**Capítulo 20: Identidade Descentralizada**

Neste capítulo, abordamos a identidade digital descentralizada. Especialmente com o advento do blockchain, a identidade digital ganhou destaque, já que o blockchain é visto como um facilitador para transformar o atual paradigma de identidade digital, que sofre com vários problemas.

Ao longo do caminho, abordaremos os seguintes tópicos:

* Identidade
* Identidade digital
* Identidade no Ethereum
* Identidade no mundo do Web3, DeFi e Metaverso
* Projetos de blockchain específicos de SSI
* Desafios

Então, vamos mergulhar no assunto.

**Identidade**

Identidade refere-se às características distintivas, atributos de personalidade, atributos físicos e expressões que definem um indivíduo, entidade ou grupo (alguém ou algo). É o conjunto único de características que tornam uma entidade reconhecível e distinta.

A identidade pode englobar diferentes aspectos, como identidade pessoal, identidade social, identidade cultural e identidade digital. É um aspecto importante de nossas vidas cotidianas e desempenha um papel fundamental na forma como indivíduos ou entidades interagem com os outros ao seu redor.

Um aspecto importante da identidade são as credenciais. Credenciais podem ser definidas como uma atestação por uma autoridade relevante de um atributo associado a uma entidade. O atributo pode ser pessoal, como idade, nome ou gênero, ou também pode ser algo que você conquistou, como uma qualificação. Credenciais emitidas para indivíduos incluem passaporte, carteira de motorista, licença para matar, diploma/grau acadêmico, entre outros. Uma organização pode receber credenciais como autorização para operar em determinada jurisdição, registros de empresa, licenças de construção e muitos outros. Tradicionalmente, todas essas credenciais são baseadas em papel.

**<p style="page-break-after: always;"></p>**

Por exemplo, uma carteira de motorista é um cartão (papel) emitido para você pela autoridade competente após aprovação no exame de direção. Você a apresenta como prova de sua habilidade para dirigir a um policial quando necessário. Tal modelo é chamado de modelo de credencial baseado em papel, que é fundamentado em documentos emitidos por uma autoridade competente. O indivíduo ao qual o documento em papel é emitido é o **titular**, a entidade que emite o documento é o **emissor**, e o **verificador** é uma entidade que verifica essa credencial para assegurar sua legitimidade e permitir acesso a alguns recursos. Por exemplo, para abrir uma conta bancária, posso precisar apresentar minha carteira de motorista como prova de endereço e identidade. Essas credenciais em papel contêm algumas **declarações**, por exemplo, suas qualificações. Essas credenciais também devem provar quem emitiu a credencial, quem é o titular da credencial e, mais importante, que o que está sendo alegado é correto e preciso, e que as declarações não foram alteradas de nenhuma forma. Em resumo, a **declaração deve ser verificável**. No entanto, no mundo real, a falsificação de documentos é bastante comum. Como esse modelo é baseado em confiança presumida entre titulares, emissores e verificadores, ele apresenta alguns desafios.

Quando uma identidade é representada eletronicamente, ela pode ser definida como uma identidade digital. Discutiremos isso a seguir.

**Identidade digital**

Identidade digital pode ser definida como uma representação eletrônica de uma entidade da vida real, como um ser humano, máquina, dispositivo ou organização. Uma identidade digital pode ser representada por um número de identificação de cidadão, ID de funcionário, nome de host e de muitas outras formas.

Em outras palavras, uma identidade digital é uma informação que identifica um indivíduo ou entidade no mundo digital, por exemplo, na internet.

**Gerenciamento de identidade** pode ser definido como o processo de criar, atualizar, deletar e armazenar contas de identidade digital e gerenciar o acesso a recursos por meio do processo de autenticação e autorização.

Existem vários modelos de identidade digital, incluindo os modelos de identidade centralizada, federada e descentralizada. Vamos explorar esses modelos a seguir.

**Modelo de identidade centralizado**

O modelo de identidade centralizado é o que mais conhecemos. É o modelo mais amplamente utilizado, com identificadores e credenciais emitidos por provedores de serviço centralizados e governos, por exemplo, logins do Facebook, logins do Google, passaportes, carteiras de cidadão e carteiras de motorista. Podemos pensar em duas variações deste modelo:

* **Modelo de identidade centralizado independente**: este modelo representa paradigmas em que um único usuário possui uma credencial de um único provedor de serviço. Há uma relação direta um-para-um, e as credenciais emitidas, como login e senha, funcionam apenas com uma única entidade da qual foram emitidas. O usuário precisa manter credenciais individuais emitidas por cada uma das entidades.
* **Modelo centralizado compartilhado**: este modelo é baseado em um paradigma em que um provedor de serviço central mantém as credenciais e elas são compartilhadas entre diferentes provedores de serviço.

**<p style="page-break-after: always;"></p>**

Este modelo é o modelo original de identidade na internet, também chamado de “modelo de identidade baseado em conta”. Aqui, um usuário estabelece sua identidade fornecendo detalhes de registro a um provedor de serviço e criando uma conta com este provedor de serviço, por exemplo, um site.

No entanto, há vários problemas com este modelo:

* É centralizado, então todos os seus dados residem do lado do provedor de serviço.
* Você não tem controle sobre seus dados, como eles são usados, onde são usados, quando são usados e para qual propósito — mesmo que haja regulamentação a respeito, não está realmente sob seu controle. E se o provedor de serviço estiver fazendo o melhor para atender seus clientes, mas um hacker conseguir roubar dados pessoais do provedor?
* A responsabilidade recai sobre o usuário para manter seus nomes de usuário e senhas para cada provedor de serviço.
* Sendo centralizado, está sujeito a invasões e vazamentos de dados. Basta procurar na internet por “vazamentos de dados em 2023” ou “vazamentos de dados recentes” para ter uma ideia da magnitude desse problema.
* Cada provedor de serviço tem suas próprias políticas de segurança para gerenciamento de contas e pode impor essas políticas aos clientes, incluindo escolha de senha, seu comprimento, uso, etc. Embora a intenção seja boa aqui, na prática torna-se bastante difícil para os clientes gerenciar suas contas.
* Modelo básico de conta — um nome de usuário e senha.

**Modelo de identidade federada**

O princípio fundamental da ideia do modelo de identidade federada é permitir que um usuário utilize a mesma conta (nome de usuário e senha) e outras credenciais para autenticar-se com vários provedores de serviço. Membros de uma federação (daí o nome identidade federada) confiam uns nos outros ou em um provedor central para autenticar seus respectivos usuários e atestar seu direito de usar os serviços. Por exemplo, uma vez que um usuário é autenticado por um membro, não precisa fazer login novamente em outro provedor de serviço com um nome de usuário e senha separados. O usuário pode fazer login usando as mesmas credenciais que foram usadas para acessar o provedor de identidade em outros serviços da rede, sem precisar inserir novamente nome de usuário e senha.

O modelo opera com uma entidade central chamada **provedor de identidade**, que fica entre a organização e o usuário. Além disso, uma vez que o usuário faz login com suas credenciais em um provedor de serviço, se estas forem aceitas por outro provedor de serviço, as mesmas credenciais podem ser usadas para autenticar-se com outros provedores. Por exemplo, fazer login em outro site usando o Google ou outro provedor de identidade de terceiros é um exemplo de modelo de identidade federada. Nesse caso, os usuários se autenticam com o provedor de identidade (Google) usando suas credenciais de login, e o provedor de identidade emite um **token de segurança** para o usuário. O usuário pode então usar esse token para acessar outros sites ou aplicativos sem criar um novo login ou compartilhar suas informações pessoais com esses outros sites ou aplicativos.

A autenticação federada permite um processo contínuo de autenticação de usuário enquanto o site ou aplicativo mantém seu domínio de segurança. Ela também reduz a necessidade de os usuários lembrarem várias senhas e credenciais. Além disso, ajuda a garantir que as informações do usuário não sejam compartilhadas entre aplicativos ou sistemas. A identidade federada é comumente usada na internet e é suportada por protocolos como **OAuth** e **OpenID Connect**.

A Figura 20.1 abaixo mostra a diferença arquitetural chave entre esses dois modelos:

**Figura 20.1: Modelo centralizado vs modelo federado**  
<IMAGEM>

O diagrama anterior mostra, no lado esquerdo, o relacionamento entre um usuário e o provedor de serviço em um modelo de identidade centralizado tradicional, onde o usuário possui uma conta com o provedor de serviço. No lado direito, a entidade provedora de identidade abaixo do usuário fornece os serviços de identidade em um modelo de identidade federada.

A ideia é usar um provedor de identidade central onde você possui apenas uma conta de identidade. No entanto, agora, por meio do IDP, você pode fazer login em múltiplos provedores de serviço, por exemplo, sites ou aplicativos. Essa coleção de todos os provedores de serviço que usam o mesmo provedor de identidade é chamada de **federação**, daí o nome do modelo. As partes de uma federação confiam umas nas outras para autenticação, por isso são chamadas de **partes confiáveis** (*relying parties*). Vários protocolos de identidade federada para permitir o *Single Sign-On* (SSO) foram desenvolvidos e funcionam muito bem, incluindo:

* **SAML (Security Assertion Markup Language)**: permite a troca de dados de autenticação e autorização entre entidades, permitindo o login único federado e acesso a recursos entre diferentes domínios. Usa mensagens baseadas em XML para transmitir com segurança informações de identidade e autorização entre um provedor de identidade e um provedor de serviço.
* **OAuth (Open Authorization)**: é um protocolo padrão aberto para autorização de acesso delegado que permite aos usuários conceder a aplicativos ou serviços de terceiros acesso a recursos específicos em seu nome, sem a necessidade de compartilhar suas credenciais de login. Ele fornece um mecanismo de autenticação baseado em token que permite aos proprietários dos recursos conceder e revogar acesso a seus recursos a qualquer momento, oferecendo uma forma segura e amigável de compartilhar informações entre diferentes plataformas e serviços.  
  Mais informações sobre o padrão estão disponíveis aqui: <https://oauth.net/2/>.
* **OpenID**: O protocolo OpenID é um mecanismo de autenticação aberto e descentralizado promovido pela fundação sem fins lucrativos OpenID Foundation. Ele permite a autenticação de usuários em vários sites cooperantes, conhecidos como **Partes Confiáveis (RPs)**, usando um serviço de **Provedor de Identidade (IDP)** de terceiros, eliminando a necessidade de os donos de sites criarem seu próprio mecanismo de login. Isso significa que os usuários podem fazer login em diferentes sites sem precisar lembrar nomes de usuário e senhas separados para cada um. Para criar uma conta, os usuários escolhem um provedor de identidade OpenID, que podem usar para acessar qualquer site que aceite autenticação OpenID. É um padrão comumente usado por muitos serviços na internet. Alguns provedores de OpenID incluem Ubuntu One e Microsoft.

Este modelo também é amplamente utilizado na internet, por exemplo, quando você usa sua conta do Twitter para fazer login em outro site ou usa sua conta da Amazon para visitar os sites de outros vendedores online.

Este modelo pode aliviar alguns dos problemas do modelo centralizado, mas ainda sofre de alguns problemas-chave, descritos a seguir:

* Não há um serviço único; existem muitos provedores de identidade, o que leva os usuários a manterem credenciais de diferentes provedores.
* Segurança entre provedores de identidade, obrigações contratuais, compartilhamento e gerenciamento de dados tornam-se difíceis e também levantam preocupações de segurança e privacidade.
* Os provedores de identidade ainda são centralizados.
* Alta dependência de um único provedor de identidade, o qual está sujeito aos mesmos problemas de invasões e vazamentos de dados do modelo de identidade centralizado.
* Além disso, se um provedor de identidade for comprometido, significa comprometer todos os provedores de serviço vinculados à mesma conta.

Em resumo, o modelo de identidade federado funciona bem, mas possui limitações principalmente devido à sua arquitetura centralizada. Em geral, ambos os modelos sofrem com problemas de confiança, gerenciamento de acesso, alocação de recursos, identidade compartilhada, privacidade e dificuldade de uso.

Para gerenciar as identidades dos usuários e seu acesso ao sistema, são utilizados os Sistemas de Gerenciamento de Identidade e Acesso (*Identity and Access Management Systems – IAMS*). Esses sistemas realizam principalmente duas funções: **autenticação** e **autorização**. Autenticação fornece garantias sobre a identidade de uma entidade, e a autorização determina os direitos de acesso, ou seja, o que um usuário autenticado pode fazer no sistema.

Existem vários atores que compõem um sistema de gerenciamento de identidade e acesso. Essas entidades incluem usuários, provedores de serviço, provedores de identidade e autoridades de atributos, como serviços de diretório, DNS e autoridades certificadoras. Esses sistemas são comuns na internet e em ambientes corporativos.

**Os IAMS enfrentam vários desafios, incluindo:**

* Privacidade e segurança
* Possibilidade de roubo de identidade
* Credenciais falsificadas
* Ainda utilizam credenciais em papel para integração ou, na melhor das hipóteses, cópias digitalizadas dos documentos de credenciais!
* Como atualizar informações pessoalmente identificáveis após mudanças nas circunstâncias pessoais
* Leva muito tempo para integração, exige atestação longa e verificação de documentos, etc.
* Centralização, sujeita a ataques cibernéticos, segmentação jurisdicional, comportamento anticompetitivo, censura, exclusão e inclusão

Então, o que mudou com o blockchain? Podemos usar o blockchain para melhorar ou construir um sistema de identidade completamente novo que não sofra com os problemas existentes que mencionamos anteriormente? Um novo modelo inspirado no blockchain surgiu que pode aliviar a maioria desses problemas, chamado de **modelo de identidade descentralizado**.

Alguns benefícios que podem ser alcançados ao se utilizar tecnologia blockchain para IAM estão listados abaixo:

* As informações sobre a identidade são auditáveis, rastreáveis e verificáveis
* Acessível publicamente por qualquer pessoa
* Resistente à censura devido às garantias de segurança do blockchain
* Combate fraudes devido à transparência
* Promove maior confiança devido à transparência
* Eficiente
* Pode reduzir a taxa de fraudes
* Reduz os custos de verificação
* Pode facilitar a provisão de Identidade Auto-Soberana (SSI) e DIDs. Pode servir como a camada na qual SSI e DIDs podem ser fornecidos com relativa facilidade em comparação com outras abordagens

No entanto, alguns desafios incluem a privacidade e proteção das informações pessoalmente identificáveis e se devemos ou não armazenar dados pessoais em uma cadeia. Esses desafios podem ser tratados de diferentes maneiras. Veja o Capítulo 18, *Privacidade no Blockchain*, para mais detalhes sobre privacidade.

Observe que **não é necessariamente o caso de que a tecnologia blockchain seja essencial para fornecer identidade descentralizada**, mas a infraestrutura sólida, segurança, integridade e benefícios de descentralização que o blockchain oferece fazem dele a primeira escolha como plataforma sobre a qual soluções robustas de identidade descentralizada podem ser construídas. O blockchain oferece uma camada fundamental forte para construir um sistema de identidade descentralizado, utilizando as propriedades de descentralização, integridade e segurança que, de outra forma, seriam difíceis — e em alguns casos simplesmente impossíveis (por exemplo, descentralização) — de alcançar por outros meios.

O blockchain pode habilitar o modelo de identidade descentralizado, facilitando a criação de identidade descentralizada, sistemas de gerenciamento de identidade e acesso baseados em blockchain e **identidade auto-soberana**.

Agora, vamos discutir o modelo de identidade descentralizado.

**Modelo de identidade descentralizado**

Este modelo não depende de um modelo de identidade centralizado ou federado, mas é descentralizado em um nível fundamental. Ele **não é baseado em contas** e funciona com **identidades reais em um formato ponto a ponto (peer-to-peer)**, onde uma relação direta entre entidades (pares) é estabelecida, em vez de passar por um provedor central. Não há “conta” que exista nesse mundo; em vez disso, uma relação direta entre pares com base em credenciais é estabelecida. Além disso, não há uma única parte controlando esse mecanismo; **todos estão no controle**. Enquanto quiserem manter uma relação com uma entidade, a “conexão” estará ativa. Se alguma das partes (par) não quiser mais, ela simplesmente encerra o canal sem deixar para trás qualquer informação pessoal.

Com essa propriedade de conectividade ponto a ponto, a descentralização pode ser alcançada porque qualquer par pode se conectar diretamente a qualquer outro par. A intenção original da internet era ser descentralizada e ponto a ponto, mas ao longo de muitas décadas ela se tornou centralizada devido aos provedores de serviço centralizados. O que mudou é se agora posso criar um mecanismo de identidade verdadeiramente descentralizado que se baseia em uma rede ponto a ponto, mas que é realmente descentralizado, e nenhuma autoridade única pode controlá-lo. A resposta para isso é — você adivinhou — **blockchain**!

O blockchain fornece uma **rede ponto a ponto** e uma **Infraestrutura de Chave Pública Descentralizada (DPKI)**, que fornece os ingredientes necessários para desenvolver um modelo de identidade descentralizado.

Os sistemas atuais de Infraestrutura de Chave Pública (PKI) são baseados em uma terceira parte confiável chamada Autoridade Certificadora (CA), que verifica a identidade dos usuários e emite certificados digitais que vinculam chaves públicas a identidades específicas. Eles são baseados em um modelo hierárquico de confiança que se estende do certificado do usuário até uma CA raiz confiável. Embora esse sistema funcione razoavelmente bem, assegure a integridade das identidades digitais e proteja a comunicação online, ele é fundamentalmente centralizado. Como resultado, sofre de problemas como **personificação** e **ataques de intermediário (man-in-the-middle)** para obter um certificado válido de forma indevida de uma CA, que confia implicitamente em uma terceira parte — tornando as CAs um ponto único e centralizado de falha.

Como o blockchain é um armazenamento de chave-valor descentralizado e distribuído, ele permite uma abordagem melhor para gerenciar identidades digitais de forma segura e transparente sem depender de uma CA centralizada. Podemos construir um sistema de emissão e revogação de certificados mais eficiente usando blockchain. Os sistemas PKI tradicionais têm um processo lento e menos seguro de revogação de certificados. No entanto, no DPKI, a revogação e a emissão podem ser feitas rapidamente, com todos os nós da rede atualizados em tempo real. Ao eliminar a dependência de uma CA centralizada, o DPKI torna os sistemas PKI mais seguros, transparentes, eficientes e descentralizados, dando aos usuários **mais controle sobre sua identidade digital**.

Fundamentalmente, ele fornece uma maneira segura de trocar chaves públicas para estabelecer uma conexão segura ponto a ponto entre duas partes e também fornece uma camada de armazenamento segura para armazenar essas chaves públicas para permitir a verificação de assinaturas digitais em credenciais de identidade digital, ou seja, **credenciais verificáveis (VCs)** que pares podem trocar para provar sua identidade no mundo real.

À medida que o interesse no paradigma de identidade descentralizada cresceu, surgiu um novo termo: **SSI** (*Self-Sovereign Identity*, ou Identidade Auto-Soberana).

**Identidade auto-soberana**

Identidade auto-soberana (SSI) pode ser definida como a identidade de uma pessoa que é **totalmente possuída e controlada pela própria pessoa**. Ela não depende nem está sujeita a qualquer outra autoridade ou terceira parte confiável.

Observe que SSI **não significa** que qualquer um pode alegar qualquer coisa sobre sua identidade e sair impune. Em vez disso, há um modelo de emissão seguro e estável por trás dessas identidades e credenciais que, com a ajuda de vários outros atores, componentes e tecnologias (especialmente criptografia), garante a integridade do sistema. Além disso, SSI não se aplica apenas a indivíduos; aplica-se a qualquer entidade, máquina, dispositivo IoT e, praticamente, qualquer coisa que precise de uma identidade no mundo digital.

O controle no modelo descentralizado de SSI **permanece com o usuário**, enquanto nos modelos tradicionais centralizados e federados, o controle está nas mãos dos provedores de serviço, emissores e verificadores.

Podemos visualizar a diferença entre os modelos de SSI centralizado e descentralizado na imagem a seguir:

**Figura 20.2: Modelo centralizado vs SSI**  
<IMAGEM>

O diagrama anterior mostra como o modelo descentralizado de SSI, exibido no lado direito, **devolve o controle ao usuário**, em vez do modelo centralizado mostrado à esquerda, onde o usuário está fora do mecanismo de controle, que é controlado por emissores e verificadores. Observe que no modelo de SSI, o usuário está **no centro do ecossistema** e no controle.

Essa mudança fundamental no controle, das entidades centralizadas para o usuário, torna o paradigma SSI tão impactante e profundamente elegante, o que resulta em casos de uso extremamente poderosos. Pode ajudar a melhorar processos de negócios, estabelecer confiança, melhorar a experiência do cliente, abordar requisitos regulatórios de maneiras inovadoras, resistir à vigilância e à pirataria de dados, e melhorar os setores de governo, finanças, saúde e praticamente todas as outras indústrias.

Há dois elementos-chave que viabilizam esse modelo: **VCs (credenciais verificáveis)** e **DIDs (identificadores descentralizados)**. Vamos agora analisar a composição do SSI.

**Componentes do SSI**

Nesta seção, exploraremos os componentes do SSI, incluindo **VCs**, que são credenciais verificáveis resistentes a adulterações e criptograficamente seguras, e **apresentações verificáveis (VPs)**, que são formatos de dados usados para compartilhar uma ou mais VCs.

**Credenciais verificáveis**

Já vimos o que é uma credencial em papel anteriormente. As VCs são os equivalentes digitais das credenciais em papel. Embora também esperemos que as credenciais em papel sejam verificáveis, devido aos problemas discutidos anteriormente, o paradigma atual de credenciais em papel **não é seguro nem à prova de fraudes**.

Formalmente, podemos definir credenciais como um conjunto de informações sobre uma entidade (**sujeito**) que uma autoridade (**emissor**) afirma ser verdadeiro, e usando essas afirmações, o sujeito pode convencer outras partes (**verificadores**) de que essas afirmações são verdadeiras. Aqui, uma relação de confiança entre o verificador e o emissor é implícita. Além disso, o sujeito confia que o emissor emitiu afirmações convincentes e verificáveis. As credenciais podem pertencer a sujeitos humanos, máquinas, organizações e qualquer outra entidade que precise convencer outras sobre a veracidade de suas afirmações. As afirmações podem ser divididas em três categorias amplas:

* **Pessoais**, como idade e etnia
* **Relacionamentos e associações**, como país de residência, cônjuge, membro de alguma instituição, etc.
* **Direitos adquiridos**, como direitos legais, aposentadoria, isenção de impostos, benefícios do Estado, etc.

Essas credenciais devem ser **verificáveis**, ou seja, devem ser capazes de convencer o verificador de que as afirmações feitas são verdadeiras. Para isso, o verificador deve ser capaz de estabelecer:

* Quem é o emissor da credencial, por exemplo, uma universidade ou um governo
* Que a credencial **não foi adulterada**
* Que a credencial é **válida**, ou seja, não expirou ou foi revogada. Pode haver outras condições de validade dependendo do caso de uso, por exemplo, uma credencial que concede isenção de imposto não é transferível. Assim, uma condição de validade poderia ser que ela não tenha sido transferida ou reutilizada, caso seja uma oferta de uso único.

Tradicionalmente, com credenciais em papel, essas garantias são fornecidas por recursos de segurança como **padrões guilhochês**, **hologramas** ou algum outro recurso que não pode ser modificado ou copiado. No mundo digital, porém, usamos **criptografia** para alcançar essas garantias. Por exemplo, o emissor, a validade e a integridade (autenticidade) podem ser verificadas usando **assinaturas digitais**.

Como dissemos anteriormente, VCs são o equivalente digital de credenciais físicas em papel. Em essência, as VCs são versões digitais das credenciais que os usuários podem carregar em seus **dispositivos móveis**, assim como eram carregadas em carteiras físicas antes. A diferença é que elas são **digitalizadas (verificáveis eletronicamente, criptograficamente seguras)** e têm várias vantagens sobre suas contrapartes físicas. Uma comparação entre credenciais físicas e VCs digitais é mostrada na tabela abaixo, destacando os benefícios das VCs:

**Comparação: Credenciais físicas em papel vs Credenciais verificáveis digitais**

| **Atributo/tipo** | **Credenciais físicas em papel** | **Credenciais verificáveis digitais** |
| --- | --- | --- |
| **Clonagem** | Possível, mesmo que em alguns casos seja extremamente difícil, por exemplo, cartões de ID nacionais e passaportes. Isso se refere a passaportes de papel, não os digitais. | Não é possível. |
| **Invasão (hacking)** | Relativamente fácil. | Extremamente difícil, a menos que o dispositivo onde as VCs estão armazenadas esteja completamente acessível ao invasor, o que é bastante difícil com esquemas de segurança modernos (2FA, biometria etc.). |
| **Preservação da privacidade** | Pouca ou nenhuma privacidade. Um agente de controle de fronteira vê tudo no seu passaporte. | Mais privacidade é possível, pois a **divulgação seletiva** é viável com VCs. |
| **Acesso e permissões** | Acesso total para qualquer pessoa; um verificador pode ver todas as declarações e atributos relacionados à entidade. | Suporte ao princípio da autoridade mínima/permissões seletivas (controle de acesso), o que permite mais segurança. |
| **Custo** | Alto, especialmente na produção de documentos de viagem e outros documentos sensíveis, devido ao alto custo dos recursos de segurança especiais. | Custo muito baixo, apenas custo inicial de infraestrutura; uma vez estabelecida, é possível emitir e verificar virtualmente quantas credenciais forem necessárias. |
| **Limitações físicas** | Como objetos físicos, podem ser difíceis de carregar, suscetíveis a roubo e perda, e difíceis de proteger contra desgaste. | Praticamente impossível de roubar; geralmente armazenadas em dispositivos móveis. Mesmo se o dispositivo for perdido, as VCs podem ser reemitidas rapidamente. |
| **Delegação** | Não é possível. | Possível, se permitido conforme a política do emissor. |

**Arquitetura**

O ecossistema de VCs consiste em vários **atores e componentes**, que são descritos a seguir:

* **Emissor**: A entidade que emite VCs para os sujeitos.
* **Sujeito**: Uma entidade cujas afirmações (atributos) são armazenadas na VC.
* **Titular**: Uma entidade que está na posse da VC e a apresenta ao verificador para verificação quando necessário.
* **Verificador**: Uma entidade que assegura que as afirmações feitas na VC são verdadeiras e corretas. O verificador recebe VCs do titular e as verifica.
* **Carteira digital**: Uma entidade que armazena as VCs para o titular.
* **Agente digital**: Um software que atua como interface entre o ecossistema de VC e o titular, por exemplo, um aplicativo em um smartphone.
* **Registro de dados verificáveis** (*Verifiable Data Registry – VDR*): É uma entidade que é a fundação do ecossistema de VC e do ecossistema de identidade descentralizada. Ele existe na internet e é acessível a praticamente todos os atores do ecossistema de VC. Armazena os dados necessários para operar com sucesso um ecossistema de VC, como **chaves públicas dos emissores**, **o esquema de todos os atributos da VC**, uma **lista de revogação**, uma **lista de expiração** e **metadados** descrevendo VCs e outros dados. Registros de Dados Verificáveis podem ser blockchains ou bancos de dados descentralizados e centralizados. No entanto, blockchains, devido às suas características de segurança inerentes, podem ser mais adequados para o ecossistema de identidade descentralizada.

A imagem a seguir mostra a arquitetura de alto nível do ecossistema de VCs:

**Figura 20.3: Ecossistema de VCs**  
<IMAGEM>

Observe que, no diagrama, o **titular** da VC está no controle, enquanto todos os outros atores ao seu redor facilitam o processo de emissão e verificação da credencial. No ecossistema de VCs, é necessário que o verificador confie nos emissores como autoridades que são autorizadas a emitir VCs; no entanto, o **registro de dados verificáveis** é confiado por todos os atores. Os verificadores podem estabelecer sua própria **relação de confiança** e **regras de verificação** para a verificação das VCs. Por exemplo, uma VC emitida por uma autoridade nacional de saúde para vacinação contra Covid é confiável e quase todos os verificadores a aceitam sem hesitação. No entanto, uma VC emitida por uma entidade privada (mesmo que seja criptograficamente correta) pode não ter o mesmo nível de autenticidade e confiabilidade que a emitida por uma autoridade nacional. Assim, os verificadores têm a liberdade de decidir **quais VCs confiam e estão dispostos a aceitar**.

O sujeito também confia na carteira onde as VCs são armazenadas para protegê-las com segurança. Além disso, o verificador e o sujeito da VC confiam no emissor para emitir credenciais legítimas e corretas e revogá-las caso expirem, não sejam mais válidas ou sejam comprometidas. A confiança envolve algum tipo de **utilização e gerenciamento de chave pública/privada e certificados**, e o blockchain oferece esse mecanismo.

Lembre-se de que o sujeito e o titular geralmente são a mesma entidade, mas às vezes podem ser diferentes, por exemplo, um passe de acesso a uma creche emitido para uma criança mas gerenciado pelos pais, ou um comprovante de vacinação de um gato emitido para o dono do animal. No entanto, pode ser realmente o **gato** que possui a VC em seu chip (etiqueta).

**Estrutura de uma VC**

Uma **VC** geralmente é composta por oito elementos:

* **Contexto (Context)**: Contém um ou mais Identificadores Uniformes de Recursos (**URIs**) que fornecem as informações necessárias para interpretar e verificar corretamente uma credencial, incluindo os padrões, modelos de dados e vocabulários usados para construir a VC.
* **Tipo (Type)**: Contém URIs que indicam o tipo da VC. O verificador lê o tipo e usa essa informação para decidir se pode interpretar e lidar corretamente com a credencial. Se o verificador encontrar um tipo desconhecido ou não reconhecido, ele simplesmente rejeita a credencial.
* **ID**: Este é o identificador único criado pelo emissor para esta VC e contém um único URI.
* **Emissor (Issuer)**: Descreve a entidade que emitiu a VC. Contém um URI que aponta para um documento que descreve o emissor.
* **Sujeito da credencial (Credential subject)**: Essa propriedade transmite informações sobre o sujeito da credencial. Inclui um identificador pseudônimo do sujeito na forma de um URI, juntamente com um conjunto de propriedades (**declarações**) que o emissor está afirmando sobre o sujeito. O pseudônimo preserva a privacidade do sujeito.
* **Prova (Proof)**: Contém a prova — geralmente uma assinatura digital — para verificar a VC.
* **Data de emissão (Issuance date)**: A data de emissão da VC, no formato **RFC3339**.
* **Status**: Permite a descoberta, por meio de um URI, de informações sobre o status atual da VC, como se ela está suspensa, revogada, etc.

**URI** é uma sequência de caracteres que identifica um nome ou recurso na internet ou em qualquer outra rede. Ele fornece uma forma padrão de identificar recursos como páginas da web, imagens, vídeos e outros arquivos. Um URI localiza um recurso em uma rede e especifica o protocolo para recuperá-lo, como HTTP, FTP ou file.

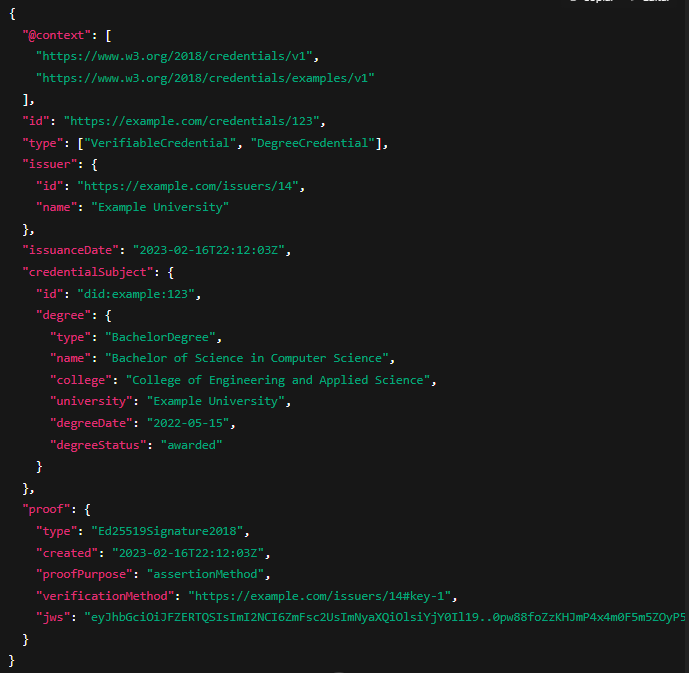
A estrutura da VC é mostrada na Figura 20.4 abaixo:

**Figura 20.4: Estrutura de uma VC**  
<IMAGEM>

A Figura 20.4 mostra a estrutura de uma VC, incluindo os componentes de metadados, declaração e prova.

Outras propriedades podem incluir data de emissão, data de expiração, status da credencial e outros campos, dependendo do caso de uso. Além disso, uma VC, dependendo do caso de uso, pode conter **múltiplas declarações** e **múltiplas provas**, e pode ser apresentada opcionalmente em algo chamado de **VP** (*Verifiable Presentation*).

Um exemplo de VC é mostrado abaixo, exibindo algumas das propriedades discutidas acima. Está codificado em **JSON-LD**:



O exemplo acima mostra uma VC para o diploma de uma pessoa da "Example University", com todos os detalhes necessários e prova criptográfica. Vamos ver o que cada elemento significa:

* "@context": Especifica o contexto JSON-LD do documento
* "id": Um identificador único para a credencial
* "type": O tipo da credencial, que neste caso é VerifiableCredential e DegreeCredential
* "issuer": A entidade que emitiu a credencial, incluindo um identificador e nome, ou seja, Example University
* "issuanceDate": A data e hora em que a credencial foi emitida pela universidade
* "credentialSubject": O sujeito da credencial, neste caso uma pessoa com diploma de Bachelor of Science in Computer Science, incluindo os detalhes do diploma como tipo, nome, faculdade, universidade, data e status
* "proof": Prova criptográfica da credencial, incluindo o tipo de assinatura, data de criação, finalidade da prova, método de verificação e a assinatura JWS da credencial

**JWS** é um padrão proposto pela IETF (descrito no RFC 7515) para assinar dados arbitrários.

**Apresentação verificável**

Uma **VP (Verifiable Presentation)** é usada para apresentar uma ou múltiplas **VCs** a um verificador para verificação, conforme mostrado na Figura 20.5 abaixo:

**Figura 20.5: Estrutura de uma VP**  
<IMAGEM>

O diagrama anterior mostra a estrutura de uma VP, contendo metadados sobre a VP, uma prova assinada pelo sujeito (titular) e um conjunto de VCs.

Uma VP também permite que um **titular de uma VC revele apenas um subconjunto de informações** da VC a fim de preservar a privacidade. Pode ser criada uma VP que contenha **somente as informações mínimas necessárias para a verificação**, e **nada além do que o titular deseja compartilhar**. Por exemplo, revelar apenas que o titular tem mais de 18 anos de idade e nada mais. Assim, o titular enviará somente essa informação como uma VP ao verificador para verificação. Isso é chamado de **divulgação seletiva** ou **divulgação mínima**.

**Identificadores descentralizados**

Um **DID** é uma identidade auto-soberana que é **permanente**, **portável** e **verificável**, e **não depende de nenhuma autoridade centralizada**. Em termos práticos, podemos pensar em um DID como um novo tipo de identificador globalmente único ou endereço que é **criptograficamente aumentado** para suportar verificação segura e descentralização.

Formalmente, podemos definir um DID como um identificador globalmente único, permanente, geralmente gerado criptograficamente e registrado criptograficamente, que **não depende de nenhuma autoridade centralizada**.

Um DID é um novo tipo de URI. A Figura 20.6 abaixo mostra como é a aparência de um DID:

**Figura 20.6: DID**  
<IMAGEM>

O diagrama anterior mostra o formato de um DID, que é composto por três elementos separados por dois-pontos. É uma estrutura simples, consistindo de:

* **Esquema (Scheme)**: Uma string fixa, "did", indicando que este é um DID.
* **Método (Method)**: O método que foi usado para gerar o DID, por exemplo, ethr, btcr, sov, web, etc. Basicamente, descreve onde o DID está localizado e em qual protocolo ele opera. Esse elemento ajuda o DID a ser resolvido para um documento DID correspondente. Pode também ser definido como uma especificação de como a sintaxe específica de um identificador é implementada. A **especificação do método DID** define as operações pelas quais DIDs e documentos DID são criados, resolvidos, atualizados e desativados.

Um **documento DID** pode ser definido como um conjunto de elementos que descrevem o sujeito do DID e o material criptográfico que o sujeito do DID pode usar para se autenticar e provar sua associação com o DID. Cada especificação de método DID define um esquema que funciona com um método DID específico.

* **String específica do método (DID method-specific string)**: Também chamada de **identificador específico do método**. A sintaxe deste identificador é definida pelo método DID. Normalmente, é uma string longa gerada por métodos criptográficos e **deve ser única**.

Exemplos:

* + O método btcr baseado em Bitcoin gera o identificador a partir da posição de uma transação no blockchain do Bitcoin.
  + O método web utiliza um nome de domínio DNS seguro com certificado TLS/SSL, com um caminho para o documento DID.  
    Exemplo: did:web:masteringblockchain.com → aqui, a resolução do documento DID é feita por meio do nome de domínio.
  + Outro exemplo:  
    did:ethr:0x2fEFA78F636002fe9B3B43A3d3672b011420ea90  
    → onde did é o esquema, ethr é a blockchain (VDR), e a string hexadecimal é o endereço do contrato inteligente.

Há muitos métodos DID já estabelecidos, e as especificações estão disponíveis aqui:  
<https://www.w3.org/TR/did-spec-registries/#did-methods>

**Requisitos dos DIDs**

Os DIDs possuem vários requisitos importantes:

* **Descentralizados**: Não dependem de nenhuma entidade centralizada, provedor de identidade ou autoridade.
* **Verificáveis criptograficamente**: Capazes de provar criptograficamente a identidade e a propriedade, associadas a um par de chaves pública/privada.
* **Persistentes**: Estão permanentemente vinculados a um sujeito e não mudam.
* **Resolvíveis**: É possível descobrir mais informações sobre o sujeito (para descoberta de metadados).
* **Fáceis de criar**: A criação de DIDs deve ser rápida, fácil e de custo muito baixo.

Os DIDs são os **contrapartes das VCs**. Eles são gerados criptograficamente e verificáveis. O formato de um DID é:  
esquema:método:string específica do método — como mostrado na Figura 20.6.

O DID possui diversos relacionamentos com várias entidades no ecossistema. A Figura 20.7 abaixo mostra a arquitetura de alto nível de um DID e os relacionamentos entre o **DID**, o **documento DID**, o **sujeito ou controlador do DID** e o **Verifiable Data Registry (VDR)**:

**Figura 20.7: Arquitetura e relacionamentos do DID**  
<IMAGEM>

Como mostrado no diagrama, um **DID** é uma string única que segue um formato específico, descrito anteriormente. Esse DID pode ser resolvido para um **documento DID**. Esse documento DID pode existir em um **blockchain** ou em algum outro armazenamento, como um **repositório global descentralizado de chave-valor**. Ele também pode existir no **IPFS**, **STORJ** ou até mesmo em um servidor web.

O documento DID contém **informações de identificação do controlador** sobre o sujeito, **material criptográfico para verificação** e outros metadados. O DID e o documento DID tornam a **verificabilidade criptográfica** possível por parte de um verificador. O funcionamento se dá da seguinte forma: o DID e o documento DID estão **criptograficamente vinculados entre si**. O documento DID pode ser serializado para armazenamento e transmissão utilizando **JSON**, **JSON-LD**, **CBOR**, ou qualquer outro mecanismo de serialização adequado ao caso de uso. DIDs funcionam principalmente em conjunto com VCs para formar um mecanismo de identidade descentralizada criptograficamente verificável.

Existem muitos tipos de DIDs e, embora todos suportem um conjunto básico de funções, cada um pode ser gerado utilizando mecanismos e conjuntos de regras diferentes, ou seja, diferem na forma como a funcionalidade é implementada. Esses diferentes mecanismos de implementação são chamados de **métodos DID**. Há muitos métodos DID.

Diversas operações podem ser realizadas com DIDs e são definidas pelas operações **CRUD**:

* **C – Create**: como gerar um DID e o documento associado – Criar
* **R – Read**: mecanismo de recuperação do documento DID – Ler
* **U – Update**: mecanismo de atualização do documento DID – Atualizar
* **D – Deactivate**: mecanismo de desativação do documento DID – Desativar

Por exemplo, em um blockchain, isso pode significar a execução de um **contrato inteligente** que gerencia essas operações.

Um **método DID** define como ler ou escrever um DID e um documento DID em um **registro de dados verificáveis (VDR)** específico, por exemplo, um blockchain. Um DID sempre identifica um **sujeito DID** e é resolvido para um **documento DID**. Esse documento contém informações sobre um DID específico. Ele pode conter dados como **chaves públicas**, **métodos de autenticação** usados para verificação, **endpoints de serviço** usados para interação, **carimbos de data/hora** (usados para auditoria) e **assinaturas digitais**, usadas para garantir a integridade dos dados.

Então, vamos agora pensar sobre qual é a **relação entre um DID e uma VC**. Devido ao relacionamento entre **chaves públicas e privadas** (ou seja, assinaturas digitais), um verificador pode verificar se a credencial realmente veio de um **emissor autêntico**.

Mas como o verificador sabe que a chave pública fornecida é a correta? Em um mundo descentralizado, isso parece impossível de resolver. No modelo PKI tradicional, temos **autoridades certificadoras (CAs)** de alto nível que todos confiam; elas assinam em nome de entidades que confiam nelas, então, de certa forma, uma terceira parte confiável certifica que a chave pública usada por uma empresa é realmente legítima, e devido à relação de confiança com a CA raiz, todos estão satisfeitos com a autenticidade da chave pública.

Mas isso é claramente um modelo **centralizado**.

Como isso funciona em um ecossistema SSI?

Como o verificador pode ter certeza, em um sistema descentralizado, de que a chave pública fornecida pelo emissor é correta e realmente pertence ao emissor autêntico? **É aí que entram os DIDs.** Os DIDs são identificadores que são vinculados à chave pública, e os verificadores podem resolvê-los (“consultar”) e, assim, terem **confiança de que a chave pública é autêntica**.

Agora, como podemos provar que o DID está vinculado à chave pública e ao **controlador do DID** (titular ou sujeito) que está no controle da chave privada correspondente?

A solução pode ser **gerar matematicamente o DID a partir da chave pública**, em vez de criar DIDs arbitrários. Assim, se existe uma relação matemática entre o DID e a chave pública, o controlador pode provar que o DID realmente pertence a ele, porque possui a chave privada e pode responder a qualquer desafio feito pelo verificador — por exemplo, por meio de um mecanismo de desafio/resposta.

Parece razoável até aqui, mas e se o controlador precisar **girar as chaves** (fazer rotação)? O DID precisaria mudar — mas isso **não está alinhado com os objetivos dos DIDs**, que vimos anteriormente: DIDs devem ser **persistentes**.

É aqui que entram os **documentos DID**.

O documento DID contém a chave pública, e como o DID **sempre é resolvido para um documento DID**, existe a possibilidade de **atualizar o documento DID** e, assim, a chave pública e outros metadados conforme necessário. Isso permite que o **DID permaneça persistente**, mas que o documento DID possa mudar — o que permite **rotação de chaves** e outras atualizações.

**O mecanismo funciona da seguinte maneira:**

1. O **controlador** (detentor da chave privada) gera o **DID** com base no par de chaves pública/privada inicial (gênese).
2. O controlador **publica o documento DID**, que contém o DID e a chave pública.
3. Agora, qualquer entidade usando o documento DID pode verificar **criptograficamente** que o DID e a chave pública associada estão **vinculados** entre si.
4. Se o controlador, por qualquer motivo, **mudar (girar) o par de chaves**, ele cria um documento DID **atualizado** e o **assina com a chave privada anterior**. Isso pode ser uma atualização no blockchain para registrar o novo documento DID. Em essência, o controlador publica um novo documento DID que contém o DID original e a nova chave pública, mas é **assinado com a chave privada original** — o que cria uma **cadeia de confiança** entre os documentos DID, que pode ser rastreada de volta através de qualquer número de atualizações até o DID original. Cada documento DID funciona como um novo certificado digital para a nova chave pública, **sem necessidade de autoridade certificadora** ou qualquer outra terceira parte confiável.

Essencialmente, o **controlador** é uma entidade que pode fazer alterações no documento DID.  
Também observe que o controlador **nem sempre** é o sujeito da identificação.

Portanto, em resumo, podemos dizer que um **documento DID** é uma **estrutura de dados** associada a um DID e que contém informações sobre a entidade que ele representa. Ele fornece uma maneira de **publicar e recuperar** informações sobre um DID, como sua chave pública, mecanismos de autenticação e outros metadados relevantes. O documento DID é normalmente hospedado em uma **URL**, conhecida como **endpoint DID**, que pode ser **resolvida por um resolvedor DID**. O resolvedor DID é responsável por buscar o documento DID associado a um determinado DID e retornar os dados associados ao solicitante.

Um exemplo genérico de **documento DID** é mostrado abaixo:



Alguns elementos-chave neste documento são:

* "@context": Indica o tipo do documento, por exemplo, um documento JSON-LD
* "id": O DID
* "authentication": Descreve esquemas criptográficos para verificação

Há muitos outros elementos possíveis; para uma referência detalhada, consulte o documento do W3C:  
📎 <https://www.w3.org/TR/did-core/>

Um **DID** é um identificador único usado para representar um indivíduo ou entidade em uma rede descentralizada. Ele é um identificador persistente que é **independente de qualquer autoridade ou registro centralizado**, e pode ser usado para **autenticar e autorizar** o detentor do identificador a acessar recursos ou participar de transações.

Uma **VC** é um **documento digital** que contém **declarações verificáveis** sobre um indivíduo ou entidade, como sua identidade, credenciais ou atributos. VCs são projetadas para serem **portáteis**, de forma que possam ser compartilhadas com outras pessoas, e são usadas para estabelecer confiança entre as partes. Elas são tipicamente emitidas por entidades confiáveis, como universidades, governos ou instituições financeiras, e podem ser verificadas por partes confiáveis usando **provas criptográficas**.

A **combinação de um identificador (DID)** e de uma **VC** (por exemplo, idade, cidadania ou qualificação) é o que constitui uma identidade digital. Essas VCs vivem em sua **carteira digital** no seu **smartphone**, e você pode apresentá-las quando quiser para qualquer parte que precise de verificação. Novamente, está **totalmente sob seu controle** decidir se irá apresentá-la ou não.

A **relação entre DIDs e VCs** é que **DIDs são usados como identificadores de sujeito** dentro das VCs.

**Carteiras digitais**

Uma **carteira digital** armazena **VCs** com segurança, protegendo-as contra roubo e corrupção, e as disponibiliza quando necessário. É semelhante a uma carteira de criptomoedas, que discutimos no Capítulo 6, *Arquitetura do Bitcoin*, embora existam algumas diferenças. Por definição, no entanto, ela é semelhante a uma carteira de criptomoedas. Existe outro tipo de carteira que é comum: a disponível em iOS e Android para armazenar cartões de pagamento e outras credenciais.

Uma carteira digital é um software ou hardware que permite ao seu proprietário **criar, armazenar, organizar e proteger chaves criptográficas, segredos e outros tipos de informações sensíveis e privadas**. Informações sensíveis podem incluir VCs, identificadores descentralizados (DIDs), dados pessoais, contas e qualquer objeto digital que se enquadre na definição de informação sensível ou privada.

Uma **carteira SSI** (Self-Sovereign Identity) é uma carteira digital usada principalmente no ecossistema de identidade descentralizada para **armazenar e gerenciar VCs**. Uma carteira SSI pode armazenar **DIDs, VCs, cartões, dados pessoais, credenciais de acesso, documentos de identidade digital, documentos de viagem** e muitos outros. Ela é baseada nos princípios de design de **portabilidade**, **privacidade em primeiro lugar** e **segurança em primeiro lugar**.

A arquitetura de alto nível de uma carteira SSI é mostrada na Figura 20.8 abaixo:

**Figura 20.8: Arquitetura de alto nível de uma carteira SSI**  
<IMAGEM>

A Figura 20.8 mostra um componente chamado **Agente**, ou seja, um **agente digital**. Esse agente intermedeia comunicações entre carteiras, usuários e outros agentes no ecossistema SSI. A **carteira em si** é o **elemento protegido central**, onde, por exemplo, as chaves e outros dados sensíveis são armazenados. Uma **vantagem fundamental desse design** é a separação de responsabilidades. Embora às vezes não se faça distinção entre esses dois elementos — e a carteira seja simplesmente chamada de "carteira", sem distinguir entre o agente e a carteira —, internamente, o **componente de interface (agente)** e a **carteira** são normalmente dois componentes distintos.

O agente consiste em:

* Armazenamento seguro
* Mecanismo de backup e recuperação
* Interface de mensagens para lidar com comunicações
* Funcionalidade de roteamento

O armazenamento seguro se comunica com os serviços da carteira via uma **API segura**. A carteira consiste em dois subcomponentes principais:

* **Armazenamento criptografado** (ambientes de execução confiáveis, elementos seguros e enclaves seguros)
* **Sistema de gerenciamento de chaves**

Carteiras de identidade podem **criar DIDs**. Lembre-se, no modelo SSI, **você cria suas próprias identidades** e **não depende de terceiros** para criar uma para você. Essas carteiras podem **armazenar esses DIDs em um blockchain** por meio de transações blockchain. Elas podem **armazenar, assinar e apresentar uma VC** a um verificador.

**Registros de Dados Verificáveis (*Verifiable Data Registries – VDRs*)**

Fundamentalmente, os **VDRs** são bancos de dados que servem como **fonte canônica de verdade e confiança** para **DIDs** e **chaves públicas**. Os VDRs são **bancos de dados distribuídos globais de chave-valor**.

Podemos usar bancos de dados centralizados para implementar SSI, mas isso talvez só seja aceitável dentro de um **grupo pequeno ou consórcio**. Para uma implementação de SSI em **escala global**, espera-se que os **blockchains se tornem o VDR preferido**.

Os VDRs podem ser:

* Um **blockchain público**
* Uma **blockchain de consórcio**
* Uma **blockchain permissionada privada**

Eles também podem ser **blockchains específicos de aplicação**, construídos propositalmente para SSI, por exemplo:

* **Evernym**
* **Sovrin public ledger**
* **Hyperledger Indy**
* **Hyperledger Ursa**
* **Hyperledger Aries**
* **Veres One**

Tais **ASBCs** (*application-specific blockchains*) para SSI suportam transações e tipos de registros que facilitam o gerenciamento de DIDs.

**Estruturas de Governança**

Fundamentalmente, a **confiança** é estabelecida em SSI por meio da **criptografia**, ou seja, **confiança criptográfica**. No entanto, isso por si só **não é suficiente**, porque o elemento da **confiança humana** também é necessário para estabelecer outra camada de relacionamentos confiáveis entre **emissores**, **verificadores** e **sujeitos**.

Isso é importante porque **VCs emitidas por uma agência desconhecida podem não ser consideradas valiosas**, a menos que haja um nível de **confiança humana** e **reconhecimento da autoridade** estabelecido no mundo real.

Essa confiança permite que os verificadores **determinem a legitimidade dos emissores** sob uma **estrutura de governança** confiável por eles.

Estruturas de governança também especificam as **políticas, padrões e procedimentos** do ponto de vista **comercial, legal e técnico**, que os emissores devem seguir para emitir, ou que os sujeitos devem seguir para obter uma VC.

Com todos os componentes discutidos até agora no ecossistema SSI, podemos pensar na **pilha SSI** como um **modelo de quatro camadas**, mostrado na Figura 20.9:

**Figura 20.9: Pilha SSI de quatro camadas**  
<IMAGEM>

No diagrama acima:

* A **camada de governança** é onde uma autoridade de governança ou um órgão profissional publica uma estrutura de governança a ser seguida por todos os atores do ecossistema.
* A **camada de credenciais** é onde o ecossistema de VCs existe com emissores, verificadores e titulares.
* A **camada de comunicações e interfaces** inclui componentes como agentes digitais, carteiras e outras interfaces de comunicação.
* A **camada de identidades e chaves** inclui DIDs, registros de DIDs e VDRs.

As **duas camadas inferiores** são responsáveis por alcançar a **confiança em nível técnico** (criptográfico), enquanto as **duas superiores** alcançam a **confiança em nível humano**.

**Identidade no Ethereum**

Contas na blockchain Ethereum podem ser vistas como **DIDs**. Isso ocorre porque qualquer número de contas pode ser gerado por qualquer pessoa **sem necessidade de permissão** e **sem a necessidade de armazená-las em um servidor centralizado**.

Embora essas contas **não sejam DIDs formais** e **não tenham VCs associadas**, devido à sua natureza permissionless, elas podem ser consideradas como DIDs.

Como analogia com os DIDs e o ecossistema SSI que discutimos anteriormente:

* Uma conta Ethereum possui uma **chave privada** e uma **chave pública**.
* A **chave pública** pode ser vista como a identidade do controlador.
* A **chave privada** é usada para assinar mensagens.

A blockchain Ethereum pode servir como um **VDR** que pode armazenar DIDs. Quaisquer VCs emitidas para o DID podem ser verificadas na blockchain **validando o DID do emissor**, que está armazenado na Ethereum. Pode ser, por exemplo, um contrato inteligente onde todos os DIDs são armazenados.

DIDs são **emitidos, mantidos e controlados por indivíduos**. Uma conta Ethereum é um exemplo de DID. Você pode criar quantas contas quiser **sem permissão de ninguém** e **sem a necessidade de armazená-las em um registro central**.

DIDs são armazenados em **ledgers distribuídos** (blockchains) ou **redes ponto a ponto**. Isso torna os DIDs:

* **Globalmente únicos**
* **Resolvíveis com alta disponibilidade**
* **Verificáveis criptograficamente**

Um DID pode ser associado a diferentes entidades, incluindo **pessoas, organizações ou instituições governamentais**.

**Identidade no mundo do Web3, DeFi e Metaverso**

Lembre-se de que, nos tempos da Web 1.0, o modelo baseado em conta com **nome de usuário e senha** era predominante. Esse ainda é o caso em um grande número de sites.

No entanto, com a Web 2.0, um novo paradigma começou a surgir onde **uma identidade podia ser usada para autenticar em outros provedores de serviço também**, em vez de apenas naquele em que foi originalmente criada. Vemos isso em uso em muitos serviços online — por exemplo, os botões “Entrar com Apple” ou “Entrar com Google”.

A Web 2.0 opera em um modelo de confiança de cima para baixo, onde **plataformas centralizadas estão no controle**.

Na Web3, o **usuário é empoderado**, e podemos simplesmente nos conectar a um provedor de serviço com base em **nossa identidade descentralizada** e nas **credenciais que possuímos**. Nada mais de modelos de conta centralizados nem de confiar em provedores de identidade centralizados. **Tudo o que você precisa é de uma carteira e das credenciais apropriadas** para se conectar a um provedor de serviço.

Vimos uma amostra disso quando usamos o **MetaMask** para nos conectar à Ethereum; no entanto, existem aplicações muito mais ricas com essa capacidade.

A seguir estão alguns exemplos de como **identidades descentralizadas** estão sendo ou podem ser usadas no mundo do **Web3**, **DeFi** e **metaverso**:

* **Acesso a sistemas DeFi com base em identidade**: em vez de conectar diretamente sua carteira a um protocolo DeFi, um **modelo baseado em identidade** pode ser usado onde **uma verificação KYC opcional** é feita com base em **VCs**. Os usuários podem então **obter acesso a serviços financeiros descentralizados**, como empréstimos, pools de liquidez, derivativos, DEXs e seguros, com base nessas credenciais. As credenciais **permitem manter a privacidade**, pois você pode escolher o que divulgar — e ainda assim cumprir a conformidade.
* **Identidade em jogos/metaverso**: Um **DID controlado por você** pode ser usado para **armazenar e acessar ativos, NFTs, pontuações, conquistas e avatares** em mundos virtuais, jogos e experiências do metaverso. Isso permite uma experiência fluida de identidade entre **múltiplos mundos virtuais**, o que seria impossível de alcançar com as identidades isoladas atuais baseadas em conta.
* **Acesso baseado em identidade à Web3**: A Web3 é centrada no usuário, mas atualmente o controle é mantido por **carteiras de criptomoedas**, o que não é ideal. Usar **DIDs + VCs** como base para identidade permite que você use **sua carteira como agente de identidade**, mas de maneira mais robusta e com suporte para **interações ricas de identidade com múltiplos serviços** — como acesso a DAOs, votação, aplicativos e outras plataformas.
* **Credenciais portáteis e compatíveis com conformidade para empresas Web3**: Uma empresa Web3 pode emitir VCs para seus colaboradores e usuários. Essas credenciais podem conter **provas de experiência, afiliação, cargo, etc.** Tais VCs podem ser usadas em várias plataformas como prova de identidade ou qualificação — sem revelar informações desnecessárias.
* **Acesso a eventos, comunidades e DAOs**: VCs podem servir como **passe de acesso** a **comunidades** com requisitos de entrada, como possuir certos NFTs, ser membro de uma DAO ou ter participado de um evento. Em vez de consultar blockchains diretamente (custoso e lento), as verificações são feitas usando as credenciais criptograficamente assinadas, que são rápidas e privadas.
* **Reputação descentralizada**: Um histórico de comportamento verificável em DAOs, protocolos DeFi, fóruns e marketplaces pode ser **convertido em VCs**, que formam uma base para **sistemas de reputação baseados em identidade descentralizada**.

**Projetos de blockchain específicos de SSI**

Há vários projetos de blockchain que focam especificamente em **identidade auto-soberana (SSI)**. Os mais conhecidos são:

* **Sovrin**
* **uPort**
* **Veres One**
* **Microsoft ION**
* **Hyperledger Indy**

Esses projetos estão sendo usados ou explorados em **governos, instituições educacionais, organizações globais**, e outros setores para emissão, gerenciamento e verificação de **identidades digitais** baseadas em DIDs e VCs.

O W3C mantém uma **lista de métodos DID** e implementações em andamento, que pode ser consultada aqui:  
📎 <https://www.w3.org/TR/did-spec-registries/#did-methods>

**Desafios da Identidade Descentralizada**

Embora a **identidade descentralizada** pareça **uma solução elegante e poderosa**, ela não está isenta de desafios. Alguns dos principais desafios são listados a seguir:

* **Escalabilidade e desempenho**: a verificação de credenciais e a resolução de DIDs podem ser **ineficientes**, especialmente se dependerem de registros públicos de blockchain. Operações de leitura e escrita podem ser lentas e dispendiosas em blockchains públicos.
* **Conformidade regulatória**: muitos governos e entidades reguladoras ainda não reconhecem **VCs e DIDs como equivalentes legais** a documentos tradicionais. A ausência de um quadro jurídico universal é um grande obstáculo para a adoção em larga escala.
* **Interoperabilidade**: há muitos esquemas e métodos de DIDs diferentes. Garantir que sistemas e implementações diversos possam interoperar com sucesso **é um grande desafio técnico e organizacional**.
* **Privacidade e controle de dados**: embora SSI seja projetado para dar controle ao usuário, o **armazenamento local de dados sensíveis**, se mal implementado, pode introduzir **novas vulnerabilidades**. Além disso, os usuários podem **não entender os riscos** de compartilhar determinadas VCs.
* **Gestão de chaves**: como os usuários agora controlam diretamente suas identidades, eles também precisam **gerenciar suas chaves privadas com segurança**. Isso inclui **backup, rotação, revogação e recuperação**, o que pode ser desafiador para usuários não técnicos.
* **Adoção pelo mercado**: apesar do crescente entusiasmo, **a adoção real de SSI ainda é limitada**. Convencer instituições a mudarem de sistemas centralizados para modelos descentralizados requer mudanças culturais, técnicas e regulatórias.
* **Confiança nos emissores**: os verificadores ainda precisam **confiar nos emissores** das credenciais. Um sistema descentralizado **não elimina a necessidade de confiança humana** em organizações e autoridades legítimas.
* **Custos de infraestrutura**: construir e manter um ecossistema de SSI pode envolver custos significativos, principalmente quando usado em grande escala.

**Resumo**

Neste capítulo, exploramos o mundo da **Identidade Descentralizada (SSI)**. Discutimos os seguintes pontos:

* Os **problemas com o modelo de identidade centralizado**
* O que é **identidade auto-soberana (SSI)**
* Os **componentes principais** do ecossistema SSI
* O conceito de **Credenciais Verificáveis (VCs)**
* O papel dos **Identificadores Descentralizados (DIDs)**
* Como os **documentos DID** se relacionam com as identidades
* O uso de **carteiras digitais** e **agentes**
* O papel de **Registros de Dados Verificáveis (VDRs)**
* A importância de **estruturas de governança**
* A relação entre SSI e **Web3/DeFi/metaverso**
* Os principais **desafios** da identidade descentralizada

Identidade descentralizada é uma das **inovações mais transformadoras** introduzidas por blockchains e tecnologias descentralizadas. Combinando **criptografia, descentralização e confiança entre pares**, SSI **tem o potencial de redefinir a forma como interagimos, nos identificamos e confiamos uns nos outros na internet**.