UNIVERSIDADE FEDERAL DE SANTA CATARINA

USO DE IDENTIFICADORES ÚNICOS DESCENTRALIZADOS PARA GESTÃO DE DISPOSITIVOS IOT NO CONTEXTO DA COMPUTAÇÃO FORENSE

Diovana Rodrigues Valim

FLORIANÓPOLIS
2023/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

USO DE IDENTIFICADORES ÚNICOS DESCENTRALIZADOS PARA GESTÃO DE DISPOSITIVOS IOT NO CONTEXTO DA COMPUTAÇÃO FORENSE

Diovana Rodrigues Valim

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

FLORIANÓPOLIS
2023/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

Diovana Rodrigues Valim

Uso de Identificadores Únicos Descentralizados para Gestão De Dispositivos IoT No Contexto Da Computação Forense

Submetido ao corpo docente do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como um dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Orientadora:

_____
Profª. Drª. Carla Merkle Westphall

Coorientador:

_____
Me. Leandro Loffi

Banca Examinadora:

_____
Prof. Dr. José Eduardo de Lucca

_____
Profª. Drª. Patricia Della Mea Plentz

Dedico este trabalho primeiramente a Deus, por sempre em Sua generosidade permitir que eu alcançasse todos os meus objetivos dentro de Seus planos. Ao universo e aos meus guias, por permanecerem ao meu lado mesmo quando sozinha eu julgava estar.

Aos meus pais e minha irmã, por todo apoio e dedicação durante todos estes anos. Sem vocês, nada disso seria possível. Sou e serei eternamente grata por me manterem de pé mesmo quando meus pés não podiam mais suportar o peso da minha própria mente.

A todos os meus ancestrais que contribuíram em sua luta para que eu pudesse estar no lugar em que estou agora.

AGRADECIMENTOS

Agradeço a todas as minhas professoras e professores, pelos ensinamentos durante esta jornada que me permitiram completar a minha jornada acadêmica. Em especial, à professora Carla Merkle Westphall, por orientar a construção deste TCC e por ter realizado este trabalho com dedicação e paciência.

Meus sinceros agradecimentos ao meu co-orientador Leandro Loffi, pela disponibilidade, paciência e solicitude.

Agradeço também à minha família, minha fortaleza em momentos difíceis, e aos meus amigos Artur Carmezini e Gabriel Cabral, por dividirem comigo este longo, porém recompensador caminho.

RESUMO

A *Internet* das coisas é um paradigma que se refere a uma malha de dispositivos conectados. Derivado do termo em inglês *Internet of Things* e comumente reduzido à sigla IoT, esta rede de dispositivos concerne à junção de inúmeros objetos interligados sobre um protocolo de telecomunicação comum. Neste contexto, trabalhando-se com uma malha robusta de nodos onde a comunicação deve ser feita de forma leve e efetiva, os dados trocados acabam sendo comunicados por meio de protocolos normalmente inseguros.

Computação forense, por sua vez, é definida como "a área da Ciência da Computação que se desenvolve gradualmente para atender à demanda oriunda da criminalística". Neste contexto, é necessário que exista controle sobre a identificação de provas digitais, para que os dispositivos móveis e embarcados envolvidos em uma cadeia de evidências sejam devidamente identificados e preservados.

Neste sentido, um sistema de identidade soberana torna-se uma das principais formas de garantir a autenticidade de um sujeito, pois permite a identificação de um usuário ou dispositivos de forma distribuída. Este é um conceito que pode, através de identificadores descentralizados únicos, DIDs, ser adaptado de forma a prover um processo de identificação digital seguro para todos os nodos de uma rede de dispositivos IoT. A *blockchain* é um recurso chave, pois permite a implementação desta estrutura, em que a descentralização caracteriza o elemento chave para garantir a segurança do sistema.

Dentro do ambiente de computação forense, é necessário garantir com máxima assertividade a autenticidade de todos os elementos do sistema. Portanto, este trabalho tem como objetivo a construção de um sistema de identidade digital descentralizada, baseado em uma construção de *blockchain*, para garantir propriedades de autenticidade e confiança nos nodos envolvidos no trânsito de evidências.

Palavras-chave: *Internet of Things*, *Blockchain*, Identidade Soberana, Computação Forense, *Decentralized Identifiers*.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1. Introdução

A computação forense, definida como o conjunto de práticas científicas auxiliares no processo de resolução de questões jurídicas e civis (COSTA & GARCIA, 2014), tem como objetivo maior o esclarecimento de crimes e violações a partir da identificação e análise de evidências e rastros deixados no ambiente digital. A volatilidade deste cenário, que trabalha não apenas profundamente associado ao uso de computadores e notebooks, como também com dispositivos móveis e embarcados, torna-se uma problemática centralizadora na garantia da veracidade de identidade de um equipamento envolvido em uma cadeia de custódia.

Nesse sentido, de acordo com o Art. 158-A da Lei nº 13.964, de 24 de dezembro de 2019, define-se cadeia de custódia como a junção de todos os processos necessários para garantir a história cronológica de um vestígio coletado em locais ou em vítimas de crimes, de forma que seja possível rastrear e entender a história do vestígio desde o seu reconhecimento até o seu descarte. Além disso, por meio de uma cadeia de custódia, é possível garantir que as evidências estejam protegidas contra violações ou adulterações durante um processo criminal.

Mesmo fora de uma cadeia de custódia, a contínua expansão dos dispositivos comuns ligados à rede através do paradigma da *Internet* das Coisas expressa a complexidade inerente a este tipo de sistema. Sua composição, formada por inúmeros nodos descentralizados comunicando-se por meio dos protocolos da *Internet* (FILHO, 2016), traz à tona grande volatilidade e flexibilidade, características problemáticas quando existe a necessidade de identificação segura e universal.

Em uma cadeia de custódia, então, não apenas a integridade dos dados é considerada uma propriedade crítica essencial ao funcionamento do sistema. O dispositivo no qual as evidências se encontram também é um ponto vulnerável neste cenário, em que existe grande apelo à necessidade de garantir que o objeto seja autêntico e disponível. Desta forma, identificar um equipamento para preservá-lo, torna-se um desafio a ser superado. No entanto, quando trata-se de dispositivos inseridos no contexto da *Internet* das Coisas, os gerenciadores de identidade comuns podem não ser a melhor alternativa para solucionar este problema (LIU et al., 2012). Isso porque este tipo de equipamento é dotado de baixa capacidade computacional, bem como pouca memória de processamento.

Muitas vezes, alguns equipamentos inseridos em uma cadeia de custódia são dispositivos embarcados. Desta maneira, é necessário que a gerência de identidade seja feita em baixo custo computacional e de maneira, preferencialmente, distribuída. Neste contexto, identificadores descentralizados, também conhecidos pelo acrônimo DID (*Decentralized Identifiers*), tornam-se úteis para garantir a identidade de um usuário ou dispositivo sem que entidades centralizadoras sejam necessárias. Seu uso torna-se interessante quando existem múltiplas partes que colaboram entre si para realizar a validação de uma identidade descentralizada.

De modo geral, uma solução distribuída tende a ser mais flexível e tolerante a falhas. E neste caso, torna-se possível utilizar construções de *blockchain* para implementar uma identidade descentralizada no contexto de validação e verificação de dispositivos em uma cadeia de custódia. O protocolo da confiança, como também é chamada a *blockchain*, permite, através de sua arquitetura, que toda e qualquer modificação feita em uma evidência inserida no sistema seja facilmente detectada pelo seu esquema de encadeamento. Para isso, a proposta deste trabalho tem como objetivo construir um sistema de identidade descentralizada, baseada em *blockchain*, para identificar e validar de forma segura todo dispositivo envolvido em uma cadeia de custódia no ambiente de computação forense.

1.1. Motivação

A identidade descentralizada desempenha um papel fundamental na Internet das Coisas (IoT), pois oferece uma solução eficiente para garantir a autenticidade e a segurança dos dispositivos e usuários conectados. Ao utilizar identificadores descentralizados únicos (DIDs) e tecnologias como *blockchain*, é possível estabelecer uma infraestrutura confiável em que cada dispositivo tenha sua própria identidade verificável e segura. Isso permite que os dispositivos IoT se comuniquem de maneira confiável e autenticada, mitigando riscos de ataques cibernéticos, falsificação de dados e comprometimento da integridade do sistema. Além disso, a identidade descentralizada facilita a interoperabilidade entre diferentes dispositivos e plataformas, promovendo um ecossistema mais aberto e colaborativo na Internet das Coisas.

1.2. Objetivos

1.2.1. Objetivo Geral

O objetivo geral deste trabalho é definido sobre a necessidade de desenvolver um sistema que possa identificar com segurança e assertividade a identidade de dispositivos embarcados envolvidos em uma cadeia de custódia no contexto da computação forense. Neste sentido, o propósito final é construir uma identidade descentralizada para estes dispositivos, e para tanto, considera-se o uso de tecnologias distribuídas que dispensam o uso de validações de entidades centralizadoras, como construções de *blockchain*, para garantir a preservação da informação mesmo na ausência de entidades centralizadoras.

1.2.2. Objetivos Específicos

A seguir, estão listados os objetivos específicos deste trabalho:

- Conhecer e entender o funcionamento dos dispositivos inseridos no paradigma da *Internet* das Coisas;
- Conhecer e entender o funcionamento das tecnologias de registro distribuído implementadas através da *blockchain*;
- Estruturar, de forma sistemática, o estado da arte para as tecnologias envolvidas na identificação e gerenciamento de dispositivos embarcados no contexto da computação forense;
- Compreender a implementação e uso de Identificadores Descentralizado; e,
- Implementar um sistema de gerenciamento de identidade para dispositivos embarcados baseado em uma construção de *blockchain*.

1.3. Organização

O presente trabalho está estruturado em 5 capítulos. No capítulo de número 1 estão descritos a Introdução, a Motivação e os Objetivos, necessários para compreensão do conteúdo apresentado neste trabalho. No capítulo de número 2, está a Fundamentação Teórica, que apresenta os conceitos utilizados como base para o desenvolvimento de uma proposta de solução. No capítulo 3, estão os Trabalhos Correlatos, em que apresenta-se uma Revisão Sistemática Bibliométrica e a análise de três artigos. No capítulo de número 4, está descrito o

processo de desenvolvimento do Sistema de Manutenção de Identidade Descentralizada. No capítulo 5, estão as Considerações Finais, uma síntese sobre os resultados das pesquisas feitas.

2. Fundamentação Teórica

2.1. IoT (*Internet of Things*)

A colaboração e o avanço de múltiplas áreas da ciência, como a microeletrônica e a computação móvel e embarcada, viabilizou o surgimento de novos paradigmas tecnológicos. A *Internet* das Coisas, comumente reduzida à sigla IoT, é um modelo computacional que se refere não a algo concreto ou único, mas sim, a uma malha de dispositivos conectados, capazes de transformar a vida das pessoas sobre a ótica da globalização em um sentido de praticidade e conexão.

2.1.1. Contexto Histórico

A interação do homem com um computador, historicamente, é profundamente estática e marcada por características como a rigidez e a pouca mobilidade (CAMPBELL-KELLY et al., 2014). Entretanto, a portabilidade consequente de unidades de processamento menores e mais poderosas (CASTELLS, 2011) criou um mundo onde objetos tornam-se peça fundamental na concepção de um espaço completamente conectado sobre os protocolos de comunicação da *Internet*.

Já em 1926, Nikola Tesla, em entrevista à revista Colliers, previu a existência de um mundo no qual as fronteiras de conexão entre os seres humanos através de dispositivos eletrônicos não ficariam restritas à tela de um computador. O avanço contínuo e exponencial das técnicas comunicação e sensoriamento permitiu com que um ecossistema peculiar surja e estabeleça-se como uma extensão da *Internet* comum. John Romkey, cientista responsável por criar o primeiro dispositivo comum acionável via *Internet* em 1990, afirmou no ano de 2017: "Anteriormente, o mundo tinha uma *Internet*. Entretanto, era uma *Internet* sem coisas" (ROMKEY, 2017).

A *Internet* das Coisas foi um termo primeiramente utilizado pelo pesquisador do MIT Kevin Ashton, que sugere o uso de etiquetas RFID para rastreamento de produtos da corporação multinacional americana de bens de consumo P&G, Procter & Gamble. Em um cenário onde IoT caracteriza-se por, antes de tudo, compreender a junção de inúmeros objetos interligados entre si sobre um protocolo de telecomunicação comum (BASSI & HORN, 2008), Ashton

busca inserir este novo ecossistema como um fragmento particular em uma sociedade que em si e em sua economia, baseia-se em "coisas". Entretanto, "a tecnologia da informação de hoje é tão dependente dos dados originados pelas pessoas que nossos computadores sabem mais sobre ideias do que coisas" (ASHTON, 2011).

A ampliação dos espaços limítrofes da *Internet* permite com que objetos antes inanimados possuam maior interação com os ecossistemas humanos. Dispositivos embarcados tornam-se cada vez mais populares à medida que as necessidades impostas pela globalização exigem conectividade constante e contínua.

> A conexão com a rede mundial de computadores viabilizará, primeiro, controlar remotamente os objetos e, segundo, permitir que os próprios objetos sejam acessados como provedores de serviços. Estas novas habilidades, dos objetos comuns, geram um grande número de oportunidades tanto no âmbito acadêmico quanto no industrial (SANTOS et al., 2016).

Desde 1950, o matemático e cientista Alan Turing, pai da computação moderna, defendia a possibilidade da existência de uma inteligência artificial. Para ele, uma máquina seria perfeitamente capaz de competir com um ser humano em escala intelectual, desde que ela possuísse os melhores órgãos sensoriais e fosse dotada da capacidade de compreender e comunicar-se por meio de uma linguagem natural. A perspectiva levantada por Turing em meados do século passado, hoje, converge para a hipótese de que um objeto operando em um meio, comunicando-se com o exterior e dando os melhores resultados mediante uma pequena capacidade de processamento transforma a realidade a sua volta, construindo um ecossistema inteligente que facilita os seus processos em si mesmo.

Para (EASTERLING, 2012), a *Internet* das Coisas é um conjunto de “dispositivos digitais, em que o espaço entre eles não consiste em circuitos obscuros, mas sim o espaço da própria cidade”. Ainda segundo o mesmo autor, “é como se o computador estivesse extrapolando os seus próprios limites, e incorporando-se a tudo aquilo que julgamos comum.” Por este motivo, em (KUROSE & ROSS, 2012), surge um debate cujo tema permanece centralizado no uso do termo "Redes de Computadores". Os autores julgam que esta expressão começa a soar antiquada frente aos novos equipamentos e seus respectivos usos não tradicionais, associados à *Internet.* Os recursos, funcionalidades e informações requeridos, controlados ou providos por estes equipamentos implicam na existência de novos usos e aplicações, sobre os quais, de forma concomitante, emergem novas características, riscos e vulnerabilidades.

2.1.2. Arquitetura

No *background* do cenário da *Internet* das Coisas, segundo (NOOR & HASSAN, 2019), o ecossistema de *Internet* das Coisas é formado, fundamentalmente, por uma arquitetura de três camadas. A primeira delas é a camada de *hardware,* onde estão os sensores responsáveis por coletar e armazenar de forma temporária as informações do ambiente. A segunda camada é a de rede, por meio da qual os dados são comunicados. Finalmente, existe a camada de aplicação ou serviço, que é responsável pelo processamento de todo o conteúdo recebido a fim de gerar informação útil. A figura 1 representa a estrutura em camadas da *Internet* das Coisas.

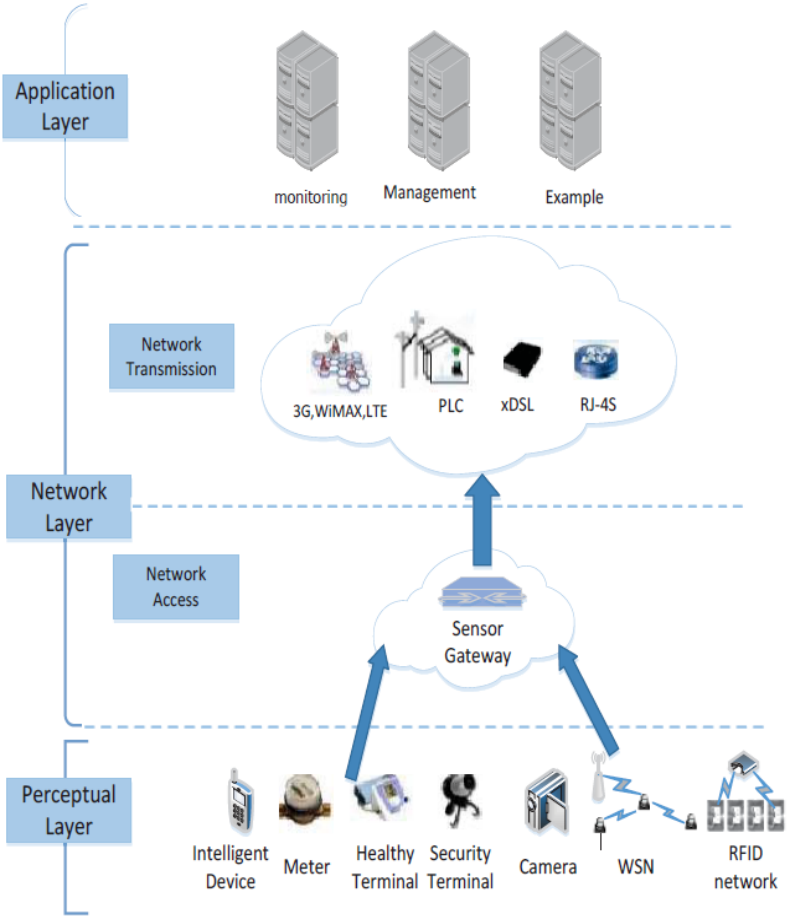

Figura 1 – Composição da arquitetura de uma rede IoT (HAI et al., 2014).

Hoje em dia, incontáveis dispositivos podem ser conectados à *Internet* de modo a gerar dados e comunicar informações. Estes equipamentos variam suas aplicações desde ambientes

domésticos, como televisores, máquinas de lavar roupa, lâmpadas e geladeiras, até usos industriais, em controladores e atuadores. Independente de sua serventia, entretanto, a aplicabilidade deste ecossistema diz respeito "a proposta de um sistema global de fácil identificação em que equipamentos poderiam ser interconectados através da *Internet*, [...] trabalhando de forma eficiente e produtiva" (GODOI & ARAÚJO, 2019).

Desta forma, segundo (MIRANDA et al., 2015), o termo *Internet* das Coisas torna-se incompleto ou ainda insuficiente para referenciar a ampla gama de aplicações disponíveis ao benefício dos seres humanos. Enquanto os dados coletados e produzidos pelos dispositivos embarcados não forem devidamente processados pelas chamadas "aplicações de consumo", normalmente funcionais sobre o paradigma de computação em névoa e em nuvem, eles não formam o arranjo de informações necessárias para compor uma *Internet* das pessoas. De acordo com (CARION & QUARESMA, 2019), "é nesse encontro de *coisas* e *pessoas* que se tem a maioria das aplicações, serviços e produtos de IoT para consumo."

2.1.3. Características

A heterogeneidade, que é uma das características principais de um ambiente de *Internet* das Coisas, e a contínua expansão dos dispositivos comuns ligados à rede através deste paradigma expressam a complexidade inerente a este tipo de sistema. Este atributo surge, principalmente, da grande multiplicidade decorrente das tecnologias de *software* e *hardware* envolvidas na produção desse ecossistema (DOMINGUESCHE, 2021). Sua composição, formada por inúmeros nodos descentralizados comunicando-se por meio dos protocolos da *Internet*, traz à tona grande volatilidade e flexibilidade, características problemáticas quando existe a necessidade de garantir uma identificação segura e universal. Garantir a integração contínua, bem como a escalabilidade e a disponibilidade também são desafios frente ao grande volume de dados que circula entre os dispositivos embarcados em um contexto de *Internet* das Coisas (DOMINGUESCHE, 2021).

A base de um sistema de *Internet* das Coisas é a comunicação entre os dispositivos que o integram. Partindo deste pressuposto, é necessário que esta característica seja garantida mesmo frente às adversidades inerentes deste ambiente, com grande destaque à heterogeneidade natural deste tipo de ecossistema, e os desafios que dela surgem (PARK et

al., 2016). Neste contexto, é interessante salientar enquanto atributo bastante particular a necessidade de processar grandes quantidades de dados em um sentido de *upload.* É imprescindível que a existência desta grande demanda pelo envio de dados - estes, coletados pelos sensores sobre o ambiente - recebam o tratamento adequado de forma a gerar informação. Este comportamento, por sua vez, vai na contramão das aplicações tradicionais da *Internet,* nas quais existe um grande apelo ao *download* de dados em detrimento das operações de *upload* (PARK et al., 2016).

2.1.4. Aplicações

De acordo com (SRINIVASA, 2020), "a *Internet* das Coisas não poderá mais ser vista enquanto um composto de sistemas individuais, mas sim como uma infraestrutura integrada e crítica sobre a qual aplicações e serviços são executados". É natural a inferência de que tamanha criticidade represente o resultado de inúmeros dispositivos, protocolos, padrões e abordagens unidos em plataformas que embora possuam o mesmo fim em si mesmas, sejam tão diferentes. É possível citar como possíveis aplicações finais deste paradigma abordagens de cidades, sistemas de cuidado com a saúde e agricultura inteligentes, bem como usos para administração de recursos naturais renováveis e não renováveis e gestão de redes de varejo e logística.

> A *Internet* das Coisas é a rede de objetos físicos ou "coisas" incorporadas aos conceitos de eletrônica, software, sensores e conectividade de forma a permitir que ela alcance maior valor e utilidade ao trocar dados com o fabricante, operador ou até mesmo outros dispositivos conectados. Neste contexto, cada "coisa" é exclusivamente identificada como objeto conectado por meio de seu sistema de computação embarcado, enquanto é capaz de interoperar dentro da infraestrutura de *Internet* existente (SRINIVASA, 2020).

O avanço das tecnologias envolvidas na produção deste ecossistema particular aumenta a sofisticação necessária no estabelecimento de conexão entre dispositivos. De forma paralela, existe também visível incremento na qualidade dos serviços prestados, ao custo de grandes desafios na integração segura entre todos os nodos de uma rede. Neste contexto, e compreendendo enquanto fato comum a existência destes dispositivos no dia a dia das pessoas, é necessário determinar os riscos gerados pela comunicação ininterrupta entre o usuário e estes equipamentos, nos quais "uma enorme quantidade de dados gerados são coletados sem seguir um padrão ou metodologia" (OLIVEIRA et al., 2019).

2.1.5. Desafios

Compreender o conceito da *Internet* das Coisas enquanto uma extensão da *Internet* comum tem por consequência direta a necessidade de assumir os novos desafios - que por sua vez, não podem receber o mesmo tratamento daqueles já conhecidos sobre a ótica da *Internet* convencional - em uma perspectiva também bastante voltada à segurança da informação, em busca de disponibilidade, confidencialidade e integridade (OLIVEIRA et al., 2019). Esta observação causa grande impacto na forma como os esforços atuais focados no desenvolvimento do paradigma de IoT são estruturados, uma vez que eles devem ser levados de forma paralela às características inerentes aos dispositivos embarcados - baixa capacidade de processamento e disponibilidade de energia somados à pouca quantidade de memória.

> Com a IoT, as ameaças à segurança vão muito além do roubo de informações ou da impossibilidade de uso de determinados serviços. Essas ameaças podem agora estar potencialmente relacionadas com as vidas reais, inclusive de segurança física. Soluções de segurança e privacidade devem ser implementadas conforme as características de dispositivos IoT heterogêneos (CHICARINO et al., 2017).

Em decorrência da popularização dos equipamentos embarcados em propósitos individuais e pessoais, como os dispositivos *wearable*, e em sua aplicação em usos relacionados ao *health care* e *smart homes,* surge um grande leque de possibilidades alimentadas, em seu cerne, pelos dados produzidos pelos usuários. Neste cenário, discussões a respeito dos aspectos de privacidade e padronização de métodos e protocolos ganham destaque sobre a ótica de situações em que o vazamento de dados em um ambiente virtual pode acarretar consequências catastróficas na vida das pessoas no mundo real.

A segurança é um tópico fundamental a ser desenvolvido em um contexto de *Internet* das Coisas, de forma a garantir que todas as partes que compõem esse sistema permaneçam íntegras e disponíveis. Idealmente, a construção de um ecossistema seguro inicia-se logo na primeira camada de um sistema de IoT, com o uso de *hardware* projetado com maior sofisticação e atenção nos processos de escolha de mecanismos criptográficos e de segurança (NOOR, & HASSAN, 2019). Entretanto, sobressaindo a escolha do custo sobre o benefício, observa-se comumente uma vã tentativa de aplicar medidas de segurança da *Internet*

tradicional em um ambiente onde estas podem não ser suficientes. De acordo com (NOOR, & HASSAN, 2019):

> Os ataques listados pela Open Web Application Security Project (OWASP) atingem as três camadas de um sistema IoT, que são *hardware*, rede e serviços. Portanto, a implementação de medidas cujo objetivo é mitigar as consequências causadas pelas vulnerabilidades em IoT devem abranger a arquitetura em todas as camadas [...].

Neste cenário, surgem desafios no âmbito de garantir a autenticidade da informação em um ambiente extremamente volátil, quiçá até mesmo, altamente modificável. Essas características são bastante problemáticas quando existe a necessidade de garantir uma identificação segura e universal. Em uma cadeia de custódia - que se refere, segundo o artigo 158-A do Código de Processo Penal, aos "procedimentos utilizados para manter e documentar a história cronológica do vestígio coletado em locais ou em vítimas de crimes, para rastrear sua posse e manuseio a partir de seu reconhecimento até o descarte" -, por exemplo, não apenas a integridade dos dados é considerada uma propriedade crítica essencial ao funcionamento do sistema. O dispositivo no qual as evidências se encontram também é um ponto vulnerável neste cenário, onde existe grande apelo à necessidade de garantir que o objeto seja autêntico e disponível. Desta forma, identificar um equipamento para preservá-lo, torna-se um desafio a ser superado.

2.1.6. Correlações

Quando se estabelece uma cadeia de custódia sobre um ambiente de *Internet* das Coisas, é necessário que a gerência de identidade - mecanismo por meio da qual a autenticidade de um dispositivo pode ser garantida - seja feita de forma leve e distribuída. Esta abordagem, por sua vez, leva em consideração todas as características anteriormente citadas que são inerentes à natureza destes sistemas, construindo uma solução que tende a ser mais flexível e tolerante a falhas.

Neste sentido, segundo (KAMAL et al., 2022), para que um dispositivo embarcado seja aceito enquanto elemento confiável em uma cadeia de custódia, a evidência precisa, além de ser relacionada ao crime e investigada por pessoas de devida competência, manter sua integridade frente a modificações, questões de transparência, confiabilidade e imutabilidade. Ainda de

acordo com os mesmos autores, "[...] qualquer alteração nos dados pode levar a um resultado de investigação falso" (KAMAL et al., 2022).

2.2. *Blockchain*

O conceito central de *blockchain* é descrito, em muitas referências, como uma abordagem inovadora no cenário da manutenção de informação a longo prazo. No ano de 2008, Satoshi Nakamoto propôs uma solução em que técnicas de criptografia são utilizadas junto a uma abstração de um livro-registro distribuído e de código aberto para criar moedas digitais (XU et al., 2019).

A característica chave de uma rede *blockchain*, é a descentralização da estrutura do sistema (SWAN, 2015). A imutabilidade e a grande tolerância à falhas também são atributos destaque, que garantem uma solução flexível e altamente disponível, em que um ecossistema inteiro estará protegido do exterior por meio de medidas eficazes de controle de acesso, que permitirão a validação dos dados por meio dos usuários individuais da rede.

A tecnologia construída por Nakamoto é o que sustenta o crescente conjunto de moedas digitais da atualidade, como o *bitcoin* e o *ethereum*, dois exemplos bastante competitivos no cenário. Entretanto, o protocolo da confiança, como também é chamada a *blockchain*, é uma estrutura cujas aplicações multiplicam-se para além do cenário de criptomoedas.

Desviar a zona limítrofe da confiança de fora para dentro do sistema é o que o manterá seguro, em uma dinâmica onde nenhum dos nodos necessariamente precisará confiar no outro e nem em uma entidade terceira centralizadora, construindo redes *peer to peer* multipropósito.

2.2.1. Arquitetura

Definida enquanto um banco de dados distribuído, conforme descrito na figura 2, sobre o qual armazena-se, de forma encadeada, informações sobre as transações realizadas na rede, a *blockchain* garante a imutabilidade da informação por meio da concatenação - em cada bloco da estrutura - do *hash* da informação contida no bloco imediatamente anterior (SCHMITT & STEIL, 2021).

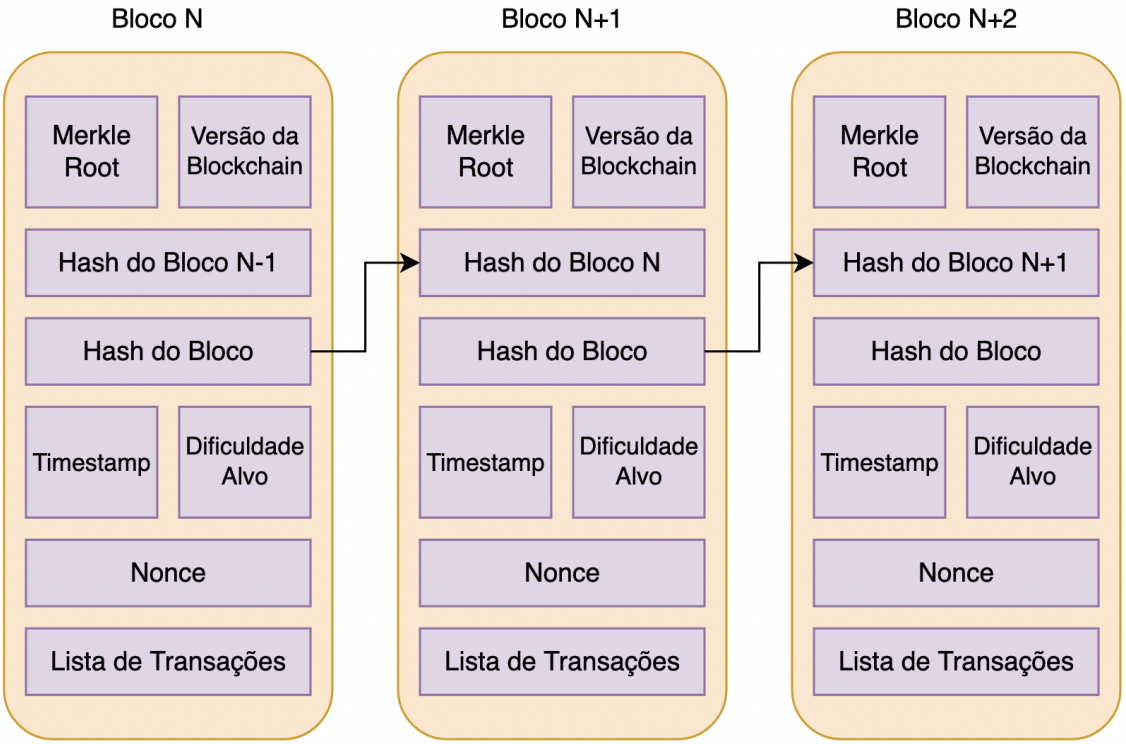

Figura 2 - Modelo de encadeamento dos blocos de uma *blockchain,* adaptado de (KHAN & SALAH, 2018).

Nos modelos mais tradicionais de *blockchain*, um bloco é estruturado em duas partes principais, cabeçalho e corpo, também chamado de cauda. O cabeçalho contém inúmeras informações utilizadas para construir uma estrutura rastreável e imutável, como a versão do *software* da *blockchain* e o campo *Merkle Root*, que "indica o nodo raiz de uma estrutura chamada *Merkle Tree,* utilizada para organizar os blocos da cadeia" (RIBEIRO & MENDIZABAL, 2019). O *nonce,* por sua vez, é um número aleatório arbitrário utilizado para garantir que o *hash* gerado para o bloco está dentro da dificuldade determinada pela rede.

Um *hash*, de forma sucinta, pode ser definido como o resumo criptográfico de um conjunto de *bytes* de tamanho variável. Esta estrutura de dados é obtida por meio de funções criptográficas irreversíveis, que garantem que a informação de saída tenha sempre um comprimento fixo. Portanto, é possível observar a propriedade de imutabilidade neste cenário, uma vez que uma função de *hash* produz um par de resultados iguais entre si para dois conjuntos de *bytes* igualmente equivalentes. Desta forma, se qualquer informação sobre uma transação for alterada dentro da estrutura encadeada, é precisamente viável detectar tal violação a partir do cálculo do *hash* do bloco adulterado, que deve ser exatamente igual ao contido no nodo

seguinte. Em uma estrutura de *blockchain*, modificar as informações de um bloco provoca o colapso da estrutura como um todo. (DI PIERRO, 2017).

Em sistemas de informação comuns, a autenticidade e a não-repudiação podem ser facilmente fornecidas através de uma autoridade confiável única, sobre a qual são centralizadas e validadas todas as transações do sistema. Entretanto, segundo (DI PIERRO, 2017), "[...] uma entidade centralizadora nem sempre é uma solução prática, pois não escala com o alto número de transações frequentes e porque requer que todos os integrantes do sistema confiem no mantenedor da entidade". A *blockchain*, portanto, encarrega-se de construir um ambiente onde os múltiplos nodos de uma rede são igualmente encarregados de registrar e validar uma transação, que é livremente auditável por qualquer participante.

2.2.2. Aplicações

Estruturas de *blockchain* podem ser utilizadas para múltiplas aplicações que não estão necessariamente relacionadas ao cenário de criptoativos e sistemas de pagamento, referência mais evidente de uso e aplicação da tecnologia de *blockchain.* O protocolo da confiança pode ser utilizado para construir sistemas de informação seguros e praticamente imutáveis, com alta resistência à corrupção e substituição de dados. Neste sentido, torna-se possível proteger informações sensíveis, assinar, autenticar e validar qualquer tipo de documento digital e prover rastreabilidade e transparência a um dado durante todo o seu ciclo de vida.

Enquanto uma tecnologia extremamente promissora, o uso da *blockchain* aplicado à manutenção de processos eleitorais, por exemplo, vêm sendo estudado para construir votações digitais transparentes e flexíveis (HJÁLMARSSON et al., 2018). Implementar sistemas eleitorais dentro destes termos é um grande desafio, e embora existam países que amplamente utilizem urnas eletrônicas de forma segura, como é o caso do Brasil e do Canadá, segundo o artigo Hard and Soft Forks, o uso de uma rede distribuída garantiria com maior eficiência que cada voto contabilizado no processo é único e autêntico.

Estratégias baseadas em *blockchain* também têm sido muito utilizadas dentro da área de saúde, em virtude da necessidade de gerenciar dados confidenciais com privacidade e transparência. Estas características fomentam a construção de um ambiente em que o risco de

ataques cujo objetivo é o vazamento de dados é bastante alto. Para garantir a segurança neste tipo de ecossistema, é possível fazer uso de estruturas de *blockchain* privadas, com foco na imutabilidade e rastreabilidade dos dados a longo prazo (DE AGUIAR et al., 2021). No entanto, questões relacionadas à privacidade da identidade não são completamente solucionadas em soluções tradicionais de *blockchain.* Inúmeras técnicas citadas pela literatura podem ser utilizadas para resolver esta questão em particular, como, por exemplo, a técnica de *zero-knowledge proof,* examinada em (SWEENEY, 2002).

Criptoativos, no entanto, correspondem ao exemplo mais tradicional de aplicação de uma estrutura de *blockchain*. Representados por meio de ativos digitais diversos, como criptomoedas, *tokens* fungíveis e não-fungíveis e protocolos de finanças descentralizadas (DeFi), estão envolvidos no contexto distribuído sobre as características fundamentais de uma *blockchain.*

2.2.3. Processo de Mineração

No cenário de criptoativos, o processo de mineração - sinônimo do conceito de *Proof Of Work* - representa a validação de uma transação e consequente adição de um novo bloco à *blockchain.* Este é o resultado do esforço concentrado na resolução de um problema matemático pré-definido e de livre conhecimento de todos os integrantes da rede na qual o processo ocorre.

Segundo (MOREIRA, 2019), "um bloco é considerado minerado todas as vezes em que o minerador encontrar um *nonce* que faça com que o hash esteja abaixo da dificuldade da rede, tendo assim uma certa quantidade de 0 no valor inicial do *hash*".

A maior parte das *blockchains* tradicionais implementa o modelo de prova de trabalho no qual entende-se que uma transação será registrada em um livro-razão somente quando algum usuário solucionar uma complexa equação matemático-computacional (YAGA et al., 2018). O resultado deste problema é a prova de que o usuário de fato cedeu seu poder de processamento, e tem direito legítimo à inserir um novo bloco na estrutura da *blockchain.*

Alterar um único *bit* do bloco minerado N resulta, consequentemente, na alteração do seu valor de *hash.* Desta forma, inicia-se um efeito em cascata em que o próximo bloco (N+1) e

todos os subsequentes tornam-se inválidos dentro da estrutura de *blockchain*, pois o *hash* utilizado na construção do bloco não condiz com o *hash* utilizado para validá-lo.

Para que um bloco possa ser inserido na rede, ele precisa possuir um resumo criptográfico de seu conteúdo. O cálculo de um *hash* dentro de estruturas de *blockchain* é uma atividade complexa, que exige alto poder computacional. Isto ocorre porque antes de tudo, é necessário encontrar um valor de *nonce* arbitrário e aleatório que produza um *hash* que possua uma quantidade de *bits* determinística, expressa em 0s (RIBEIRO & MENDIZABAL, 2019). Segundo os mesmos autores:

> Nonce é um campo de 32 bits que pode assumir qualquer valor, fazendo com que sejam necessárias muitas tentativas até que o número n de *bits* 0 seja alcançado, garantindo que grande esforço computacional foi empregado na solução do problema [...] (RIBEIRO & MENDIZABAL, 2019).

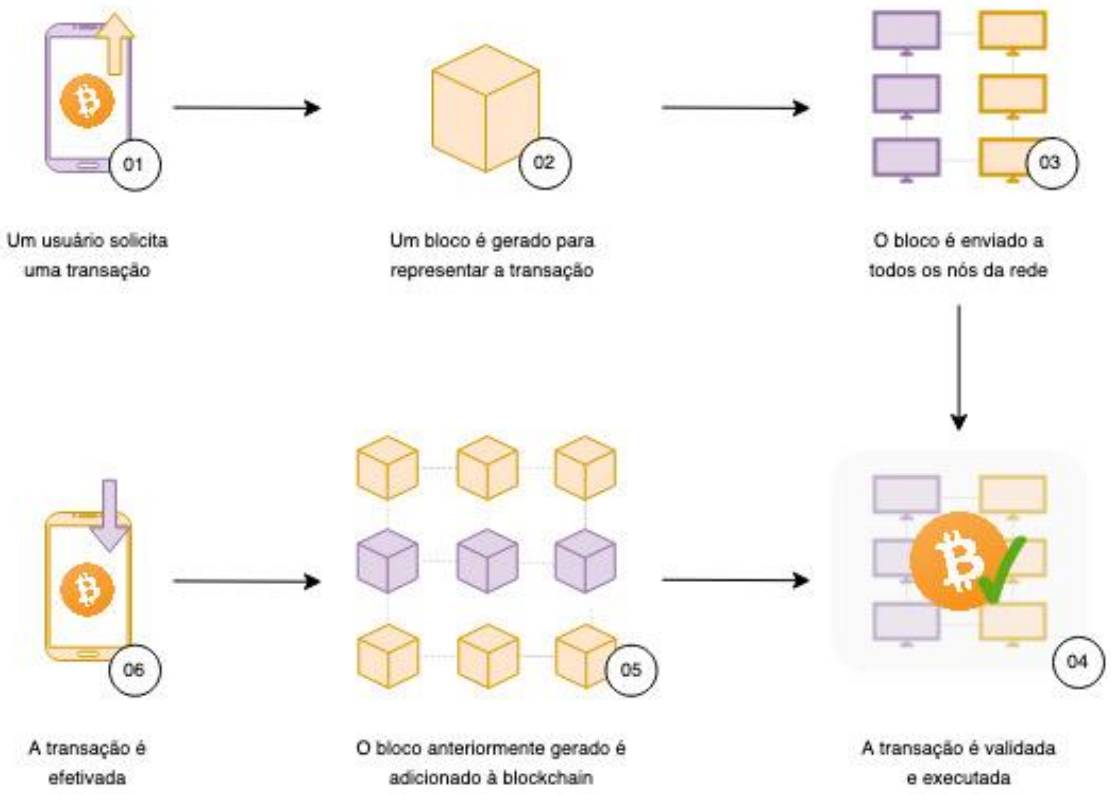

Figura 3 - Processo de mineração na *blockchain,* adaptado de (LAMOUNIER, 2018)

A figura de número 3 destaca o processo de mineração, que garante que um bloco estará válido dentro da estrutura distribuída quando "o minerador encontrar um *nonce* que faça com

que o *hash* esteja abaixo da dificuldade estabelecida pela rede" (MOREIRA, 2019). Em (YAGA et al., 2018), os autores destacam que este cálculo é estabelecido em uma estrutura em que a sua resolução seja complexa, mas a validação do resultado obtido, não. Desta forma, todo e qualquer nodo pode validar e confirmar os blocos candidatos à inserção, onde transações inválidas se não puderem ser comprovadas.

Em *blockchains* voltadas ao cenário de criptoativos, o processo de mineração retorna ao minerador que primeiro conseguir solucionar a equação uma recompensa, geralmente expressa em unidades destes ativos. Neste sentido, um *bitcoin* ou um *ethereum* são, antes de tudo, números, registrados e transacionados em livros-razão digitais com as credenciais de quem se identifica enquanto dono daquela unidade (DI PIERRO, 2017).

2.2.4. Mecanismos de Consenso

Em sistemas distribuídos, mecanismos de consenso são amplamente difundidos enquanto formas de fazer com que um conjunto de processos que são executados de forma independente concordem sobre um valor proposto (RIBEIRO & MENDIZABAL, 2019). Em uma estrutura de *blockchain*, cada nodo da rede possui, de forma independente, uma cópia do estado da rede em termos dos blocos que a compõem em um determinado momento do seu ciclo de vida. O processo de validação para inserção de novos blocos possui caráter assíncrono, e por consequência da ausência de "coordenação central, cada nó pode ter uma visão diferente do estado do *blockchain* em um dado instante" (MIERS et al., 2019).

Segundo os mesmos autores:

> Para garantir a convergência dessas visões em um espaço de tempo não muito longo, a *blockchain* utiliza um mecanismo de consenso. Isso é essencial para criar um sistema consistente, no qual todos os nós concordem com a ordem dos blocos e sobre os seus conteúdos (MIERS et al., 2019).

Neste sentido, em uma rede distribuída como a *blockchain,* que é estruturada sobre uma arquitetura de blocos encadeados, o conceito de algoritmo de inserção de blocos serve para garantir uma decisão única sobre sobre qual bloco deve ser adicionado à cauda da *blockchain*, em um ambiente descentralizado.

Em redes de *blockchain* públicas, como é o caso das estruturas utilizadas por criptomoedas como o *Bitcoin* e o *Ethereum*, normalmente o mecanismo de consenso utilizado é o de prova de trabalho (*PoW*). Neste protocolo, qualquer nodo da rede *P2P* pode participar do processo de inserção e validação de novos blocos (RIBEIRO & MENDIZABAL, 2019). É um modelo que busca equilibrar a dificuldade do critério de aceitação de novos blocos, de forma que o número de blocos minerados permaneça em um nível aceitável no decorrer do tempo.

Outra abordagem utilizada para obter consenso em sistemas de *blockchain* é aquela em que os nodos que estão aptos a participar do processo de inserção e validação de blocos são pré-definidos pelo protocolo (RIBEIRO & MENDIZABAL, 2019). Esta estratégia corresponde aos modelos de consenso mais tradicionais, e tende a ser utilizada principalmente em estruturas de *blockchain* privadas, que trabalham com conjuntos de nodos menores. Um exemplo desta abordagem é o Practical Byzantine Fault Tolerance, algoritmo de replicação que permite qualquer sistema distribuído lidar com falhas bizantinas (MIERS et al., 2019).

Segundo os mesmos autores:

> Os nós se organizam para operar em rodadas, de modo que em cada rodada um nó primário é selecionado de acordo com certas regras. O nó primário fica então responsável por inserir o próximo bloco na cadeia. O processo é dividido em três fases: Pré-Preparado, Preparado e Comprometido. Para passar de uma fase a outra, um nó precisa receber o voto de ⅔ de todos os nós (MIERS et al., 2019).

No entanto, para que seja possível criar um ecossistema favorável a esta dinâmica, é fundamental que todos os nodos participantes da rede sejam conhecidos. Neste cenário em particular, aspectos computacionais não são componentes necessários para o estabelecimento de consenso dentro da *blockchain*.

2.2.5. Imutabilidade e Bifurcações

Embora a imutabilidade seja um dos pilares sobre o qual uma rede de *blockchain* está estruturada, o conceito aplicado a este cenário pode ser, em algumas vias, ambíguo. Segundo (YAGA et al., 2018), *blockchain ledgers* são resistentes a alterações e invioláveis, mas não podem ser considerados completamente imutáveis. Isso porque os blocos da cauda de uma estrutura distribuída de *blockchain* podem ser facilmente substituídos na existência de conflito

entre os resultados produzidos de forma simultânea por dois nós validadores diferentes, onde ambos são encadeados para o mesmo bloco imediatamente anterior.

Uma bifurcação pode ocorrer de forma ocasional dentro de uma *blockchain,* quando dois nós mineram, simultaneamente, o mesmo bloco. Nestes cenários, por um curto espaço de tempo, a estrutura da rede ficará dividida em dois estados distintos. Sempre que uma bifurcação ocorre, ela deve ser automaticamente resolvida pelo *software* que implementa a *blockchain,* de forma a normalizar o histórico transacional do sistema.

Entretanto, como o ecossistema está implementado sobre uma estrutura distribuída, o *ledger* central não poderá obrigar nenhum nó a seguir uma verdade absoluta, onde diferentes versões da mesma *blockchain* coexistem em paralelo. De forma geral, alcançar tal violação é ou improvável, ou extremamente difícil (HOFMANN et al., 2017).

Para além das bifurcações acidentais que ocorrem neste tipo de rede distribuída, a ocorrência de *bugs* no código fonte que implementa a estrutura da *blockchain* também podem ocasionar problemas para garantir um sistema em estado conciso. Alterações no conjunto de regras utilizado para validar novas entradas, sejam elas propositais ou acidentais, pode provocar dois tipos diferentes de bifurcações em uma *blockchain*, *soft forks* e *hard forks.*

Em um *soft fork,* descrito na figura 4, as alterações nas regras utilizadas para validação de blocos são retrocompatíveis. Desta forma, os novos registros que são inseridos na *blockchain* continuam sendo válidos na estrutura inicial desta. Por outro lado, em um *hard fork,* descrito na figura 5, isto não acontece. Nestes casos, os blocos criados de acordo com as novas regras não podem ser validados pela regra antiga, e duas estruturas paralelas de *blockchain* passam a coexistir. A promoção de cenários como este pode provocar violação na propriedade de imutabilidade da rede, bem como minar a legitimidade da *blockchain*.

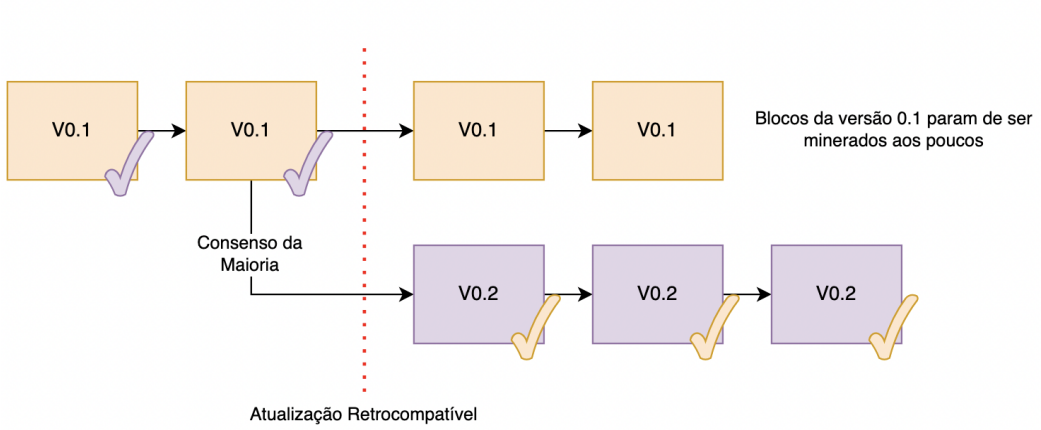

Figura 4 - *Soft fork,* adaptado do artigo Hard and Soft Forks.

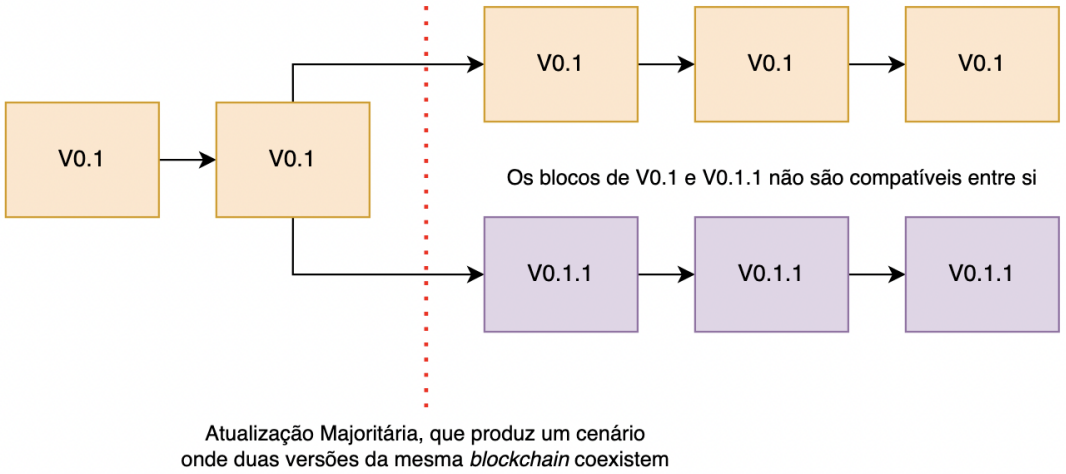

Figura 5 - *Hard fork,* adaptado do artigo Hard and Soft Forks.

2.2.6. Smart Contracts

O conceito de *smart contracts* está normalmente associado de forma paralela ou extensível às aplicações de *blockchain*. Primeiramente citado em 1994 por Nick Szabo enquanto "um protocolo de transação computacional executado nos termos de um contrato". Seu objetivo geral é minimizar acidentes em termos de segurança, para que múltiplos participantes de um sistema não permaneçam presos na ideia ou necessidade de uma entidade terceira de igual confiança.

Smart contracts podem ser encarados enquanto um conjunto que une código e dados a ser implantado em uma rede *blockchain* usando criptografia assimétrica (YAGA et al., 2018). São

auto-executáveis dentro do cumprimento de condições pré-acordadas, e inseridos em um ambiente digital, não podem ser perdidos ou violados. Os usuários dentro de uma rede distribuída como a *blockchain* podem realizar transações sobre seus ativos digitais através de funções públicas, disponíveis através de *smart contracts.* O contrato inteligente fica responsável por executar de forma apropriada uma operação pré-determinada sobre um conjunto de dados que encontra-se em um estado inviolável (YAGA et al., 2018). Desta forma, e dentro de inúmeros outros propósitos, um *smart contract* representa uma terceira entidade confiável dentro dos processos transacionais de uma rede *peer to peer.*

Embora o conceito de contratos inteligentes combine muito com a ideia de *blockchain* e redes distribuídas, em (YAGA et al., 2018) os autores destacam que nem toda estrutura de *blockchain* estará apta a executar *smart contracts*. Representando transações que precisam ser validadas entre múltiplas partes, trazem consigo os benefícios da transparência e da possibilidade de fornecer dados atestáveis. O seu ponto positivo de maior destaque, no entanto, reside em sua estrutura, imaginada de forma a tornar dispensável a presença de entidades centralizadoras. Uma vez publicado, um *smart contract* irá permitir que as melhores decisões de negócio sejam tomadas de forma automática, imparcial e inviolável.

Para que isso aconteça, no entanto, contratos inteligentes devem ser determinísticos (YAGA et al., 2018). Isso significa que a partir de uma entrada comum, o resultado produzido pelo processamento de dados, seja ele qual for, será sempre o mesmo. Desta forma, é possível garantir que todos os nós dentro de uma rede distribuída têm a possibilidade de concordar no estado final produzido pela execução de um contrato inteligente, permitindo que as redes que implementam sistemas de *blockchain* ultrapassassem o estigma das criptomoedas em detrimento do uso para aplicações não convencionais (WESTPHALL, 2021).

Atualmente, segundo (ZHANG et al., 2020):

> Muitos esforços têm sido feitos para proteção da privacidade dos dados e rastreamento confiável de informações, no entanto, as soluções convencionais ainda são vulneráveis à perda de informações, vazamento de dados e outros ataques.

Isso ocorre em virtude da distinção natural entre a arquitetura e organização dos sistemas distribuídos frente aos grandes monólitos convencionais. As tentativas de manter este tipo de rede confiável e segura a longo prazo, independente de sua aplicação, precisam considerar os

vários nodos do sistema enquanto agentes principais na execução deste trabalho. Neste cenário, o uso da *blockchain* torna-se uma alternativa para preservar dados e informações de forma segura ao longo do tempo, afastando-se de sua aplicação convencional, que permanece voltada ao contexto de moedas e ativos digitais.

A popularização das criptomoedas*,* por sua vez, chama a atenção para o uso da *blockchain* na construção de sistemas rastreáveis anti-falsificação, de gerência de propriedade intelectual ou de cadeias de ativos, todos baseados na descentralização. A *internet* das coisas pode ser vista como o exemplo mais contundente onde a aplicação de um sistema de *blockchain* traz ganhos à segurança dos dados trocados entre os nós da rede distribuída. Muitas pesquisas procuram estudar formas de melhorar pontos específicos no cenário da *internet* das coisas, como os protocolos de comunicação ou as formas de transferir e armazenar dados. No entanto, quase sempre, estes representam gargalos no ecossistema, pois dependem de serviços centralizados (ZHANG et al., 2020). Segundo (UDDIN et al., 2021):

> Nas características atuais, as grandes quantidades de dados produzidas pelo crescente número de dispositivos IoT podem formar um gargalo nos sistemas de *internet* das coisas, resultando em uma baixa qualidade de serviço. Um ponto único de falha se refere a um componente do sistema que pode interromper todo o fluxo de processamento da rede, o que é inconveniente em todo sistema em que se deseja alcançar alta disponibilidade e confiabilidade.

Adotar sistemas de *blockchain* para identificar dados e ativos dentro da *internet* das coisas é uma das soluções propostas para evitar pontos únicos de falha, de forma segura. Todos os participantes da rede executam o *software* em posição de autoridade sobre os dados, e deixam de depender de entidades terceiras que podem ser corruptíveis frente à manutenção da informação. É uma solução que resolve o isolamento e a falta de confiança, pontos cruciais ao funcionamento de um sistema distribuído, em uma cadeia que envolve várias partes (ZHANG et al., 2020).

2.2.7. Hyperledger Fabric

Neste sentido, para criar uma estrutura de rede distribuída como a *blockchain,* é possível trabalhar com *frameworks* próprios para este fim ou utilizar linguagens de programação puras através de bibliotecas de terceiros. O *Hyperledger Fabric* é um projeto de *software* livre utilizado para desenvolver aplicações *blockchain,* levando em consideração a sua capacidade

de adaptação a uma ampla gama de cenários (BLUMMER et al., 2018, apud WESTPHALL, 2021).

A implementação do *Hyperledger Fabric* é uma extensão do conjunto de ferramentas originais (mais amplo e genérico), o *Hyperledger,* e neste contexto, é importante ressaltar a facilidade com a qual integra-se *smart contracts* escritos em linguagens de programação terceiras, como Java, Go ou ainda JavaScript.

Segundo a introdução ao *Hyperledger Fabric* feita pela IBM, *What is Hyperledger Fabric?***,** este projeto visa construir uma arquitetura flexível e escalável, dentro de aspectos que buscam trabalhar, sobretudo, a modularidade e a resiliência. A estrutura de consenso dentro deste *framework* também é produzida de modo a permitir relacionamentos independentes e privados, promovendo o isolamento total das transações através dos canais fornecidos pelo *Hyperledger,* que podem ser compartilhados em subgrupos específicos dentro de uma rede distribuída. Esta abordagem vai de encontro com as arquiteturas de *blockchain* tradicionais, que são projetadas, a princípio, apenas para uso público.

Substancialmente, um sistema de *blockchain* construído sobre a estrutura do *Hyperledger Fabric* possui uma arquitetura fundamentalmente diferente das redes de *ledger* distribuídas tradicionais. Isso porque este tipo em particular baseia-se em três tipos de nodos (solicitante, endossante e validador), que possuem funções diferentes dentro do ecossistema; uma *blockchain* tradicional, por sua vez, opera com todos os pares executando funções de um contrato inteligente, validando transações e adicionando novos blocos ao *ledger* (WESTPHALL, 2021).

Segundo o mesmo autor:

> Primeiro, um cliente envia uma proposta de transação para os nodos endossantes. Estes, por sua vez, executam a transação e devolvem o seu resultado ao cliente, assinado digitalmente por eles. Em seguida, o cliente envia a transação assinada aos solicitantes, que seguem um protocolo para decidir quais transações formarão o próximo novo bloco. Quando eles chegam a um acordo, propagam o novo bloco pela rede e todos os pares o validam (WESTPHALL, 2021).

O processo de validação é igualmente determinístico e é executado exclusivamente pelos nodos cuja responsabilidade é legitimar cada transação que irá inserir um novo bloco no

ledger. Seguindo três etapas sequenciais paralelamente, cada nó deve assegurar que a transação segue as políticas de endosso da rede e que não existem conflitos no *software* da *blockchain* utilizado pelo nó solicitante da transação. Por fim, se ambos os critérios são atendidos, o *ledger* é atualizado e o bloco produzido pela transação é inserido na rede (WESTPHALL, 2021).

Existem ainda outros *frameworks* que são utilizados pela comunidade na construção de estruturas de *blockchain,*como o Corda e o *Meter*. O *Hyperledger Fabric,* no entanto, por ser mantido pela *Linux Foundation,* possui maior suporte e documentação. Segundo o site da fundação mantenedora, o "*Hyperledger* é um esforço colaborativo de código aberto criado para promover as tecnologias de blockchain de vários setores". Considerado uma colaboração global de inúmeros serviços somados à tecnologia, recebe apoio de gigantes da tecnologia como a IBM e atende muito bem seu propósito original[4], que é suprir as necessidades da indústria atreladas à *blockchain* que não poderiam ser supridas por estruturas de *ledgers* públicos.

2.3. Identidade Soberana

A autenticação dentro de um sistema de informação é o processo que fornece garantias quanto à identidade de quem o acessa. É a forma pela qual um sujeito (usuário, organização ou objeto) pode ser identificado com confiança em um sistema. A identidade digital, por sua vez, é um atributo fundamental e único deste processo, e pode ser estabelecida por meio de diferentes modelos de acordo com a natureza e das necessidades do sistema no qual é implementada.

A proposta de identidade descentralizada surge como uma referência a parte das abordagens tradicionais de autenticação, relacionadas às entidades centralizadas ou federadas. Este processo, no modelo centralizado que é descrito na figura 6, ocorre de forma direta entre o usuário e o provedor de serviços, enquanto que no federado, descrito na figura 7, é feito por meio de um provedor de identidade. Em ambos os modelos, mesmo que em níveis distintos, o usuário torna-se responsável pela manutenção de suas credenciais, em propostas nas quais não possui domínio real sobre sua identidade digital[6]. Neste sentido, de maneira geral, é possível que ocorram cenários em que o acesso do usuário aos serviços finais seja negado em virtude da negligência relacionada a sua identidade.

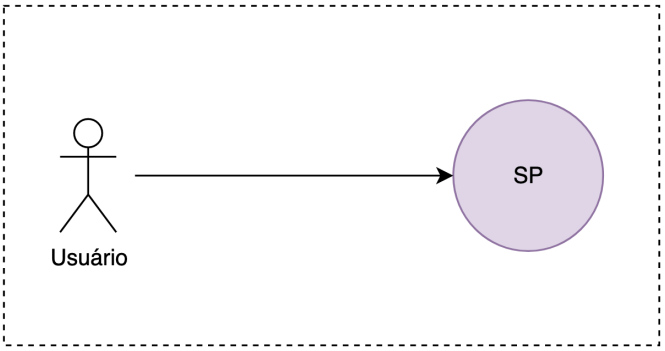

Figura 6 – Modelo de Identidade Digital Centralizado, adaptado de (PREUKSCHAT & REED, 2021).

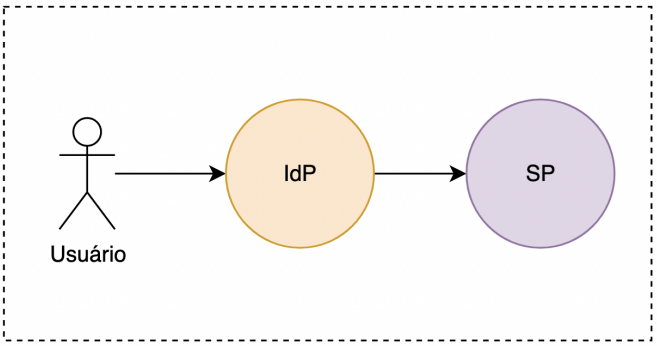

Figura 7 – Modelo de Identidade Digital Federado, adaptado de (PREUKSCHAT & REED, 2021).

Segundo (REVOREDO, 2019) "a identidade digital é a soma total de todos os atributos que existem sobre nós no mundo digital". Muitos dos processos de autenticação em sistemas de informação da *Internet* tradicional suportam soluções de *Single Sign-On,* que permitem que o usuário seja identificado por meio de serviços providos por partes terceiras autorizadas a fim de facilitar o processo de autenticação. No entanto, quando outras corporações estão envolvidas neste processo, aumenta a possibilidade de que o propósito final atrelado ao uso dos dados do usuário não esteja completamente alinhado às expectativas deste (LUX et al., 2020).

2.3.1. Identificador Descentralizado

Uma alternativa segura, flexível e altamente disponível, responsável por gerenciar a identidade do usuário enquanto trabalha pela sua privacidade é a Identidade Soberana. A ideia

central desta abordagem é o conceito de descentralização da identidade digital de um usuário, através de identificadores únicos verificáveis por meio do compartilhamento de chaves criptográficas públicas (LUX et al., 2020). Segundo (XAVIER et al., 2021):

> Nas implementações de Identidade Auto-Soberana - *Self-Sovereign Identity* (SSI) que utilizam *blockchain*, as entidades publicam suas chaves públicas num registro distribuído e, através desse processo, obtém um identificador, o Identificador Descentralizado - *Distributed Identifiers* (DID). Quando uma entidade precisa verificar a autenticidade dos emissores, [...] o DID pode ser utilizado para recuperar tais chaves.

Um DID (Identificador Descentralizado) é um tipo de identificador que permite a descentralização da identidade de forma verificável. Ele pode se referir a qualquer coisa, como pessoas, organizações, dispositivos ou mídias físicas e digitais. Segundo (OMAR & BASIR, 2020), os "DIDs são construídos para serem a base da identidade digital descentralizada e da infraestrutura de chave pública (PKI) para a Internet".

2.3.1.1. Princípios Essenciais para Construção de um Identificador Descentralizado segundo Christopher Allen

Enquanto peça chave de ecossistemas ainda maiores, Identificadores Descentralizados precisam estar adequados dentro de um conjunto de características que garantem ao indivíduo total autoridade sobre sua própria identidade, segundo a especificação W3C sobre DIDs. Segundo Christopher Allen, especialista em segurança digital responsável pela criação do conceito inicial de *Self-Sovereign Identity*:

> A Identidade Auto-Soberana é o próximo passo além da identidade centrada no usuário, e isso significa que começa no mesmo lugar: o usuário deve ser central para a administração da identidade. Isso requer não apenas a interoperabilidade da identidade de um usuário em vários locais, com o seu devido consentimento, mas também o verdadeiro controle deste a respeito de sua identidade digital, criando autonomia para o usuário (ALLEN, 2020).

Neste sentido, para que o usuário possua o controle e a autoridade inicialmente planejados sobre sua identidade, Allen determina 10 características essenciais que um Identificador Descentralizado padronizado deve possuir:

Característica	Descrição
Independência	Os usuários devem ser independentes uns dos outros.
Controle	Os usuários devem controlar suas próprias identidades.
Acesso	Os usuários devem ter livre acesso às suas identidades, com alta disponibilidade.
Transparência	Sistemas e seus algoritmos devem ser transparentes aos usuários
Persistência	A identidade de um usuário deve possuir um longo ciclo de vida.
Portabilidade	Deve ser possível comunicar informações e serviços a respeito da identidade entre múltiplos sistemas.
Interoperabilidade	A Identidade Auto-Soberana de um usuário deve poder ser utilizada na maior quantidade de serviços possíveis por ele.
Consentimento	O uso da identidade de um usuário em um sistema deve ser devidamente autorizado por ele.
Minimização	A exposição de permissões relacionadas à identidade do usuário deve ser minimizada.
Proteção	As informações bem como os direitos do usuário e de sua

	identidade devem ser protegidos.

Tabela 1 - Os 10 princípios da Identidade Auto-Soberana, adaptado de (SIQUEIRA et al., 2021)

Identificadores únicos e credenciais verificáveis são os alicerces fundamentais de uma estrutura de Identidade Soberana. DIDs são identidades únicas inseridas em um escopo global (GRÜNER et al., 2019). Por serem permanentes, descentralizados, localizáveis e criptograficamente verificáveis (REED, 2018), permitem que o usuário possa ser identificado com maior privacidade e controle sobre seus dados, em um modelo que garante mais confiança quanto a veracidade do processo de autenticação (LUX et al., 2020).

2.3.1.2. Estrutura de um Identificador Descentralizado

De acordo com a W3C, consórcio mundial cujo objetivo é a padronização da *World Wide Web*, um identificador descentralizado está estruturado de acordo com a Figura 8, em um formato de URI (*Uniform Resource Identifier*), que pode ser extensível a um modelo de URL para que seja possível incorporar outros recursos úteis, como *path* e *query params,* no intuito de acrescentar mais informações a uma Identidade Descentralizada, como uma estrutura de chave pública-privada.

> Semelhante à resolução DNS em que uma URL é fornecida como entrada e o endereço IP correspondente é retornado como saída, a resolução DID usa o DID como entrada e retorna um documento DID como saída (FDHILA et al., 2021).

Segundo a documentação da RNP, qualquer função geratriz de um identificador descentralizado é válida, desde que sigam as regras gerais de formação pré-estabelecidas e que sua execução resulte em um documento DID válido, segundo o documento para Descentralização da Identidade Digital da RNP.

Um DID necessariamente é representado por meio de uma string. Um identificador descentralizado é uma especificação baseada na sintaxe formal de um schema e resolve, por meio de seu método, um Documento DID, que descreve as propriedades do identificador

descentralizado (OMAR & BASIR, 2020). Um schema, por sua vez, é a definição das propriedades de um Documento DID. Na figura 8, é possível observar a estrutura base de um Identificador Descentralizado:

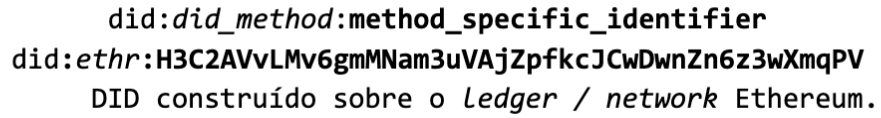

Figura 8 - Exemplo de um Identificador Descentralizado, retirado da especificação W3C sobre DIDs.

A sintaxe formal de um Identificador Descentralizado inicia-se com o prefixo *did.* Em seguida, é apresentado o método DID da estrutura. Um método é um ledger ou rede, que possui especificações particulares para as operações CRUD de um DID. Segundo (FDHILA et al., 2021), "existem inúmeros métodos que podem ser utilizados para gerenciar um Documento DID. Eles podem ser diferenciados por meio de quatro parâmetros: Infraestrutura, Governança, Operações e Segurança". Um método define as operações que estruturam o ciclo de vida de um Identificador Descentralizado, como funções de criação, resolução, atualização e desativação.

De acordo com a especificação da W3C sobre Identificadores Descentralizados, um DID pode ser resolvido em inúmeros métodos que suportem a tecnologia. Alguns exemplos relacionados são *Ethereum, Sovrin, IPFS e Veres One.* As atualizações feitas sobre um Documento DID são feitas de acordo com a política de inserção e atualização descritas pelo método, *e.g.,* CRUD. No entanto, de qualquer forma, uma atualização feita sobre um Documento DID é uma re-declaração do JSON-LD do registro original.

Após a criação de um Identificador Descentralizado, cria-se um registro deste DID em um livro-razão distribuído, e.g. *Blockchain*. Este processo é feito através do envio de uma transação, que inclui o DID e os metadados (DDO, DID Document) associados ao registro (XAVIER et al., 2021).

> O Identificador Descentralizado é o ID exclusivo para procurar um Documento DID. Um Documento DID é armazenado em um local central,

> para que possa ser facilmente pesquisado. Espera-se que o DID seja "persistente e imutável" para que esteja fora da influência de qualquer pessoa que não seja seu proprietário (POWERS, 2018).

O fluxo de criação de um Identificador Descentralizado está descrito na figura 9.

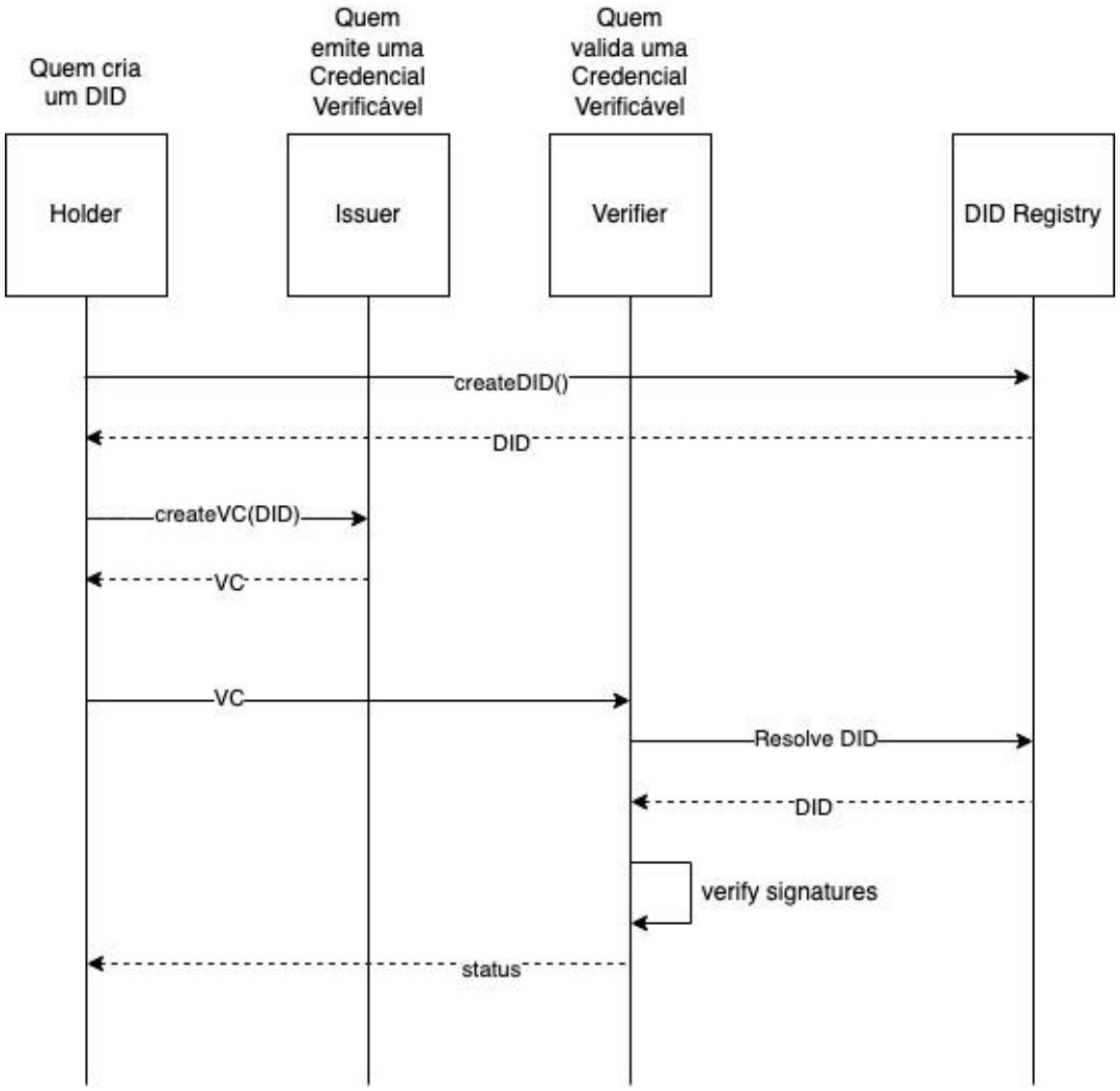

Figura 9 - Processo de criação de um DID, adaptado de (BRUNNER et al., 2020).

Um documento de um Identificador Descentralizado é representado por meio da especificação JSON (*Javascript Object Notation*), contendo informações sobre a estrutura em si, bem como formas de garantir a autenticidade do documento em um controlador DID, entidade que possui autoridade suficiente para alterar um documento de um Identificador Descentralizado, segundo a especificação W3C sobre DIDs. Na figura 10 está descrita a estrutura de um documento DID.

```
{
        "@context": [
                "https://www.w3.org/ns/did/v1",
                "https://w3id.org/security/suites/ed25519-2020/v1"
         ]
         "id": "did:example:123456789abcdefghi",
         "authentication": [
                {
                         "id": "did:example:123456789abcdefghi#keys-1",
                        "type": "Ed25519VerificationKey2020",
                        "controller": "did:example:123456789abcdefghi",
                        "publicKeyMultibase":
                "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
                }
        ]
}
```

Figura 10 - Estrutura de um Documento Descentralizado

Qualquer Identificador Descentralizado gerado sobre um método DID deve ser um valor único, pois deve ser referenciado e desreferenciado a um documento DID, de acordo com a figura 11. O documento DID contém toda e qualquer informação relacionada ao recurso que se busca representar. De acordo com a especificação da W3C sobre Identificadores Descentralizados:

> As funções de resolução de um DID resolvem um Identificador Descentralizado em um documento DID por meio da operação "Read" do método DID. Os detalhes de como esse processo é realizado estão fora do escopo desta especificação, mas todos os resolvedores DID em conformidade implementam estas interfaces de resolução.

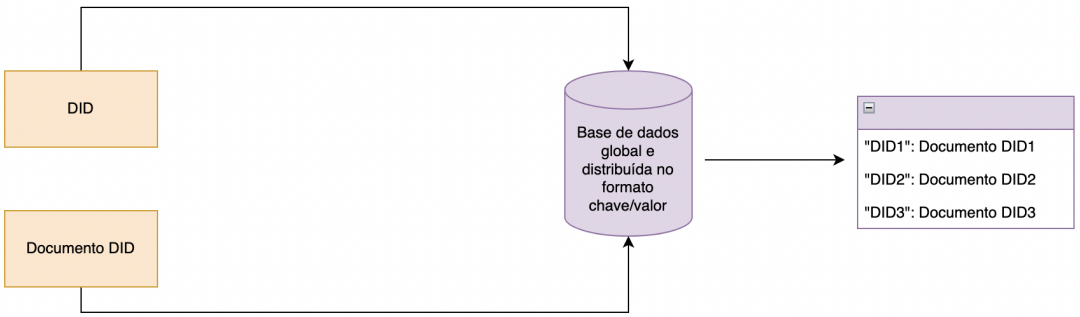

Figura 11 - Representação do armazenamento e resolução de um DID, adaptado do documento para Descentralização da Identidade Digital da RNP.

Documentos DID são planejados para que aplicações terceiras os utilizem para validação da identidade digital de um usuário ou recurso. Neste sentido, segundo o guia da RNP para Descentralização da Identidade Digital, a estrutura de um documento possui, necessariamente, as seguintes informações:

- Identificador Descentralizado Único (DID)
- Conjunto de chaves-privadas (para verificação) Conjunto de métodos de autenticação (para autenticação)
- Conjunto de *endpoints* de serviço (para interação)
- Timestamp (para auditoria)
- Assinatura (para integridade)

2.3.2. Trust Over IP

A Identidade Descentralizada e as Credenciais Verificáveis são dois conceitos que viabilizam uma nova era de confiança digital, na qual a autenticação é feita por meio de abordagens distribuídas. No entanto, torna-se necessária a existência de um sistema de padronização que habilite a interoperabilidade entre sistemas de autenticação distribuídos. A fundação Trust Over IP, que faz parte da Linux Foundation, utilizou como base o protocolo de comunicação padrão da Internet TCP/IP para criar uma arquitetura de quatro camadas que viabiliza a confiança digital em escala global, integrando confiança criptográfica aos sistemas de comunicação da Identidade distribuída.

A implementação do protocolo TCP/IP seguida pela criação do SSL/TLS permitiu uma série de inovações que construíram a Internet como conhecemos hoje. No entanto, existe uma lacuna na qual mesmo a segurança destes protocolos não pôde resolver, que é a identificação do usuário, por mais que exista confiança no site que se está acessando (CAMERON, 2005).

De acordo com a especificação do Hyperledger Aries, o propósito do Trust Over IP é preencher esta lacuna ao definir uma estrutura padronizada por meio da qual seja possível estabelecer comunicação segura em sistemas de identificação distribuídos.

A segunda camada da pilha Trust Over IP é justamente a troca de mensagens padronizada pelo DIDComm (HARDMAN, 2019).

2.3.3. DIDComm

DIDComm (*Decentralized Identifiers Communications*) é o protocolo que permite a comunicação segura entre duas partes (DID para DID). É uma especificação que provê às entidades responsáveis pelo controle sobre a criação e armazenamento do DID um canal de comunicação seguro (KIM et al., 2021). É definido sobre uma arquitetura robusta que permite autenticação mútua entre duas partes, em que a confiança está enraizada nos Identificadores Descentralizados e depende das próprias mensagens, e não das propriedades externas dos transportes usados (CURREN, 2022).

Baseado em um conjunto padrão de especificações hoje mantidas pela *Decentralized Identity Foundation,* o DIDComm foi inicialmente desenvolvido através do framework Aries, da fundação Hyperledger. Seu primeiro objetivo era tornar a comunicação dentro do Indy-CLI transparente, provendo também interoperabilidade entre tecnologias e protocolos definidos dentro do contexto das Identidades Descentralizadas, como VCs, Documentos e Métodos DID.

A característica mais importante do DIDComm é que todas as conexões entre dois agentes são protegidas por meio de pseudônimos (HARDMAN, 2019). Por meio deste protocolo, é possível trocar DIDs e Documentos DID de forma a manter uma conexão privada e segura, estabelecendo um relacionamento de confiança entre as partes. Segundo a especificação do Aries Cloud Agent, a conexão estabelecida não tem data de validade e não necessita de intermediários.

2.3.3. Uso de Identificadores Descentralizados e sua Arquitetura

A partir da compreensão da ideia inicial dos Identificadores Descentralizados, que correspondem a uma forma de garantir a identidade de um sujeito de forma digital, distribuída e verificável, é possível não apenas observar as propostas de uso mais tradicionais relacionadas aos DIDs como também analisar o impacto da *blockchain* nas implementações deste conceito dentro de sistemas de informação modernos. Segundo (REED, 2018),

Identificadores Descentralizados representam apenas o topo da pilha do conceito e aplicação da Identidade Descentralizada.

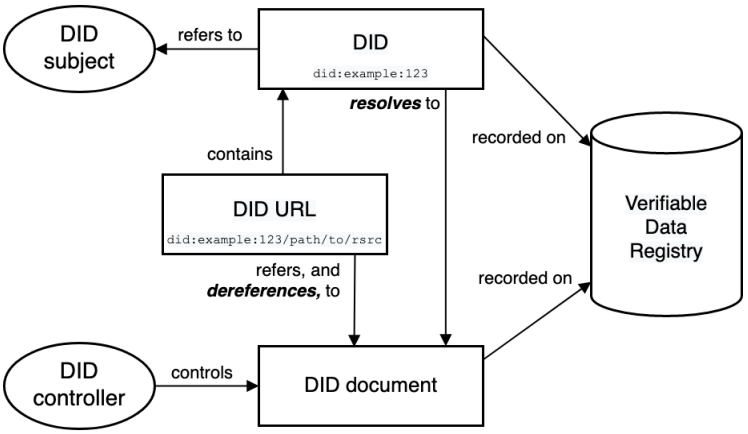

Figura 12 - *Overview* da arquitetura DID e do relacionamento entre seus componentes, retirado da Recomendação Oficial da W3C para Identificadores Descentralizados.

Este modelo de referência expresso na figura 12 apresenta o componente URL como objeto central e de suma importância para resolução de DIDs. A URL contém o atributo URI - observado enquanto a camada inicial da arquitetura - que representa o sujeito proprietário da identidade em questão. A partir de ambos, é possível obter o documento DID, gerenciado por um controlador. Este contém, finalmente, a informação a respeito do portador do identificador. Tanto o documento DID quanto a URI DID são guardados em um banco de dados, devidamente auditado e verificado.

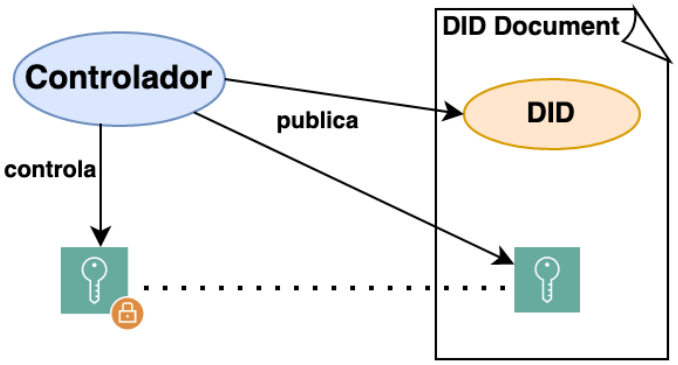

Figura 13 - Relacionamento entre a entidade Controladora, o Documento DID e a estrutura de Identificador Descentralizado, retirado do guia da RNP para Descentralização da Identidade Digital.

Neste cenário, de acordo com a figura 13, um controlador DID é a entidade dentro de um sistema de autenticação descentralizado que tem permissões suficientes para alterar o conteúdo de um documento DID. Um documento DID pode ter mais de uma entidade o controlando.

Segundo (NAKAMURA et al., 2019), "aplicações relacionadas à identidade digital utilizando tecnologia *blockchain* permitem a verificação, autorização e gerenciamento de identidade inalterados, resultando em eficiências significativas e redução de fraudes". Este processo torna-se possível em virtude da manutenção das chaves públicas (armazenadas na camada de *Verifiable Data Registry* (VDR), segundo a Recomendação Oficial da W3C para Identificadores Descentralizados) de cada entidade participante do sistema de identificação e validação em um registro distribuído (XAVIER et al., 2021). De acordo com os mesmos autores:

> Quando o DID se refere ao emissor, armazená-lo em locais públicos como as *blockchains* é necessário, uma vez que outras entidades, como os portadores e verificadores, podem facilmente obtê-las e garantir que estão se comunicando com a entidade na qual confiam. Nesse contexto, o VDR atua como um repositório confiável de chaves públicas (XAVIER et al., 2021).

Nos casos em que o processo de comunicação e validação do registro envolvem não o emissor, e sim o portador, alternativas em que a identidade do usuário é preservada devem ser consideradas a fim de garantir a sua privacidade (XAVIER et al., 2021). Apesar desta recomendação, é possível encontrar fluxos em que a *blockchain* é usada para estes fins, de qualquer forma.

2.3.4. Tecnologias

2.3.4.1. Hyperledger Indy

A principal ferramenta utilizada para implementar sistemas de Identidade Descentralizada é o *Hyperledger Indy*. Tal qual o *Hyperledger Fabric,* este *framework* está sobre o domínio do projeto *Hyperledger,* iniciativa *open source* mantida pela *Linux Foundation* cujo objetivo é avançar nos campos de estudo e implementação de soluções distribuídas na área de *blockchain.*

Neste sentido, o *Hyperledger Indy* é baseado nos códigos fonte *Sovrin Network,* que em 2017 foram doados à *Linux Foundation* (SIQUEIRA et al., 2021). Segundo o site da ferramenta, é possível definir o *Hyperledger Indy* como um *software* que fornece acesso à bibliotecas e componentes reutilizáveis para construir Identidades Descentralizadas digitais, baseadas em tecnologias como a *blockchain.* O *Hyperledger Indy,* de acordo com a definição do *site Hyperledger Foundation,* é interoperável, e pode ser utilizado em qualquer ambiente de forma independente.

Segundo (SIQUEIRA et al., 2021):

> Dessa forma, o Hyperledger Indy define uma camada de software para lidar com identidades soberanas. Uma vez construída sobre a base Blockchain, a infraestrutura de identidades digitais Hyperledger Indy ganha em confiança, descentralização e auditabilidade.

O *Hyperledger Indy* é construído sobre uma estrutura baseada em cinco componentes fundamentais. São eles:

Componente	**Descrição**
Criptografia	Implementação *Zero Knowledge Proof* que permite que um usuário garanta a integridade de outro.
Nodos	Conjunto de participantes da rede. Todos os nodos possuem uma cópia completa do *ledger* da rede, mas somente o nodo mestre mantém-se responsável pela governança nesta. O consenso é garantido dentro da *blockchain* por meio do algoritmo RBFT (*Redundant Byzantine Fault Tolerance*).

Ledger	Estrutura sobre a qual são armazenados os registros da rede.
Estado	Estado atual da *blockchain,* gerenciado por meio da árvoce de *Merkle.*
Armazenamento	Armazenamento da rede em um banco de dados no formato de chave/valor.

Tabela 2 - Componentes fundamentais da arquitetura do *Hyperledger Indy,* retirada do artigo *A kickstart to Hyperledger Indy.*

O *Hyperledger Indy* funciona de forma a transformar a identificação inicial de um usuário (seu nome) em uma chave exclusiva descentralizada, um DID. Neste processo está inclusa a criação do documento DID correspondente. Segundo o artigo "*A kickstart to Hyperledger Indy*", de forma semelhante ao fluxo de interação da *blockchain,* os usuários poderão interagir entre si utilizando pares de chaves público-privada.

Existem quatro papéis dentro do ecossistema implementado pela arquitetura do *framework Hyperledger Indy.* São eles que possibilitam a execução do fluxo de autenticação e verificação da identidade de um usuário.

Papel	**Descrição**
Titular (*Holder*)	Entidade em posse de uma credencial.
Emissor (*Issuer)*	Entidade responsável por emitir e conceder credenciais a um solicitante após verificação.
Verificador (*Verifier)*	Entidade que verifica a credibilidade de um solicitante.

Cadastro (*Registry)*	Repositório em que são armazenados os registros de todos os usuários em posse de um Identidade Descentralizada digital.

Tabela 3 - Papéis em um ecossistema de uso e validação de Identificadores Descentralizados no *Hyperledger Indy*, retirado do artigo *"A kickstart to Hyperledger Indy"*.

Desta forma, o *Hyperledger Indy* implementa a estrutura da figura 14:

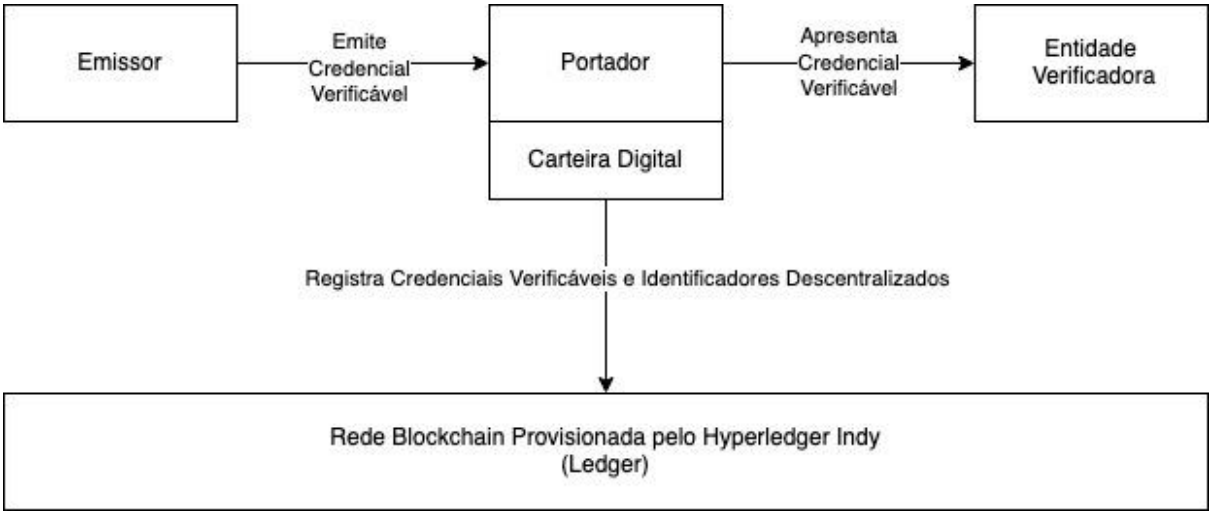

Figura 14 - Arquitetura do *Hyperledger Indy*, adaptado do artigo *"A kickstart to Hyperledger Indy"*.

2.3.4.2. Indy-CLI

O Indy-CLI é uma especificação que implementa uma interface de linha de comando utilizada para interagir com a plataforma Hyperledger Indy. A partir do Indy-CLI é possível que os usuários de uma rede distribuída estejam aptos a criar e gerenciar Carteiras Digitais e Identificadores Descentralizados, utilizáveis por meio da emissão de Credenciais Verificáveis. Desta forma, os usuários da rede podem interagir entre si por meio do compartilhamento ou *discovery* de suas Identidades Descentralizadas.

Esta ferramenta pode ser utilizada por meio da instalação direta no sistema operacional ou em uma versão organizada em containers. A partir da Von-Network, ferramenta descrita na seção 2.3.4.3, é possível obter acesso à esta versão conteinerizada.

Para desempenhar suas principais funcionalidades, o Indy-CLI é capaz de fornecer dois parâmetros essenciais e obrigatórios, a *seed* e a *key.* Este par de valores é criado como uma sequência aleatória de 32 e 64 caracteres, respectivamente. A *seed* é um valor privado associado a uma Identidade Descentralizada, utilizada para derivar o DID, enquanto a *key,* por sua vez, é utilizada para criptografar a carteira digital em que o Identificador Descentralizado será armazenado.

É importante ressaltar que a *seed* deve ser mantida segura pois representa a chave privada do usuário que detém o DID. Além disso, qualquer pessoa em sua posse tem a chance de gerar outra vez o mesmo par de DID e Chave Verificadora, e.g., *verkey*. A *verkey* é a chave pública correspondente à chave privada gerada a partir da *seed*.

A *key* também deve ser guardada com cuidado, uma vez que permite acesso a todas as credenciais gerenciadas pela carteira do indivíduo. Por isso, a recomendação oficial é de que estes dois parâmetros (*seed* e *key*) sejam gerenciados exclusivamente de forma *offline*, de modo a mantê-los isolados da rede distribuída na qual os Identificadores serão manuseados.

Ao criar um Identificador Descentralizado, um par de chaves pública e privada, associadas ao DID, são derivadas a partir do algoritmo ED25519. Cada DID é associado, por tanto, a uma *signing key* (chave privada) e a uma *verkey* (chave pública). Enquanto a *signing key* deve ser armazenada na carteira digital do portador, em segredo, o DID e sua *verkey* associada são armazenados no *ledger* para acesso público (TAM, 2019).

2.3.4.3. Von-Network

A Von-Network é uma ferramenta baseada no componente Indy da fundação Hyperledger. Possuindo inúmeras funcionalidades, seu principal objetivo é fornecer ao usuário um banco de dados distribuído por meio do qual é possível autenticar e gerenciar Identificadores Descentralizados, de forma prática e simplificada.

Além de contar com uma aplicação de linha de comando, possui uma interface gráfica que permite gerenciar o status de todos os nodos do *ledger*, bem como visualizar o estado de todas as transações feitas na rede. A Von-Network conta com um *schema* padrão para registro de Identificadores Descentralizados, que funciona integrado ao Indy-CLI.

Além de ser possível registrar Identificadores Descentralizados, de acordo com a especificação da Von-Network, é possível inserir no *ledger* transações que identificam *schemas* personalizados para a criação de DIDs.

2.3.4.4. ACA-Py

O componente Aries Cloud Agent Python é um *framework* baseado no Hyperledger Aries, por meio do qual é possível criar e gerenciar Identificadores Descentralizados e Credenciais Verificáveis. Para fazer uso desta ferramenta, é necessário criar um controlador que seja capaz de se comunicar com o ACA-Py, que opera por meio do DIDComm. O controlador pode ser escrito em qualquer linguagem que possa se comunicar por meio de *requests* HTTP.

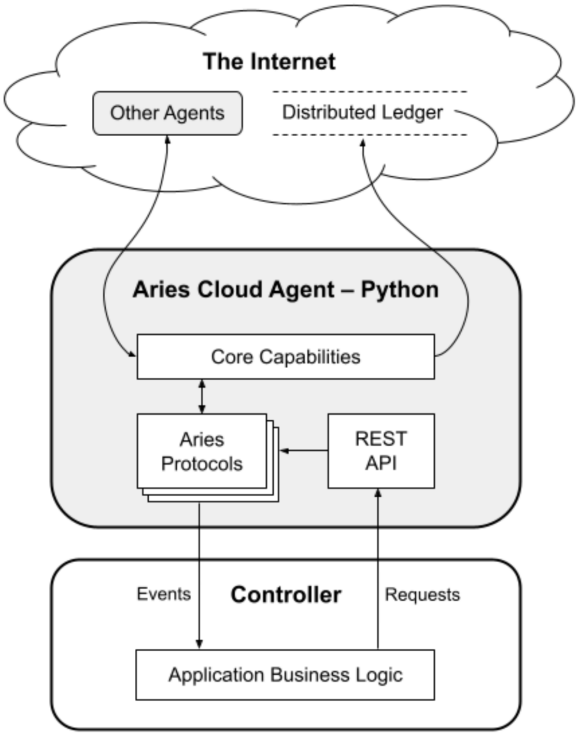

Figura 15 - Arquitetura do Aries Cloud Agent - Python, retirada da documentação do framework ACA-Py.

De acordo com a especificação do ACA-Py, o Aries Cloud Agent funciona como um intermédio personalizável por meio do qual é possível criar e gerenciar os processos de emissão e validação de Credenciais Verificáveis e Identificadores Descentralizados, abstraindo as camadas de baixo nível relacionadas a estes meios. É uma solução *multi-tenant* e escalável, que trabalha em conjunto com outras ferramentas para criar um ambiente de gerenciamento seguro, funcionando em um esquema de emissor como serviço.

Por si só, o Hyperledger Aries é uma ferramenta que provê um conjunto de funcionalidades específicas para criar um ambiente distribuído, interoperável e reutilizável. A especificação do Aries Cloud Agent é um mediador, que facilita o gerenciamento de Identificadores Descentralizados, Credenciais Verificáveis e comunicação DID para DID.

2.3.5. Credenciais Verificáveis

Credenciais Verificáveis e Identificadores Descentralizados são conceitos atrelados de forma intrínseca. A especificação de uma Credencial Verificável, por sua vez, determina, segundo a recomendação *Verifiable Credentials Data Model* v1.1 da W3C, que toda e qualquer informação que uma credencial física (Carteira Nacional de Habilitação, Carteira de Identidade, etc.) pode ter também pode ser representada em uma Credencial Verificável para uso em ambiente digital.

Para que uma VC seja emitida, é necessário a existência de um DID. Isso ocorre porque uma Credencial Verificável será atrelada a um Identificador Descentralizado, uma vez que, em uma VC, estão presentes os metadados necessários para identificar o emissor, enquanto que o sujeito portador será identificado pelo seu DID e todos os dados a ele associados (KIM et al., 2020).

> Uma Credencial Verificável é criada pelo emissor e enviado ao receptor. Ele contém um conjunto de declarações sobre atributos, por exemplo, nome, data de nascimento, grau, ID ou outras informações que o emissor deseja atribuir ao destinatário (BRUNNER et al., 2020).

A especificação da W3C no contexto de *Verifiable Credentials* está dentro da camada de Registro de Dados Verificáveis, que, entre inúmeros fins, tem por objetivos em destaque a

possibilidade de estabelecer relações transparentes de confiança entre as entidades de um sistema e a divulgação segura dos esquemas que definem o conteúdo de uma credencial (XAVIER et al., 2021). Segundo os mesmos autores:

> A relação de confiança entre emissores (*issuers*) e portadores (*holders*) é importante para que as credenciais sejam emitidas para quem tem direito de gerenciá-las e sejam guardadas com segurança e confidencialidade (XAVIER et al., 2021).

Em uma estrutura de *Verifiable Credential*, segundo o guia da RNP para Descentralização da Identidade Digital, é possível guardar informações a respeito do usuário detentor da Credencial Verificável, como foto, nome, número de identificação, bem como aquelas referentes à autoridade que emitiu a credencial, como agência governamental e cidade. Outros dados sobre o tipo de credencial também podem estar presentes neste tipo de identificação. É desta forma que, de modo independente, o usuário possui total controle sobre seus dados, bem como sobre estes são compartilhados com possíveis serviços terceiros, criando um padrão de arquitetura centrado no usuário.

No processo de emissão de uma Credencial Verificável, são identificados três principais papéis: *holder, verifier* e *issuer.*

Os *holders* são as entidades às quais um Identificador Descentralizado se refere. O portador de um DID pode criar um Identificador Descentralizado para cada Credencial Verificável que ele possua, no intuito de preservar sua identidade e privacidade. Um *holder* pode compartilhar seu Identificador com outras entidades, para que entre elas seja possível estabelecer uma comunicação segura.

Os *verifiers* são entidades que validam as informações contidas em um Identificador Descentralizado. Desta forma, tornam-se responsáveis por validar a identidade de um *holder,* seja este uma pessoa ou organização. A confirmação de identidade é feita sempre que há necessidade de concessão de acesso a um serviço ou recurso por meio do DID.

Um *issuer* é uma organização responsável por emitir uma VC. No entanto, para isso, primeiramente se faz necessário a criação de um Identificador Descentralizado. O ato de criar

um DID envolve a seleção e uso de um método, que será posteriormente identificado no *schema* deste Identificador Descentralizado (KIM et al., 2020). Por consequência, esta entidade fica responsável por, além de criar o DID, atualizar suas informações quando necessário. Emissores podem ser governos, empresas e/ou indivíduos.

A estrutura de uma Credencial Verificável é normalmente expressa através da sintaxe JSON-LD, que especifica um documento JSON associado à LD *Signatures* ou BBS+ *Signatures* para que provas de conhecimento do tipo Zero Knowledge Proof possam ser estabelecidas, segundo o documento para Descentralização da Identidade Digital da RNP.

3. Trabalhos Correlatos

3.1. Revisão Bibliográfica Bibliométrica

Para elaborar o presente trabalho, foi necessário realizar um levantamento no intuito de determinar o estado da arte no uso de tecnologias de *blockchain* no cenário de sistemas de Identidade Soberana para IoT. No entanto, para tornar possível a construção de uma fundamentação teórica rica e atualizada, pesquisas em cenários mais específicos dentro de cada uma das grandes áreas que este trabalho trata também foram feitas.

Os dados utilizados para construir esta revisão bibliográfica sistemática foram retirados das plataformas *Google Scholar* e *Scopus*. Enquanto o primeiro é uma ferramenta de pesquisa e indexação sobre conteúdo acadêmico na *Internet,* o segundo é um banco de dados de resumos e artigos de periódicos e revistas acadêmicas. Os dados da tabela 1 apresentam as palavras-chave pesquisadas em ambos os *sites*, e a soma de aparições em resumos, artigos e trabalhos. É possível determinar, a partir destes resultados, a relevância de cada termo no ambiente acadêmico.

Termo	**Quantidade**
"Blockchain"	556.548
"Self-Sovereign Identity"	1.030.000
"Internet of Things"	4.192.541
"Decentralized Identifiers"	252.000
"Internet of Things" "Self-Sovereign Identity"	143.041
"Internet of Things" "Forensics"	45.687
"Hyperledger Indy"	1.124
"Internet of Things" "Security"	560.902
"Internet of Things" "Blockchain"	140.326

"Internet of Things" "Blockchain" "Self-Sovereign Identity"	3.623
"Internet of Things" "Decentralized Identifiers"	38.600
"Decentralized Identifiers" "Blockchain"	23.300

Tabela 4 - Revisão Bibliométrica Sistemática

Partindo dos dados coletados e expostos da Tabela 1, é possível observar que o tema central deste trabalho, "*Internet of Things",* possui grande relevância expressa através do grande número de trabalhos acadêmicos publicados. Este valor é de 4.192.541 resultados. Ao combinar este termo com outras palavras-chave relacionadas ao desenvolvimento do trabalho, obteve-se, no entanto, números mais baixos. A pesquisa por "*Internet of Things"* junto de "*Self-Sovereign Identity*" e "*Internet of Things"* junto de "*Blockchain"* resultou em 143.041 e 140.326 resultados. A busca pelo termo "*Internet of Things"* associado a outras palavras-chave centrais ao tema deste trabalho, como "*Security"* e "*Forensics"*, trouxe 560.902 e 45.687 resultados, respectivamente.

Adicionalmente, a pesquisa pelo termo "*Decentralized Identifiers*", que é um dos temas essenciais ao desenvolvimento deste trabalho, resultou em 252.000 registros. A junção deste termo com outras expressões reduz significativamente o número de resultados obtidos: "*Decentralized Identifiers"* em conjunto com "*Internet of Things*" trouxe 38.600 resultados. O mesmo termo associado a "*Blockchain"* reduz o número de registros obtidos a 23.300.

Blockchain, no entanto e apesar de possuir grande relevância no cenário das tecnologias modernas - principalmente pelo seu uso atrelado às criptomoedas -, possui um número relativamente baixo de publicações. Foram obtidos 556.548 resultados na busca deste termo. *Self-Sovereign Identity,* terceiro e último tema central deste trabalho, possui um número extremamente baixo de publicações. A partir do valor de 5.416 resultados é possível observar que o tema é relativamente novo. A combinação dos três termos anteriormente citados, "*Internet of Things*", "*Blockchain*" e "*Self-Sovereign Identity*", obteve 3.623 trabalhos, demonstrando a especificidade do tema escolhido.

Outras buscas, referentes às ferramentas necessárias para o desenvolvimento deste trabalho, resultaram em números relativamente baixos comparados aos valores de outros termos. Na pesquisa pelo termo "*Hyperledger Indy",* ferramenta utilizada na criação de sistemas de identificação digital descentralizada, obteve-se apenas 1.124 resultados.

Nas seções 2.2, 2.3 e 2.4 estão descritos resumos de três trabalhos da literatura, provindos da revisão bibliográfica sistemática. Estes foram escolhidos com base na proximidade de cada tema com o objetivo final deste trabalho.

3.2. The Immutability Concept Of Blockchains And Benefits Of Early Standardization

A *blockchain* pode ser descrita como uma das tecnologias mais promissoras da atualidade. Esta tecnologia surge com a ascensão das criptomoedas, mas diz respeito a uma maneira diferente de armazenar e controlar registros, independente de sua natureza, de forma distribuída. Por meio dela, é possível que inúmeros pares em uma rede confiem uns nos outros sem a interferência de um terceiro. A tendência é que cada vez mais, novas soluções tecnológicas sejam possibilitadas pela *blockchain,* considerando sua sólida estrutura, que provê dados imutáveis e consenso distribuído (HOFMANN et al., 2017).

Formalmente, é possível definir a tecnologia da *blockchain* como um banco de dados distribuído, praticamente imutável sendo mantido sobre uma rede *peer-to-peer*. A validação dos dados, por sua vez, é feita através de um algoritmo de consenso pré-determinado, enquanto que a sua integridade histórica é garantida por meio da arquitetura encadeada. O objetivo central desta tecnologia é descentralizar o controle sobre um fluxo de dados, de forma a garantir a imutabilidade dos dados.

Dentro de uma *blockchain*, nenhum nodo precisa conhecer ou confiar nos demais integrantes da rede distribuída. A confiança no sistema, no entanto, só é obtida através da imutabilidade dos dados inseridos em uma *blockchain*. Entretanto, existe uma possibilidade para que exista inconsistência nos dados de uma *blockchain: bugs* no *software* que a implementa. Esta possibilidade é considerada improvável ou extremamente difícil. Nos casos em que os dados de um *ledger* distribuído foram violados, houve uma característica comum: diferentes versões

para uma mesma *blockchain* executando em nodos distintos. Justamente por ser um sistema distribuído, não é possível garantir que todos os pares estejam executando o mesmo estado.

Possíveis inconsistências no *software* de uma *blockchain* decorrem, principalmente, da falta de padronização na construção da estrutura da rede *peer-to-peer*. Falar a respeito da padronização da tecnologia dentro de aspectos como arquitetura, taxonomia e ontologia significa discutir a respeito de modelos que podem ser desenvolvidos e implantados com clareza nestes três pilares fundamentais. Neste sentido, é preciso esclarecer o significado de imutabilidade dentro dos contextos de taxonomia e ontologia, para que não se perca a confiança do público alvo, seja ele o usuário final enquanto consumidor ou o mercado no geral.

Grandes consórcios de tecnologia, como o *Hyperledger Fabric*, estiveram envolvidos no desenvolvimento de uma solução padrão para construção de *blockchains*. O que se infere como fundamental para o desenvolvimento de uma rede de *ledger* distribuído é que cada nó parte da estrutura pode confiar na imutabilidade dos dados por ela transacionados. Neste sentido, para alcançar o consenso, o envolvimento precoce dos usuários é necessário na definição de requisitos.

O conhecimento existente sobre a *blockchain* é fomentado em uma estrutura de código aberto. Neste modelo, existe um processo de produção descentralizado, em que qualquer pessoa pode modificar ou compartilhar tecnologias. Consequentemente, torna-se difícil, em virtude da grande quantidade de pessoas envolvidas, avaliar e chegar a um consenso se uma solução é ou não suficiente para o escopo de aplicações atuais da *blockchain*. Criar padrões de qualidade, arquitetura, taxonomia e ontologia são métodos por meio dos quais será possível lidar com a complexidade inerente deste cenário.

O primeiro avanço em direção à padronização é eliminar a ambiguidade onde quer que ela exista. O principal ponto debatido em (HOFMANN et al., 2017) diz respeito à ambiguidade dos dados, no qual destaca-se que, embora tecnicamente os dados sejam considerados imutáveis, eles ainda podem estar errados. O mecanismo de consenso deve estar alinhado para verificar a informação de entrada, de forma a tornar o sistema confiável para que os usuários possam deliberar sobre os dados com segurança.

A imutabilidade do mecanismo de consenso, no entanto, é algo bastante controverso. Isso porque nenhum código é criado em seu perfeito estado desde a primeira implantação. A adaptabilidade do *software* da *blockchain* é a forma de garantir que o algoritmo permaneça funcional durante todo seu ciclo de vida. Correções e melhorias são o diferencial na manutenção dentro deste tipo de programa, muito porque a tecnologia ainda tem bastantes pontos a melhorar. Neste cenário, os *smart contracts* tornam-se uma boa abordagem para trabalhar com dados dentro de uma *blockchain,* pois garante a validação automática sobre determinada condição para toda entrada do ecossistema.

3.3. Current Research On Internet Of Things (IoT) Security: A Survey

O avanço do paradigma da *Internet* das Coisas ocorre com rapidez em virtude da expansão contínua dos novos meios de comunicação. A segurança neste ambiente, no entanto, é notadamente um problema. A integração do físico com o digital, aumenta os riscos de ciberataques, e neste cenário, aplicar medidas de proteção pertencentes à *Internet* tradicional não é a solução mais adequada. Isso ocorre em virtude da natureza do ecossistema de IoT, que mistura inúmeros dispositivos, protocolos e recursos somados à grande escalabilidade de nodos em uma rede distribuída (NOOR & HASSAN, 2018).

Os ataques que existem dentro do cenário de *Internet* das Coisas estão distribuídos de acordo com a arquitetura de três camadas deste ambiente. A primeira delas é a camada de *hardware,* a segunda é a de rede e finalmente, a terceira, é a camada de aplicação ou serviço. A construção de um ecossistema seguro, neste sentido, inicia-se a nível de arquitetura. Planejar componentes com base em protocolos de segurança visa reduzir o número de vulnerabilidades e riscos, seja pelo acesso direto ou indireto ao dispositivo.

Neste contexto, redes inseguras permitem a operação de *hackers* mesmo em ambientes teoricamente controlados. Sistemas de nuvem vulneráveis ou desatualizados, *gateways* que operam entre dispositivos e falta de mecanismos de criptografia na transmissão de dados são fatores chave para a ocorrência de ataques que visam violar a privacidade, confidencialidade e disponibilidade do sistema.

Para garantir um processo seguro de fim-a-fim, é possível trabalhar com mecanismos de criptografia de dados, *trust management* e *secure routing*. Estes, por sua vez, têm o intuito de

garantir que os recursos que circulam por uma rede distribuída permaneçam seguros. Os processos de cibersegurança devem ser implementados como medida de proteção neste cenário, mas não devem ser encarados como suficientes para garantir a segurança completa do ambiente. Novas vulnerabilidades estão surgindo de forma constante, e elas introduzem sempre novas ameaças. Os dispositivos embarcados inseridos no contexto da *Internet* das Coisas permanecerão vulneráveis se a manutenção de códigos de correção em *software* e *firmware* não for feita corretamente.

De forma geral, a segurança em IoT envolve todas as camadas de sua arquitetura. Atualmente, a produção de dispositivos e sistemas neste cenário trabalha com soluções que propõem a mitigação dos danos ao invés de sua eliminação. Modelar riscos e propostas que sejam adequadas ao ambiente de *Internet* das Coisas é a alternativa mais segura para contornar os cada vez mais frequentes ataques a este ecossistema.

3.4. Sobre o uso de Blockchain em soluções com Credenciais Verificáveis e Identidades Auto-Soberanas

O conceito de Identidade Auto-Soberana é uma nova solução que surge para colocar o usuário em total controle sobre seus dados, tornando-o auto-suficiente em relação à gestão de suas credenciais. Este modelo torna-se relevante através das implementações de Credenciais Verificáveis, especificação prática elaborada pela *World Wide Web Consortium* (W3C). O estudo feito pelos autores neste trabalho identifica profunda relação entre as implementações de Identidades Auto-Soberanas com *blockchain*, que utilizam esta tecnologia para estruturar o componente responsável por verificar a autenticidade das credenciais (XAVIER et al., 2021).

A camada de Registro de Dados Verificáveis é responsável, dentre inúmeras finalidades, por criar e manter relações de confiança entre as entidades do sistema através da divulgação dos esquemas que definem uma Credencial Verificável. Esta relação é importante para que uma credencial seja emitida somente em nome do portador autorizado. Desta forma, todas as partes envolvidas em uma transação que envolve a troca e validação de credenciais estão seguras quanto a autenticidade da identidade de cada sujeito.

A troca de chaves criptográficas é uma das principais maneiras de estabelecer relações confiáveis em um sistema de identidade descentralizada. A W3C mantém como referência de

algoritmos de criptografia assimétrica para assinatura digital o RSA e o ED25519, amplamente utilizados na indústria. Como em qualquer processo de validação de assinatura digital, é necessário que a chave pública associada à entidade portadora seja utilizada para garantir a autenticidade da credencial.

Em estruturas de Identidade Auto-Soberana, cada entidade publica suas chaves públicas em um banco de dados distribuído. O retorno deste processo é um identificador único descentralizado - *Decentralized Identifier,* DID -, por meio do qual o sujeito pode recuperar as chaves públicas de outros emissores. Desta forma, cada usuário inserido no sistema pode validar a identidade de qualquer outro envolvido no processo de verificação de identidade.

Quando um identificador descentralizado refere-se a um emissor, é necessário que o seu armazenamento seja feito em estruturas de *blockchain,* para garantir a imutabilidade dos dados. Desta forma, verificadores e portadores podem facilmente ter certeza que se comunicam com a entidade correta, na qual confiam. Neste cenário, a camada de Registro de Dados Verificáveis é essencial para armazenar as chaves públicas das entidades emissoras.

A comunicação envolvendo o portador, por sua vez, é ligeiramente diferente, pois não é aconselhável que os mecanismos de armazenamento exponham a identidade do sujeito. Desta forma, o uso de *blockchain* não é aconselhado pelos autores; no entanto, é possível encontrar fluxos em que o uso de estruturas distribuídas são utilizadas de qualquer forma, principalmente em redes distribuídas privadas.

Existem ainda outros usos para a tecnologia de *blockchain* associada ao contexto de Identidade Auto-Soberana. São cenários em que esta estrutura é utilizada para controlar um banco de dados de revogações sem divulgar informações da credencial em si. Segundo os autores, é possível observar este cenário de forma paralela a outras tecnologias utilizadas para gestão de revogação, como registros centralizados, por exemplo. O uso da *blockchain* torna possível distribuir a solução como um todo, e não apenas em partes determinadas, de forma a suprir a demanda completa por descentralização.

4. Desenvolvimento de um Sistema de Manutenção de Identidade Descentralizada

A ideia central para o desenvolvimento deste trabalho recorre da necessidade de autenticar dispositivos móveis e embarcados em um sistema de coleta, identificação e preservação de evidências físicas e digitais. A arquitetura proposta, que é adaptada de (SCHMITT & STEIL, 2021) e está descrita na figura 16, utiliza as ferramentas anteriormente citadas para construir este sistema. A estrutura de criação e manutenção da Identidade Descentralizada está posicionada entre o dispositivo e o Gerenciador de Coleta e Visualização de Evidências.

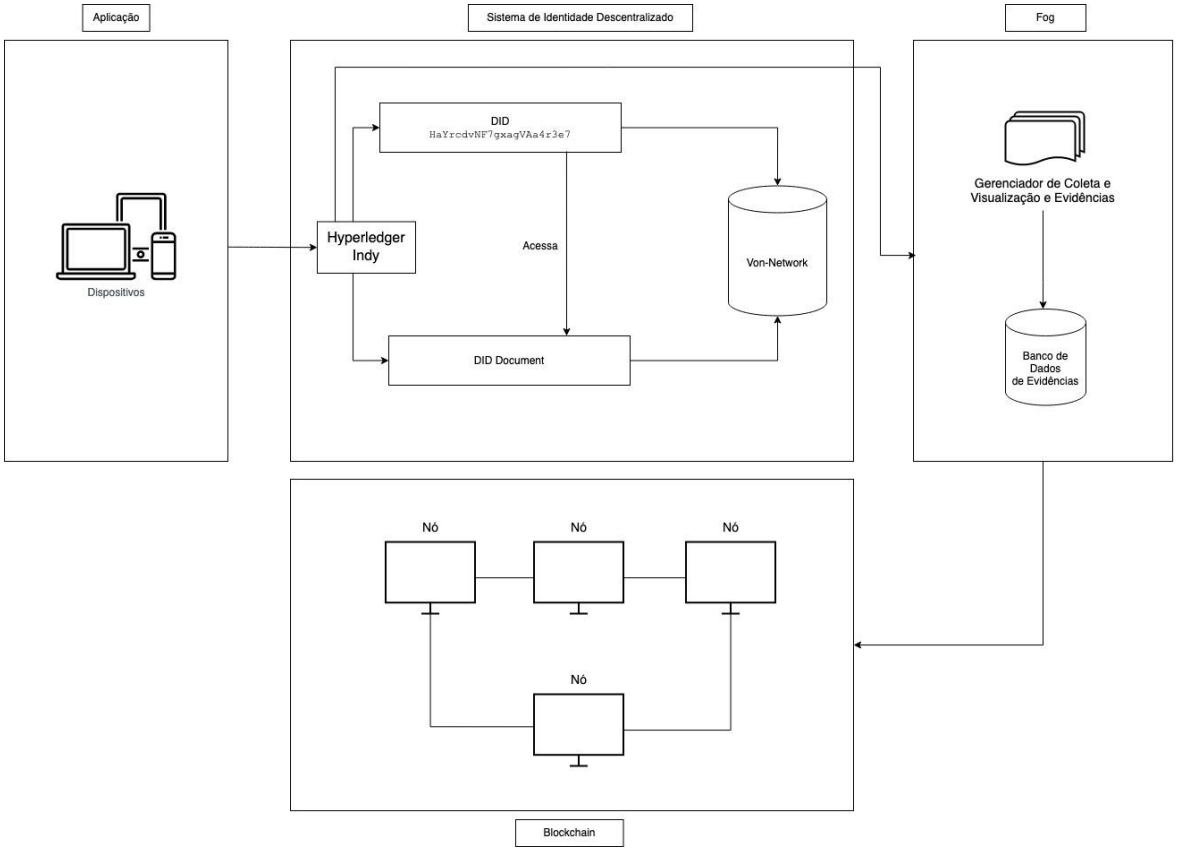

Figura 16 - Arquitetura de um Sistema de Identidade Descentralizado baseado na Recomendação Oficial da W3C para Identificadores Descentralizados e aplicado no cenário descrito em (SCHMITT & STEIL, 2021).

Dentro deste contexto, o gerenciador da Identidade Descentralizada torna-se um componente essencial à implementação do sistema como um todo, pois é responsável pela criação de DIDs no processo de identificação de evidências. Desta forma, o objetivo final deste trabalho foi criar um Sistema de Manutenção de Identidade Descentralizada, baseado na especificação do

framework Hyperledger Indy. Este sistema funciona integrado com a Von-Network, utilizada para gerenciar Identificadores Descentralizados. Na figura 17 está descrito o recorte da arquitetura final do projeto.

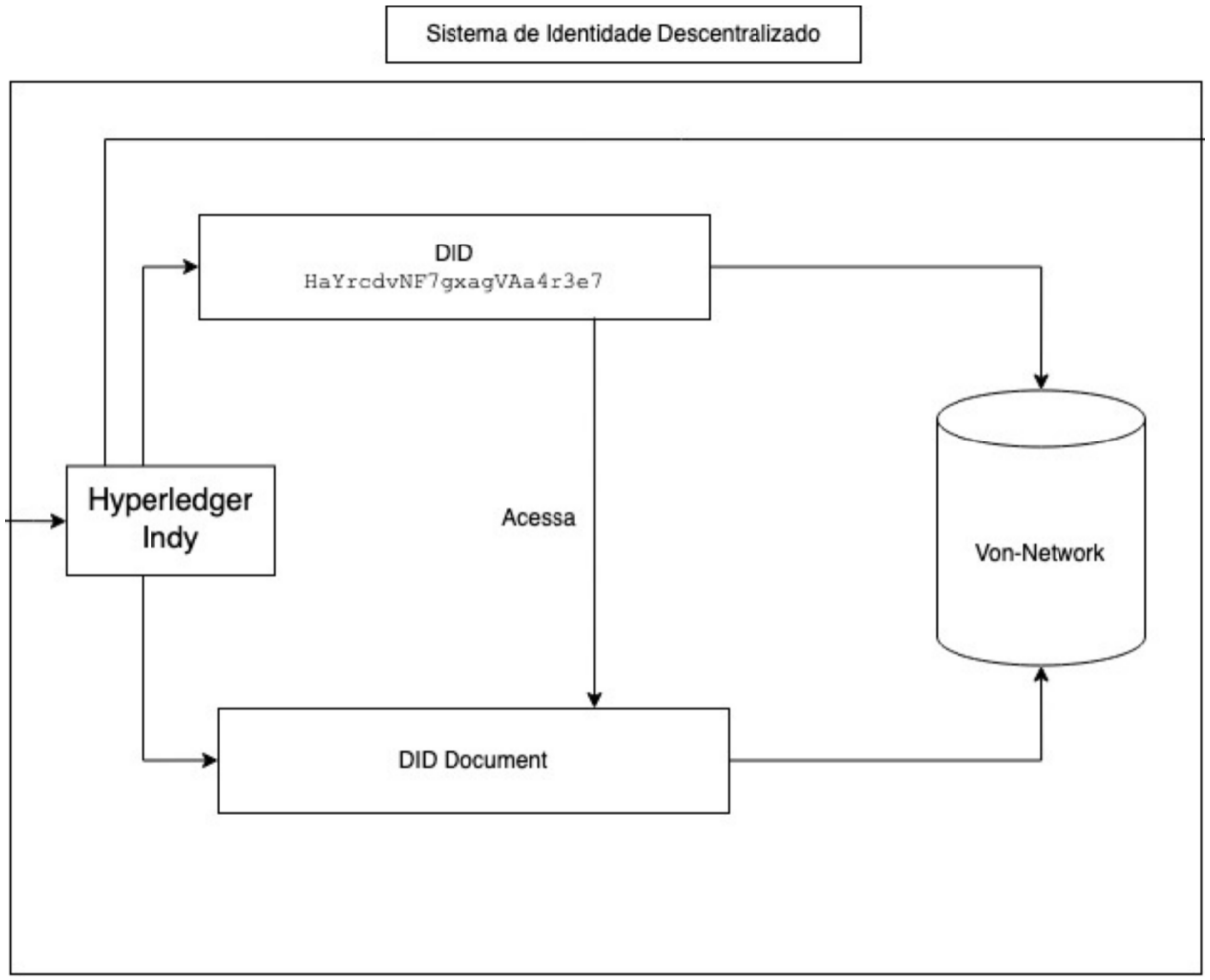

Figura 17 - Sistema de Manutenção de Identidade Descentralizada

Na primeira versão desta arquitetura, refletiu-se sobre o uso do componente Aries, *framework* também provisionado pela Hyperledger Foundation, para criar e manter em funcionamento um sistema de Identidade Descentralizada. Por meio da implementação do Hyperledger Aries, o ACA-Py, é possível criar e gerenciar Identificadores Descentralizados gerados a partir de *schemas* personalizados, que permitem especificar Documentos DID voltados ao contexto da autenticação de dispositivos móveis e embarcados como evidências físicas.

No entanto, não foi possível dar sequência na construção do sistema com o uso deste *framework* devido a alta complexidade do ACA-Py. Esta ferramenta, componente essencial na

integração do sistema, é uma implementação extensa e complexa. Por este motivo, decidiu-se prosseguir com um sistema mínimo viável, através da implementação do componente Hyperledger Indy, também utilizado para emissão e validação de Identidades Descentralizadas.

Neste sentido, prosseguiu-se com o estudo da especificação Indy-CLI integrado à Von-Network para posterior desenvolvimento do Sistema de Manutenção de Identidade Descentralizada, baseado em um *schema* padrão disponibilizado pela estrutura da Von-Network. Este sistema funciona a partir do mecanismo de linha de comando do *Hyperledger Indy*.

4.4. Análise de Código: Indy-CLI

O processo de criação de um DID começa no arquivo `generate_did.py`, em que a função `generate_did(seed)`, descrita na figura 18, cria uma instância do holder por meio da classe `AnchorHandle()`. A classe `AnchorHandle` possui inúmeros parâmetros, entre eles o DID e a *verkey* do usuário. Estes atributos iniciam-se vazios, pois somente serão populados após a chamada do método `seed_to_did(seed)`.

```
def generate_did(seed):
        TRUST_ANCHOR = anchor.AnchorHandle()
        did, verkey = TRUST_ANCHOR.seed_to_did(seed)

        print(f"\nSeed: {seed}")
        print(f"DID: {did}")
        print(f"Verkey: {verkey}")

        return did, verkey
```

Figura 18 - Código da função `generate_did(seed)`.

O método `seed_to_did(seed)` é responsável por gerar um Identificador Descentralizado e uma *verkey*, chave pública verificável. Este método chama a função

`nacl_seed_to_did(seed)`. Utilizando uma versão decodificada em bytes da *seed*, criada por meio da função `seed_as_bytes(seed)` que é descrita na figura 19, a função `nacl_seed_to_did(seed)`, a partir da biblioteca *nacl*, do Python, produz uma chave de assinatura com base na *seed* recebida. Por meio da chave de assinatura, também chamada de "Signing Key", é extraida a *verkey*. Uma *verkey*, em português "Chave de Verificação", é uma chave pública verificável associada a um DID e derivada a partir da *seed*, utilizada para verificar assinaturas digitais para provar as propriedades de um Identificador Descentralizado.

```
def seed_as_bytes(seed):
        if not seed or isinstance(seed, bytes):
        return seed
        if len(seed) != 32:
        return base64.b64decode(seed)
        return seed.encode("ascii")
```

Figura 19 - Código da função `seed_as_bytes(seed)`.

Após derivar a Chave de Verificação, a função `nacl_seed_to_did(seed)`, descrita na figura 20, cria um Identificador Descentralizado para o usuário. A criação de um DID é feita a partir da *seed* em *bytes*, por meio do algoritmo de codificação *base58*, que codifica um trecho da *verkey* em bytes em um DID, e decodifica o resultado no padrão *ascii*. O algoritmo base58 é utilizado para compactar a representação de dados em forma binária.

Por fim, a Chave de Verificação completa é codificada através do algoritmo base58 para obter a *verkey*. A função retorna então o Identificador Descentralizado e a Chave de Verificação.

```
def nacl_seed_to_did(seed):
        seed = seed_as_bytes(seed)
        vk = bytes(nacl.signing.SigningKey(seed).verify_key)
        did = base58.b58encode(vk[:16]).decode("ascii")
        verkey = base58.b58encode(vk).decode("ascii")
        return did, verkey
```

Figura 20 - Código da função `nacl_seed_to_did(seed)`.

O *ledger* é o local em que o Documento DID é armazenado. Cada Identificador Descentralizado registrado na Blockchain da Von-Network possui todos os seus metadados associados disponíveis para consulta. Na Von-Network, quando entramos na página **Domain**, descrita na figura 21, temos acesso a todos os DIDs registrados na rede distribuída, e na aba **Raw Data**, descrita na figura 22, estão disponíveis os *schemas* e Documentos DID registrados no *ledger*.

Figura 21 - Estados do *ledger* da Von-Network.

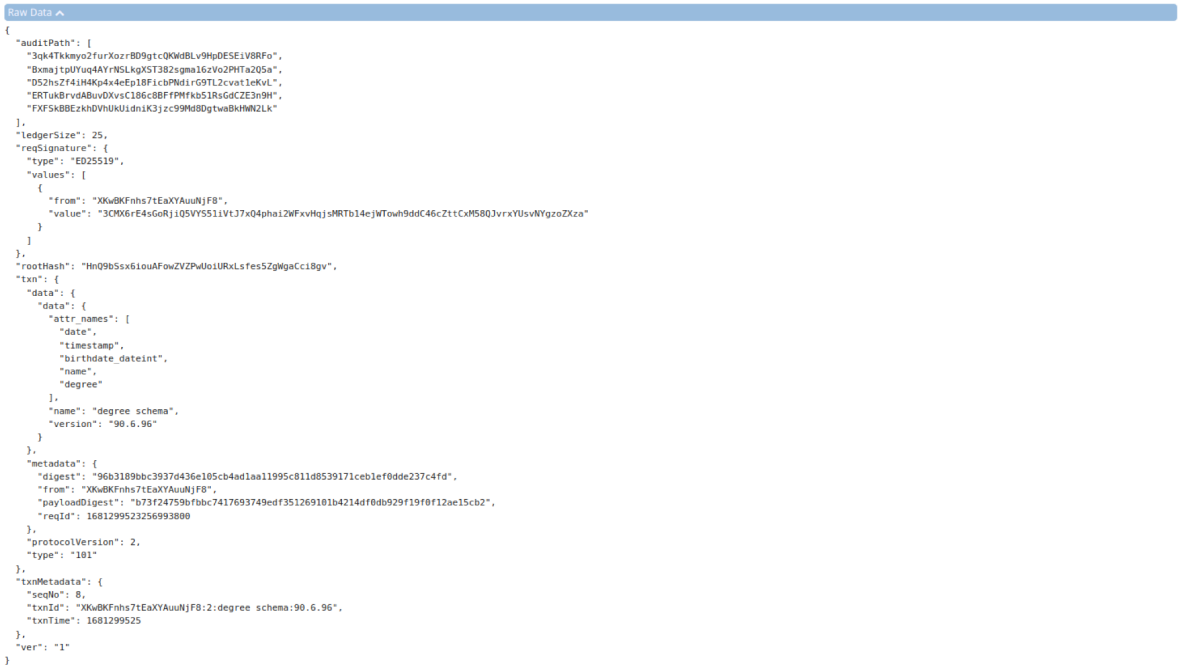

Figura 22 - Aba Raw Data.

Porque o Documento DID é armazenado como uma transação no *ledger*, o Indy CLI também é usado para interagir com este tipo de registro. Quando o Indy CLI é utilizado para criar ou atualizar um Documento DID, uma transação é enviada para a rede, que inclui a atualização no *ledger*. A transação é então processada pelo algoritmo de consenso da rede e, uma vez

alcançado o consenso, o Documento DID atualizado torna-se parte do histórico do *ledger,* conforme descrito na figura 23. Posteriormente, outros participantes da rede podem acessar e recuperar o documento DID consultando o *ledger* usando o DID associado.

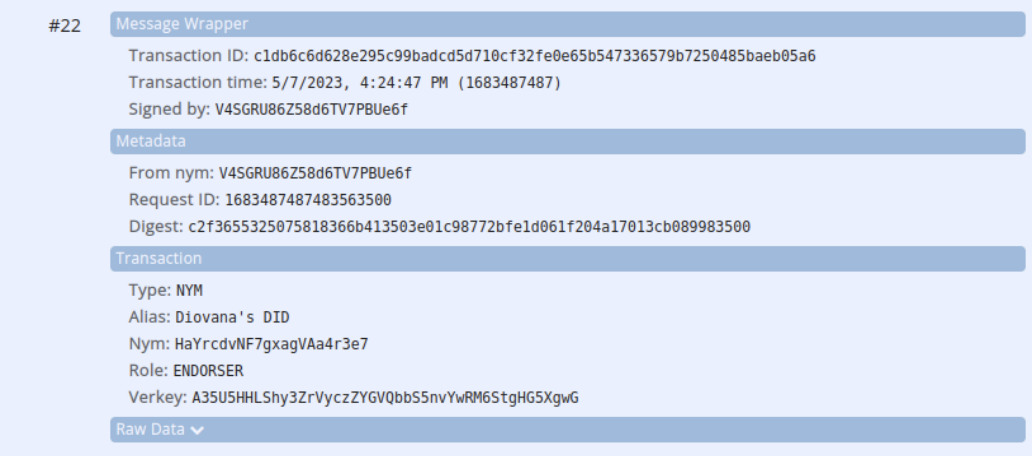

Figura 23 - Registro do Identificador Descentralizado “Diovana’s DID” no *ledger*.

Na rede distribuída, existem dois tipos de DID. O primeiro tipo é NYM. Um DID desta classe é o Identificador Descentralizado de um sujeito. A segunda classe é do tipo pseudônimo, utilizada para garantir a privacidade dentro da comunicação entre dois participantes. Quando um pseudônimo é utilizado apenas para o fim anteriormente citado, ele pode ser também chamado de Identificador *Pairwise*, responsável por manter uma conexão entre dois clientes segura.

De acordo com o texto do tutorial **Explore HyperLedger Indy Command Line Interface**, no sistema da Von-Network, Identificadores Descentralizados podem ser diferenciados em dois papéis diferentes: *Steward* e *Trust Anchor.* O sujeito cuja função é operar a rede distribuída Indy possui o papel *Steward*, o qual é responsável por criar e configurar os nós da rede e também por criar, atualizar ou excluir DIDs por meio de transações NYM. O papel Trust Anchor, por sua vez, é utilizado para somente publicar Identificadores Descentralizados e *schemas* no ledger.

4.5. Sistema de Manutenção de Identidade Descentralizada

Para este trabalho, desenvolveu-se um Sistema de Manutenção de Identidade Descentralizada baseado no código do Indy-CLI. Utilizou-se as funções do *framework* para criar um servidor que gera *seeds* e Identificadores Descentralizados. Este servidor é uma API Rest de duas rotas GET, sendo elas `[GET] /seed` e `[GET] /did?seed={seed}`.

Este servidor foi construído na linguagem Python, utilizando o *microframework* Flask. O Flask é destinado a aplicações simples que possuem poucos requisitos que não sejam muito complexos. Construído sobre um núcleo simples e expansível, permite ao desenvolvedor selecionar quais *features* serão utilizadas, de forma modular.

No código do servidor de Manutenção de Identidade Descentralizada, a rota `[GET] /seed`, descrita na figura 24, é responsável por criar uma *seed* válida para o usuário. Para isso, uma nova instância da classe `AnchorHandle`, retirada do código do Indy-CLI, é criada. A partir dela dois métodos são chamados: `generate_key(key_length)`, descrito na figura 25, e `generate_seed()`, descrito na figura 26 Estes foram criados a partir do script da aplicação de linha de comando do Indy-CLI.

```
@app.route('/seed',methods=['GET'])
def create_seed():
        anchor = AnchorHandle()

        return jsonify(**{
        "seed": anchor.generate_seed(),
        "key":anchor.generate_key(48)
        })
```

Figura 24 - Função `create_seed()`.

```
def generate_key(self, key_lenght=48):
stream = os.popen(f'openssl rand -base64 {key_lenght}')
output = stream.read()
return output[:-1]
```

Figura 25 - Método `generate_key(key_length)`.

```
def generate_seed(self):
key = self.generate_key(32)
stream = os.popen(f'echo "{key}" | fold -w 32 | head -n 1')
output = stream.read()
return output[:-1]
```

Figura 26 - Método `generate_seed()`.

A função `popopen(cmd)` do pacote nativo *os* do Python é utilizada para abrir um canal de um comando. Por meio desta, na função `generate_key(key_length)`, utilizou-se o comando `openssl rand -base64 {key_length}` para gerar um conjunto de caracteres pseudo-aleatórios através do método de codificação *base64*. O retorno da função *`generate_key(key_length)`* é a string gerada pelo comando, cortando o último caractere, que corresponde a uma quebra de linha (**\n**).

Já a função `generate_seed()` cria, por meio do método `generate_key(key_length)`, uma chave de 32 caracteres. Em um canal de comando, são aplicadas regras de formatação sobre a string recém criada. Por fim, retorna-se a chave, mais uma vez cortando o último caractere, que também corresponde a uma quebra de linha.

A figura 27 demonstra o fluxo de interação entre o cliente e a aplicação, descrita em suas camadas de *handler* e *service*.

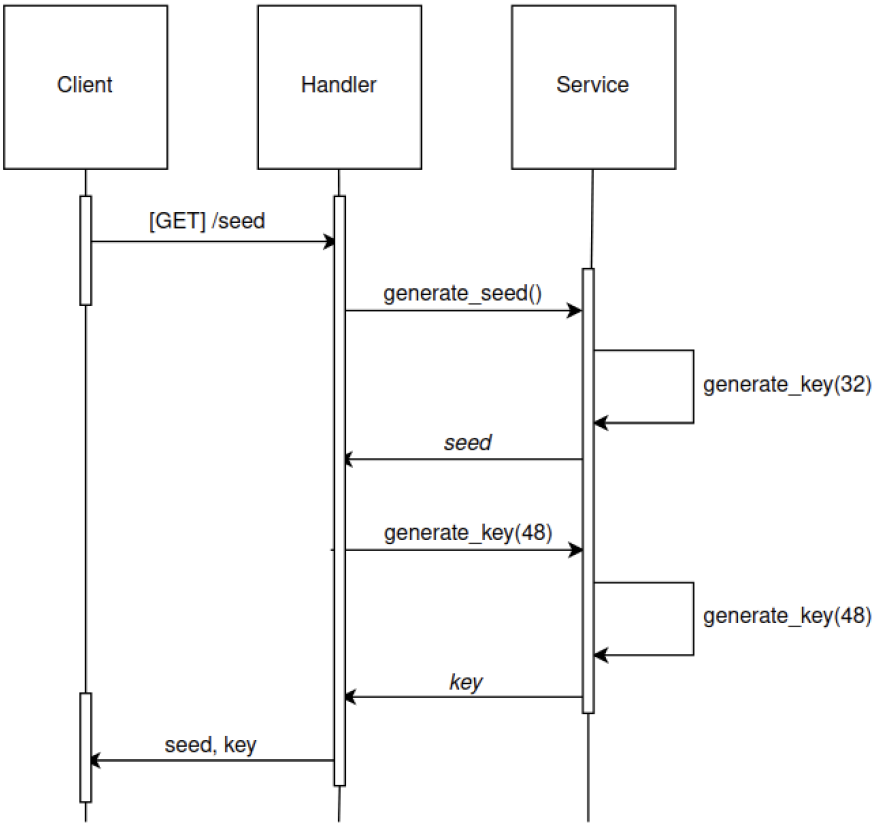

Figura 27 - Fluxo de Execução do Endpoint `[GET] /seed`

Já a rota `[GET] /did`, descrita na figura 28, é o *path* em que é gerado um novo Identificador Descentralizado a partir de uma *seed*. Essa função recebe um *query param seed*.

```
@app.route('/did',methods=['GET'])
def create_did():
      seed = request.args.get('seed')

      if seed is None:
      return jsonify(
            message="missing seed query param",
            status="bad request"
      ), 400

      did, verkey = generate_did.generate_did(seed)
      return jsonify(
      did=did,
      verkey=verkey
      ), 200
```

Figura 28 - Função `create_did()`.

A função `create_did()` começa recuperando o *query param seed*. Se este parâmetro for nulo, o servidor retorna um status 400 Bad Request, informando ao cliente que a função funciona somente quando o parâmetro seed existe. Se a condição `if seed is None` for falsa, o processamento prossegue, chamando a função `generate_did(seed)`, descrita na figura 29, que é importada do módulo `generate_did`, definido dentro do código do Indy-CLI.

```
def generate_did(seed):
      TRUST_ANCHOR = anchor.AnchorHandle()
      did, verkey = TRUST_ANCHOR.seed_to_did(seed)

      print(f"\nSeed: {seed}")
      print(f"DID: {did}")
      print(f"Verkey: {verkey}")

      return did, verkey
```

Figura 29 - Função `generate_did(seed)`.

A função `generate_did(seed)` cria uma instância da classe AnchorHandle, por meio da qual o Identificador Descentralizado e a *verkey* serão derivados através do método `seed_to_did(seed)`. Conforme citado anteriormente, este método utiliza a função `nacl_seed_to_did(seed)`, descrita figura 30, para produzir a chave privada do usuário com base na *seed* recebida, e por meio desta, extrair a *verkey*. O DID é criado por meio da codificação da *verkey* em bytes, que por fim será decodificada no padrão *ascii*. A função retorna então o Identificador Descentralizado e a Chave de Verificação.

```
def nacl_seed_to_did(seed):
      seed = seed_as_bytes(seed)
      vk = bytes(nacl.signing.SigningKey(seed).verify_key)
      did = base58.b58encode(vk[:16]).decode("ascii")
      verkey = base58.b58encode(vk).decode("ascii")
      return did, verkey
```

Figura 30 - Função `nacl_seed_to_did(seed)`.

A figura 31 demonstra o fluxo de interação entre o cliente e a aplicação, descrita em suas camadas de *handler* e *service*.

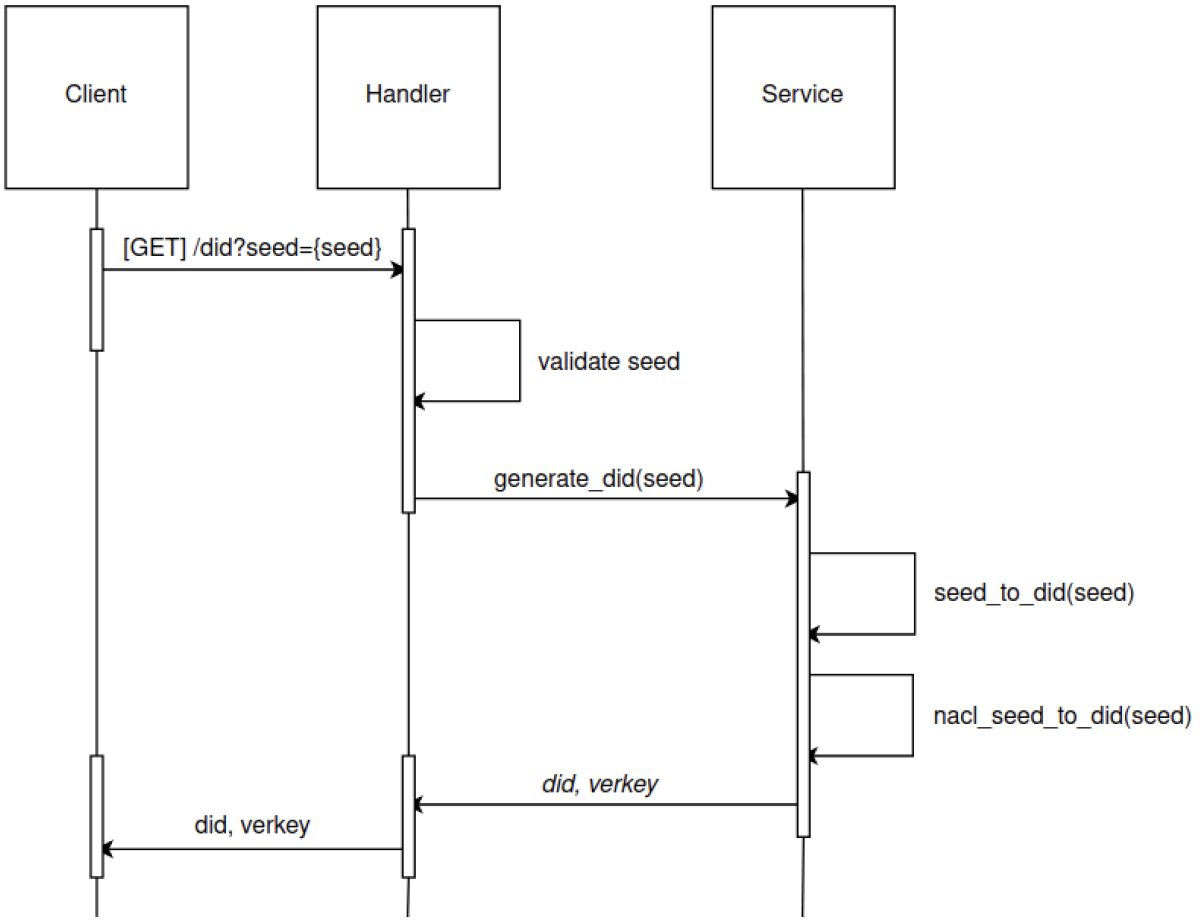

Figura 31 - Fluxo de Execução do Endpoint `[GET] /did?seed={seed}`

5. Conclusões

A proposta final deste trabalho consistiu no estudo das ferramentas Indy-CLI e Von-Network, utilizadas em conjunto para criar um Sistema de Manutenção de Identidade Descentralizada, no intuito de criar Identificadores Descentralizados e gerenciar *seeds* e chaves de verificação. Em tese, utilizou-se as linhas de comando do Indy-CLI para emitir os requisitos necessários para a criação de um DID. Tanto o Identificador Descentralizado como o Documento DID são armazenados na Von-Network, que funciona como o *ledger* do sistema.

Inicialmente, este sistema deveria funcionar com um *schema* personalizado voltado ao contexto da identificação de dispositivos IoT em uma cadeia de custódia. No entanto, o *framework* por meio do qual a implementação deste sistema dependia demonstrou grande complexidade, impeditiva para a sequência deste trabalho na forma como ele foi planejado. No entanto, apesar de tudo, o estudo das ferramentas anteriormente citadas gerou um conhecimento satisfatório sobre o funcionamento prático da criação e registro de Identificadores Descentralizados, tema central deste trabalho.

A blockchain proporciona uma base sólida para a implementação do Sistema de Manutenção de Identidade Descentralizada proposto, pois os componentes-chave necessários para garantir a emissão de identificadores únicos de dispositivos são amplamente apoiados pelos princípios da arquitetura distribuída da *blockchain*. Tanto o Indy-CLI quanto o Von-Network se beneficiam da natureza descentralizada da *blockchain*, permitindo a criação de Identificadores Descentralizados e o gerenciamento seguro de *seeds* e chaves de verificação.

De forma sucinta, foi possível compreender a importância do Indy-CLI no contexto de um Sistema de Manutenção de Identidade Descentralizada a partir do amplo estudo da ferramenta. É por meio desta que ocorre a interação com os componentes necessários para a criação e o gerenciamento dos DIDs, proporcionando uma compreensão mais aprofundada dos processos envolvidos. Este *framework* é, em sua essência, escrito para funcionar com outras ferramentas, como a Von-Network, por fornecer uma interface padrão e compatível com a especificação deste componente.

Também neste contexto, a Von-Network possui extrema importância, uma vez que foi o mecanismo por meio do qual tornou-se possível armazenar DIDs e seus respectivos

documentos de forma segura, garantindo integridade e proteção contra acesso não autorizado. Isso ocorre devido à integração com a *blockchain,* que proporciona imutabilidade dos registros e a capacidade de auditoria, para garantir a confiabilidade e a rastreabilidade das informações armazenadas. A simulação de um ambiente descentralizado também contribuiu na demonstração de um ambiente resiliente e disponível.

A combinação do Indy-CLI e do Von-Network em um Sistema de Manutenção de Identidade Descentralizada ofereceu uma abordagem promissora para a emissão e gestão de DIDs. Ao aproveitar os princípios da *blockchain*, esse sistema proporciona confiança, segurança e eficiência na identificação de dispositivos sobre um *schema* padrão.

Imagina-se como prováveis trabalhos futuros a realização de estudos aprofundados sobre a ferramenta ACA-Py para a implementação de *schemas* personalizados voltados ao contexto da Internet das Coisas. A criação de *schemas* personalizados possibilita a gestão de dispositivos IoT no contexto da Computação Forense por meio de Identificadores Descentralizados, colaborando para a manutenção de um ambiente à prova de violações.

6. Referências

ALLEN, C. "**Ideology & Architecture of Self-Sovereign Identity**", 2020. In: Odyssey Connect 2020. Disponível em: https://bit.ly/3AFD92X. Acesso em: 20/11/2022.

AVELLANEDA, Oscar. et al., "**Decentralized Identity: Where Did It Come From and Where Is It Going?**", 2019. In: IEEE Communications Standards Magazine, vol. 3, no. 4, pp. 10-13, doi: 10.1109/MCOMSTD.2019.9031542.

ASHTON, K. "**That 'Internet of Things' Thing**", 2011. In: RFID Journal, v. 22, n. 7.

BASSI, A.; HORN, G. "**Internet of Things in 2020: A Roadmap for the Future**", 2008. In: European Commission: Information Society and Media, v. 22, p. 97-114.

BLUMMER, Tamas et al. "**An Introduction to Hyperledger**", 2018. [S.l.: s.n.].

BRASIL, Lei nº 13.964, de 24 de dezembro de 2019. **Aperfeiçoa a legislação penal e processual penal**. Diário Oficial da União: seção 1, Brasília, DF.

CAMPBELL-KELLY, Martin et al. "**Computer: A History of the Information Machine**", 2014. 3a Edição. Westview Press.

CASTELLS, M. "**The Rise of the Network Society: The Information Age: Economy, Society, and Culture**", 2011. John Wiley & Sons.

CARRION, P.; QUARESMA, M. "**Internet das Coisas (IoT): Definições e aplicabilidade aos usuários finais**", 2019. Florianópolis: Human Factors in Design.

CHICARINO, Vanessa R. et al. "**Uso de Blockchain para Privacidade e Segurança na Internet das Coisas**", 2017. In: Rio de Janeiro: Universidade Federal do Rio de Janeiro.

COSTA, Rafael Paes; GARCIA, Raphael. "**Princípios e Análises da Computação Forense"**, 2014. In: Toledo Prudente Centro Universitário.

CURREN, Sam. "What is DIDComm?", 2022. Disponível em: https://indicio.tech/what-is-didcomm-with-pictures/

DE AGUIAR, Erikson Júlio, et al., "**A Survey of Blockchain-Based Strategies for Healthcare**", 2021. ACM Comput. Surv. 53, 2, Article 27, 27 pages.

DE GODOI, Maiko Gustavo; DE ARAÚJO, Liriane Soares. "**A INTERNET DAS COISAS: evolução, impactos e benefícios**", 2019. Bom Retiro: FATEC.

DI PIERRO, M., "**What Is the Blockchain?**", 2017. In: Computing in Science & Engineering, vol. 19, no. 5, pp. 92-95.

DOGAN, Omer. "**Developing Digital Identity Applications Using Hyperledger Indy, Ursa and Aries Frameworks**", 2021. In: Medium. Disponível em: https://medium.com/stm-blockchain/developing-applications-using-hyperledger-indy-ursa-and-aries-frameworks-concepts-7c5955b8cc1

DOMINGUESCHE, Felipe Barbosa. "**Desenvolvimento de uma plataforma de Internet das Coisas (IoT) integrada a redes de sensores sem fio**", 2021. São Paulo: FATEC.

EASTERLING, Keller. "**An Internet of Things**", 2012. In: New Haven: E-flux Journal.

FDHILA, W., et al. "**Methods for Decentralized Identities: Evaluation and Insights**", 2021. In: González Enríquez, J., Debois, S., Fettke, P., Plebani, P., van de Weerd, I., Weber, I. (eds) Business Process Management: Blockchain and Robotic Process Automation Forum. BPM 2021. Lecture Notes in Business Information Processing, vol 428. Springer, Cham. https://doi.org/10.1007/978-3-030-85867-4_9

GRÜNER, A., et al., "**A comparative analysis of trust requirements in decentralized identity management**", 2019. In: International Conference on Advanced Information Networking and Applications. pp. 200–213. Springer, Cham.

GUSSON, Cássio, "**O que é o Hyperledger? Conheça a plataforma para construção de blockchains privadas**", In: CriptoFacil, Disponível em: https://www.criptofacil.com/o-que-e-o-hyperledger-conheca-a-plataforma-para-construcao-de-blockchains-privadas/. Acesso em 28/10/2022.

HAI, Huang et al., "**An SDN based management framework for IoT devices**", 2014.

HARDMAN, Daniel. "**DID Communication**", 2019.

HJÁLMARSSON, F. Þ., et al., "**Blockchain-Based E-Voting System**", 2018. In: IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 983-986, doi: 10.1109/CLOUD.2018.00151.

HOFMANN, Frank et al. "**The immutability concept of blockchains and benefits of early standardization**", 2017. In: ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K), pp. 1-8.

KAMAL, Randa et al. "**Forensics chain for evidence preservation system: An evidence preservation forensics framework for internet of things-based smart city security using blockchain**", 2022. In: Menoufia, Egypt: Wiley.

KHAN, Minhaj Ahmad & SALAH, Khaled. "**IoT security: Review, blockchain solutions, and open challenges**". 2018. In: Future Generation Computer Systems, Vol. 82, Pages 395-411, ISSN 0167-739X.

KIM, Bong Gon et al. "A Security Analysis of Blockchain-Bases DID Services", 2021. In: Institute of Information and Communications Technology.

KUROSE, J. F. and ROSS, K. W. "**Computer Networking: A Top-Down Approach**", 2012. Pearson, 6th edition.

LAMOUNIER, Lucas. "**O Guia Definitivo da Tecnologia Blockchain: Uma Revolução Para Mudar o Mundo**", 2018. In: 101 Blockchains. Disponível em: https://101blockchains.com/pt/tecnologia-blockchain-guia/. Acesso em 10/10/2022.

LIU, J., XIAO, Y., e CHEN, C. P. (2012). "**Authentication and access control in the internet of things**". In Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, pages 588–592. IEEE.

LUX, Zoltán András, et al., "**Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials**" 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2020.

MAURO, Faccioni Filho. "**Internet das Coisas**", 2016. In: Universidade do Sul de Santa Catarina.

MIERS, Charles et al. "**Análise de Mecanismos para Consenso Distribuído Aplicados a Blockchain**", 2019. In: Minicursos do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais.

MIRANDA, Javier et al. "**From the Internet of Things to the Internet of People**", 2015. In: IEEE Internet Computing, Vol. 19, n. 2, p. 40-47.

MOREIRA Kleber Brito. "**Blockchain - Tecnologia, Arquitetura e Aplicações**", 2019. Brasília: Universidade de Brasília.

NAKAMURA, Emilio Tissato et al., "**Identidade Digital Descentralizada: Conceitos, aplicações, iniciativas, plataforma de desenvolvimento e implementação de caso de uso**", 2019. In: XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, set 2019.

NOOR, Mardiana binti Mohamad & HASSAN, Wan Haslina. "**Current research on Internet of Things (IoT) security: A survey**", 2019. In: Computer Networks, Vol. 148, p. 283-294.

OLIVEIRA, Nairobi Spiecker De et al. "**Segurança da Informação para Internet das Coisas (IoT): uma Abordagem sobre a Lei Geral de Proteção de Dados (LGPD**)", 2019. São Leopoldo: Universidade do Vale do Rio dos Sinos.

OMAR, A. Sghaier & BASIR, O. "**Capability-Based Non-fungible Tokens Approach for a Decentralized AAA Framework in IoT**", 2020. In: Choo, KK., Dehghantanha, A., Parizi, R. (eds) Blockchain Cybersecurity, Trust and Privacy. Advances in Information Security, vol 79. Springer, Cham. https://doi.org/10.1007/978-3-030-38181-3_2

PARK, Taehyeun et al. "**Learning How to Communicate in the Internet of Things: Finite Resources and Heterogeneity**", 2016. In: Virginia: IEEE Access, Vol 4.

PREUKSCHAT, A. & REED, D. "**Self-Sovereign Identity**", 2021. In: Manning Publications.

RIBEIRO Lucas & MENDIZABAL Odorico Machado. "**Introdução à Blockchain e Contratos Inteligentes: Apostila para Iniciantes**" 2019. Florianópolis, Universidade Federal de Santa Catarina.

REED, D. "**Decentralized Identifiers (DIDs): The Fundamental Building Block of Self-Sovereign Identity**", 2018. SSI MeetUP.

REVOREDO, Tatiana. "**Identidade Digital Auto-Soberana: O Gerenciamento de Identidades e a Capacidade de Provar Quem Somos**", 2019. In: Medium, Disponível em:

https://medium.com/global-blockchain-strategy/identidade-digital-auto-soberana-32f3ea297089

ROMKEY, J. "**Toast of the IoT: The 1990 Interop Internet Toaster",** 2017. In: IEEE Consumer Electronics Magazine, Vol. 6, No. 1, p. 116-119.

SANTOS, Bruno P. et al. "**Internet das Coisas: da Teoria à Prática**", 2016. Belo Horizonte: Universidade Federal de Minas Gerais.

SIQUEIRA, Alexandre et al., "**Blockchain e Identidades Digitais Descentralizadas: Fundamentos e Oportunidades**", 2021. In: Fundamentos e Tendências em Inovação Tecnológica, Vol. 2. Kindle Direct Publishing.

SCHMITT Henrique Hermes & STEIL Pedro Paulo Chiarelli. "**Preservação de Dados em Blockchain**", 2021. Florianópolis: Universidade Federal de Santa Catarina.

SRINIVASA, Roopha Shree Kollolu. "**A Review on Wide Variety and Heterogeneity of IoT Platforms**", 2020. In: The International journal of analytical and experimental modal analysis, analysis, Vol. 12, Issue 1, jan, Page No:3753-3760.

SWAN, Melanie. "**Blockchain: Blueprint for a new economy**", 2015. In: O'Reilly Media, Inc..

SWEENEY, Latanya. "**K-Anonymity: A Model for Protecting Privacy**". In: International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 10, No. 5 (2002) 557-570.

TAM, K.C. "**Exploring Hyperledger Indy through indy-dev Example**", 2019. In: https://kctheservant.medium.com/exploring-hyperledger-indy-through-indy-dev-example-10075d2547ae

UDDIN, Md Ashraf et al. "**A survey on the adoption of blockchain in IoT: challenges and solutions**", 2021. Blockchain: Research and Applications, Vol. 2, Issue 2, 100006, ISSN 2096-7209.

WESTPHALL, Johann. "**Blockchain privacy and scalability in a decentralized validated energy trading context with Hyperledger Fabric**", 2021. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2021.

XAVIER, Marcus R. et al., "**Sobre o uso de Blockchain em soluções com Credenciais Verificáveis e Identidades Auto-Soberanas**", 2021. In: XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, out 2021.

XU, Min et al. "**A systematic review of blockchain**", 2019. In: Financial Innovation. Vol. 5, No. 27.

YAGA, Dylan et al. Blockchain Technology Overview. "**National Institute of Standards and Technology Internal Report**", 2019. [*s. l.*].

ZHANG, Jingyu et al. "**Blockchain-based Systems and Applications: A Survey",** 2020. Journal of Internet Technology, [S.l.], v. 21, n. 1, p. 1-14, jan. ISSN 2079-4029.

0289: The Trust Over IP Stack. In: https://github.com/hyperledger/aries-rfcs/blob/main/concepts/0289-toip-stack/README.md. Acesso em: 30/06/2023.

A kickstart to Hyperledger Indy. In: Blockchain Simplified, 2020. Disponível em: https://blockchainsimplified.com/blog/a-kickstart-to-hyperledger-indy/. Acesso em 20/11/2022.

Aries RFC 0005: DID Communication. In: https://github.com/hyperledger/aries-rfcs/blob/main/concepts/0005-didcomm/README.md. Acesso em: 30/06/2023.

Como Instalar e Utilizar o Docker no Ubuntu 20.04
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-pt

Demonstration of Hyperledger Aries Cloud Agent. In: https://kctheservant.medium.com/demonstration-of-hyperledger-aries-cloud-agent-6e476a5426b0. Acesso em: 30/06/2023.

DIDComm Presentation at DID F2F, June 2020: https://youtu.be/s1Gum-hN3QM

Decentralized Identifiers (DIDs): Core architecture, data model, and representations. In: World Wide Web Consortium, 2022. Disponível em: https://www.w3.org/TR/did-core/. Acesso em 29/10/2022.

Descentralização da Identidade Digital. In: RNP Wiki Confluence, 2022. Disponível em https://wiki.rnp.br/pages/viewpage.action?pageId=160712659#. Acesso em: 29/10/2022.

Explore Hyperledger Indy Command Line Interface
https://www.myhsts.org/tutorial-learn-how-to-work-with-hyperledger-indy-command-line-interface.php

Hard and Soft Forks. In: CryptoGraphics, 2022. Disponível em: https://cryptographics.info/cryptographics/blockchain/hard-soft-forks/. Acesso em 29/10/2022.

How To Install Python 3 and Set Up a Programming Environment on Ubuntu 20.04
https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-programming-environment-on-ubuntu-20-04-quickstart

Hyperledger Indy. In: Hyperledger Foundation, 2022. Disponível em: https://www.hyperledger.org/use/hyperledger-indy. Acesso em 15/11/2022.

Indy-SDK. In: https://github.com/hyperledger/indy-sdk/blob/main/cli/README.md. Acesso em: 30/06/2023.

Representing IoT devices in BigchainDB using DID. In: https://github.com/WebOfTrustInfo/rwot5-boston/blob/master/draft-documents/did%20-%20iot%20-%20bigchaindb/iot%20devices%20with%20dids%20on%20bigchaindb.md

Trust Over IP Stack. In: https://www.thoughtworks.com/pt-br/radar/platforms/trust-over-ip-stack. Acesso em: 30/06/2023

Using the containerized Indy-CLI https://github.com/bcgov/von-network/blob/main/docs/Indy-CLI.md. Acesso em: 30/06/2023.

Verifiable Credentials Data Model 1.0: Expressing verifiable information on the Web. In: World Wide Web Consortium, 2019. Disponível em: https://www.w3.org/TR/vc-datamodel. Acesso em: 20/09/2022.

Verifiable Credentials Data Model v1.1. In: World Wide Web Consortium, 2022. Disponível em: https://www.w3.org/TR/2022/REC-vc-data-model-20220303/. Acesso em 20/09/2022.

Von-Network. In: https://github.com/bcgov/von-network. Acesso em: 30/06/2023.

What is Hyperledger Fabric?. In: IBM**,** Disponível em: https://www.ibm.com/downloads/cas/0XMOQJNP. Acesso em: 25/10/2022.

Apêndices

APÊNDICE A - RELATÓRIO NO FORMATO SBC

USO DE IDENTIFICADORES ÚNICOS DESCENTRALIZADOS PARA GESTÃO DE DISPOSITIVOS IOT NO CONTEXTO DA COMPUTAÇÃO FORENSE

Diovana Rodrigues Valim[1]

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

diovana.r.valim@grad.ufsc.br

Abstract. *The Internet of Things (IoT) is a network of interconnected devices using a common telecommunication protocol. However, due to the need for efficient communication in a robust network, the data exchanged is often transmitted over insecure protocols. On the other hand, computer forensics is the area of Computer Science that aims to meet the demands of criminalistics, requiring control over the identification and preservation of digital evidence. In this sense, a sober identity system, which uses decentralized unique identifiers (DIDs) and blockchain, emerges as a way to guarantee the security of users and devices in an IoT network. In the context of forensic compatibility, it is essential to ensure the certification of all elements of the system, and therefore this work seeks to build a decentralized digital identity system, based on blockchain, to guarantee the security and trust of the devices involved in the transfer of vouchers. .*

Resumo. *A Internet das Coisas (IoT) é uma rede de dispositivos interconectados que utiliza um protocolo de telecomunicação comum. No entanto, devido à necessidade de uma comunicação eficiente em uma rede robusta, os dados trocados são frequentemente transmitidos por meio de protocolos inseguros. Por outro lado, a computação forense é a área da Ciência da Computação que visa atender às demandas da criminalística, requerendo controle sobre a identificação e preservação de provas digitais. Nesse sentido, um sistema de identidade soberana, que utiliza identificadores descentralizados únicos (DIDs) e blockchain, surge como uma forma de garantir a autenticidade dos usuários e dispositivos em uma rede IoT. No contexto da computação forense, é essencial assegurar a autenticidade de todos os elementos do sistema, e, portanto, este trabalho busca construir um sistema de identidade digital descentralizada, baseado em blockchain, para garantir a autenticidade e confiança dos dispositivos envolvidos na transferência de evidências.*

1. Introdução

A computação forense é uma prática que visa resolver questões jurídicas e civis por meio da identificação e análise de evidências digitais (COSTA & GARCIA, 2014). A volatilidade desse cenário, que abrange não apenas computadores, mas também dispositivos móveis e embarcados, representa um desafio para garantir a autenticidade dos equipamentos envolvidos em uma cadeia de custódia.

A cadeia de custódia, de acordo com o Art. 158-A da Lei nº 13.964, de 24 de dezembro de 2019, consiste em um conjunto de processos que asseguram a história cronológica de um vestígio coletado em locais de crime, permitindo rastrear sua trajetória desde o reconhecimento até o descarte. Ela também garante a proteção das evidências contra violações ou adulterações durante processos criminais.

A expansão da Internet das Coisas introduz ainda mais complexidade nesse contexto. Com inúmeros nodos descentralizados se comunicando por meio de protocolos de internet (FILHO, 2016), surge a necessidade de uma identificação segura e universal.

Em uma cadeia de custódia, não apenas a integridade dos dados é essencial, mas também a autenticidade dos dispositivos onde as evidências estão armazenadas. No entanto, os gerenciadores de identidade convencionais podem não ser adequados para dispositivos IoT devido à sua baixa capacidade computacional (LIU et al., 2012). Nesse sentido, os Identificadores Descentralizados (DIDs) se mostram úteis, permitindo a validação de identidades sem depender de entidades centralizadas.

A utilização da Blockchain para implementar uma identidade descentralizada oferece uma solução flexível e tolerante a falhas. A arquitetura da Blockchain, também conhecida como "protocolo da confiança", facilita a detecção de modificações em evidências, tornando-se uma base sólida para o sistema de identidade descentralizada proposto. O objetivo deste trabalho é construir um sistema baseado em blockchain que identifique e valide de forma segura todos os dispositivos envolvidos em uma cadeia de custódia no campo da computação forense.

2. Metodologia

A preparação deste trabalho iniciou-se com o desenvolvimento de uma sólida fundamentação teórica, levantando referências de trabalhos e publicações científicas para compreender o cenário da autenticação de dispositivos IoT no contexto da computação forense. Estas referências teóricas foram fundamentais para o desenvolvimento prático do Sistema de Manutenção da Identidade Descentralizada.

Além disso, levantou-se uma série de trabalhos correlatos que auxiliaram na compreensão do estado da arte nos temas propostos, contextualizando o tema escolhido e possibilitando a identificação de lacunas, que facilitaram a definição dos objetivos do trabalho bem como compreender sua viabilidade. Estes trabalhos também serviram como forma de comparar e validar os resultados obtidos, permitindo verificar a consistência do sistema final.

Para compreender o necessário a ser feito, realizou-se uma pesquisa acerca das ferramentas utilizadas para a criação e gerenciamento de Identificadores Descentralizados. O entendimento dos *frameworks* Hyperledger Indy e Aries, bem como de suas implementações Indy-CLI e ACA-Py, associadas ao uso da ferramenta Von-Network serviram como base teórica e prática para o desenvolvimento do sistema final. O Sistema de Manutenção da Identidade Descentralizada, resultado de uma extensa revisão do cenário atual do tema deste trabalho, foi essencial para compreender as possibilidades de futuras pesquisas e desenvolvimentos.

3. Conceitos Básicos

3.1. Internet das Coisas

O termo Internet das Coisas, em inglês *Internet of Things,* comumente reduzido à sigla IoT*,* refere-se não à algo concreto e/ou único, mas sim a uma malha de dispositivos conectados capazes de transformar a vida dos seres-humanos em algo mais sensível, prático e conectado, sob a ótica da globalização. Referir-se à IoT é juntar um protocolo de telecomunicação com os objetos mais banais e acessíveis ao homem: uma cafeteira, uma geladeira, um relógio, dentre outras inúmeras coisas disponíveis, seja no meio doméstico, corporativo ou escolar. À vista disso, semanticamente, "Internet das Coisas" significa uma rede mundial de objetos interligados, com base em protocolos de comunicação (BASSI & HORN, 2008).

Para (EASTERLING, 2012), a Internet das Coisas (Internet of Things, IoT), é "um mundo embutido com tantos dispositivos digitais que o espaço entre eles não consiste em circuitos obscuros, mas sim no espaço da própria cidade". Ainda segundo o autor, "é como se o computador estivesse extrapolando os seus próprios limites, e incorporando-se a tudo aquilo que julgamos comum."

Desta forma, para além de trabalhar com dispositivos heterogêneos, as redes IoT também estão inseridas em ambientes de comunicação que fazem uso de inúmeros protocolos bastante discrepantes entre si. Protocolos de comunicação mais leves e adaptáveis ao contexto da Internet das Coisas, tais quais Zigbee, LoRaWAN, MQTT, NFC, Bluetooth dentre tantos outros, podem ser integrados a fim de produzir um ambiente flexível, extensível, e antes de tudo, *lightweight*, em um sentido de fazer bom uso da limitada capacidade de processamento e memória destes dispositivos. No entanto, justamente devido a ausência de padrões de comunicação, os processos de autenticação e autorização tornam-se mais complexos. Segundo (WANGHAM et al., 2013), o uso de certificados digitais, familiares à Internet comum, são impraticáveis para autenticação de dispositivos na Internet das Coisas.

No *background* deste contexto, de acordo com (NOOR & HASSAN, 2019), a Internet das Coisas é fundamentalmente uma arquitetura de três camadas. A primeira delas é a camada de *hardware*, onde estão os sensores responsáveis por coletar e armazenar de forma temporária as informações do ambiente. A segunda camada é a de rede, por meio da qual os dados são comunicados. Finalmente, existe a camada de aplicação ou serviço, que é responsável pelo processamento de todo o conteúdo recebido a fim de gerar informação útil.

Para garantir o bom funcionamento desta arquitetura, a segurança torna-se um tópico fundamental a ser desenvolvido, de forma a garantir que todas as partes que compõem esse sistema permaneçam íntegras e disponíveis. Idealmente, a construção de um ecossistema seguro inicia-se logo na primeira camada de um sistema de IoT, com o uso de hardware projetado com maior sofisticação e atenção nos processos de escolha de mecanismos criptográficos e de segurança (NOOR, & HASSAN, 2019). Entretanto, sobressaindo a escolha do custo sobre o benefício, observa-se comumente uma tentativa de aplicar medidas de segurança da Internet tradicional em um ambiente onde estas não se aplicam e/ou não são suficientes.

Neste cenário, surgem desafios no âmbito de garantir a autenticidade da informação em um ambiente extremamente volátil, quiçá até mesmo, altamente modificável. Essas características são bastante problemáticas quando existe a necessidade de garantir uma identificação segura e

universal. Em alguns cenários, não somente a integridade dos dados é considerada uma propriedade crítica essencial ao funcionamento do sistema, como também o dispositivo. Desta forma, identificar um equipamento para preservá-lo, torna-se um desafio a ser superado.

3.2. Blockchain

A Blockchain é um banco de dados distribuído, no qual um usuário participante da rede possui uma cópia exata desta em um determinado momento do tempo. As características chave desta estrutura são a imutabilidade e a grande tolerância à falhas (SWAN, 2015), que garantem uma solução flexível e altamente disponível.

Segundo (NAKAMOTO, 2008), a Blockchain é uma estrutura de blocos encadeados que forma um grande banco de dados distribuído, cujo objetivo é garantir a imutabilidade da informação. Além disso, uma rede Blockchain é extremamente tolerante à falhas (SWAN, 2015), o que garante uma solução flexível e altamente disponível.

Cada bloco inserido em uma Blockchain possui uma marcação única. Essa marcação corresponde ao resumo criptográfico do bloco. Um *hash* é obtido por meio de funções criptográficas irreversíveis, que garantem que a informação de saída para um determinado conjunto de dados seja sempre a mesma. Desta forma, é possível observar a propriedade de imutabilidade neste cenário, uma vez que uma função de *hash* produz um par de resultados iguais entre si para dois conjuntos de *bytes* igualmente equivalentes.

O encadeamento neste cenário é construído de modo que um bloco N possua o *hash* do bloco N - 1 inserido em sua estrutura. Por consequência, qualquer alteração no *hash* do bloco N - 1 será facilmente percebida pelo cálculo do *hash* do bloco N, que precisa corresponder a um valor único independente do momento do cálculo. Segundo (DI PIERRO, 2017), em um sistema de Blockchain, modificar as informações de qualquer bloco provoca o colapso da estrutura como um todo.

Estruturas de Blockchain podem ser utilizadas para múltiplas aplicações que não estão necessariamente relacionadas ao cenário de criptoativos e sistemas de pagamento, referência mais evidente de uso e aplicação da tecnologia de Blockchain. O protocolo da confiança pode ser utilizado para construir sistemas de informação seguros e praticamente imutáveis, com alta resistência à corrupção e substituição de dados. Neste sentido, torna-se possível proteger informações sensíveis, assinar, autenticar e validar qualquer tipo de documento digital e prover rastreabilidade e transparência a um dado durante todo o seu ciclo de vida.

A principal atividade de um usuário em uma Blockchain é a mineração de blocos. Segundo (MOREIRA, 2019), "um bloco é considerado minerado todas as vezes em que o minerador encontrar um *nonce* que faça com que o *hash* esteja abaixo da dificuldade da rede, tendo assim uma certa quantidade de 0 no valor inicial do *hash*".

Para que um bloco possa ser inserido na rede, ele precisa possuir um resumo criptográfico de seu conteúdo. O cálculo de um hash dentro de estruturas de blockchain é uma atividade complexa, que exige alto poder computacional. Isto ocorre porque antes de tudo, é necessário encontrar um valor de *nonce* arbitrário e aleatório que produza um *hash* que possua uma quantidade de bits determinística, expressa em 0s (RIBEIRO & MENDIZABAL, 2019).

O processo de validação para inserção de novos blocos possui caráter assíncrono, e por consequência da ausência de "coordenação central, cada nó pode ter uma visão diferente do estado do blockchain em um dado instante" (MIERS et al., 2019).

Neste sentido, em uma rede distribuída como a Blockchain, que é estruturada sobre uma arquitetura de blocos encadeados, o conceito de algoritmo de inserção de blocos serve para garantir uma decisão única sobre sobre qual bloco deve ser adicionado à cauda da Blockchain, em um ambiente descentralizado.

Em Blockchains públicas, o mecanismo de consenso é o de Prova de Trabalho. Neste protocolo, qualquer nodo da rede Blockchain pode participar do processo de inserção e validação de novos blocos (RIBEIRO & MENDIZABAL, 2019). Segundo os mesmos autores, outra abordagem comum é aquela em que os nodos que estão aptos a participar do processo de inserção e validação de blocos são pré-definidos pelo protocolo. Enquanto a primeira estratégia tende a ser utilizada em sistemas de Blockchain públicos, a segunda é normalmente encontrada em redes Blockchain privadas.

3.3. Identidade Soberana

A proposta da Identidade Soberana surge como uma referência a parte das abordagens tradicionais de autenticação, relacionadas à entidades centralizadas ou federadas. É uma alternativa segura, flexível e altamente disponível, responsável por gerenciar a identidade do usuário enquanto trabalha pela sua privacidade. A ideia central desta abordagem é a descentralização da identidade de um usuário através de Identificadores Descentralizados verificáveis por meio do compartilhamento de chaves criptográficas públicas (LUX et al., 2020).

Identificadores Descentralizados e Credenciais Verificáveis são os alicerces fundamentais de uma estrutura de Identidade Soberana. DIDs são identidades únicas inseridas em um escopo global (GRÜNER et al., 2019). Por serem permanentes, descentralizados, localizáveis e criptograficamente verificáveis (REED, 2018), permitem que o usuário possa ser identificado com maior privacidade e controle sobre seus dados, em um modelo que garante mais confiança quanto a veracidade do processo de autenticação (LUX et al., 2020).

Um Identificador Descentralizado possui a estrutura de uma URI extensível a um modelo de URL para que seja possível indexar por meio deste outros recursos úteis. Um DID necessariamente é representado por meio de uma string, baseada na sintaxe formal de um *schema* e resolve, por meio de seu método, um Documento DID, que descreve as propriedades do identificador descentralizado (OMAR & BASIR, 2020).

Um método é um ledger ou rede, que possui especificações particulares para as operações CRUD de um DID. Segundo (FDHILA et al., 2021), "existem inúmeros métodos que podem ser utilizados para gerenciar um Documento DID. Eles podem ser diferenciados por meio de quatro parâmetros: Infraestrutura, Governança, Operações e Segurança". Um método define as operações que estruturam o ciclo de vida de um Identificador Descentralizado, como funções de criação, resolução, atualização e desativação.

Após a criação de um Identificador Descentralizado, cria-se um registro deste DID em um livro-razão distribuído, e.g. Blockchain. Este processo é feito através do envio de uma

transação, que inclui o DID e os metadados (DDO, DID Document) associados ao registro (XAVIER et al., 2021).

Um Identificador Descentralizado é representado por meio de um Documento DID, que possui informações adicionais e metadados associados ao Identificador Descentralizado. Escrito em formato JSON-LD, contém em si mesmo formas de garantir a autenticidade do DID e do próprio documento.

Qualquer Identificador Descentralizado gerado sobre um método DID deve ser um valor único, pois deve ser referenciado e desreferenciado a um Documento DID. O Documento DID contém toda e qualquer informação relacionada ao recurso que se busca representar.

Documentos DID são planejados para que aplicações terceiras os utilizem para validação da identidade digital de um usuário ou recurso. Neste sentido, segundo o guia da RNP para Descentralização da Identidade Digital, a estrutura de um documento possui, necessariamente, as seguintes informações:

- Identificador Descentralizado Único (DID)
- Conjunto de chaves-privadas (para verificação) Conjunto de métodos de autenticação (para autenticação)
- Conjunto de endpoints de serviço (para interação)
- Timestamp (para auditoria)
- Assinatura (para integridade)

3.4. Indy-CLI

O Indy-CLI é uma especificação que implementa uma interface de linha de comando utilizada para interagir com a plataforma Hyperledger Indy. A partir do Indy-CLI é possível que os usuários de uma rede distribuída estejam aptos a criar e gerenciar Carteiras Digitais e Identificadores Descentralizados, utilizáveis por meio da emissão de Credenciais Verificáveis. Desta forma, os usuários da rede podem interagir entre si por meio do compartilhamento ou discovery de suas Identidades Descentralizadas.

Esta ferramenta pode ser utilizada por meio da instalação direta no sistema operacional ou em uma versão organizada em containers. A partir da Von-Network, é possível obter acesso a esta versão conteinerizada.

Para desempenhar suas principais funcionalidades, o Indy-CLI é capaz de fornecer dois parâmetros essenciais e obrigatórios, a *seed* e a *key*. Este par de valores é criado como uma sequência aleatória de 32 e 64 caracteres, respectivamente. A *seed* é utilizada para derivar o DID, enquanto a *key*, por sua vez, é utilizada para criptografar a carteira digital em que o Identificador Descentralizado será armazenado.

É importante ressaltar que a *seed* deve ser mantida segura pois qualquer pessoa em sua posse tem a chance de gerar outra vez o mesmo par de DID e Chave Verificadora, e.g., *verkey*. A *key* também deve ser guardada com cuidado, uma vez que permite acesso a todas as credenciais gerenciadas pela carteira do indivíduo. Por isso, a recomendação oficial é de que estes dois parâmetros sejam gerenciados exclusivamente de forma *offline*, de modo a mantê-los isolados da rede distribuída na qual os Identificadores serão manuseados.

Ao criar um Identificador Descentralizado, um par de chaves pública e privada, associadas ao DID, são derivadas a partir do algoritmo ED25519. Cada DID é associado, por tanto, a uma *signing key* (chave privada) e a uma *verkey* (chave pública). Enquanto a *signing key* deve ser armazenada na carteira digital do portador, em segredo, o DID e sua *verkey* associada são armazenados no *ledger* para acesso público (TAM, 2019).

3.5. Von-Network

A Von-Network é uma ferramenta baseada no componente Indy da fundação Hyperledger. Possuindo inúmeras funcionalidades, seu principal objetivo é fornecer ao usuário um banco de dados distribuído por meio do qual é possível autenticar e gerenciar Identificadores Descentralizados, de forma prática e simplificada.

Além de contar com uma aplicação de linha de comando, possui uma interface gráfica que permite gerenciar o status de todos os nodos do *ledger*, bem como visualizar o estado de todas as transações feitas na rede. A Von-Network conta com um *schema* padrão para registro de Identificadores Descentralizados, que funciona integrado ao Indy-CLI.

Além de ser possível registrar Identificadores Descentralizados, de acordo com a especificação da Von-Network, é possível inserir no *ledger* transações que identificam schemas personalizados para a criação de DIDs.

4. Desenvolvimento

A implementação deste trabalho está baseada no desenho de uma arquitetura prévia descrita em (SCHMITT & STEIL, 2021). Utilizando as ferramentas Indy-CLI e Von-Network estruturou-se um Sistema de Manutenção da Identidade Descentralizada.

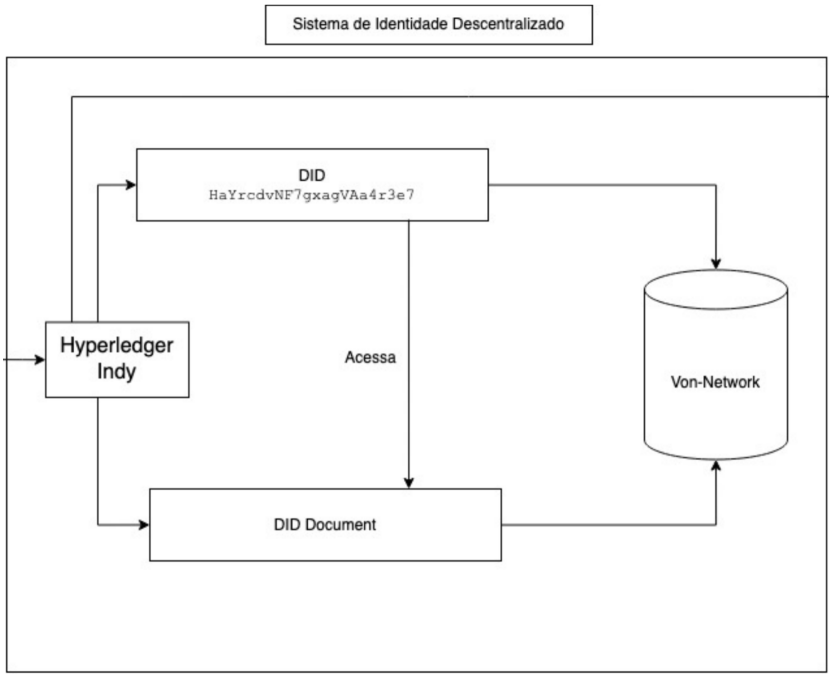

Figura 1 - Sistema de Manutenção de Identidade Descentralizada

O Indy-CLI é uma especificação que implementa uma interface de linha de comando utilizada para interagir com a plataforma Hyperledger Indy. A partir desta ferramenta, é possível que os usuários de uma rede distribuída estejam aptos a criar e gerenciar Carteiras Digitais e Identificadores Descentralizados, utilizáveis por meio da emissão de Credenciais Verificáveis. Desta forma, os usuários da rede podem interagir entre si por meio do compartilhamento ou *discovery* de suas Identidades Descentralizadas.

Utilizou-se de forma integrada ao Indy-CLI a ferramenta Von-Network, cujo principal objetivo é fornecer ao usuário uma base de dados distribuída para registrar e armazenar Identificadores Descentralizados. A Von-Network é uma aplicação de linha de comando que possui também uma interface gráfica, por meio da qual é possível observar o estado da rede bem como as transações feitas sobre ela.

O Sistema de Manutenção da Identidade Descentralizada foi construído na linguagem Python por meio do *microframework* Flask e utiliza as funções do Indy-CLI para derivar os parâmetros necessários para a criação de um DID, bem como para, de fato, criar Identificadores Descentralizados. É uma API Rest de duas rotas:

```
[GET] /seed
```

```
[GET] /did?seed={seed}
```

No código do servidor de Manutenção de Identidade Descentralizada, a rota `[GET] /seed` é responsável por criar uma *seed* válida para o usuário. A *seed* é uma sequência de caracteres aleatórios usada para criar um Identificador Descentralizado único. É um parâmetro que deve ser mantido *offline*, para impedir o acesso não autorizado à identidade associada ao DID. A *seed* é gerada a partir da função `generate_seed()`, que cria, por meio do método `generate_key(key_length)`, uma chave. Esta, por sua vez, é um conjunto de 32 caracteres pseudo-aleatórios gerados a partir do comando `openssl rand -base64 {key_length}`.

A rota `[GET] /did?seed={seed}` é responsável por gerar um novo Identificador Descentralizado a partir de uma *seed* válida. A primeira etapa deste processo é validar o *query param seed.* Se a *seed* for válida, dá-se sequência ao processamento, cujo objetivo é gerar um DID. Para isso, utiliza-se a função `nacl_seed_to_did(seed)`, que produz a chave privada do usuário com base na *seed* recebida e, por meio desta, extraí a chave de verificação do Identificador Descentralizado. O DID é criado por meio da codificação da *verkey* em *bytes*, que por fim será decodificada no padrão *ascii*. A função retorna então o Identificador Descentralizado e a Chave de Verificação.

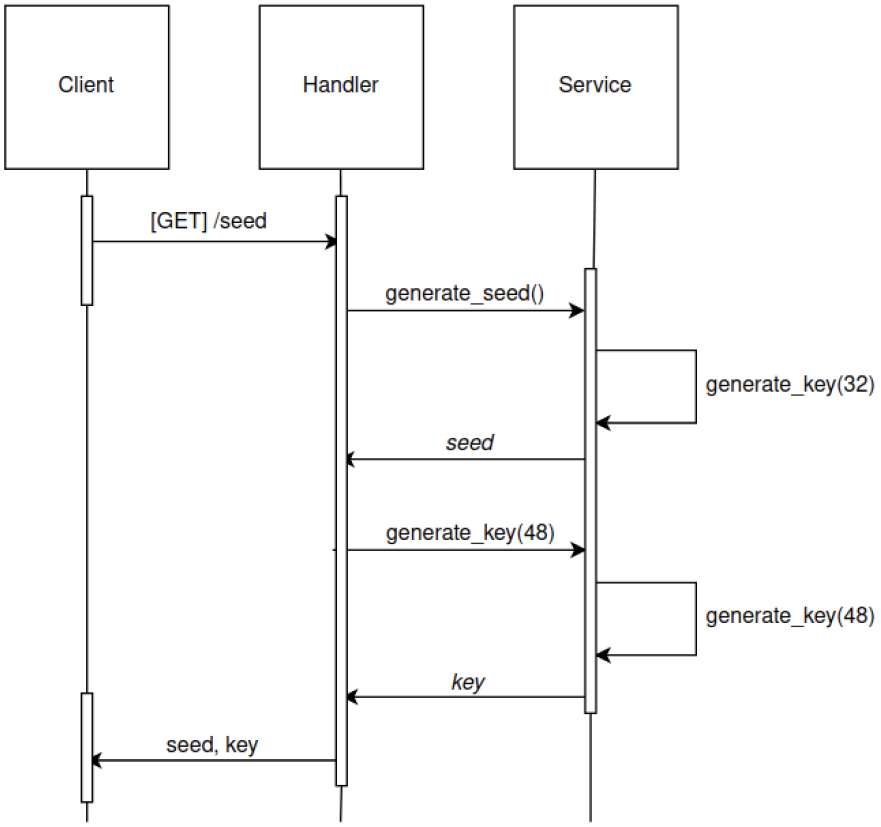

Figura 2 - Fluxo de Execução do Endpoint `[GET] /seed`

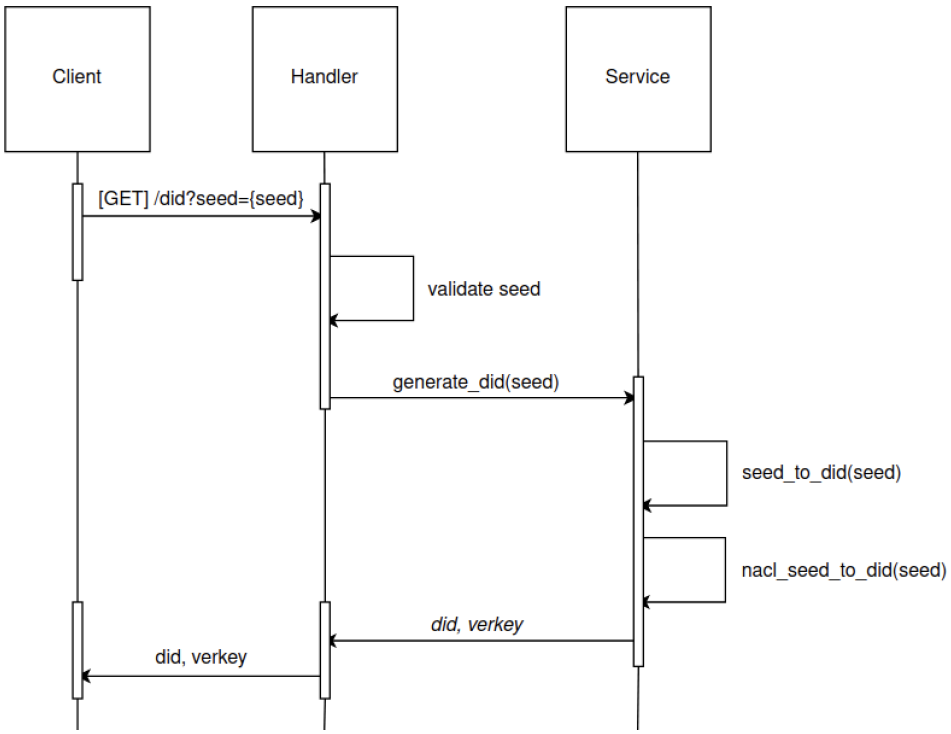

Figura 3 - Fluxo de Execução do Endpoint `[GET] /did?seed={seed}`

5. Conclusões

Compreende-se, por fim, que a Blockchain constitui uma base sólida para a implementação do Sistema de Manutenção da Identidade Descentralizada. Isso porque os componentes-chave necessários para garantir a emissão de Identificadores Descentralizados são primeiramente fundamentados nesta tecnologia. Tanto o Indy-CLI quanto a Von-Network são beneficiados pela natureza descentralizada da Blockchain, que permite a criação de Identificadores Descentralizados.

Neste contexto e por consequência, é possível afirmar que a Von-Network e o Indy-CLI são dois componentes essenciais para a fundamentação do sistema proposto neste trabalho. Enquanto o Indy-CLI permite a criação e o gerenciamento de DIDs, a Von-Network fornece o ambiente distribuído para registro e gerenciamento de Identificadores Descentralizados e Documentos DID.

A combinação do Indy-CLI e do Von-Network em um Sistema de Manutenção de Identidade Descentralizada ofereceu uma abordagem promissora para a emissão e gestão de DIDs. Ao aproveitar os princípios da Blockchain, esse sistema proporciona confiança, segurança e eficiência na identificação de dispositivos sobre um *schema* padrão.

6. Referências

BASSI, A.; HORN, G. "**Internet of Things in 2020: A Roadmap for the Future**", 2008. In: European Commission: Information Society and Media, v. 22, p. 97-114.

BRASIL, Lei nº 13.964, de 24 de dezembro de 2019. **Aperfeiçoa a legislação penal e processual penal**. Diário Oficial da União: seção 1, Brasília, DF.

COSTA, Rafael Paes; GARCIA, Raphael. "**Princípios e Análises da Computação Forense"**, 2014. In: Toledo Prudente Centro Universitário.

DI PIERRO, M., "**What Is the Blockchain?**", 2017. In: Computing in Science & Engineering, vol. 19, no. 5, pp. 92-95.

EASTERLING, Keller. "**An Internet of Things**", 2012. In: New Haven: E-flux Journal.

FDHILA, W., et al. "**Methods for Decentralized Identities: Evaluation and Insights**", 2021. In: González Enríquez, J., Debois, S., Fettke, P., Plebani, P., van de Weerd, I., Weber, I. (eds) Business Process Management: Blockchain and Robotic Process Automation Forum. BPM 2021. Lecture Notes in Business Information Processing, vol 428. Springer, Cham. https://doi.org/10.1007/978-3-030-85867-4_9

FILHO, Faccioni Mauro. "**Internet das Coisas**", 2016. In: Universidade do Sul de Santa Catarina.

GRÜNER, A., et al., "**A comparative analysis of trust requirements in decentralized identity management**", 2019. In: International Conference on Advanced Information Networking and Applications. pp. 200–213. Springer, Cham.

LIU, J., XIAO, Y., e CHEN, C. P. (2012). "**Authentication and access control in the internet of things**". In Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, pages 588–592. IEEE.

LUX, Zoltán András, et al., "**Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials**" 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2020.

MIERS, Charles et al. "**Análise de Mecanismos para Consenso Distribuído Aplicados a Blockchain**", 2019. In: Minicursos do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais.

MOREIRA Kleber Brito. "**Blockchain - Tecnologia, Arquitetura e Aplicações**", 2019. Brasília: Universidade de Brasília.

NAKAMOTO, Satoshi. "**Blockchain: Blueprint for a new economy**", 2015. In: O'Reilly Media, Inc..

NOOR, Mardiana binti Mohamad & HASSAN, Wan Haslina. "**Current research on Internet of Things (IoT) security: A survey**", 2019. In: Computer Networks, Vol. 148, p. 283-294.

OMAR, A. Sghaier & BASIR, O. "**Capability-Based Non-fungible Tokens Approach for a Decentralized AAA Framework in IoT**", 2020. In: Choo, KK., Dehghantanha, A., Parizi, R. (eds) Blockchain Cybersecurity, Trust and Privacy. Advances in Information Security, vol 79. Springer, Cham. https://doi.org/10.1007/978-3-030-38181-3_2

REED, D. "**Decentralized Identifiers (DIDs): The Fundamental Building Block of Self-Sovereign Identity**", 2018. SSI MeetUP.

RIBEIRO Lucas & MENDIZABAL Odorico Machado. "**Introdução à Blockchain e Contratos Inteligentes: Apostila para Iniciantes**" 2019. Florianópolis, Universidade Federal de Santa Catarina.

SCHMITT Henrique Hermes & STEIL Pedro Paulo Chiarelli. "**Preservação de Dados em Blockchain**", 2021. Florianópolis: Universidade Federal de Santa Catarina.

SWAN, Melanie. "**Blockchain: Blueprint for a new economy**", 2015. In: O'Reilly Media, Inc..

TAM, K.C. "**Exploring Hyperledger Indy through indy-dev Example**", 2019. In: https://kctheservant.medium.com/exploring-hyperledger-indy-through-indy-dev-example-10075d2547ae

WANGHAM, Michelle S.; DOMENECH, Marlon Cordeiro; MELLO, Emerson Ribeiro de. "**Infraestruturas de Autenticação e Autorização para Internet das Coisas**", 2013. In: Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais.

XAVIER, Marcus R. et al., "**Sobre o uso de Blockchain em soluções com Credenciais Verificáveis e Identidades Auto-Soberanas**", 2021. In: XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, out 2021.

Decentralized Identifiers (DIDs): Core architecture, data model, and representations. In: World Wide Web Consortium, 2022. Disponível em: https://www.w3.org/TR/did-core/. Acesso em 05/07/2023.

Descentralização da Identidade Digital. In: RNP Wiki Confluence, 2022. Disponível em https://wiki.rnp.br/pages/viewpage.action?pageId=160712659#. Acesso em 05/07/2023.

A kickstart to Hyperledger Indy. In: Blockchain Simplified, 2020. Disponível em: https://blockchainsimplified.com/blog/a-kickstart-to-hyperledger-indy/. Acesso em 05/07/2023.

Explore Hyperledger Indy Command Line Interface https://www.myhsts.org/tutorial-learn-how-to-work-with-hyperledger-indy-command-line-interface.php. Acesso em 05/07/2023.

Hyperledger Indy. In: Hyperledger Foundation, 2022. Disponível em: https://www.hyperledger.org/use/hyperledger-indy. Acesso em 05/07/2023.

Indy-SDK. In: https://github.com/hyperledger/indy-sdk/blob/main/cli/README.md. Acesso em: 30/06/2023.

Using the containerized Indy-CLI https://github.com/bcgov/von-network/blob/main/docs/Indy-CLI.md. Acesso em 30/06/2023.

APÊNDICE B - CONFIGURANDO O CENÁRIO DE EXECUÇÃO

Configuração da máquina virtual utilizada para realização deste trabalho:

- Sistema Operacional: Linux Ubuntu 20.04.2 LTS (64 bits)
- Processador: Intel® Core™ i7-9750H CPU @ 2.60GHz x 2
- Memória Principal: 3.8GB
- Armazenamento: 72.2 GB
- Dispositivo de Rede: Qualcomm QCA9377 802.11ac Wireless Adapter em modo Bridge. (?)

A máquina virtual possui o Python 3 pré-instalado. Esta linguagem será necessária para criar e ativar um ambiente virtual no qual a ferramenta Von-Network será executada. No entanto, o *pip,* ferramenta que permite a instalação e gerenciamento de pacotes Python, não vem instalada. Por isso, foi necessário executar os seguintes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install python-pip
```

Para verificar se a instalação do pacote foi bem-sucedida, pode-se executar o seguinte comando:

```
pip --version
```

Cujo retorno será:

```
pip 22.0.2 from {path}/lib/python3.10/site-packages/pip (python 3.10)
```

Tendo ambas as ferramentas, Python e o gerenciador de pacotes pip, será necessário criar um ambiente virtual Python para execução da Von-Network. Um ambiente virtual permite a

existência de um espaço isolado, utilizado para fins específicos. O tutorial a seguir é baseado no artigo **How To Install Python 3 and Set Up a Programming Environment on Ubuntu 20.04,** publicado no site Digital Ocean.

Primeiro passo: *update* e *upgrade:*

- É necessário atualizar as dependências do sistema antes de continuar. Para isso:
    - `sudo apt update`
    - `sudo apt -y upgrade`

Segundo passo: Python e pip:

- É necessário garantir que o Python e o pip estejam corretamente instalados em sua máquina. O *pip* foi instalado previamente, e o Python é uma instalação padrão do Ubuntu. Desta forma, precisamos apenas checar se ambas as ferramentas estão configuradas corretamente. Para isso:
    - `python3 -V (Python 3.10.6)`
    - `pip -V (pip 22.0.2)`

Terceiro passo: Instalar ferramentas adicionais:

- Para garantir que o ambiente virtual seja configurado corretamente, é necessário adicionar ferramentas adicionais. Para isso:
    - `sudo apt install build-essential libssl-dev libffi-dev python3-dev`

Quarto passo: Instalar o *venv*:

- O ambiente virtual será criado a partir da biblioteca *venv*, que é parte do do pacote original do Python. Para o instalar:
    - `sudo apt install -y python3-venv`

Quinto passo: Criar um ambiente virtual:

- Para criar um ambiente virtual, será utilizado o comando *venv*. O nome `my_env` pode ser substituído pelo nome desejado para denominar o ambiente virtual. O nome do ambiente virtual deste projeto será `dvalim-did`.
    - `python3 -m venv my_env`

Sexto passo: Ativar o ambiente virtual;

- Após a criação do ambiente virtual, será necessário ativá-lo. Para isso, será usado o comando source direcionado ao diretório de ativação criado no passo anterior. A partir do momento em que o ambiente estiver ativado, o prompt de comando possuirá o prefixo do nome do ambiente virtual.
    - `source my_env/bin/activate`
        - `(my_env) name@ubuntu: ls -la`

APÊNDICE C - USO DA VON-NETWORK

A Von-Network é o *framework* por meio do qual é possível registrar Identificadores Descentralizados em um banco de dados distribuído. É uma ferramenta necessária ao desenvolvimento deste trabalho. Para instalá-la, é necessário clonar o repositório do projeto.

```
git clone https://github.com/bcgov/von-network
```

Para executar uma instância da Von-Network, será necessário possuir o Docker instalado no sistema e o ambiente virtual anteriormente configurado ativado. Finalmente, será possível criar uma instância da Von-Network. Para isso, é necessário estar no diretório clonado:

```
cd von-network
```

Para que seja possível montar e rodar em containers as imagens do framework. Para isso, é necessário executar os seguintes comandos:

```
sudo ./manage build
```

```
sudo ./manage start
```

Desta forma, no endereço de localhost, na porta 9000, estará rodando uma instância da Von-Network.

APÊNDICE D - USO DO INDY-CLI

Para fazer uso da Von-Network, será necessário ter um DID, que deverá ser registrado na rede distribuída. Para isso, é preciso criar e iniciar uma instância do Indy-CLI, que irá gerar um Identificador Descentralizado.

Para este processo, é possível utilizar a versão organizada em containers do Indy-CLI, provisionada pela Von-Network. O comando `./manage build` cria os containers da Von-Network, enquanto o comando `./manage start` os instância.

O comando `./manage` possui a opção `generateSecrets`. Seu objetivo é gerar um par de *seed* e *key* válidas. Para tanto:

```
./manage generateSecrets
```

Cujo retorno é:

```
Seed:18/TNCcY3rkntjbz+hQvEqyTKeNLc1a5
Key:u7ObnJGoe7DjCaKxSoAtWYJkXw3tUbWWxbuwxQcALjCbTNdruvSlUGKWBriBf0Cs
```

A partir do par de *seed* e *key,* é possível gerar um Identificador Descentralizado:

```
./manage generateDid 18/TNCcY3rkntjbz+hQvEqyTKeNLc1a5
```

```
Seed: 18/TNCcY3rkntjbz+hQvEqyTKeNLc1a5
DID: HaYrcdvNF7gxagVAa4r3e7
Verkey: A35U5HHLShy3ZrVyczZYGVQbbS5nvYwRM6StgHG5XgwG
```

No Indy-CLI, também é possível registrar e manusear uma carteira digital. Para isso, é preciso garantir que a pasta *cli-scripts* possua as permissões necessárias de leitura, escrita e execução.

```
chmod a+rws cli-scripts
```

O *bit setuid* (s) significa que quem executa esse comando ganha os direitos de acesso efetivos do proprietário do arquivo - neste caso, o ID delta. Com essa configuração de permissão efetuada, torna-se viável dar sequência no processo de criação de uma *wallet*.

```
./manage indy-cli create-wallet walletName=wallet_name
```

Durante o processo de criação da *wallet*, será solicitada uma chave de acesso para criar e abrir a carteira digital.

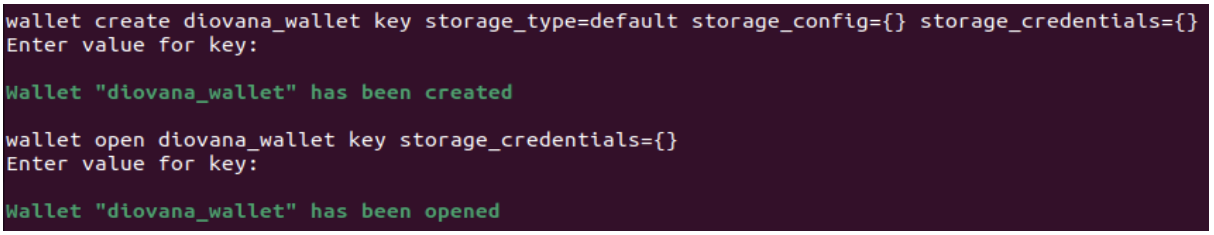

Figura 1 - Criação e Abertura de uma Carteira Digital.

Em sequência, a *wallet* dá início ao processo de criação de um DID. Para isso, será solicitada a *seed* criada no processo anterior.

Também é possível abrir uma sessão interativa da linha de comando do Indy-CLI.

```
sudo ./manage indy-cli
```

Neste momento, é possível interagir com inúmeros comandos da ferramenta de forma direta. Para mais informações, o comando `help` auxilia no entendimento e provê informações sobre os comandos disponíveis.

APÊNDICE E - INSTALAÇÃO E EXECUÇÃO DO SISTEMA DE MANUTENÇÃO DA IDENTIDADE DESCENTRALIZADA

Com o Python, o gerenciador de pacotes *pip* e o *venv* instalados, o primeiro passo é ativar o ambiente virtual anteriormente criado. Este processo pode ser feito por meio do comando abaixo, em que a variável *my_env* é o nome do ambiente virtual.

```
source my_env/bin/activate
```

Possuindo o Git instalado na máquina virtual, o segundo passo é clonar o repositório do projeto para execução local. Para isso:

```
git clone https://github.com/ddvalim/python-identity-manager.git
```

Com o ambiente virtual ativo e com a pasta do projeto disponível, finalmente, será possível instanciar o Sistema de Manutenção de Identidade Descentralizada. Para isso, é necessário estar no diretório clonado:

```
cd python-identity-manager
```

Dentro do diretório do projeto, iremos instalar os pacotes que são pré-requisitos para a execução do programa. Isso pode ser feito com o comando:

```
pip install -r requirements.txt
```

Com os pacotes necessários instalados, é possível executar uma instância do servidor. Com este comando, o programa estará rodando no endereço `http://127.0.0.1:5000`.

```
FLASK_APP=main.py flask run
```

APÊNDICE F - TESTE DO SISTEMA DE MANUTENÇÃO DA IDENTIDADE DESCENTRALIZADA

Para testar a aplicação, é possível utilizar algum cliente HTTP (como o Postman ou o Insomnia) para simular requisições como um cliente do sistema. Com esta finalidade, é necessário importar, por meio do cURL (ferramenta utilizada para dados para um servidor e que pode funcionar acoplada de um cliente HTTP), as duas requisições. Para isso, basta copiar o código abaixo e, em uma nova requisição, colar na aba da URL.

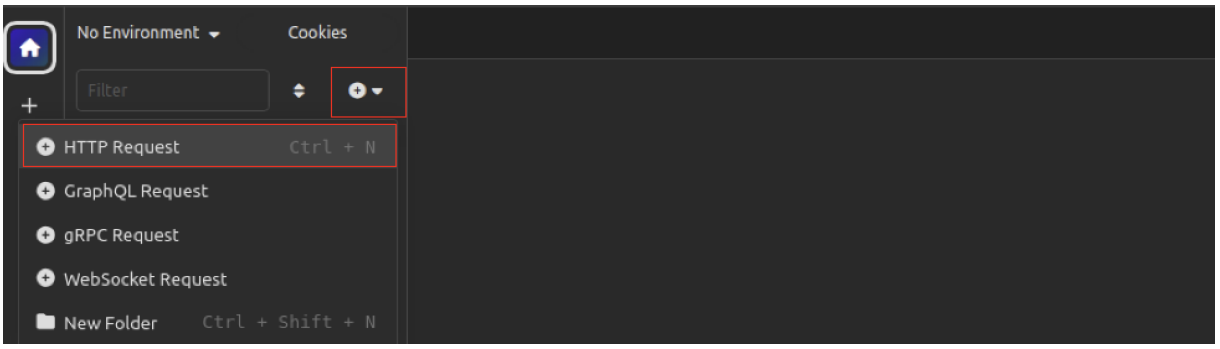

Figura 1 - Criando uma nova requisição no Insomnia.

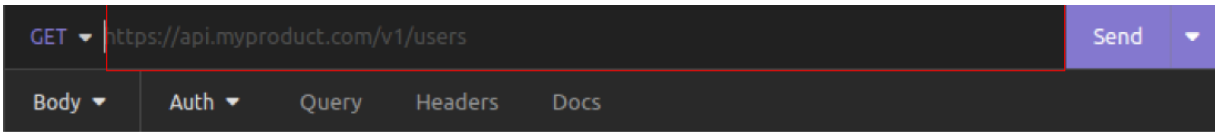

Figura 2 - Importando um cURL no Insomnia.

```
[GET] /seed
curl --request GET \
  --url 'http://127.0.0.1:5000/seed'

[GET] /did?seed={seed}
curl --request GET \
  --url
'http://127.0.0.1:5000/did?seed=966aNFshS8ISYybdiWvgba3BkwRk1aTv'
```

APÊNDICE G - CÓDIGO FONTE

Estrutura do Projeto

```
python-identity-manager/
      did/
      holder/
main.py
```

main.py

```
from flask import Flask, request, jsonify
from did import generate_did
from holder.anchor import AnchorHandle

app = Flask(__name__)

@app.route('/did',methods=['GET'])
def create_did():
      seed = request.args.get('seed')

      if seed is None:
      return jsonify(
            message="missing seed query param",
            status="bad request"
      ), 400

      did, verkey = generate_did.generate_did(seed)
      return jsonify(
      did=did,
      verkey=verkey
      ), 200

@app.route('/seed',methods=['GET'])
def create_seed():
      anchor = AnchorHandle()

      return jsonify(**{
      "seed": anchor.generate_seed(),
      "key":anchor.generate_key(48)
      })
```

anchor.py

```
import base64
import os
import base58
import nacl.signing

def nacl_seed_to_did(seed):
      seed = seed_as_bytes(seed)
      vk = bytes(nacl.signing.SigningKey(seed).verify_key)
      did = base58.b58encode(vk[:16]).decode("ascii")
      verkey = base58.b58encode(vk).decode("ascii")
      return did, verkey

def seed_as_bytes(seed):
      if not seed or isinstance(seed, bytes):
      return seed
      if len(seed) != 32:
      return base64.b64decode(seed)
      return seed.encode("ascii")


class AnchorHandle:
      def __init__(self):
      self._did: str = None
      self._verkey: str = None

      @property
      def did(self):
      return self._did

      def seed_to_did(self, seed):
      return nacl_seed_to_did(seed)

      def generate_key(self, key_lenght=48):
      stream = os.popen(f'openssl rand -base64 {key_lenght}')
      output = stream.read()
      return output[:-1]

      def generate_seed(self):
      key = self.generate_key(32)
      stream = os.popen(f'echo "{key}" | fold -w 32 | head -n 1')
      output = stream.read()
      return output[:-1]
```

generate_did.py

```
from holder import anchor

def generate_did(seed):
      TRUST_ANCHOR = anchor.AnchorHandle()
      did, verkey = TRUST_ANCHOR.seed_to_did(seed)

      print(f"\nSeed: {seed}")
      print(f"DID: {did}")
      print(f"Verkey: {verkey}")

      return did, verkey
```