerada interna quando usuários autorizados a acessar um sistema abusam de suas permissões para comprometer o sigilo, a integridade ou a disponibilidade dos recursos de uma organização (CAPPELLI; MOORE; TRZECIAK, 2012). Estes usuários podem causar danos de maneira intencional ou devido a erros por falta de treinamento ou negligência.

Em geral os mecanismos de controle de acesso não são capazes de detectar usuários que abusam de suas permissões e, mesmo quando ferramentas externas são empregadas para detectar tais situações, os mecanismos de controle de acesso não são capazes de mitigá-las (SALEM; HERSHKOP; STOLFO, 2008; HOMOLIAK et al., 2019). Esta é uma situação que exige mecanismos de controle de acesso dinâmicos, no sentido que políticas

[41] https://github.com/welkson/PEP4Django

de acesso possam ser dinamicamente modificadas em respostas a ameaças internas. É importante não confundir a modificação dinâmica de políticas de acesso com a tomada de decisão de uma política baseado em informações em tempo de execução.

Alguns trabalhos têm focado neste problema nos últimos anos. Por exemplo, Bailey, Chadwick e Lemos (2014) apresentaram um *framework* para suportar a reconfiguração dinâmica de políticas de controle de acesso. Já Silva, Diniz et al. (2018) aplicam esses conceitos para detectar abusos por usuários autorizados em plataformas de nuvem Openstack, realizando uma análise do impacto de se efetivar determinadas mudanças como auxílio ao processo de tomada de decisão para responder a uma ameaça interna.

Outra abordagem para tal foi apresentada por Silva, Silva et al. (2017). O SARBAC explora *logs* de processos de negócio implementados por um sistema para construir um modelo que capture o comportamento de seus usuários. Com isso, modelos probabilísticos são utilizados para se comparar o comportamento de um usuário com o restante dos usuários de um sistema. Através do cálculo do intervalo de confiança dos modelos sendo comparados é possível afirmar que o comportamento de um usuário é estatisticamente diferente dos outros usuários, o que indica uma possível anomalia que é respondida de acordo com uma política de adaptação como, por exemplo, remover o papel do usuário em questão ou remover uma determinada permissão de um papel.

1.3. Considerações finais

As questões apresentadas neste capítulo se referem à busca de soluções para um velho problema: provar sua identidade *online*, garantindo acesso aos recursos ou serviços de maneira segura e somente ao que se tem autorização, através do estabelecimento de uma relação de confiança entre as partes. Este problema é profundamente enraizado, pois conforme as palavras de Kim Cameron, Chefe de Arquitetura de Identidade da Microsoft de 2004 à 2019, a Internet foi construída sem uma camada de identidade (CAMERON, 2005) – não há como saber com quem ou o que a conexão está sendo estabelecida. Saber apenas o endereço IP do dispositivo (o qual pode ser forjado) não informa nada em termos de qual entidade está acessando o serviço.

Questões como se a entidade tem os atributos mínimos como idade, por exemplo, precisam ser tratados de outra maneira. E essa maneira historicamente tem sido a construção de diversos modelos de GId, combinando tecnologias diversas de autenticação e autorização. A constante transformação digital da sociedade mantém esse antigo problema ainda não resolvido em evidência e busca por novas soluções, como a proposta da Identidade Autossoberana, a qual busca colocar o ser humano no centro do controle de sua identidade digital, representando um modelo de GId totalmente descentralizado. Entretanto, ainda é difícil dizer se o usuário está preparado para (e/ou disposto a) assumir a grande responsabilidade e os custos de gerir seus dados pessoais de forma totalmente autônoma e soberana.

As ações recentes realizadas pela FIDO Alliance e W3C, que permitem o uso de telefones inteligentes como autenticadores externos e a proposta de sincronismo de chaves por múltiplos dispositivos FIDO, também chamado de *passkey*, foram feitas com o intuito de acabar com a dependência do par usuário e senha pelos usuários. Porém, como observado no estudo conduzido por Owens et al. (2021), a ampla adoção WebAuthn ainda

dependerá da percepção relativa dos usuários sobre a usabilidade *versus* segurança, quando comparado com as tradicionais senhas. Usuários estão habituados com senhas, mesmo que façam um mau uso destas. Assim tem-se uma resistência ou dificuldade implícita para aprender uma nova forma de autenticar-se nos serviços, uma vez que o benefício do WebAuthn não é algo que possa ser facilmente observado por usuários leigos.

Cabarcos, Krupitzer e Becker (2019) e Ryu et al. (2021) apresentam revisões sistemáticas da literatura sobre autenticação adaptativa. No primeiro estudo, os autores afirmam que os sistemas analisados são difíceis de estender ou reutilizar (por exemplo, incluir novos autenticadores, estratégias de adaptação ou contextos) e concluem que estes são desafios significativos para o uso prático de soluções de autenticação dinâmica. Na segunda revisão, que aborda especificamente autenticação biométrica multimodal contínua, os autores concluem que muitos sistemas não avaliam adequadamente a usabilidade (aceitação e satisfação do usuários) e a viabilidade das soluções (avaliação extensiva com dados reais sem restrições), requisitos-chave para o sucesso e implantação real da solução.

Com relação à identidades de software, a combinação de padrões e boas práticas permitem ter identidades mais fortemente vinculadas a propriedades do componente de software e a limitar o impacto de vazamentos de identidades. Consequentemente, torna-se mais fácil ter ambientes com máquinas confiáveis e que executam código de origem conhecida. Tais garantias são especialmente úteis para componentes que acessam dados confidenciais. Além disso, a utilização de ferramentas que aplicam conceitos como atestação e que emitem identidades de forma automatizadas permite que esta confiança não dependa de mudanças no código ou de atividades repetitivas de operadores. As identidades podem então ser a cola que permitirá que mecanismos de autenticação e autorização isolem serviços do ambiente e, ao mesmo tempo, integrem componentes de uma mesma aplicação de forma segura e transparente.

Referências

ACQUISTI, Alessandro et al. Nudges for privacy and security: Understanding and assisting users' choices online. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 50, n. 3, p. 1–41, 2017.

ADKINS, Heather et al. **Building secure and reliable systems: best practices for designing, implementing, and maintaining systems**. First edition. Beijing [China] ; Boston [MA]: O'Reilly, 2020. 519 p. ISBN 9781492083122. Disponível em: <https://sre.google/books/building-secure-reliable-systems/>.

ALLAN, Ant. **Hype Cycle for Identity and Access Management Technologies**. Gartner, jul. 2020. Disponível em: <https://www.gartner.com/en/documents/3987655/hype-cycle-for-identity-and-access-management-technologi>. Acesso em: 5 mai. 2021.

ALLEN, Christopher (Ed.). **The Path to Self-Sovereign Identity**. Abr. 2016. Disponível em: <http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html>. Acesso em: 14 mai. 2021.

ANGULO, Julio et al. Towards Usable Privacy Policy Display & Management-The PrimeLife Approach. In: HAISA. 2011. p. 108–118.

ANSI. **Role Based Access Control**. 2004. ANSI/INCITS 359-2004.

ANSI. **Role Based Access Control - Policy-Enhanced**. 2012. ANSI/INCITS 494-2012.

APPLE. **Apple, Google, and Microsoft commit to expanded support for FIDO standard to accelerate availability of passwordless sign-ins**. Mai. 2022. Disponível em: <https://www.apple.com/newsroom/2022/05/apple-google-and-microsoft-commit-to-expanded-support-for-fido-standard/>. Acesso em: 3 ago. 2022.

ATHERTON, Christopher John et al. **Federated Identity Management for Research Collaborations**. Jun. 2018. DOI: 10.5281/zenodo.1307551. Disponível em: <https://doi.org/10.5281/zenodo.1307551>.

ATHERTON et al. **Academic Interfederation into the 2030s**. Zenodo, mai. 2022. DOI: 10.5281/zenodo.6584587. Disponível em: <https://doi.org/10.5281/zenodo.6584587>.

BAILEY, Christopher; CHADWICK, David W.; LEMOS, Rogério de. Self-adaptive federated authorization infrastructures. **Journal of Computer and System Sciences**, v. 80, n. 5, p. 935–952, 2014. ISSN 0022-0000. DOI: 10.1016/j.jcss.2014.02.003. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022000014000154>.

BARTH, A. **HTTP State Management Mechanism**. Abr. 2011. Disponível em: <http://www.rfc-editor.org/rfc/rfc6265.txt>.

BARTH, Susanne; IONITA, Dan; HARTEL, Pieter. Understanding Online Privacy—A Systematic Review of Privacy Visualizations and Privacy by Design Guidelines. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 3, fev. 2022. ISSN 0360-0300. DOI: 10.1145/3502288. Disponível em: <https://doi.org/10.1145/3502288>.

BASNEY, Jim et al. CILogon: Enabling federated identity and access management for scientific collaborations. English (US). **Proceedings of Science**, Sissa Medialab Srl, v. 351, 2019. ISSN 1824-8039. DOI: 10.22323/1.351.0031.

BRADLEY, John et al. (Ed.). **Client to Authenticator Protocol (CTAP)**. Jun. 2021.

BRAMHALL, Susan et al. **CAS Protocol 3.0 Specification**. Dez. 2017. Disponível em: <https://apereo.github.io/cas/6.5.x/protocol/CAS-Protocol-Specification.html>. Acesso em: 3 ago. 2022.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Diário Oficial da República Federativa do Brasil**, Brasília, DF, 14 ago. 2018. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em: 28 mai. 2021.

BROEDER, Daan et al. **Federated Identity Management for Research Collaborations**. Geneva, abr. 2012. CERN-OPEN-2012-006. Disponível em: <https://cds.cern.ch/record/1442597>.

CABARCOS, Patricia; KRUPITZER, Christian; BECKER, Christian. A Survey on Adaptive Authentication. **ACM Computing Surveys**, v. 52, p. 1–30, set. 2019. DOI: 10.1145/3336117.

CAMERON, Kim. **The Laws of Identity**. 2005. Disponível em: <https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf>. Acesso em: 28 fev. 2022.

CAPPELLI, Dawn; MOORE, Andrew; TRZECIAK, Randall. **The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes (theft, sabotage, fraud)**. Upper Saddle River, NJ: Addison-Wesley, 2012. 389 p. (The Sei series in software engineering). OCLC: ocn752067994. ISBN 9780321812575.

CAVOUKIAN, Ann. **Privacy by Design: The 7 Foundational Principles**. Ago. 2009. Information and Privacy Commissioner of Ontario. Disponível em: <https://www.ipc.on.ca/wp-content/uploads/Resources/7foundationalprinciples.pdf>. Acesso em: 3 ago. 2022.

CHAGAS, M. et al. SM4VO: A Security Management Mechanism for Virtual Organizations. **2019 9th Latin-American Symposium on Dependable Computing (LADC)**, p. 1–10, 2019. DOI: 10.1109/LADC48089.2019.8995732.

DASGUPTA, Dipankar; ROY, Arunava; NAG, Abhijit. Multi-Factor Authentication. In: ADVANCES in User Authentication. Cham: Springer International Publishing, 2017. p. 185–233. ISBN 978-3-319-58808-7. DOI: 10.1007/978-3-319-58808-7_5. Disponível em: <https://doi.org/10.1007/978-3-319-58808-7_5>.

DUNPHY, Paul; PETITCOLAS, Fabien AP. A first look at identity management schemes on the blockchain. **IEEE security & privacy**, IEEE, v. 16, n. 4, p. 20–29, 2018.

FALCÃO, Eduardo et al. Autenticando aplicações nativas da nuvem com identidades SPIFFE. In: MINICURSOS XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. 2021. p. 100–144.

FELDMAN, Daniel et al. **Solving the Bottom Turtle: a SPIFFE way to establish trust in your infrastructure via universal identity**. 2020. ISBN 978-0-578-77737-5. Disponível em: <thebottomturtle.io>.

FERRAIOLO, David F.; KUHN, D. Richard. Future directions in role-based access control. In: THE FIRST ACM WORKSHOP. **Proceedings of the first ACM Workshop on Role-based access control - RBAC '95**. Gaithersburg, Maryland, United States: ACM Press, 1996. 8–es. ISBN 9780897917599. DOI: 10.1145/270152.270165. Disponível em: <http://portal.acm.org/citation.cfm?doid=270152.270165>. Acesso em: 12 ago. 2022.

FIDO. **How FIDO Addresses a Full Range of Use Cases**. Mar. 2022. Disponível em: <https://media.fidoalliance.org/wp-content/uploads/2022/03/How-FIDO-Addresses-a-Full-Range-of-Use-Cases-March24.pdf>. Acesso em: 3 ago. 2022.

FIELDING, R.; NOTTINGHAM, M.; RESCHKE, J. **HTTP Semantics**. Jun. 2022. Disponível em: <http://www.rfc-editor.org/rfc/rfc9110.txt>.

FIELDING, Roy Thomas. **REST: Architectural Styles and the Design of Network-based Software Architectures**. 2000. Doctoral dissertation – University of California, Irvine. Disponível em: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

FORUM, World Economic. **A Blueprint for Digital Identity**. 2016. Disponível em: <https://www3.weforum.org/docs/WEF_A_Blueprint_for_Digital_Identity.pdf>.

GERADIN, Damien; KATSIFIS, Dimitrios; KARANIKIOTI, Theano. **Google as a de facto privacy regulator: Analyzing Chrome's removal of third-party cookies from an antitrust perspective**. TILEC Discussion Paper No. DP2020-034, 2020.

GOODIN, Dan. **Lapsus$ and SolarWinds hackers both use the same old trick to bypass MFA**. Disponível em: <https://arstechnica.com/information-technology/2022/03/lapsus-and-solar-winds-hackers-both-use-the-same-old-trick-to-bypass-mfa/>. Acesso em: 3 ago. 2022.

GRASSI, Paul; GARCIA, Michael; FENTON, James. **NIST Special Publication 800-63-3 Digital Identity Guidelines**. 2020. https://doi.org/10.6028/NIST.SP.800-63-3.

GRIMES, Roger. **12+ ways to hack multi-factor authentication**. KnowBe4, 2019. Disponível em: <https://info.knowbe4.com/12-way-to-hack-two-factor-authentication>. Acesso em: 3 ago. 2022.

GROUP, World Bank. **Principles on identification for sustainable development: toward the digital age - First Edition (English)**. 2018.

GRÜNER, A. et al. A Comparative Analysis of Trust Requirements in Decentralized Identity Management. **Advances in Intelligent Systems and Computing**, v. 926, p. 200–213, 2020. ISBN: 9783030150310 Publisher: Springer Verlag. ISSN 21945357. DOI: 10.1007/978-3-030-15032-7_18.

GUIDORIZZI, Richard P. Security: Active Authentication. **IT Professional**, v. 15, n. 4, p. 4–7, 2013. DOI: 10.1109/MITP.2013.73.

HARDT, D. **The OAuth 2.0 Authorization Framework**. Out. 2012. http://www.rfc-editor.org/rfc/rfc6749.txt. Disponível em: <http://www.rfc-editor.org/rfc/rfc6749.txt>. Acesso em: 3 ago. 2022.

HAYASHI, Eiji et al. CASA: Context-Aware Scalable Authentication. In: PROCEEDINGS of the Ninth Symposium on Usable Privacy and Security. Newcastle, United Kingdom: Association for Computing Machinery, 2013. (SOUPS '13). ISBN 9781450323192. DOI: 10.1145/2501604.2501607. Disponível em: <https://doi.org/10.1145/2501604.2501607>.

HODGES, Jeff et al. (Ed.). **Web Authentication: An API for accessing Public Key Credentials Level 2**. Abr. 2021.

HOMOLIAK, Ivan et al. Insight Into Insiders and IT: A Survey of Insider Threat Taxonomies, Analysis, Modeling, and Countermeasures. **ACM Computing Surveys**, v. 52, n. 2, 30:1–30:40, 2 abr. 2019. ISSN 0360-0300. DOI: 10.1145/3303771. Disponível em: <https://doi.org/10.1145/3303771>. Acesso em: 9 jun. 2022.

HU, Vincent C. et al. **Guide to Attribute Based Access Control (ABAC) Definition and Considerations**. Jan. 2014. DOI: 10.6028/NIST.SP.800-162. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-162.pdf>. Acesso em: 12 ago. 2022.

IBM. **Cost of a Data Breach Report 2022**. 2022. Disponível em: <https://www.ibm.com/security/data-breach>.

IEEE Standard for a Software Quality Metrics Methodology. **IEEE Std 1061-1998**, 1998. DOI: 10.1109/IEEESTD.1998.243394.

ITU. **NGN identity management framework**. International Telecommunication Union (ITU), 2009. Recommendation Y.2720. Disponível em: <https://www.itu.int/rec/T-REC-Y.2720-200901-I>.

JAKOBSSON, Markus et al. Implicit authentication for mobile devices. In: USENIX ASSOCIATION. PROCEEDINGS of the 4th USENIX conference on Hot topics in security. 2009. v. 1, p. 25–27.

JONES, M.; BRADLEY, J.; SAKIMURA, N. (Ed.). **Introduction to JSON Web Tokens**. Disponível em: <https://jwt.io/introduction/>. Acesso em: 2 jun. 2020.

JØSANG, Audun; ZOMAI, Muhammed Al; SURIADI, Suriadi. Usability and privacy in identity management architectures. In: AUSTRALIAN COMPUTER SOCIETY, INC. PROCEEDINGS of the fifth Australasian symposium on ACSW frontiers-Volume 68. 2007. p. 143–152. Disponível em: <http://dl.acm.org/citation.cfm?id=1274548>.

LAWSON, Bruce et al. **HTML 5.3**. Jan. 2021. Disponível em: <https://www.w3.org/TR/html53/>. Acesso em: 11 ago. 2022.

LEACH, Paul J.; MEALLING, Michael; SALZ, Rich. **A Universally Unique IDentifier (UUID) URN Namespace**. Jul. 2005. Disponível em: <https://tools.ietf.org/html/rfc4122>.

LÓPEZ, M Allende. **Self-sovereign identity: The future of identity: Self-sovereignity, digital wallets, and blockchain**. v. 10. 2020. p. 0002635. Disponível em: <https://publications.iadb.org/publications/english/document/Self-Sovereign-Identity-The-Future-of-Identity-Self-Sovereignity-Digital-Wallets-and-Blockchain.pdf>. Acesso em: 6 ago. 2022.

M'RAIHI, D. et al. **TOTP: Time-Based One-Time Password Algorithm**. Mai. 2011. Disponível em: <http://www.rfc-editor.org/rfc/rfc6238.txt>.

MACHANI, Salah et al. (Ed.). **FIDO UAF Architectural Overview**. Out. 2020.

MAQBALI, Fatma Al; MITCHELL, Chris J. **Web password recovery — a necessary evil?** arXiv, 2018. DOI: 10.48550/ARXIV.1801.06730. Disponível em: <https://arxiv.org/abs/1801.06730>.

MELLO, Emerson Ribeiro de; CHAVES, Shirlei Aparecida. O uso do endereço de email pelos sites mais acessados pelo público brasileiro e os possíveis impactos na privacidade de seus usuários. In: XI Computer on the Beach. 2020.

MEREWOOD, Rowan. **Cookies SameSite explicados**. Mai. 2020. Disponível em: <https://web.dev/samesite-cookies-explained/>. Acesso em: 3 ago. 2022.

MICROSOFT. **From cookie theft to BEC: Attackers use AiTM phishing sites as entry point to further financial fraud**. Jul. 2022. Disponível em: <https://www.microsoft.com/security/blog/2022/07/12/from-cookie-theft-to-bec-attackers-use-aitm-phishing-sites-as-entry-point-to-further-financial-fraud>. Acesso em: 3 ago. 2022.

NAKAMURA, Emili Tissato et al. Identidade Digital Descentralizada: conceitos, aplicações, iniciativas, plataforma de desenvolvimento e implementação de caso de uso. In: MINICURSO - SBSeg 2019 - Petrópolis - RJ. 2019. Disponível em: <https://sol.sbc.org.br/livros/index.php/sbc/catalog/view/85/372/636-1>.

NIST. **Digital Identity Guidelines**. National Institute of Standards e Technology, jun. 2017. NIST Special Publication 800-63-3. DOI: https://doi.org/10.6028/NIST.SP.800-63-3.

_____. **Digital Identity Guidelines: Authentication and Lifecycle Management**. National Institute of Standards e Technology, jun. 2017. NIST Special Publication 800-63B. DOI: https://doi.org/10.6028/NIST.SP.800-63b.

NOTTINGHAM, M. **URI Design and Ownership**. Jun. 2020. Disponível em: <https://www.rfc-editor.org/rfc/rfc8820>.

OPEN, OASIS. **OASIS Security Services (SAML) TC**. Disponível em: <https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security>. Acesso em: 8 ago. 2022.

OPENID. **Specifiations**. Disponível em: <https://openid.net/developers/specs/>. Acesso em: 8 ago. 2022.

OSTERN, Nadine; CABINAKOVA, Johana. Pre-prototype testing: empirical insights on the expected usefulness of decentralized identity management systems. In: PROCEEDINGS of the 52nd Hawaii International Conference on System Sciences. 2019.

OWENS, Kentrell et al. User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators. In: SEVENTEENTH Symposium on Usable Privacy and Security (SOUPS 2021). USENIX Association, ago. 2021. p. 57–76. ISBN 978-1-939133-25-0. Disponível em: <https://www.usenix.org/conference/soups2021/presentation/owens>. Acesso em: 9 ago. 2022.

PHILLIPS, Tricia. **Hype Cycle for Identity and Access Management Technologies**. Gartner, jul. 2021. Disponível em: <https://www.gartner.com/en/documents/4004062>. Acesso em: 3 ago. 2022.

PREUKSCHAT, Alex; REED, Drummond. **Self-sovereign identity**. Manning Publications, 2021.

REESE, Ken et al. A Usability Study of Five Two-Factor Authentication Methods. In: FIFTEENTH Symposium on Usable Privacy and Security (SOUPS 2019). Santa Clara, CA: USENIX Association, ago. 2019. p. 357–370. ISBN 978-1-939133-05-2. Disponível em: <https://www.usenix.org/conference/soups2019/presentation/reese>.

RISSANEN, Erik (Ed.). **eXtensible Access Control Markup Language (XACML) Version 3.0 Plus Errata 01**. Jul. 2017. OASIS Standard incorporating Approved Errata. Disponível em: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html>.

ROSE, Scott et al. **Zero Trust Architecture**. Gaithersburg, MD, 2020. DOI: 10.6028/NIST.SP.800-207. Disponível em: <https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420>.

RYU, Riseul et al. Continuous Multimodal Biometric Authentication Schemes: A Systematic Review. **IEEE Access**, v. 9, p. 34541–34557, 2021. DOI: 10.1109/ACCESS.2021.3061589.

SALEM, Malek Ben; HERSHKOP, Shlomo; STOLFO, Salvatore J. A Survey of Insider Attack Detection Research. In: STOLFO, Salvatore J. et al. (Ed.). **Insider Attack and Cyber Security**. Boston, MA: Springer US, 2008. v. 39. p. 69–90. DOI: 10.1007/978-0-387-77322-3_5. Disponível em: <http://link.springer.com/10.1007/978-0-387-77322-3_5>. Acesso em: 12 ago. 2022.

SCHAAR, Peter. Privacy by Design. **Identity in the Information Society**, v. 3, n. 2, p. 267–274, 2010. DOI: 10.1007/s12394-010-0055-x.

SEAMLESSACCESS. **SeamlessAccess enables true Single Sign-On**. Disponível em: <https://seamlessaccess.org>. Acesso em: 5 mai. 2021.

SERASA. **Pesquisa Global de Identidade e Fraude 2021**. 2021. Disponível em: <https://www.serasaexperian.com.br/images-cms/wp-content/uploads/2021/06/Pesquisa-Global-de-Identidade-e-Fraude-2021.pdf>.

SILVA, Carlos Eduardo da; DINIZ, Thomás et al. Self-adaptive authorisation in OpenStack cloud platform. **Journal of Internet Services and Applications**, v. 9, n. 1, p. 19, 16 set. 2018. ISSN 1869-0238. DOI: 10.1186/s13174-018-0090-7. Disponível em: <https://doi.org/10.1186/s13174-018-0090-7>.

SILVA, Carlos Eduardo da; MEDEIROS, Welkson Renny de; SAMPAIO, Silvio Costa. PEP4Django - a policy enforcement point for python web applications. In: IX workshop de gestão de identidades digitais (WGID). São Paulo, SP, Brazil, 2019.

SILVA, Carlos Eduardo da; SILVA, José Diego Saraiva da et al. Self-adaptive role-based access control for business processes. In: PROCEEDINGS of the 12th international symposium on software engineering for adaptive and self-managing systems (SEAMS2017). Buenos Aires, Argentina: IEEE Press, 2017. (SEAMS 2017), p. 193–203. ISBN 978-1-5386-1550-8. DOI: 10.1109/SEAMS.2017.13. Disponível em: <https://dl.acm.org/citation.cfm?id=3105537>.

SRINIVAS, S. et al. (Ed.). **Universal 2nd Factor (U2F) Overview**. Abr. 2017. Disponível em: <https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.pdf>.

TAYLOR, John. News from the e-Science Programme, first phase. **Social Science Information**, RCUK website, v. 47, n. 2, p. 131–157, 2001.

UNION, European. Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho. **Jornal Oficial da União Europeia**, European Union, 4 mai. 2016. Disponível em: <https://eur-lex.europa.eu/legal-content/PT/TXT/?uri=uriserv%5C%3AOJ.L_.2016.119.01.0001.01.POR>.

UTZ, Christine et al. (Un)Informed Consent: Studying GDPR Consent Notices in the Field. In: PROCEEDINGS of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, United Kingdom: Association for Computing Machinery, 2019. (CCS '19), p. 973–990. ISBN 9781450367479. DOI: 10.1145/3319535.3354212.

VERMA, Abhishek et al. Large-scale cluster management at Google with Borg. In: PROCEEDINGS of the Tenth European Conference on Computer Systems. 2015. p. 1–17.

W3C. **A JSON-based Serialization for Linked Data**. Jul. 2020. Disponível em: <https://www.w3.org/TR/json-ld/>. Acesso em: 23 jul. 2022.

_____. **Decentralized Identifiers (DIDs) v1.0**. 2022. Disponível em: <https://www.w3.org/TR/did-core/>. Acesso em: 21 jul. 2022.

_____. **Verifiable Credentials Data Model v1.1**. 2022. Disponível em: <https://www.w3.org/TR/vc-data-model/>. Acesso em: 23 jul. 2022.

WANGHAM, Michelle S; DOMENECH, Marlon C; MELLO, Emerson R de. Infraestruturas de Autenticação e de Autorização para Internet das Coisas. In: MINICURSOS do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg 2013. 2013.

WANGHAM, Michelle Silva; MELLO, Emerson Ribeiro de et al. Gerenciamento de identidades federadas. In: MINICURSOS X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. 2010. p. 1–52.

WARD, Rory; BEYER, Betsy. BeyondCorp: A new approach to enterprise security, 2014. https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43231.pdf.

WIEFLING, Stephan; DÜRMUTH, Markus; LO IACONO, Luigi. More Than Just Good Passwords? A Study on Usability and Security Perceptions of Risk-Based Authentication. In: ANNUAL Computer Security Applications Conference. Austin, USA: Association for Computing Machinery, 2020. (ACSAC '20), p. 203–218. ISBN 9781450388580. DOI: 10.1145/3427228.3427243. Disponível em: <https://doi.org/10.1145/3427228.3427243>.

WILANDER, John. **Intelligent Tracking Prevention 2.2**. Abr. 2019. Disponível em: <https://webkit.org/blog/8828/intelligent-tracking-prevention-2-2/>. Acesso em: 3 ago. 2022.

WU, Cong et al. ICAuth: Implicit and Continuous Authentication When the Screen Is Awake. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). 2019. p. 1–6. DOI: 10.1109/ICC.2019.8761435.

YADRON, Danny. **Man Behind the First Computer Password: It's Become a Nightmare**. Edição: The Wall Street Journal. Mai. 2014. Disponível em: <https://www.wsj.com/articles/BL-DGB-35227>. Acesso em: 22 jul. 2022.

