Capítulo 4

Blockchain e Contratos Inteligentes para Aplicações em IoT, Uma Abordagem Prática

Jauberth Weyll Abijaude, Fabíola Greve, Péricles de Lima Sobreira

Abstract

The Internet of Things aggregates devices able to capture information and interfere in the environment, in such a way to obtain, generate and send data on a large scale to different domain application systems, such as agriculture, industry, trade, and governments. These systems need a security layer to guarantee, among other requirements, the irrefutability of transactions and the integrity of the manipulated data. In this sense, an integration with the blockchain technology, through smart contracts, would meet this need. Blockchain is a disruptive technology that offers a digital trust network for conducting transactions between peers, often unknown. This chapter presents a recent boarder research of IoT with blockchain; introduces a classification of layered blockchain technology and conducts a comprehensive study on consensus strategies and IoT applications with blockchain. In the end, it offers a guide with information that allows interested parties to design training in this area, including the realization of practical exercises.

Resumo

A Internet das Coisas agrega dispositivos capazes de capturar informações e interferir no ambiente, de maneira a obter, gerar e enviar dados em larga escala para sistemas de domínios de aplicações diferentes, tais como agricultura, indústria, comércio e governos. Estes sistemas precisam de uma camada de segurança para garantir, dentre outras características, a irrefutabilidade das transações e a integridade dos dados manipulados. Neste sentido, a integração com a blockchain, através dos contratos inteligentes, atenderia a esta necessidade. A blockchain é uma tecnologia disruptiva que oferece uma rede de confiança digital para a realização de transações entre pares, muitas vezes desconhecidos. Este capítulo apresenta pesquisas recentes na fronteira da IoT com a blockchain; apresenta uma classificação da tecnologia de blockchain em camadas e realiza um estudo amplo sobre as estratégias de consenso e aplicações IoT com blockchain. Ao final, fornece um guia com informações que permitam aos interessados a concepção de treinamentos nesta área, contemplando, inclusive, a realização de exercícios práticos.

4.1. Introdução

O avanço de inúmeras tecnologias, incluindo sensores; atuadores; computação embarcada, na nuvem e na névoa; e o surgimento de uma nova geração de dispositivos sem fio, faz com que muitos objetos ou coisas em nosso dia a dia se tornem interoperáveis. A Internet das Coisas, (do inglês *Internet of Things* - IoT), adota processamento, arquitetura de comunicação, tecnologias inteligentes e estratégias de gerenciamento para integrar um grande número de objetos inteligentes à Internet [Dorri et al. 2016].

As aplicações da IoT permitem a comunicação direta e a interação entre os dispositivos pela Internet. Os dispositivos IoT atuais incluem smartphones, eletrodomésticos inteligentes, veículos e sensores internos e externos. No entanto, com o grande número de dispositivos, os aplicativos IoT tradicionais estão enfrentando desafios em muitos aspectos, incluindo integridade de dados, segurança e sua robustez [Lao et al. 2020].

A blockchain e as tecnologias de registro distribuído (*Distributed Ledger Technologies*, DLTs) oferecem suporte distribuído e confiável para a realização de transações entre participantes, que não necessariamente têm confiança entre si, e que se encontram dispersos numa rede P2P. É considerada uma tecnologia disruptiva e com potencial para substituir entidades certificadoras e centralizadoras das transações de negócios, tais como bancos, governos, cartórios, etc., conferindo-lhes atributos de segurança como a auditabilidade, irrefutabilidade, não-repudiação, rastreabilidade, consistência e (um certo nível de) anonimato [Greve et al. 2018].

Por outro lado, os Contratos Inteligentes (SC do inglês *Smart Contracts*) servem como um mecanismo que torna os sistemas blockchain flexíveis e escalonáveis para lidar com tarefas relacionadas a contratos entre partes [Christidis and Devetsikiotis 2016], permitindo o desenvolvimento de aplicações verdadeiramente descentralizadas. Em termos gerais, os SCs são vistos como procedimentos gerais para construir sistemas de automação e controle de dispositivos IoT, e atualmente, muitas empresas fornecem soluções de IoT usando SCs [Buterin et al. 2014].

Integrar a blockchain aos SCs e às aplicações de IoT consiste numa engrenagem que não é trivial, onde se observa situações peculiares, dignas de um olhar atencioso e margeado por algumas questões a serem aprimoradas [Bogner et al. 2016]. Diante de tal cenário, este capítulo explora, de forma teórica e prática, o casamento da blockchain, SCs e IoT, trazendo o estado da arte das pesquisas recentes na área.

O capítulo está estruturado em oito seções. A Seção 4.2 introduz conceitos de blockchain, SCs, IoT e blockchain para IoT. A Seção 4.3 aprofunda o tema de blockchain, dividindo-a em 4 camadas (dados, rede, consenso e aplicação) e discorrendo sobre cada uma delas [Wu et al. 2019]. A Seção 4.4 apresenta os protocolos de consenso e a correlação dos mesmos com IoT [Wu et al. 2019, Lao et al. 2020]. A Seção 4.5 explora as aplicações da blockchain, inclusive para IoT [Mistry et al. 2020, Zhang and Chen 2020] e apresenta um exemplo prático. A Seção 4.6 discute e sugere como abordar estes temas em cursos relacionados. A Seção 4.7 apresenta os desafios de pesquisa e a última, Seção 4.8 conclui o trabalho.

4.2. Principais Conceitos

Esta seção possui caráter introdutório com o objetivo de revisar conceitos fundamentais para o acompanhamento do minicurso e nivelar os conhecimentos. São quatro subseções que exploram a blockchain, os contratos inteligentes, a internet das coisas e blockchain para IoT.

4.2.1. Blockchain

A primeira rede blockchain, apresentada em 2008, permitia transacionar valores digitais através de uma estrutura computacional distribuída [Nakamoto 2008]. Tal trabalho estabelecia as bases de um sistema econômico alternativo à base de uma moeda digital (ou criptomoeda), o Bitcoin. Em 2009, através da implementação de uma máquina de estados simplificada, foi lançada a primeira versão do software com um arranjo até então inédito, que proporcionou eliminar a terceira parte de confiança, necessária para as transações financeiras tradicionais.

Os elementos básicos e seminais desta tecnologia, combinados de forma engenhosa, sustentam de forma teórica/prática o desenvolvimento de aplicações descentralizadas, dentre elas as diversas criptomoedas. São eles:

- Criptografia: Satisfaz os requisitos de segurança do sistema e das aplicações. Dentre os recursos mais utilizados, destacam-se os resumos criptográficos (funções *hash*) e as assinaturas digitais;
- Consenso distribuído: Permite que participantes distribuídos coordenem as suas ações, de forma a alcançar decisões comuns, e assim garantir a manutenção da consistência dos seus estados (*safety*) e o progresso do sistema (*liveness*), apesar da existência de falhas [Greve 2005];
- Livro razão distribuído: O livro-razão (*ledger*) é uma estrutura de dados imutável, em que transações são registradas e o estado global do sistema é mantido replicado em todos os nós da rede P2P.

Como consequência desta composição tecnológica, a blockchain (ou BC) garante algumas propriedades que contribuem de forma inovadora para o desenvolvimento de aplicações descentralizadas e sistemas, como por exemplo [Greve et al. 2018]:

- Descentralização: Sistemas e aplicações que usam a BC não precisam de uma entidade central para coordenar as ações, as tarefas são executadas de forma distribuída;
- Disponibilidade e integridade: Os dados e as transações são replicados para todos os participantes da BC, mantendo o sistema seguro e consistente;
- Transparência e auditabilidade: A cadeia de blocos que registra as transações é pública e pode ser auditada e verificada;
- Imutabilidade e Irrefutabilidade: os registros são imutáveis e a correção só pode ser feita a partir de novos registros. O uso de recursos criptográficos garante que os lançamentos não podem ser refutados;

- Privacidade e Anonimidade: As transações são anônimas, com base nos endereços dos usuários. Os servidores armazenam apenas fragmentos criptografados dos dados do usuário;
- Desintermediação: A BC consegue eliminar terceiros em suas transações, atuando como um conector de sistemas de forma confiável e segura;
- Cooperação e incentivos: Oferta de um modelo de negócios à base de incentivos, à luz da teoria dos jogos. O consenso sob demanda passa a ser oferecido como serviço em diversos níveis e escopos.

Em 2013, surge uma nova plataforma, a *Ethereum*, que evolui para além das transações de uma criptomoeda. Implementada sob um modelo de máquina de *turing* completa, com uma nova criptomoeda, e ancorada sob alguns conceitos de seu antecessor, esta nova plataforma inova ao permitir que programas de computador possam ser armazenados e executados nas cadeias de blocos. Tais programas, conhecidos como contratos inteligentes não são em si uma novidade, pois já haviam sido propostos e definidos como um um conjunto de cláusulas contratuais, especificado em formato digital, incluindo protocolos nos quais as partes cumprem estas cláusulas[Szabo 1997].

A rede *Ethereum* oferece uma máquina de estados determinística completa, que consiste em um estado único acessível globalmente, e uma máquina virtual que aplica mudanças a esse estado. Sob uma perspectiva mais prática, a *Ethereum* é uma infraestrutura de computação globalmente descentralizada e de código aberto que executa programas chamados contratos inteligentes. Ela usa a blockchain para sincronizar e armazenar as mudanças de estado do sistema, incorpora a criptomoeda *ether* e *gas*, sendo esta última para medir e restringir os custos dos recursos de execução [Antonopoulos 2017].

As transações, compostas de mensagem com remetente, destinatário, valor e carga útil de dados, dentre outros, são processadas pela Máquina Virtual Ethereum (EVM, do inglês *Ethereum Virtual Machine*). Esta máquina virtual é baseada em uma pilha que executa os *bytecodes* (instruções em linguagem de máquina), resultantes do processo de compilação dos contratos inteligentes [EVM 2020]. O estado da rede *Ethereum* é armazenado localmente em cada nó como um banco de dados, que contém as transações e o estado do sistema em uma estrutura de dados em *hash* serializada chamada de *Merkle Patricia Tree* [Merkle-Patricia-Tree 2020].

O modelo de consenso utilizado pela *Ethereum* é o mesmo do Bitcoin. Assim, ele emprega blocos sequenciais, com uma impressão digital de dados (*hash*) único, ponderados em importância pelo protocolo de consenso *Ethash* [Etash 2020], baseado em prova de trabalho (PoW, do inglês *proof of work*), para determinar a cadeia mais longa e, portanto, o estado atual. No entanto, por diversas razões, a *Ethereum* 2.0 usa o *Casper*, baseado em prova de posse ou participação (PoS, do inglês *Proof of Stake*).

Estas duas blockchains, a Bitcoin e a *Ethereum*, são de caráter público. Elas são também conhecidas como não permissionadas ou de acesso aberto, com acesso anônimo, sem nenhum controle sobre a entrada e saída de nós na rede e sem confiança mútua entre si. Existem também as blockchains permissionadas ou federadas, onde os nós são conhecidos e precisam ser autenticados. São redes voltadas normalmente para ambientes

corporativos, onde cada participante tem um papel definido. A BC *Hyperledger Fabric* é um exemplo de BC privada.

4.2.2. Contratos Inteligentes

O termo contrato inteligente foi mencionado pela primeira vez em 1997 por Nick Szabo [Szabo 1997]. Apesar de estar definido teoricamente, não havia, à época, uma infraestrutura computacional com custos e recursos equilibrados para a sua implementação.

Com o surgimento da plataforma *Ethereum*, a ideia de implementar tais contratos foi então consolidada, hospedando-os na blockchain. Desta forma, os contratos inteligentes passam a ser sistemas que movem ativos digitais automaticamente, de acordo com regras pré-especificadas [Buterin et al. 2014].

Os Contratos inteligentes são normalmente escritos em uma linguagem de alto nível, como a Solidity, suportado na rede *Ethereum*. Para serem executados, eles são compilados, e como resultado são geradas duas saídas: os *bytecodes* e a ABI (do inglês *Application Binary Interface*). Enquanto os *bytecodes* precisam ser enviados para a rede *Ethereum* usando uma transação específica de criação de contrato, a ABI constitui uma interface pela qual as aplicações podem acessar as funções e dados dos contratos. Tal acesso é feito em conjunto com o endereço *Ethereum* que identifica o contrato.

É importante ressaltar que os contratos precisam ser chamados por uma transação para serem executados. Isto pode ser feito por uma transação iniciada a partir de uma conta *Ethereum* ou por um contrato que pode chamar outro contrato e assim por diante, ressalvando que, em tal encadeamento, a primeira execução sempre será originada por uma conta *Ethereum*. Estas transações são atômicas e o estado só será modificado se todas as transações forem executadas com sucesso. Em caso de falha, todas as operações serão desfeitas e o estado da rede refeito como se nenhuma transação tivesse sido executada. Portanto, os contratos nunca funcionarão "por conta própria"ou "em segundo plano". Eles ficam dormentes até que uma transação acione-os através da ABI, de forma direta ou indireta.

Como os contratos são imutáveis, consequentemente não conseguimos alterar o seu código. Para excluí-lo, é necessário, antecipadamente, programar uma função com a opção de autodestruição. Isto remove o código e seu estado de seu endereço, deixando a conta em branco. Quaisquer transações enviadas para esse endereço de conta após a exclusão do contrato não resultam em nenhuma execução de código, porque não há mais nenhum código para executar. O uso desta função implica em reembolso de taxas, como forma de incentivar a liberação de recursos na rede. É necessário esclarecer que as transações já realizadas pelo contrato não são apagadas, pois a blockchain é imutável.

Segundo [Bartoletti and Pompianu 2017], os contratos podem ser classificados em cinco categorias: financeiro, notário, jogo, carteira e biblioteca.

Na categoria financeira, os contratos gerenciam, reúnem ou distribuem dinheiro como característica principal. Alguns contratos certificam a propriedade de um ativo do mundo real, endossam seu valor e controlam as negociações. O projeto *Ethereum* DAO foi o mais representativo desta classe, até o seu colapso devido a um problema grave de segurança em junho de 2016, obrigando um *hard fork* para reembolso de valores pagos

indevidamente. Alguns contratos fornecem um seguro que cobre contratempos provados digitalmente (por exemplo, *Etherisc* vende apólices de seguro para voos; se um voo atrasar ou for cancelado, obtém-se um reembolso). Outros contratos publicam mensagens publicitárias (por exemplo, *PixelMap* é inspirado na Million *Dollar Homepage*).

Os contratos classificados como Notariais exploram a imutabilidade da blockchain para armazenar alguns dados persistentemente e, em alguns casos, para certificar sua propriedade e procedência. Alguns contratos permitem que os usuários gravem o *hash* de um documento na blockchain, para que possam provar a existência e integridade do documento. Outros permitem declarar direitos autorais sobre arquivos de artes digitais, como fotos ou música.

A categoria jogo reúne contratos que implementam jogos de azar (por exemplo, loterias, dados, roleta, etc.) e *games*. A categoria Carteira (*wallet*) trata de chaves, envia transações, gerencia dinheiro, implanta e monitora contratos, a fim de simplificar a interação com o blockchain. Por último, na categoria Biblioteca os contratos implementam operações de propósito geral (como, por exemplo, transformações matemáticas e manipulação de *strings*), para serem usadas por outros contratos.

4.2.3. Internet das Coisas (IoT)

A Internet das Coisas é composta por dispositivos físicos com funções de rede, componentes micro-computadorizados e itens incorporados com funções de conectividade [Atzori 2016]. Junto com os rápidos avanços em software e hardware, as tecnologias relacionadas à IoT são indispensáveis na sociedade moderna.

Tais dispositivos permitem o monitoramento de ambientes industriais, domésticos e públicos, através de câmeras de vigilância, sensores, atuadores e/ou monitores. Isto permite o desenvolvimento de novas aplicações que exploram uma quantidade de dados muito grande gerados por esses dispositivos [Díaz et al. 2016, Mehmood et al. 2017].

Estas aplicações estão pulverizadas em áreas com diferentes propósitos. Por exemplo, um grande número de empresas de eletrônicos está projetando dispositivos para casas inteligentes [Stojkoska and Trivodaliev 2017]. As cidades inteligentes são uma realidade, oferecendo mais conforto e conveniência ao público [Mehmood et al. 2017]. De modo geral, para implementar estas soluções, a utilização de serviços de middleware abstraem as dificuldades de acesso aos dispositivos devido a heterogeneidade de protocolos e interfaces. A Figura 4.1 ilustra uma organização geral de elementos para Internet das Coisas [Sztajnberg et al. 2018]. Os sensores e atuadores são agrupados em dispositivos e serviços, que por sua vez precisam das interfaces de comunicação para enviar/receber dados das camadas superiores.

Existe uma variedade de protocolos e padrões de comunicação que podem ser empregados, entre eles, destacam-se o MQTT [Mqtt 2017], CoAP [Shelby et al. 2014], AMQP [Vinoski 2006], XMPP [Saint-Andre et al. 2004], WebSocket, REST e Lorawan [Sornin et al. 2015], etc.

As práticas deste minicurso utilizam o protocolo REST. Ele permite o uso da infraestrutura do HTTP para acionar ou obter recursos, apenas dando uma interpretação diferente para os métodos GET, PUT, POST e DELETE, e valendo-se da possibilidade de

Figura 4.1. Organização de elementos na IoT. Fonte [Sztajnberg et al. 2018]

enviar informações adicionais numa mensagem HTTP [Sztajnberg et al. 2018].

4.2.4. Blockchain para IoT

Blockchains para IoT são sistemas de blockchain personalizados e otimizados para aplicações de IoT. Estas aplicações são desenvolvidas em muitos campos. No entanto, boa parte desses aplicativos possui problemas como vazamento e confiabilidade de dados. Para mitigar esses efeitos problemáticos, a blockchain pode ser usada para fornecer maior segurança e estabilidade nos aplicativos IoT tradicionais [Lao et al. 2020].

Os dispositivos de IoT possuem uma série de limitações relativas à memória, processamento, potência e comunicação que precisam ser ponderados antes de se aplicar uma camada de blockchain. Nos últimos anos, muitas pesquisas foram publicadas com tentativas de abordar este casamento de forma que seja possível adotá-lo no mundo real [Bahga and Madisetti 2016, Sharma et al. 2017, Huh et al. 2017].

Segundo [Lao et al. 2020], as aplicações de blockchain e IoT podem ser categorizadas em 3 grupos: (a) pagamento digital, (b) serviço de contratos inteligentes, e, (c) armazenamento.

A categoria de pagamentos digitais foi a primeira e mais ampla utilização no campo da blockchain. Atualmente blockchains como Bitcoin e *Ethereum* podem ser utilizadas em *smartphones*, os quais funcionam como dispositivos capazes de processar (e armazenar localmente) uma parte da cadeia de blocos, o que ajudou a popularizar a manipulação de criptomoedas em dispositivos móveis.

A categoria dos contratos inteligentes emprega esta tecnologia para construir sistemas de automação e controle com base na IoT, eliminando a terceira parte de confiança [Christidis and Devetsikiotis 2016]. Há muitas empresas que fornecem este serviço, como por exemplo, a LeewayHertz [1] é uma empresa que fornece soluções para *startups* de IoT

[1] www.leewayhertz.com

e empresas usando o contrato inteligente *Ethereum*. A Ecotrace [2], empresa brasileira que atua na área de rastreabilidade de alimentos emprega blockchain, inteligência artificial e IoT para unir os elos da cadeia produtiva usando a blockchain *Hyperledger*.

A última categoria, a de armazenamento, aplica-se a aplicativos de armazenamento de dados que vêem a blockchain como um banco de dados seguro e distribuído. A Factom [3] é um desses exemplos, que utiliza APIs *(Application Programming Interface)* bem definidas para persistir informação de plataformas Web tradicionais na blockchain.

Apesar dos evidentes benefícios proporcionados pela integração tecnológica entre blockchain e IoT, há muito ainda a ser feito. Uma das iniciativas promissoras é a blockchain IOTA [4], que promete alta escalabilidade, ausência de protocolos de consenso baseado em provas e promessa de realização de transferências quase instantâneas a custo zero [Silvano and Marcelino 2020].

4.3. Arquitetura da Blockchain e de Aplicações Blockchain-IoT

A tecnologia blockchain envolve muitos elementos além de simplesmente conectar blocos em uma cadeia. Ao adicionar elementos de IoT, esta complexidade ganha novos contornos que precisam ser desvendados. Esta composição blockchain-IoT é discutida em [Lin and Liao 2017, Zheng et al. 2017, Wu et al. 2019, Lao et al. 2020], que servem de base para esta seção, subdividida em duas partes: 4.3.1. Arquitetura blockchain, 4.3.2. Arquitetura para aplicações blockchain-IoT.

4.3.1. Arquitetura Blockchain

A blockchain, segundo [Wu et al. 2019], tem uma arquitetura dividida em quatro camadas: (i) Dados, onde se encontram os blocos, o armazenamento de dados e a estrutura de árvore utilizada; (ii) Rede, onde se encontra a rede P2P e os mecanismos de comunicação; (iii) Consenso, onde naturalmente estão os protocolos de consenso; e, (iv) Aplicação, onde estão os contratos inteligentes, as criptomoedas e as *sidechains*.

A Figura 4.2 ilustra esta divisão. Na camada de dados estão a estrutura, organização e armazenamento de dados. Fatores como desempenho e acesso são cruciais para a rede blockchain. Cada uma destas redes utiliza estruturas diferentes. De um modo geral, o banco de dados para armazenamento é o Google LevelDb. A rede Bitcoin usa a árvore de *Merkle* como forma de organizar e armazenar as informações, enquanto a *Ethereum* usa a *Merkle Patricia Tree*.

A *Merkle Patricia tree* fornece uma estrutura de dados autenticada criptograficamente que pode ser usada para armazenar as ligações (chave, valor). Elas são totalmente determinísticas, o que significa que uma *tree Patricia* com as mesmas ligações (chave, valor) tem a garantia de ser exatamente a mesma até o último byte e, portanto, ter o mesmo *hash* de raiz com eficiência $O(log(n))$ para inserções, pesquisas e exclusões.

A Camada de rede da blockchain é autonomamente mantida e gerenciada por uma rede P2P composta por mineradores e usuários. É uma estrutura descentralizada e sem

[2]https://ecotrace.info/

[3]www.factom.com

[4]www.iota.org

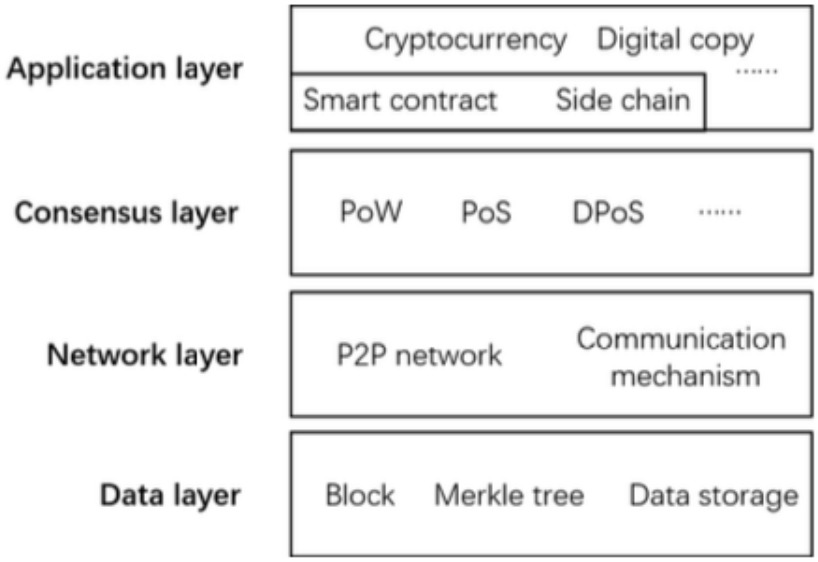

Figura 4.2. Arquitetura blockchain em quatro camadas. *Fonte* [Wu et al. 2019]

a necessidade de controle de entrada e saída, com tolerância a falhas A comunicação e a autenticação devem ser protegidas em caso de ataques.

Na plataforma *Ethereum*, existe uma variedade de redes baseadas em conformidade com a especificação formal definida em [Buterin et al. 2014], mas que podem ou não interoperar umas com as outras. Entre elas, estão a própria *Ethereum*, *Ethereum Classic*, *Ella*, *Expanse*, *Ubiq*, *Musicoin*, etc. Embora haja compatibilidade a nível de protocolo, estas redes costumam ter recursos ou atributos que exigem que os mantenedores do software cliente *Ethereum* façam pequenas alterações para dar suporte a cada rede. Por causa disso, nem todas as versões do software cliente *Ethereum* executam todos as blockchains baseadas em *Ethereum* [Antonopoulos 2017].

Atualmente, existem seis implementações principais do protocolo *Ethereum*: (1) *Parity*, escrito em *Rust*; (2) *Geth*, escrito em *Go*; (3) *cpp-ethereum*, escrito em C++; (4) *pyethereum*, escrito em Python; (5) *Mantis*, escrito em *Scala*; (6) *Harmony*, escrito em Java.

Os nós completos da *Ethereum* podem ajudar outros novos nós a obterem os dados do bloco para inicializar sua operação, além de oferecer ao operador uma verificação autorizada e independente de todas as transações e contratos. Para isto exige-se largura de banda e hardware especializados.

Os clientes remotos *Ethereum* não armazenam uma cópia local da blockchain nem validam blocos ou transações. Eles oferecem a funcionalidade de uma carteira eletrônica, podem criar e também transmitir transações. Os clientes remotos mais comuns são o *MetaMask*, *Emerald Wallet*, *MyEtherWallet* ou *MyCrypto*.

Há diferenças entre o cliente remoto e a carteira. Normalmente, o cliente remoto oferece a funcionalidade de transação de uma carteira e uma API (como `web3`, descrita em 4.5.4. Outro conceito que merece atenção é o de uma carteira remota no *Ethereum*

como de um cliente leve (análogo a um cliente de Verificação de Pagamento Simplificado em Bitcoin). Os clientes leves validam cabeçalhos de bloco e usam provas *Merkle* para validar a inclusão de transações no blockchain e determinar seus efeitos, dando-lhes um nível de segurança semelhante a um nó completo. Por outro lado, os clientes remotos *Ethereum* não validam cabeçalhos de bloco ou transações. Eles confiam inteiramente em um cliente completo para lhes dar acesso ao blockchain e, portanto, perdem garantias significativas de segurança e anonimato [Antonopoulos 2017].

A rede *Ethereum* é composta da rede principal e de redes de teste, estas últimas utilizadas como ambientes de estudos e pesquisa para desenvolvimento. A rede principal, endereçável na porta TCP 30303 trabalha com *ethers* que precisam ser comprados com dólares e as transações sofrem consequências reais. As redes de teste trabalham com *ethers* que não possuem valor real e que podem ser adquiridos em geradores de *ethers* na Internet sem custos financeiros. A rede *Ropsten* é uma rede de teste pública de blockchain. A rede de teste *Kovan* é uma rede de teste pública que usa o protocolo de consenso Aura com prova de autoridade (Esta é uma rede permissionada). A rede de teste *Rinkeby* utiliza o protocolo de consenso *"Clique"* com prova de autoridade (Esta também é uma rede permissionada).

Além disto, exite a opção de uma rede *localhost* 8545 que se conecta a um nó em execução no mesmo computador que o navegador, usando uma blockchain privada local como a *Ganache*, descrita em 4.5.4. A opção *Custom RPC* permite conexão a qualquer nó com uma interface de Chamada de Procedimento Remoto (RPC) compatível com *Geth*.

A camada de consenso é fundamental em uma rede blockchain. Chegar a um consenso não é uma tarefa trivial e muitos algoritmos de consenso foram propostos para atingir esse objetivo. Esses algoritmos ou mecanismos podem ser classificados em: PoW, PoS e suas variantes; BFT (do inglês *Bizantine Fault Tolerance*) e suas variantes. Esta camada é tratada com detalhes em 4.4

Por fim, a camada de Aplicação estende a capacidade do blockchain e torna mais fácil para os desenvolvedores construir aplicativos blockchain através dos contratos inteligentes, explicados anteriormente, das *sidechains* e do emprego de BaaS (do inglês *Blockchain as a Service*) [Samaniego et al. 2016], por exemplo. Nesta camada também estão as criptomoedas e uma série de aplicações incluindo *Fintechs*, seguros, pagamentos, governo, etc. Existe inclusive a possibilidade de ser combinada com inteligência artificial, big data, Computação quântica, IoT etc. As aplicações na plataforma *Ethereum* serão detalhadas em 4.5.

4.3.2. Arquitetura para Aplicações Blockchain-IoT

A arquitetura de aplicações para blockchain-IoT é composta de 5 camadas: Física, Rede, Blockchain, Middleware e Aplicação [Lao et al. 2020]. A Figura 4.3 ilustra estas camadas.

A camada física da arquitetura blockchain-IoT é a mesma que a camada física da IoT [Lee and Lee 2015]. Inclui os sensores, atuadores, dispositivos inteligentes, etiquetas RFID, telefones celulares, câmeras de monitoramento e quaisquer outros dispositivos de IoT relacionados à aplicações blockchain-IoT. Os protocolos empregados também são os

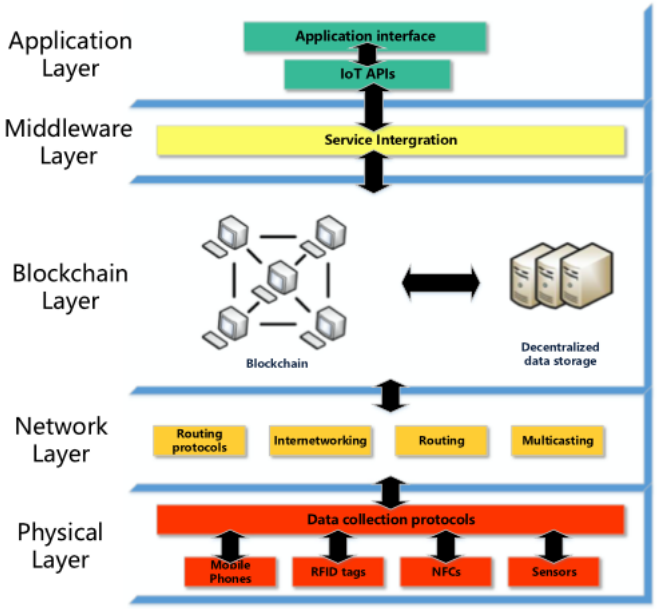

Figura 4.3. Arquitetura blockchain-IoT em 5 camadas. Fonte [Lao et al. 2020]

mesmos usados em IoT, como MQTT, COAP, REST, etc.

A camada de rede agrega funções de roteamento, interconexão de redes e *multicasting* [Jiang et al. 2016]. Esta camada é muito similar à camada de rede tradicional da blockchain. A camada blockchain representa as funções de consenso, armazenamento de dados e compartilhamento de dados, podendo inclusive ser uma plataforma de blockchain customizada [Nakamoto 2008, Underwood 2016, Ethereum 2014].

A camada de middleware é responsável por gerenciar a integração de IoT com blockchain e o fornecimento de serviço de segurança adicional [Alphand et al. 2018]. A camada de aplicativo é semelhante ao sistema IoT e às arquiteturas blockchain tradicionais. Para isto são fornecidas interações, serviços de abstração e APIs para os usuários [Dorri et al. 2017].

Para este tipos de aplicação, a blockchain é vista como um banco de dados seguro e distribuído, protegido contra violação de dados e ataques maliciosos, suportando, quando possível, contratos inteligentes que podem recepcionar os dados gerados pela IoT, mantendo um registro confiável e sem a necessidade de elementos certificadores. Isto permite uma variedade de aplicações e de soluções, nas mais diversas áreas do conhecimento, antes inimagináveis.

A Tabela 4.1 compara algumas arquiteturas blockchain-IoT, informando as características na camada de aplicação, middleware, blockchain, rede e física. A última coluna da tabela classifica as arquiteturas nos grupos de aplicação específica e blockchain como serviço.

As arquiteturas de aplicações específicas são relativas a softwares ou sistemas comerciais que usam IoT e blockchain como partes essenciais em suas operações. O *Smart Home* possui na camada física diversos dispositivos de IoT registrados na rede

Tabela 4.1. Arquiteturas de Blockchain-IoT. Fonte [Lao et al. 2020]

Blockchain-IoT	Aplicação	Middleware	Blockchain	Rede	Física	Classe
Smart Home	Smart Home App	Gerenciamento	Blockchain comercial	P2P	Dispositivos inteligentes	Aplicação específica
LO3 Energy	Energy Shopping	Energy Token	Blockchain pública	Rede de baixa latencia	Painéis solares	Aplicação específica
Slock.it	DApp	Não usa	Ethereum	Rede comercial	Travas eletrônicas	Aplicativo específica
Hybrid-IoT	Aplicação IoT	Plataforma Hybrid-IoT	POW blockchain, BFT blockchain	P2P	Sensores	Aplicativo como serviço
PBIIoT	DApp	C	Blockchain	P2P	Dispositivos de IoT	Aplicativo como serviço
JD.COM	JD.com	Blockchain Gateway	BFT blockchain	P2P	Dispositivos de IoT	Aplicativo como serviço
IoT Data Service Framework	Aplicação para Usuários	Framework	Ethereum	P2P	Dispositivos de IoT	Aplicativo como serviço
IoT Chain	Acesso com Autorização	Framework	Ethereum	Rede Comercial	Dispositivos de IoT	Aplicativo como serviço

blockchain através de uma estrutura LSB (*Lightweight Scalable Blockchain*) que garante a segurança e privacidade [Dorri et al. 2019].

LO3 Energy [5] oferece energia solar usando uma rede P2P. A camada física é composta por painéis solares. Eles capturam a geração de energia e remetem para a blockchain usando uma *token* própria (*Exergy Token*) e uma rede de baixa latência. Os clientes compram a energia por meio de aplicativos.

O "Slock.it" [6], adquirido pela Blockchains Company, controla fechaduras eletrônicas que são desbloqueadas através de um *token*. A arquitetura é simples e consiste em aplicativos distribuídos, blockchain *Ethereum*, rede comercial e bloqueios eletrônicos.

Os aplicativos como serviço são softwares de suporte que se conectam em um sistema blockchain-IoT. Tendo como camada central um middleware, estes aplicativos integram dispositivos IoT e blockchain com uma plataforma de gerenciamento simples para desenvolvedores. Rotinas como verificação de contratos e de informações precisam estar disponíveis e acessíveis nas camadas de blockchain e rede.

O "Hybrid-IoT" [Sagirlar et al. 2018] é uma plataforma classificada nesta categoria que implementa consenso baseado nos algoritmos PoW e BFT. A plataforma para a IoT aplicada à indústria BPIIoT [Bahga and Madisetti 2016] permite a criação de DApps. Os dispositivos IoT precisam ser registrados na rede blockchain.

A plataforma de blockchain "*JD Blockchain Open Platform*" [7] fornece serviços de *gateway*, de nó e de consenso na blockchain. Baseada em consenso BFT, possui protocolo de autenticação para controlar o número de acessos à rede blockchain.

Uma estrutura para implementar integridade e segurança de dados de IoT com base na plataforma *Ethereum*, onde o dispositivo de IoT é responsável por gerar e gravar dados na blockchain, dispensando uma autoridade certificadora, e posteriormente, proporcionando ao usuário a verificação da integridade dos dados por meio de um aplicativo

[5] https://lo3energy.com/

[6] https://www.blockchains.com/

[7] http://ledger.jd.com/

de usuário de dados é definida em [Liu et al. 2017].

A IotChain combina a arquitetura OSCAR [Vučinić et al. 2015] à estrutura de autorização ACE [Seitz et al. 2017]. Cada usuário registrado tem um *token* autorizado que identifica um conjunto de recursos.O dispositivo IoT é responsável pela geração de dados. O proprietário dos dados é responsável por enviar os dados para a blockchain, a arquitetura OSCAR e a estrutura de autorização ACE responsáveis por garantir a segurança dos dados do usuário [Alphand et al. 2018].

4.4. Consenso

O consenso é um problema fundamental em computação distribuída e permite com que um conjunto de participantes (ou nós) numa rede chegue a um acordo sobre um conjunto de transações, ou sobre um determinado estado do sistema, apesar da ocorrência de falhas ou da presença de nós maliciosos, que podem subverter o sistema [Greve et al. 2018]. O consenso, portanto, mantém o estado consistente das réplicas e a disponibilidade do sistema. No contexto da IoT, o consenso da blockchain precisa ser bem elaborado para atender aos requisitos de falta de recursos (computacional, espaço, etc.). Desta forma, as exigências de aplicativos de IoT, com altos custos de manutenção e fraco suporte a usos de tempo crítico, podem ser resolvidas com a introdução de um mecanismo de consenso distribuído adequado.

Esta seção apresenta os protocolos de consenso Prova de Trabalho - PoW, PoS, Variantes do PoW e PoS, Consenso Tolerante à Falhas Bizantinas - BFT e Grafo Direcionado Acíclico - DAG (*Directed Acyclic Graph*). Em seguida faremos um comparativo entre os protocolos de consenso, ilustrados na Figura 4.4 e abordamos as características de protocolos de consenso para IoT. O termo PoX (*Proof of Somethings*) é uma forma de referir-se genericamente aos protocolos que necessitam de alguma prova para alcançar o consenso [Lao et al. 2020].

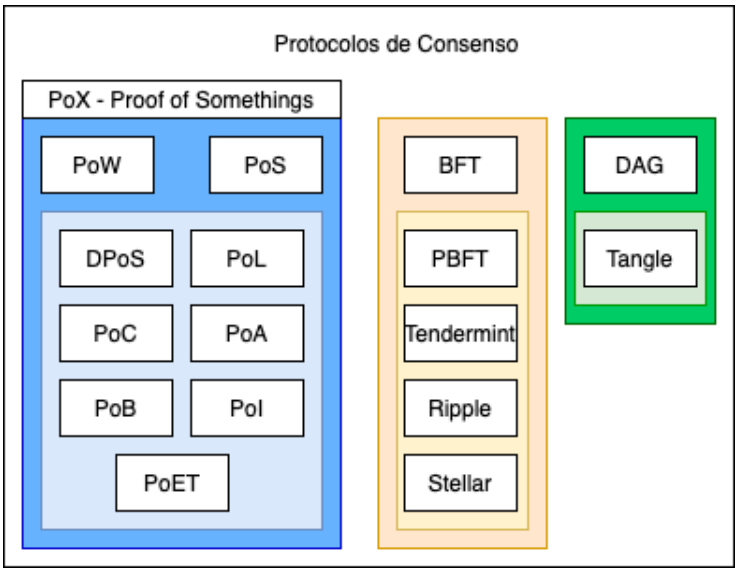

Figura 4.4. Diagrama com alguns protocolos de consenso.

4.4.1. Prova de Trabalho - Proof of Work (PoW)

O protocolo de consenso PoW surge com a blockchain do Bitcoin, no famoso artigo [Nakamoto 2008]. Desde então, diversas variações apareceram. No geral, o PoW adota a seguinte estratégia: cada nó da rede precisa resolver um desafio computacional (um *puzzle* criptográfico) para poder propor à rede um bloco de transações. Assim, através de um mecanismo de competição, em um processo exaustivo, o nó que resolver o quebra-cabeça matemático obterá uma recompensa na forma de criptomoeda, o bitcoin. Após a formação do bloco, o nó irá encaminhá-lo à rede, e todos os nós irão agregá-lo a uma "blockchain", estrutura de dados contendo toda a cadeia de blocos até então acordada pelos nós, de tal forma que o bloco recentemente transmitido aponta para o anterior.

Desta forma, observa-se dois princípios básicos que fazem o consenso PoW funcionar [Lao et al. 2020]: (i) *A regra da cadeia mais longa*: o nó considera a cadeia mais longa como a cadeia certa. Isso porque, como mais de um nó pode resolver o puzzle ao mesmo tempo, mais de um bloco é transmitido na rede para estender a cadeia. Por princípio, os nós irão sempre estender a cadeia mais longa, e portanto, após um tempo, todos estarão com a mesma estrutura de blockchain, obtendo-se assim o acordo. (ii) *A regra de incentivo*: um nó será recompensado ao encontrar um bloco adequado. Desta forma, os nós estarão motivados a despender recursos computacionais participando da competição (ou mineração de blocos). Essa estratégia mantém a rede disponível e operante.

Estas premissas são a base de funcionamento da rede Bitcoin. Elas garantem a exatidão e exclusividade da cadeia de blocos, evitando o duplo gasto e a manipulação da cadeia de blocos (ou livro razão) por um nó malicioso. Ainda, elementos criptográficos são adicionados a esse processo para garantir a segurança, privacidade e integridade dos dados. Evidentemente, há outros desafios a serem tratados em um sistema complexo que movimenta ativos digitais tão valiosos, como as propriedades da criptomoeda, onde se emprega criptografia de chave assimétrica. Para uma melhor base sobre esses elementos de segurança, recomendamos o livro [Narayanan et al. 2016].

Sagirlar et al. [Sagirlar et al. 2018] propõem o uso da blockchain na IoT com a intenção de alcançar um sistema IoT baseado em consenso distribuído adequado que supere as desvantagens. No sistema, de nome Hybrid-IoT, os subgrupos de dispositivos IoT formam blocos de blocos PoW, chamados de sub-blocos de PoW, conforme ilustrado na Figura 4.5. Em seguida, a conexão entre as sub-blockchains PoW emprega uma estrutura interconectora de consenso bizantino BFT, como a Polkadot[8] ou Cosmos [9]. Os autores propõem a formação de sub-blockchains PoW guiadas por um conjunto de diretrizes baseadas em parâmetros de dimensões, métricas e limites. Para comprovar a validade da abordagem, realizaram uma avaliação de desempenho e de segurança.

4.4.2. Prova de Participação - *Proof of Stake* (PoS)

O PoS [Kiayias et al. 2017] é um dos algoritmos em ascensão para muitas aplicações de blockchain. Após anos de uso do PoW, algumas desvantagens ficaram evidentes, como segurança (ataques de duplo gasto possíveis), alto consumo energético e desperdício de recursos (no processo de competição/mineração) ou baixa vazão - *throughput* (pouca quan-

[8]https://polkadot.network

[9]https://cosmos.network/

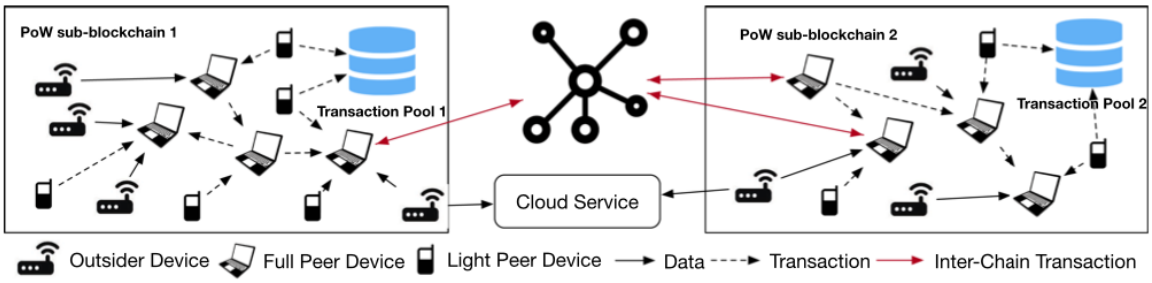

Figura 4.5. Arquitetura do Hybrid-IoT com emprego de PoW e o conceito de sub-blockchains. Fonte [Sagirlar et al. 2018]

tidade de transações acordadas no tempo).

De forma simplista, o PoS baseia-se na hipótese de que os usuários com a posse de mais moedas (ou recursos computacionais) são mais propensos a garantir a confiabilidade do sistema e têm menos probabilidade de se comportar como nós maliciosos. Assim, o algoritmo PoS considera a porcentagem do número total de moedas (ou recursos) que um nó envolvido na competição detém, e eventualmente considera o tempo que o nó leva com o montante de moedas (ou recursos) para estabelecer uma porcentagem de direito de participação no consenso. Quanto mais moedas (ou recursos), mais probabilidade o nó terá de participar do consenso para decidir sobre os blocos.

A fim de permitir que cada bloco seja gerado mais rapidamente, o mecanismo PoS elimina o processo exaustivo de resolução do *puzzle* criptográfico. Mas, há problemas que também tonaram-se ou podem se tornar evidentes, como aconteceu com o PoW. Por exemplo, os usuários com mais moedas por um longo período têm maior possibilidade de serem selecionados pelo sistema para gerar o próximo bloco, ocasionando um elitismo e centralização das decisões. Existem muitas plataformas de blockchain que adotam o PoS, como a Cardano [10], Algorand [11] e a *Ethereum* 2.0 [12], que passa a adotar o PoS, devendo herdar boa parte das aplicações com IoT que usam contratos inteligentes dessa rede, num futuro próximo.

4.4.3. Variantes do PoW e PoS

Apesar de eficientes, os protocolos baseados em PoW e PoS tem alguns pontos negativos, como por exemplo, a alta concentração do controle dos nós que possuem mais recursos, o alto consumo energético e as constantes atualizações de hardware. Desta maneira, alguns protocolos surgem em resposta a estes desafios. Alguns deles são apresentados a seguir.

Prova de Participação Delegada - DPoS (*Delegated Proof of Stake*)

O DPoS [Larimer 2014] resolve o problema de centralização através da introdução do mecanismo de delegação. Os nós da rede elegem nós especiais, os super nós ou delegados, que passam a gerar e assinar os blocos. Com esta estratégia, o tempo de se confirmar uma transação melhora, pois o protocolo DPoS elimina a necessidade de se aguardar a confirmação de nós não confiáveis. Parece um paradoxo tentar centralizar de-

[10] https://cardano.org/
[11] https://www.algorand.com/
[12] https://ethereum.org/en/eth2/

cisões em um sistema descentralizado, no entanto qualquer nó pode ser alçado à condição de delegado. Quando um deles viola quaisquer regras do protocolo, seus direitos são negados e outro delegado será eleito.

Como exemplo de uso do DPoS temos as plataformas *BitShares* [13], *Steem* [14] e EoS [15], que apresentam 101, 21 e 21 delegados, respectivamente. Comparado com o PoW, o DPoS é mais rápido e eficiente. Além disso, é mais democrático e flexível do que o PoS. Os delegados podem ser selecionados de forma justa e os delegados ruins podem ser expulsos rapidamente, caso haja uma boa governança. As desvantagens são que o DPoS não é tão descentralizado quanto o esperado para os delegados confiáveis existentes. Além disso, as pessoas relutam em votar para indicar delegados, a menos que haja incentivos e, caso não haja votos suficientes, grandes interessados podem facilmente dominar a votação para obter benefícios [Wu et al. 2019].

Prova de Sorte - PoL (*Proof of Luck*)

Este protocolo emprega funções TEE(*Trustworthy Execution Environment*) para fornecer justiça de mineração, segurança de tempo e certeza da identidade do nó. Surge como uma alternativa ao PoW, que exige cada vez mais poder de processamento e consumo de energia. Através da geração de um número aleatório executado em um TEE, estas funções bloqueiam as plataformas que podem ser usadas para processamento das operações como forma de limitar o poder desigual da computação. Após a geração do número, o PoL escolhe um líder para o consenso, e, com isto, obtém-se economia no consumo de energia, baixa latência para confirmação de transações e equidade na mineração [Milutinovic et al. 2016].

Prova de Capacidade ou Prova de Espaço - PoC (*Proof of Capacity* ou *Proof of Space*)

O PoC [Dziembowski et al. 2015] usa o espaço disponível no disco rígido para definir privilégios ao invés do poder computacional dos nós concorrentes. A probabilidade de propor um bloco é proporcional ao espaço de armazenamento cedido à rede por um nó minerador. Quanto maior a capacidade de armazenamento em disco, maior a probabilidade de decidir sobre um bloco no consenso.

Prova de Atividade - PoA (*Proof of Activity)*

Os algoritmos de PoA contam com um conjunto de N nós confiáveis chamados de autoridades. Cada autoridade é identificada por um único id e a maioria delas é considerada honesta, ou seja, pelo menos N / 2 + 1. As autoridades chegam a um consenso para ordenar as transações emitidas pelos clientes. O consenso em algoritmos de PoA depende de um esquema de rotação de mineração, uma abordagem amplamente usada para distribuir de forma justa a responsabilidade da criação de blocos entre as autoridades. O tempo é dividido em etapas, cada uma das quais tem uma autoridade eleita como líder de mineração [De Angelis et al. 2018].

As principais implementações de PoA são o *Clique* [16] e o *Aura* [17]. Ambas têm um

[13] https://wallet.bitshares.org/#/
[14] https://steem.com/
[15] https://eos.io/
[16] https://eips.ethereum.org/EIPS/eip-225
[17] https://openethereum.github.io/Aura

primeiro turno onde o novo bloco é proposto pelo líder atual (proposta de bloco); então o *Aura* requer uma nova rodada (aceitação do bloco), enquanto o *Clique* não.

Prova de Queima - PoB (*Proof of Burn*)

Neste protocolo de consenso, os mineradores devem comprovar que queimaram algumas moedas, enviando-as para alguns endereços onde não podem ser gastos. A quantidade dessas moedas destruídas determina a probabilidade de um minerador emitir um novo bloco. O PoB funciona como uma espécie de mineração virtual, queimando moedas virtuais [Frankenfield 2018].

Prova de Importância - PoI (*Proof of Importance*)

Empregando o conceito de importância, este protocolo consegue medir a capacidade de uma conta de minerar um bloco. Para isto, a quantidade de moedas que possui e o número de transações realizadas são considerados. Há uma certa similaridade com o PoS, sob o ponto de vista do saldo em criptomoedas quando se observa o uso do saldo como critério decisivo para eleger nó, no entanto há de se observar que no PoI também se considera o volume de transação realizada. Para minerar um blocoos nós devem realizar transações ativamente. Ao aplicar PoI, a blockchain ganha vantagens de eficiência energética e alta taxa de transação [Bach et al. 2018]. Um exemplo de aplicação deste protocolo é o NEM [18].

Prova de Tempo Decorrido - PoET (*Proof of Elapsed Time*)

O algoritmo de consenso PoET [PoET 2018] faz com que os nós eleitos estocasticamente aguardem um tempo de espera aleatório criado pelo sistema. O nó que primeiro esgotar o tempo será eleito o líder para a criação do novo bloco. O PoET é um algoritmo semelhante a uma loteria que atende à justiça, ao investimento e à verificação. Para evitar trapaças, dois requisitos precisam ser verificados: O primeiro é que o líder realmente espera por um tempo aleatório em vez de um curto período de tempo para vencer. O segundo é que o líder realmente espera pelo tempo de espera determinado pelo protocolo.

4.4.4. Tolerância a Falhas Bizantinas - BFT(*Byzantine Fault Tolerance*)

Os protocolos baseados em BFT pertencem a uma classe que conseguem obter um acordo em um sistema, onde os processadores podem falhar de forma arbitrária, denominado Problema Geral Bizantino [Lamport et al. 1982]. A seguir, define-se os seguintes protocolos baseados em BFT: PBFT, *Tandermint*, *Ripple* e *Stellar*.

PBFT (*Practical Byzantine Fault Tolerance*)

O PBFT [Castro et al. 1999] foi o primeiro algoritmo prático a tolerar falhas bizantinas e adaptou-se para ser usado em ambientes assíncronos. O *BFT-Smart* é um outro projeto promissor em bom estágio de maturidade [Bessani et al. 2014]. O PBFT é oferecido pelo Hyperledger Fabric como camada de acordo (ordenação de transações). Além disso, o *BFT-Smart* [Sousa et al. 2018] também foi recentemente incorporado ao projeto [Greve et al. 2018].

[18] https://nem.io/

Tendermint

O *Tendermint* [Kwon 2014] é um protocolo de consenso BFT quase assíncrono, baseado em validadores, propondo blocos de transações e votando neles. Ele requer apenas duas rodadas de votação para chegar a um consenso. Em cada rodada, há três etapas (ou seja, propor, prevenir, pré-comprometer). Quando mais de 2/3 dos votos pré-comprometidos forem recebidos para alcançar o consenso em uma rodada, o consenso para a próxima rodada começará.

Ripple

O *Ripple Protocol Consensus Algorithm* (RPCA) [Todd 2015] utiliza sub-redes confiáveis coletivamente dentro da rede maior para chegar a um consenso para o Problema Geral Bizantino. No Ripple, a Unique Node List (UNL) é um conjunto de outros servidores mantidos por cada servidor, que desempenha um papel importante quando um servidor faz consultas para determinar o consenso. Apenas os votos dos servidores na UNL são considerados na determinação do consenso. Esta é uma diferença óbvia de muitos algoritmos de consenso. A UNL representa um subconjunto da rede que exige sabedoria coletiva para chegar a um consenso. A premissa da RPCA é que cada servidor confia nos outros servidores da UNL e acredita que eles não entrarão em conluio. A RPCA procede em várias rodadas para chegar a um consenso. Em cada rodada, cada servidor primeiro coleta o máximo de transações para se preparar para o consenso e torná-las públicas na forma de "conjunto de candidatos". Em seguida, cada servidor faz uma união dos conjuntos candidatos dos servidores em seu UNL e vota em cada transação. De acordo com o resultado da votação, as transações que obtiverem votos abaixo de um percentual mínimo serão descartadas ou colocadas em candidatos definidos no próximo consenso para o próximo bloco do livro-razão, enquanto aqueles que obtiverem votos suficientes irão para o próximo turno [Wu et al. 2019]

Stellar

O *Stellar Consensus Protocol* (SCP) [Mazieres 2015] é um protocolo do acordo bizantino federado (FBA). Ele é considerado o primeiro mecanismo de consenso comprovadamente seguro a desfrutar simultaneamente de quatro propriedades principais: controle descentralizado, baixa latência, confiança flexível e segurança assintótica. Segurança assintótica significa que a segurança do SCP depende de assinaturas digitais e famílias de *hash* cujos parâmetros podem ser ajustados de forma realista para proteger contra adversários com um poder de computação inimaginavelmente vasto.

4.4.5. Grafo Direcionado Acíclico - DAG (*Directed Acyclic Graph*)

Existe uma categoria de protocolos, a exemplo do Dagcoin [Lerner 2015] e do Tangle [Popov 2018], que apresentam uma estratégia visando explorar o paralelismo do sistema tradicional de blockchain de cadeia única, e empregam uma estrutura de dados de grafo direcionado acíclico para conectar blocos. O mecanismo de consenso, distinto dos demais abordados anteriormente, consiste em que cada transação fique vinculada aos dois registros de transações anteriores através do grafo. Desta forma, a conformidade da transação atual pode ser comprovada referenciando-se às transações anteriores. Se comparado com outros protocolos de consenso que estabelecem algum tipo de prova, observa-se que o

DAG preocupa-se apenas com as transações vinculadas, constituindo um modo bem mais simples do que as diferentes provas a que se submetem os nós. O IOTA é uma plataforma de blockchain que através do Tangle utiliza este conceito.

4.4.6. Comparativo entre os Protocolos de Consenso

Nas seções anteriores, descreveu-se alguns algoritmos de consenso mais populares e atuais em blockchain. Aqui é feita uma comparação desses algoritmos, ilustrada na Tabela 4.2. A linha Tipo de Blockchain, classificada em permissionada e não permissionada, pode também ser classificada em pública, consórcio e privada. A blockchain de consórcio é gerenciada por algumas organizações ou institutos e permitem que as pessoas participem. Ao custo de alguma descentralização, a blockchain de consórcio pode fornecer a eficiência e a segurança da blockchain pública, mantendo algum controle central, monitoramento e proteção. O Hyperledger é um dos aplicativos de blockchain de consórcio mais famosos. Na blockchain privada, as permissões são até mesmo limitadas e concentradas em um indivíduo ou em uma organização. Portanto, em alguns casos, algumas regras ou mesmo dados podem ser modificados ou adulterados pela autoridade. Comparado com a blockchain pública e a blockchain de consórcio, a blockchain privada ganha as vantagens em alta velocidade e taxa de transferência. A blockchain privada é frequentemente aplicada a empresas individuais [Wu et al. 2019].

Pode-se observar, que de modo geral, os protocolos baseados em BFT tem o desempenho prejudicado à medida que a rede aumenta a quantidade de nós. Isto ocorre devido ao custo adicional de comunicação entre os nós novos e os existentes. No entanto há vantagens de extensibilidade, já que se pode combinar com vários tipos de algoritmos de melhoria para atender a necessidades específicas [Lao et al. 2020].

As blockchains baseadas em protocolos do tipo PoX são indicadas para projetos públicos, porque a exigência de poder computacional ou moeda é útil no sentido de prevenir ataques de negação de serviço, abuso de serviço e tornar a cadeia mais segura e confiável, mas por outro lado, este conjunto de protocolo também sofre de ineficiência e alta sobrecarga computacional.

O DAG é uma nova tendência em sistemas de blockchain. Ele consegue atingir alto rendimento, típico de protocolos BFT e características de flexibilidade dos protocolos PoX. Isto implica em um mecanismo de consenso com pouca exigência computacional e altos rendimentos de eficiência. No entanto, a aplicação do DAG não está madura. Por ser um protocolo novo, ainda são necessárias mais pesquisas para que se possa de fato comprovar sua aplicação e descobrir mais sobre possíveis vulnerabilidades. O IOTA tem se mostrado um ambiente adequado para este processo de validação.

Como pode-se observar, cada um destes protocolos possui vantagens e desvantagens, que permeiam critérios como desempenho, escalabilidade, segurança, poder computacional, eficiência energética, etc. Cabe ao pesquisador ou engenheiro definir o escopo de sua aplicação, e com base nisto, escolher aquele que ofereça o melhor custo x benefício.

Tabela 4.2. Comparação dos principais protocolos de consenso. Fonte:[Wu et al. 2019, Lao et al. 2020]

	PoW	**PoS**	**DPoS**	**PoC**	**PoA**	**PoI**	**PoB**	**BFT**	**PBFT**	**Ripple**	**Stellar**	**Tendermint**	**DAG**
Ano	1999	2012	2014	-	-	-	-	-	1999	2012	2015	2014	2015
Tipo de blockchain	Não permissionada	Não permissionada	Permissionada	Não permissionada	Não permissionada	Não permissionada	Não permissionada	Não permissionada	Permissionada	Permissionada	Permissionada	Permissionada	Privada
Conceito Chave	Poder Computacional	Participação, idade da moeda	Delegados, Participação						Réplica	UNL	Quórum	Validadores	
Latência	Alta	Baixa	Baixa					Alta	Baixa	Baixa	Baixa	Baixa	Baixa
Troughput	Baixo 7 tps	Baixo 100 tps	Alto 100.000 tps	Baixa	Baixo	Baixo	Baixo		Baixo 100s tps	Alta 1500 tps	Alta 3000 tps	Alta 10.000 tps	Alta 800 tps
Finalização da transação	Probabilística	Probabilística	Probabilística	Probabilística	Probabilística	Probabilística	Probabilística	Determinístico	Imediata	Determinística	Imediata	Imediata	Probabilística
Mineradores?	Sim	Sim	Sim						Não	Não	Não	Não	Não
Token	Sim	Sim	Sim						Não	Não	Não	Não	Não
Eficiência energética	Não	Sim	Sim	Sim	Sim				Sim	Sim	Sim	Sim	Sim
Tolerância a invasão	50% do poder computacional	50% de participação	<50%	50% de espaço	50% participação online	50% Participação	50% de moedas	33% de falha das réplicas	33% de falha das réplicas	20/% de falhas dos nós		33% fallhas	33% poder computacional
Escalabilidade	Baixa	Baixa	-	Baixa	Alta	Baixo	Baixo	Baixa	Baixo	Alta			Alta
Vantagem	Livre adesão ao consenso	Eficiência em Energia	Incremento de Transações	Eficiência em Energia	Eficiência em Energia	Menos chance de acumulação	Incentivo a longo prazo	Baixo custo de transação	Finalização de bloco de alto rendimento	Rápida finalização do Bloco		baixo custo de transação	Alto Throughput
Desvantagem	Baixa vazão Consumo energia Alta taxa forks	Overhead na Comunicação	-	Desperdício de espaço em disco	Cenários de aplicativos limitados	Requisitos de Confiança	Baixa latência de confirmação	Centralização de despesas gerais	Centralização de despesas gerais	Requisitos de confiança		Centralização de despesas gerais	Overhead
Vunerabilidade	Mineração egoísta	Ataque de Longo Alcance	-	Mineração Egoísta	Simples ponto de falha	Simples ponto de falha	Ataque de negação de gastos	33% de ataque	33% de ataque	Simples ponto de falhas		33% de ataque	Ataque Sybil
Aplicações	Bitcoin Ethereum Peercoin Litecoin Dogecoin	Ethereum Percoin NXT, NVC Cosmos coin	Bitshares Ark	IPFS	Decred	NEM	XCP	Tendermint	Hyperledger	Ripple	Stellar	Cosmos Network, Tendermint	

4.4.7. Características de Protocolos de Consenso para IoT

Os dispositivos de IoT possuem limitações computacionais, energéticas e restrições de armazenamento de dados. Os protocolos de consenso precisam, além destas características, de um ambiente distribuído para garantir validade e consistência. Atingir este equilíbrio é o desafio para tornar este casamento duradouro e estável. A alta eficiência energética e um processo de consenso leve podem atenuar estes problemas.

Aplicações que envolvem blockchain-IoT precisam então driblar tais limitações e herdar os benefícios da blockchain. Se as restrições dos dispositivos de IoT forem premissas, nem os clientes leves nem os mineradores são indicados.

A blockchain IOTA é um exemplo de plataforma desenvolvida para IoT. Ela adota o consenso Tangle, baseado em DAG. Esta rede não possui mineradores, portanto o consumo energético é mais eficiente. Cada nó participante desta blockchain que necessita criar/enviar transações, primeiramente deve participar ativamente do processo de consenso aprovando duas transações anteriores.

A rede no grafo Tangle é composta por nós que são entidades que emitem e validam transações, e cada nó também representa uma transação [Popov et al. 2020]. Para que um nó adicione uma transação à rede, primeiro ele deve escolher duas transações para que possa aprová-las, conforme o algoritmo *Markov Chain Monte Carlo* (MCMC); Em seguida, o nó verifica se as transações escolhidas estão em conflito. Se isto ocorre, o nó deve desaprovar as transações conflitantes, e assim prevenir o gasto duplo; o próximo passo é resolver uma espécie de prova de trabalho (PoW), encontrando um valor *nonce*, tal que, seu *hash* seja concatenado com alguns dados da transação aprovado. É necessário destacar que este esforço computacional é uma versão muito mais leve do que a realizada pelos protocolos PoW tradicionais; Após conseguir este valor, o usuário envia sua transação para a rede, e ela se torna um *tip* (transação não aprovada); Por fim, o *tip* aguarda a confirmação por meio de aprovação direta ou indireta até que seu peso acumulado atinja o limite predefinido.

As transações são assíncronas e os nós não veem o conjunto de todas as transações. A Figura 4.6 ilustra um processo de confirmação. Em (a), o nó "m"não conhece o histórico dos ramos que contêm as transações "o", "n", "l"e "g", mas ainda validou as transações "k"e "j".

Ser assíncrono implica na possibilidade da existência de transações conflitantes. É possível que a transação "k"entre em conflito com "g"porque ambas não foram validadas em conjunto.

Quando a transação "p"valida "m"e "o", em 4.6(b), cada transação vinculada a elas é verificada, direta ou indiretamente, incluindo "k"e "g". Neste momento, se houver o conflito, ele é identificado e é necessário escolher um ramo. Segundo Popov [Popov et al. 2020], a principal regra usada para decidir entre transações conflitantes é executar o algoritmo de seleção várias vezes, identificando qual das duas transações tem maior probabilidade de ser indiretamente aprovada pelo *tip* escolhido, assim garante-se a escolha do ramo com maior probabilidade [Silvano and Marcelino 2020], enquanto o outro é abandonado.

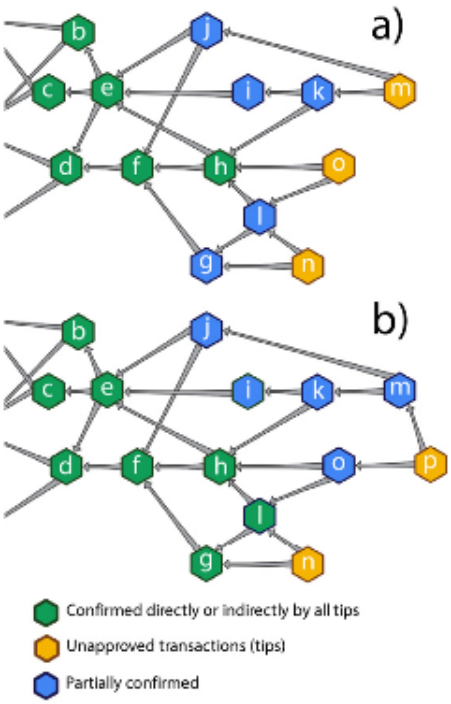

Figura 4.6. Grafo ilustrando um estado do Tangle. Em (a) um estado aleatório. Em (b) um estado após uma transação *p*. Fonte: [Silvano and Marcelino 2020]

A rede *Ethereum*, embora empregue o consenso PoW, e em breve deva migrar para o PoS é usada por Atonomi [Atonomi 2019] e consegue manter um ecossistema saudável que permite a desenvolvedores e fabricantes de IoT um ambiente de confiança universal. Através da blockchain, a aplicação valida a identidade imutável dos dispositivos de forma interoperável. O sistema *Ethereum* é detalhado na Seção 4.5.

4.5. Aplicações

A blockchain é amplamente utilizada em campos como indústria, governo, saúde, logística, comércio, etc. Esta seção descreve a evolução da blockchain desde o seu lançamento até os dias atuais, classificando as aplicações em sidechains e contratos inteligentes. Ao final, será demonstrada uma aplicação de IoT com a blockchain *Ethereum*, como forma de ilustrar os conhecimentos aqui abordados.

4.5.1. Evolução da Blockchain

Esta subseção analisa a evolução das redes blockchain desde a sua versão 1.0, originalmente concebida por Nakamoto para dar suporte à criptomoeda Bitcoin. Aqui será mostrada a evolução da blockchain bitcoin e da criação de novas redes como a *Ethereum*, *Hyper Ledger*, *Dash*, *Litecoin*, etc. Em seguida, analisa-se a Blockchain 2.0 cujo marco principal são os contratos inteligentes. A rede *Ethereum* aceita tais contratos através da EVM (*Ethereum Virtual Machine*). Ela funciona como uma Máquina de Turing completa, executando os contratos de forma determinística. Na sequência, a Blockchain 3.0 e as DApps (*Descentralized Applications*), que tem causado grandes impactos tanto na

indústria quanto na academia.

A Figura 4.7 ilustra a terminologia usada no ambiente computacional para identificar a evolução da blockchain e seus principais marcos. A primeira versão da blockchain tem como marco principal o lançamento do Bitcoin, em 2009. Com ela, a comunidade tomou conhecimento de um engenhosa combinação de técnicas como o uso de uma rede P2P, elementos criptográficos, livro-razão público e a eliminação da terceira parte de confiança sustentando as transações de uma criptomoeda, até então desconhecida e sem representatividade financeira.

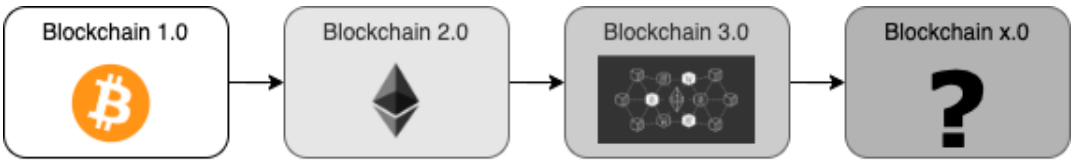

Figura 4.7. A evolução da blockchain. Fonte: [Wu et al. 2019]

A rede P2P é composta por nós que disputam o direito de publicar um bloco com as transações feitas pelos usuários e premia o vencedor com novas moedas recém emitidas. Através deste mecanismo de incentivo, do livro razão distribuído e da prova de trabalho, era possível então manter o estado da rede e evitar possíveis fraudes. Isto serviu de inspiração para pesquisas e novas aplicações, inclusive o lançamento de novas moedas digitais.

O *Litecoin* apareceu como uma das primeiras alternativas ao Bitcoin. Em 2011, a mineração de Bitcoin exigia um hardware cada vez mais especializado e caro, tornando difícil para as pessoas comuns minerarem Bitcoin. O *Litecoin* é basicamente o mesmo que o Bitcoin em tecnologia, mas é mais leve. O algoritmo Litecoin tenta permitir que qualquer pessoa com um computador comum participe do processo de mineração.

Outras moedas digitais também ganham seus próprios recursos. Por exemplo, *Dash* é uma moeda digital que suporta transações instantâneas e protege a privacidade do usuário. Ela é baseada em Bitcoin e possui uma rede única de camada dupla que inclui mineradores e nós mestres. Ela melhora o Bitcoin em dois aspectos principais: velocidade de transação e anonimato. Esta tecnologia de pagamento permite que as transações sejam concluídas quase que instantaneamente e usa técnicas de combinação de moedas para garantir a privacidade das transações [Wu et al. 2019].

A rede *Ethereum*, lançada em 2015, implementou os contratos inteligentes, permitindo transacionar, em suas redes, programas de computador, além do *ether*, que é a sua criptomoeda. Tais contratos são os representantes da Blockchain 2.0, ampliando a funcionalidade da rede. Na blockchain *Ethereum*, as pessoas também podem minerar para obterem criptomoedas recém emitidas, pois o mecanismo base foi importado do Bitcoin.

O núcleo desta plataforma é a Máquina Virtual Ethereum, que pode realizar a codificação de algoritmos complexos. O *Ethereum* é adequado para a construção de aplicativos que interagem automaticamente e diretamente entre pares ou que facilitam as atividades de coordenação de aplicativos P2P. Em teoria, qualquer atividade ou transação financeira complexa pode ser codificada de forma automática e confiável no *Ethereum*.

Além dos aplicativos financeiros, a Internet das coisas será fortemente beneficiada e influenciada, pois possui cenários que requerem uma camada de segurança, principalmente nos dados gerados pelos dispositivos.

O projeto *Hyperledger* também é uma blockchain representante do ciclo 2.0. Composta por núcleos técnicos, Capítulos e apoio da *Linux Foundation*, esta blockchain é composta de uma biblioteca de código-fonte aberta, permitindo que os interessados criem soluções personalizadas. É uma comunidade crescente, com implementações relevantes, entre elas a *Hyperledger Sawtooth* e a *Hyperledger Fabric*.

A terceira versão da blockchain é marcada pelo avanço das DApps. Estas aplicações descentralizadas podem ser definidas como um aplicativo que é maioritariamente ou totalmente descentralizado [Antonopoulos 2017]. Esta possibilidade de desenvolver aplicativos que se apóiam em um sistema distribuído, em especial a blockchain e os contratos inteligentes, causam um enorme impacto na economia e na academia. Enquanto que no primeiro eles modificam boa parte das relações contratuais vigentes, entregando a um sistema computacional a decisão e execução de determinadas ações, no segundo é uma fonte de pesquisa e estudos para criar novos conhecimentos, melhorar o desempenho, segurança e ampliar os horizontes para os mais diversos domínios. A literatura está repleta de publicações que abrangem a tecnologia, finanças econômicas, contabilidade, impostos e regulamentação, saúde, etc.

A blockchain X.0 sugere a continuidade desta evolução. A próxima sessão irá detalhar as categorias de aplicação para a Blockchain 3.0, com enfase em aplicações que envolvem IoT.

4.5.2. Categorias de Aplicações para Blockchain

As aplicações que envolvem blockchain podem ser agrupadas em *sidechains*, que permitem ao desenvolvedor anexar recursos à rede principal através de uma cadeia separada, e os contratos inteligentes, que são programas de computador representando interesses contratuais distintos executados na blockchain.

As *sidechains*, diferente dos *forks* que atualizam a blockchain, foram propostas em 2014 para que fosse possível transferir bitcoins e outros ativos entre várias blockchains [Back et al. 2014]. Ainda existem desafios técnicos para a implementação de *sidechains* [Croman et al. 2016] como por exemplo a coordenação do poder de mineração entre as cadeias, a pressão adicionada na cadeia principal e os efeitos na escalabilidade e alta latência.

Embora as *sidechains* sejam conectadas à blockchain, elas permanecem isoladas, e caso algum problema ocorra na *sidechain*, apenas ela fica comprometida, não interrompendo a cadeia principal. Alguns exemplos são: O Lisk Restaurante [19] permite que os clientes peçam comida online através de uma *sidechain* exclusiva com tipos específicos de transações personalizadas para restaurantes [Alves 2021]. O *Loom* [20] é uma plataforma para rodar DApps e jogos em sidechains conectados à *Ethereum*. O *POA Network* [21] é

[19] www.liskrestaurant.com:3000/History
[20] https://loomx.io/
[21] https://www.poa.network/

uma *sidechain Ethereum* pública de código aberto para o desenvolvimento de contratos inteligentes que usa PoA. O *Liquid* [22] é uma *sidechain* comercial que permite a movimentação instantânea de fundos entre as *exchanges*. Por último, o *Root Stock-RSK*[23] é uma *sidechain* de código aberto atrelada à rede principal do Bitcoin para a execução de contratos inteligentes.

Os contratos inteligentes são sistemas que movem ativos digitais automaticamente de acordo com regras pré-especificadas. Normalmente são escritos em uma linguagem de alto nível, como Solidity. Os contratos só são executados se forem chamados por uma transação. Um contrato pode chamar outro contrato que pode chamar outro contrato e assim por diante, mas o primeiro contrato nesta cadeia de execução sempre terá sido chamado por uma transação de uma conta de usuário.

As transações dos contratos inteligentes, independente da cadeia de execução, são sempre atômicas. Quaisquer mudanças no estado global (contratos, contas, etc.) são efetivadas apenas se toda a execução for encerrada com sucesso. Se a execução falhar devido a um erro, todos os seus efeitos (mudanças no estado) são "revertidos" como se a transação nunca tivesse sido executada. Uma transação com falha ainda é registrada como tentativa e terá de pagar as taxas de execução [Antonopoulos 2017].

Como as aplicações blockchain-IoT passam por contratos inteligentes, mais detalhes e exemplos encontram-se descritos na seção seguinte.

4.5.3. Aplicações que Utilizam Blockchain

Esta subseção apresenta aplicações que usam a blockchain. O objetivo é exemplificar a variedade de aplicações e suas relações com a blockchain. Segundo [Wu et al. 2019], é possível agrupar os contratos inteligentes conforme ilustrado na Figura 4.8, com grifo nosso, para o grupo de IoT. Em cada um dos grupos, cita-se ao menos uma referência da área.

Governança Corporativa: A Governança corporativa, as preocupações atuais e a evolução com relação à adoção da tecnologia blockchain nas áreas de serviços financeiros e governança corporativa e pública são discutidas em [Almatarneh 2020]. O autor avalia os riscos e benefícios da utilização de contratos inteligentes e avalia sua adequação (em termos de transparência, prestação de contas, responsabilidade e justiça), evidenciando as questões regulatórias. Este ensaio avalia as implicações potenciais dessas mudanças para administradores, investidores institucionais, pequenos acionistas, auditores e outras partes envolvidas na governança corporativa. Em [Yermack 2017], o custo mais baixo, a maior liquidez, a manutenção de registros mais precisa, a transparência de propriedade oferecidos por blockchains e suas consequências são demonstradas.

Bancos: A referência [Dashkevich et al. 2020] apresenta os tipos de casos de uso por Bancos Centrais considerados para adaptação com a blockchain, listando pesquisas em moeda Digital emitida por Bancos Centrais; Conformidade Regulatória; Sistemas de Compensação e Liquidação de Pagamentos operados por bancos centrais; transferência propriedade de ativos e auditoria. Em [Guo and Liang 2016] os autores afirmaram que a

[22] https://blockstream.com/liquid/
[23] www.rsk.co

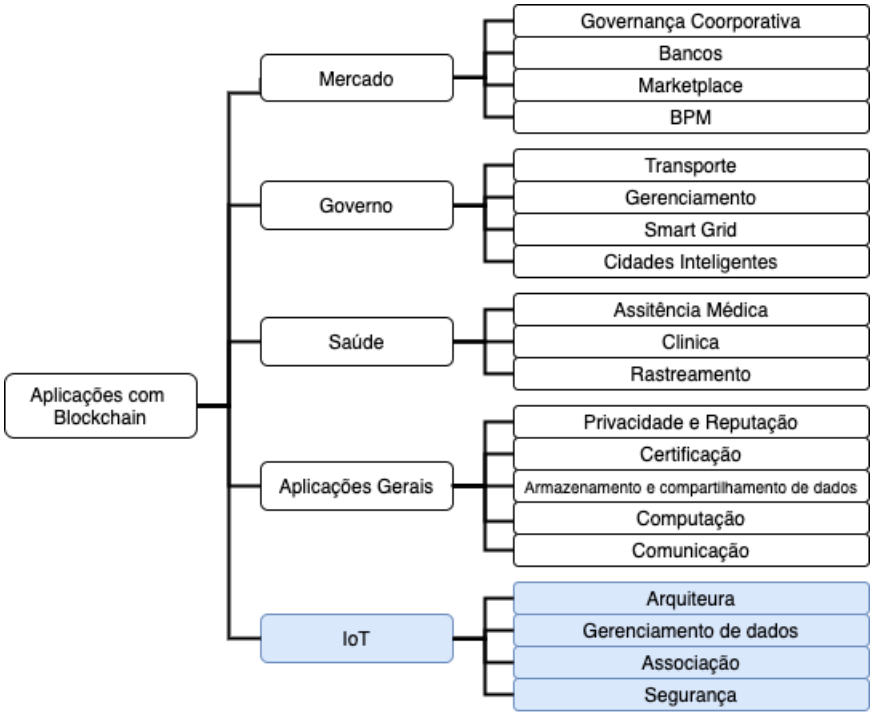

Figura 4.8. Classificação de aplicações que usam blockchain. Fonte:[Wu et al. 2019]

blockchain pode revolucionar as técnicas de pagamento e os sistemas de informação de crédito nos bancos. Em [Cocco et al. 2017] aponta-se os desafios e oportunidades de implementação da tecnologia blockchain em todos os bancos e como blockchain pode lidar com os processos financeiros como o Bitcoin.

***Marketplace*:** O *Silk Road*, o primeiro mercado da *darknet*, pode ser um exemplo típico de aplicação de blockchain em *marketplace*. Em [Christin 2013], analisa-se e exemplifica-se como os itens eram vendidos e como a população de vendedores evoluiu ao longo do tempo. Uma proposta para facilitar a recuperação eficaz de dados e transações automáticas entre vários provedores e consumidores de dados em um sistema *marketplace* está em [Yoo and Ko 2020].

BPM (Gerenciamento de Processos de negócios, do inglês *Business Process Management*): Mendling et al [Mendling et al. 2018] descreveu os desafios e oportunidades de blockchain para gerenciamento de processos de negócios (BPM) para simular pesquisas utilizando blockchain. O trabalho deles refletiu que não apenas a blockchain poderia ser usada no BPM estabelecido, mas também além dele. Eles também discutiram a tecnologia de blockchain e recursos de BPM em áreas como estratégia, governança, tecnologia da informação, pessoas e cultura

Transporte: Sistemas de transporte inteligente aliam a blockchain para tornar os dados imutáveis e tornarem os dados uma prova digital, empregando contratos inteligentes vinculados às seguradoras para coletar dados [Balasubramaniam et al. 2020].

Gerenciamento: O artigo [Hou 2017] considera a realidade na China e discute a aplicação da tecnologia blockchain no governo eletrônico chinês, descobrindo que a tecno-

logia blockchain pode trazer melhorias nos serviços governamentais, maior transparência e acessibilidade de informações do governo, desenvolvimento de compartilhamento de informações entre diferentes organizações e assistência na construção de um sistema de crédito individual, ressaltando as integrações com sistemas de gestão.

Smart Grids: A referência [Xu et al. 2017] propõe uma estrutura de gerenciamento de recursos baseada em blockchain para economizar o consumo de energia em *datacenters*. Alladi, em [Alladi et al. 2019] apresenta uma revisão de diversas aplicações para *smart grids* com uso de blockchain, entre elas o uso de energia pré-paga e o controle remoto de consumo de energia.

Cidades Inteligentes: O termo cidade inteligente envolve diversas aplicações em muitas áreas diferentes, em [Kuperberg et al. 2019] tem-se uma análise completa das oportunidades e do estado da arte da tecnologia blockchain e documentos de identidade eletrônica (eIDs) emitidos pelo governo de diversos países, incluindo implementações e provas de conceito existentes.

Assistência Médica: A assistência médica remota baseada em blockchains para sistemas de informação e assistência médica de tele-medicina é abordada em [Ji et al. 2018]. Uma aplicação baseada em blockchain que permite aos pacientes possuírem, controlarem e compartilharem de forma fácil e segura seus próprios dados de saúde sem violar a privacidade para assistência médica é detalhada em [Yue et al. 2016]. Azaria et al. [Azaria et al. 2016] propôs o MedRec, um sistema distribuído de gerenciamento de registros para processamento de registros médicos eletrônicos usando a tecnologia blockchain. A referência [Ahmad et al. 2021] explora as oportunidades e os desafios de adaptabilidade para a tecnologia de blockchain no setor de teles-saúde e tele-medicina mostrando como a blockchain pode desempenhar para fornecer a segurança e privacidade das informações necessárias, transparência operacional, imutabilidade de registros de saúde e rastreabilidade para detectar fraudes relacionadas a pedidos de seguro de pacientes e credenciais médicas.

Clínica: Em [Shae and Tsai 2017] propõe-se uma arquitetura de plataforma de blockchain para ensaios clínicos e medicina de precisão de forma a permitir um ecossistema de dados médicos confiável para pesquisa colaborativa. A referência [Liu 2016] discute as vantagens e desvantagens das tecnologias de blockchain e big data para registros médicos.

Rastreamento: Um exemplo de rastreamento de dados médicos com o uso de blockchain integrado no rastreamento de um paciente desde a primeira visita a um médico de atenção primária, rastreando seus dados até o diagnóstico final, pode ser encontrado em [Angraal et al. 2017].

Privacidade e Reputação: Dennis e Owen [Dennis and Owen 2015] propuseram o primeiro sistema de reputação generalizado que poderia ser aplicado a várias redes baseadas em blockchain. O uso de blockchain para gerenciar reputação para identificar usuários mal-intencionados e proteger a privacidade dos usuários simultaneamente no cenário *crowdsourcing* com dispositivos móveis é explorada em [Zhang et al. 2020].

Certificação: Um esquema desenhado para construir um sistema de certificação descentralizado baseado em blockchain e contratos inteligentes com o objetivo de fornecer serviços de certificado blockchain para a competição de inovação e empreendedorismo de

estudantes universitários é explicado em [Xie et al. 2020]. A proposição de blockchain e certificado digital, junto com um protocolo de autenticação seguro para dados de privacidade em blockchains sem verificar a assinatura de identidade criptografada do terceiro participante é detalhada em [Liu et al. 2020].

Armazenamento e compartilhamento de dados: Chowdhury et al. detalha uma estrutura de compartilhamento de dados que garante a autenticidade dos dados compartilhados em tempo real e fornece privacidade transacional em uma rede blockchain, melhorando o processo de tomada de decisão e reduzindo o custo geral [Chowdhury et al. 2018].

Computação: BeCome usa a tecnologia blockchain na computação de ponta para garantir a integridade dos dados [Xu et al. 2019]. O artigo [Ikeda 2018] explica um novo sistema de informação que acomoda estados quânticos de forma ponto a ponto com o apoio da blockchain.

Comunicação: A referência [Chaer et al. 2019] discute oportunidades e desafios da rede 5G com a camada de blockchain empregada para segurança.

Arquitetura: Ta-Shma et al. [Ta-Shma et al. 2017] propõem uma arquitetura usando componentes *open source* otimizados para aplicativos de Big Data empregados em cidades inteligentes.

Gerenciamento de dados: Em [Park et al. 2018] está descrito um sistema para gerenciar dados no mercado de P2P baseado em contratos inteligentes *Ethereum*.

Segurança de dados: A plataforma *Ethereum* junto com um contrato inteligente é comparada com uma comunicação usando MQTT quanto ao seu nível de segurança e simulando ataques em [Fakhri and Mutijarsa 2018]. O uso de blockchain para segurança na comunicação é também aplicado na Indústria 4.0, dados marítimos e cidades inteligentes [Rathee et al. 2021, Rahimi et al. 2020, Biswas and Muthukkumarasamy 2016].

Combinações de técnicas: Uma nova Arquitetura Orientada a Serviços (SOA) baseada em uma blockchain semântica para registro, descoberta, seleção e pagamento é apresentado em [Ruta et al. 2017].

4.5.4. Prática

Esta prática auxilia na criação de um ambiente de desenvolvimento e ilustra a interação entre um dispositivo de IoT ou um emulador que envia dados de temperatura e umidade e um contrato inteligente implementado na blockchain *Ethereum*. O material completo e um tutorial pode ser encontrado na página com informações complementares deste capítulo em [Abijaude et al. 2021].

Ainda não há um Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) ou um ambiente amigável para o desenvolvimento dos contratos e de DApps. Pode-se usar editores on-line, como por exemplo o Remix ou preparar um ambiente de desenvolvimento local. Esta última solução é a utilizada pelos autores e já testada em cursos na Universidade Estadual de Santa Cruz (UESC), Universidade Estadual do Sudoeste da Bahia (UESB) e Universidade Federal da Bahia (UFBA).

Para tanto, usa-se o o Node.js e um editor de código de sua preferência instalados

e configurados no computador. Após isto, deve-se instalar uma carteira para ter acesso à rede *Ethereum*. Aqui usamos a extensão Metamask.

Os seguintes pacotes adicionais para o Node.js precisam ser instalados: a) `solc`: compilador *Solidity*; b) `mocha`: *framework* para testar os contratos antes de implementá-los em uma rede BC; c) `web3`: coleção de bibliotecas que permite interagir com um nó *Ethereum* local ou remoto usando HTTP; d) `ganache-cli`: é uma BC pessoal para desenvolvimento rápido de aplicativos distribuídos *Ethereum* e *Corda* em um ambiente seguro e determinístico; e) `truffle-hdwallet-provider`: para realizar as assinaturas usando as palavras mnemônicas.

Estas palavras são informadas ao usuário no momento da instalação do metamask e devem ser guardadas, pois através delas conseguiremos autorizar as transações.

Com o ambiente pronto (na página do curso existe um tutorial completo para isto), deve-se criar uma pasta, fazer o download do projeto e descompactá-lo.

A pasta `deploySensor` possui o contrato inteligente (`Sensor.sol`, ilustrado na Figura 4.9) e os arquivos necessários para compilação (`compile.sol`) e implementação (`deploy.js`). O processo de compilação gera duas saídas: os bytecodes que precisam ser enviados para a rede blockchain e a ABI (do inglês *Aplication Binary Interface*) que precisa ser fornecida à aplicação, para que se tenha acesso aos contratos.

```
27      constructor(
28          string memory _name,
29          MeasureSetting[] memory _settings
30      ) {
31          require(_settings.length > 0, "Settings empty");
32          owner = msg.sender;
33          name = _name;
34          for(uint8 i; i<_settings.length;i++){
35              settings.push(_settings[i]);
36          }
37      }
38      // Cadastra as medições
39      function insertMeasure(Measure[] memory newMeasure) public {
40          for(uint i; i < newMeasure.length; i++){
41              require(newMeasure[i].value.length == settings.length,
42              "SettingsSize is different of new measure value");
43              measures.push(newMeasure[i]);
44          }
45      }
46      // Envia todas medições
47      function getAllMeasure() public  view returns (Measure[] memory measure)
48          return measures;
```

Figura 4.9. Parte do contrato `Sensor.sol`

O contrato `Sensor.sol` é responsável por receber e registrar medidas de temperatura e umidade conforme as Linhas 38 e 43. Ele armazena em uma matriz os valores de temperatura, umidade e timestamp enviados pelo sensor. A Linha 47 envia os valores que foram armazenados na matriz.

A pasta `simple-api` é composta de duas outras pastas: (a) *api* e (b) frontend. Em (a) temos o arquivo `server.js`, ilustrado na Figura 4.10, que cria uma interface

```
19  // Rota de inserção de dados
20  app.post("/insert-data", function (req, res) {
21    // Captura as variáveis no corpo da requisição
22    var { hash, temperature, humidity } = req.body;
23    // Captura o timestamp (data e horário) da requisição
24    const timestamp = new Date().getTime();
25
26    try {
27      // Compara o hash do sensor
28      if (hashSensor != hash) throw { message: "Inválid Hash" };
29      // Verifica se os dados foram enviado
30      // Se for vazio retorna erro
31      if (!temperature && !humidity)
32        throw { message: "Empty temperature or humidity" };
33
```

Figura 4.10. Parte do arquivo `server.js` onde se implementa a URL para cadastro de dados pelo sensor

REST, simulando o middleware. Através dela conseguiremos enviar os dados para serem recepcionados, e em seguida enviados para o contrato inteligente.

A Linha 20 cria a rota */insert_data* para receber os dados enviados pelo sensor. Na Linha 27 comparamos se o hash enviado pelo sensor é previamente conhecido para que os dados sejam aceitos.

Em (b) temos a pasta `src` onde estão os arquivos `sensor.js, web3.js, App.js e index.js`. O primeiro arquivo, `sensor.js` informa o endereço do contrato inteligente implementado na rede blockchain e a ABI gerada no momento da compilação do contrato. O arquivo `web3.js` cria uma instância da `web3` e a injeta na aplicação. Este passo é fundamental para que o arquivo `App.js` possa utilizar funções de manipulação do contrato.

```
3   // Endereço do contrato gerado no deploy
4   const address = "0×fC272950a29c2f2307174e82C07566c1B231C951";
5   // Abi gerada no deploy do contrato
6   const abi = [
7     {
8       inputs: [
9         {
10          internalType: "string",
11          name: "_name",
12          type: "string",
13        },
```

Figura 4.11. Parte do arquivo `sensor.js` responsável por informar o endereço do contrato e a ABI.

Na Figura 4.11, temos uma parte do código do arquivo `sensor.js`. Na Linha 4 é responsável por informar à aplicação qual o endereço do contrato para onde serão enviadas as requisições. O valor da variável `address` deve ser atualizado para o novo valor obtido após o processo de implementação de um novo contrato. O conteúdo da variável *abi* também dever ser atualizada pela nova ABI gerada no processo de compilação.

A Figura 4.12 mostra a parte do código onde a aplicação interage com o contrato inteligente. Na Linha 60 as contas ativas no metamask são recuperadas para fins de

identificação e cobrança. A Linha 63 aciona o método `insertMeasure` no contrato inteligente para registrar as medidas de temperatura, umidade e *timestamp*.

```
60      // Recupera as contas do metamask
61      const contas = await web3.eth.getAccounts();
62      // Insere medidas no contrato passando a temperatura, humidade e o timestamp
63      const leitura = await sensor.methods
64        .insertMeasure([
65          [
66            [data.temperature.toFixed(0), data.humidity.toFixed(0)],
67            data.timestamp,
68          ],
69        ])
70        .send({ from: contas[0] });
```

Figura 4.12. Código que demonstra a interação do sistema com um contrato inteligente.

É possível encontrar os tutoriais para compilação e implementação dos contratos; instalação e configuração da API que representa o middleware; e, instalação e configuração do front-end com a página da aplicação ilustrada na Figura 4.13.

Esta tela exibe os dados recebidos pelo sensor e os dados armazenados no contrato inteligente. Na parte superior são listadas as medidas recebidas pelos sensores. Ao clicar no botão `Sim!`, os dados são automaticamente enviados para a blockchain. Os endereços exibidos na tela representam a conta *Ethereum* de quem enviou a transação e o endereço da conta do contrato. O link "`Veja a Transação no etherscan`"permite ao usuário encontrar o registro de sua transação em um site especializado em manter o registro público de todas as transações da rede.

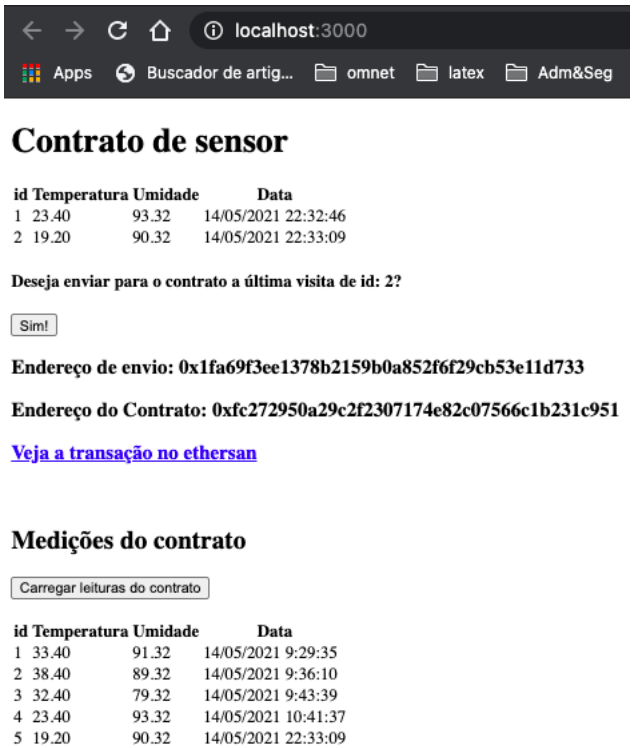

Figura 4.13. Tela da DApp que envia/recebe dados do contrato.

Existe uma diferença entre as aplicações web tradicionais e as aplicações que envolvem blockchain.

Nas aplicações web tradicionais, conforme ilustrado na Figura 4.14, geralmente o usuário usa um servidor web para acessar uma página ou um serviço web, o qual eventualmente pode até consultar outro servidores em cadeia. Comumente, um usuário envia um POST para o servidor web que em seguida envia de volta os recursos solicitados.

Em contrapartida, com o uso de uma DApp e da blockchain este cenário muda um pouco. A Figura 4.15 exemplifica a prática, a qual é uma das situações possíveis. Em (1), um sensor envia para o middleware uma coleta de dados. O middleware, por sua vez, envia estes dados para a blockchain (2) para que os mesmos sejam armazenados em um contrato e aguarda a confirmação da transação em (3). Isto possibilita armazenar informações provenientes de IoT na blockchain, o que por conseguinte, permite a herança das propriedades de segurança a nível de aplicação.

O usuário pode acessar estes dados quando em (4) envia uma solicitação para o servidor web resgatar estes valores lidos pelos sensores e já armazenados na blockchain. Ao receber esta solicitação, o servidor web, através da ABI e das funções programadas solicita à blockchain que envie estes dados(5), os quais são exibidos para o usuário.

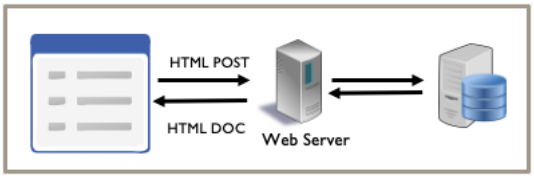

Figura 4.14. Arquitetura web.

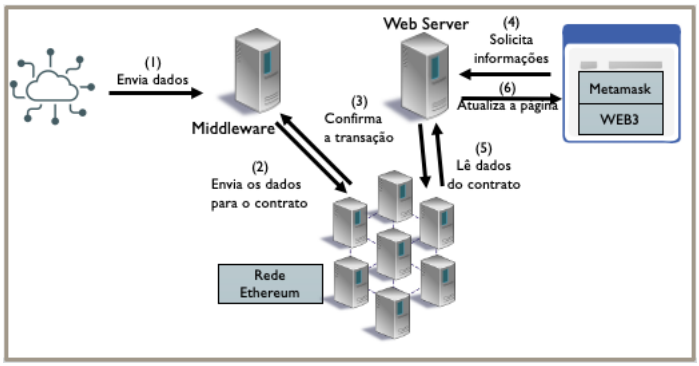

Figura 4.15. Arquitetura que exemplifica uma das formas de uma DApp trocar informações com a IoT.

Ao acessar a página do curso, teremos um tutorial completo para a execução da prática. Esta prática pode ser feita de duas formas: usando um dispositivo de IoT com interface de rede ou através de um programa que envie mensagens como o Postman ou Insomnia.

4.6. Como Montar um Curso de Blockchain e IoT

A programação de aplicações para blockchain e IoT não é uma tarefa trivial. Envolve conceitos, propriedades e conhecimentos que vão além da blockchain, dos contratos inteligentes e da linguagem de programação *Solidity* .

Existe uma escassez de material teórico e prático para o ensino de tecnologias emergentes como blockchain, CIs e desenvolvimento de DApps. Uma alternativa para isto são as plataformas MOOC (*Massive Open Online Course*), como Udemy, Coursera, edX. Algumas Universidades promovem cursos de extensão ou treinamentos para seus alunos [Rao and Dave 2019, Dettling 2018, Araujo et al. 2019].

Os autores deste minicurso elaboraram e ministraram um curso em três universidades na Bahia: A Universidade Estadual de Santa Cruz (UESC), a Universidade Estadual do Sudoeste da Bahia (UESB) e a Universidade Federal da Bahia (UFBA), e com base nesta experiência, propõem um roteiro.

4.6.1. Infraestrutura

O hardware empregado para o desenvolvimento dos laboratórios é bastante simples, uma vez que não há plataformas que necessitem de muitos recursos computacionais. Nos 3 treinamentos ministrados havia computadores equipados com processadores que vão desde o Core 2 Duo com 4 Gb de RAM, até o Core i7 com 32 Gb de RAM. O espaço em disco também não é um fator limitante, uma vez que a maioria das máquinas possui espaço de armazenamento suficiente para os experimentos realizados.

O conjunto de softwares necessários à realização de todas as atividades práticas do treinamento é composto por navegadores, extensões para navegadores, ferramentas, pacotes e aplicativos hospedados em sites. Os softwares são compatíveis com praticamente todos os sistemas operacionais, como por exemplo Windows, Linux, Unix e MacOs.

4.6.2. Conteúdo a ser Abordado

Os cursos são compostos de três módulos. O primeiro, básico, aborda conceitos de BC, CIs e DApps, criando um contrato simples e uma DApp. O segundo, explora mais profundamente as funções da linguagem *Solidity*, construindo um contrato bem mais complexo, utilizando técnicas de engenharia de software, e criando uma DApp multipágina. O terceiro módulo serviu de base para a criação deste minicurso e prevê a integração com a Internet das Coisas, coletando dados dos sensores, armazenando-os em CIs e disparando ações quando determinadas situações forem alcançadas.

O curso básico foi ministrado com carga horária de 16h. Em 2 dias de aula, com 4 horas no período da manhã e 4 horas no período da tarde. No primeiro dia, durante o período da manhã foi abordada toda a parte teórica e conceitual da blockchain com ênfase na plataforma Ethereum, além da apresentação do editor de contratos on-line *Remix*. Durante o período da tarde, configuramos as máquinas locais e realizamos rotinas de testes.

No segundo dia, durante o período da manhã apresentamos mais um pouco de teoria com foco na interação com as redes *Ethereum*. Aprendemos mais um pouco sobre a linguagem de programação *Solidity* e escrevemos, compilamos, testamos e implementamos um contrato na rede *Rinkeby*. No período da tarde desenvolvemos uma DApp que interagia com o contrato implementado.

É possível encontrar mais recursos na internet, como por exemplo:

a) *CryptoZombies*: uma plataforma online onde o intuito é ensinar sobre contratos

inteligentes de forma interativa. O usuário desenvolve um jogo com foco em zumbis onde a logística é administrada por um contrato e com interação visual através de html, css e javascript. Disponível em `https://cryptozombies.io/pt/`.

b) *Ethernaut*: uma plataforma online que apresenta diversos tutoriais voltados para jogos. Ela foca puramente na criação de contratos, sem implementação visual. Alguns exemplos possibilitam a interação através da ferramenta do desenvolvedor do navegador. Disponível em `https://ethernaut.openzeppelin.com/`.

c) *Vyper Tutorials*: semelhante à ideia do CryptoZombies, esta plataforma propõe a criação de um jogo de *pokémon*. Atualmente está em fase de desenvolvimento, mas já é possível aprender como funciona a criação de contratos. Futuramente serão adicionadas interações através de interface assim como *CryptoZombies*. Disponível em `https://vyper.fun/#/`.

d) *Ethereum Studio*: uma ferramenta para desenvolvedores que desejam aprender sobre como construir aplicações na rede *Ethereum*. Os modelos ensinam como escrever um contrato inteligente, implementá-lo e interagir com os CIs por meio de um aplicativo baseado na web. Disponível em `https://studio.ethereum.org/`.

4.6.3. Práticas Propostas

Propõe-se dividir as práticas na exploração de um editor web e suas funcionalidades. Em seguida, a montagem de um ambiente local de desenvolvimento e explorando suas vantagens e desvantagens. Após isto, introduz-se o conceitos de testes para em seguida, construir contratos inteligentes mais sofisticados, integrados a uma DApp com apenas uma página web. Na sequência, aprofunda-se mais o desenvolvimento de contratos, usando-se por exemplo técnicas de engenharia de software e a integração com DApps multipáginas, para então finalizar com contratos interagindo com dispositivos IoT.

A primeira prática apresenta o Remix, o ambiente de desenvolvimento e o primeiro contrato. Este é um exemplo didático e serve para o aluno se familiarizar com o ambiente e começar a entender os conceitos de compilação, implementação, rede de testes, etc.

A segunda prática, recomenda-se que seja a montagem de ambiente local de desenvolvimento, ressaltando as vantagens e desvantagens desta abordagem. Nessa prática, o aluno vai lidar com scripts de compilação e implementação, compreender conceitos de ABI, bytecodes, instalar uma carteira Ethereum e abastecê-la com ethers sem valor comercial.

O conceitos de testes de contratos e seus benefícios devem ser introduzidos como o terceiro momento da prática. É importante deixar claro que os contratos são imutáveis, e uma vez implementados não é possível modificá-los e nem mesmo apagá-los. Aqui o aluno aprende a configurar o ambiente de testes, montar uma rede blockchain local em sua máquina e executar testes básicos.

Na sequência, as práticas evoluem para contratos mais sofisticados, que representam exercícios mais elaborados e envolvem a transferência de moedas entre as contas. Novos testes também são propostos aqui e, ao final, a integração com um sistema Web simples, com apenas uma página, criando a primeira DApp.

Após a criação deste contrato, sugere-se criar contratos mais complicados, que envolvem conceitos de engenharia de software, como padrão *fabric*. Questões de quem vai pagar por eventuais operações nos contratos e recursos mais avançados da linguagem solidity serão tratados nesta prática. Depois de novas rotinas de teste, constrói-se uma DApp multipágina, que representa um sistema mais elaborado e com grau de dificuldade maior.

A última e mais desafiadora prática e construir um pequeno sistema que seja capaz de enviar dados de dispositivos IoT para um contrato e exibí-los em um sistema, empregando, por exemplo o estilo arquitetural REST. Caso não seja possível ter os dispositivos de IoT, pode-se optar pela geração de dados que simule tais equipamentos.

4.7. Desafios de Pesquisa

Blockchain, Internet das coisas e contratos inteligentes são uma área em evolução e com muitos desafios de pesquisa em aberto, inclusive relacionados à segurança das aplicações. Estes desafios foram categorizados e discutidos nas próximas subeções.

4.7.1. Atividades Ilegais

Uma blockchain pública é mantida por uma comunidade ao invés de autoridades. Sua natureza descentralizada também a torna uma plataforma ideal para atividades ilegais e sem censura. A economia baseada em criptomoedas facilita a lavagem de dinheiro e outras atividades ilegítimas, como a compra de bens ilícitos, compartilhamento de pornografia infantil e jogos de azar na Internet. Por exemplo, o site *Silk Road* era um mercado negro online com uma plataforma para vendedores e compradores fazerem,entre outras coisas, o tráfico ilegal de drogas [Matthews 2015]. Meiklejohn et al. [Meiklejohn et al. 2013] explora a heurística para agrupar carteiras de Bitcoin com base em evidências de autoridade compartilhada e, em seguida, usando ataques de re-identificação para classificar os operadores e quem busca usar Bitcoin para fins criminosos ou fraudulentos em grande escala.

4.7.2. *Throughput* de Transações

Os principais fatores que afetam o *throughput* das transações na blockchain são a comunicação por *broadcast*, o mecanismo de consenso e a verificação da transação. A parte mais sensível entre eles é o mecanismo de consenso. Devido às características de confiança da blockchain, a garantia da exatidão e unicidade, exige-se que cada nó comprove trabalho suficiente para que possa expressar sua confiabilidade e autenticidade de sua própria mensagem. Embora o algoritmo PoW tenha resolvido o problema do gasto duplo, ele desperdiça muitos recursos e leva muito tempo para verificar as transações. Considerando que o número de aplicativos blockchain-IoT cresce a cada ano, novos mecanismos de consenso para aprimorar o desempenho da rede blockchain são necessários. Além disso, as soluções para resolver problemas de escalabilidade, capacidade de processamento ou armazenamento do dispositivo IoT na rede blockchain é um tópico que se mostra importante. Outra direção interessante de pesquisa futura é a análise do modelo de tráfego blockchain-IoT. No entanto, pode exigir uma grande escala de simulação, coleta de dados e análise de dados [Lao et al. 2020].

Tabela 4.3. Tamanho esperado dos dados em transações com o aumento do throughput. Fonte [Wu et al. 2019]

TPS	Tamanho do bloco	Taxas	Tamanho anual
1	0.3MB	0,12 BTC	15 GB
3	0.9MB	0,36 BTC	47 GB
10	3 MB	1.2 BTC	150 GB
100	30 MB	12 BTC	1.5 TB
1.000	300 MB	120 BTC	15 TB
10.000	3 GB	1.200 BTC	150 TB
100.000	30 GB	12.000 BTC	1.500 TB

4.7.3. Problemas Relativos ao Armazenamento

Na blockchain, todas as transações são registradas em blocos, e diferentes nós no sistema distribuído possuem uma cópia de toda esta informação. Isto pode impedir efetivamente que o livro-razão seja violado. Porém, com o acúmulo de transações e o crescimento dos dados, o espaço restante em cada bloco está ficando cada vez menor.

O volume de dados originados das transações pode impactar severamente no armazenamento e na capacidade da blockchain. Segundo [Wu et al. 2019], ao fazer uma estimativa simples, supondo que cada transação tenha 512 bytes e a unidade da taxa seja 0,0004 / KB, o tamanho dos dados da transação anual ficará muito caro para armazenar, conforme ilustrado na Tabela 4.3. De acordo com o registro da VISA em 2015, um total de 92.064 milhões de transações de pagamento foram geradas ao longo do ano. Se convertermos esse volume de dados de transações da rede em Bitcoins, o tamanho anual poderia ser 47 TB, o que está muito além da capacidade da máquina/banco de dados comum. Se aumentarmos o tamanho do bloco para 30 MB, o volume de dados de transação anual também pode ser de 1,5 TB, o que também é um número expressivo. Portanto, também é um problema muito desafiador expandir a capacidade da blockchain.

4.7.4. Privacidade e Segurança

Em blockchains públicas, cada participante pode obter um backup completo dos dados transacionados, que são abertos e transparentes, garantindo, desta forma, a confiabilidade da blockchain de uma certa maneira. No entanto, para muitos aplicativos de blockchain, esse recurso pode ser fatal. Em [Berdik et al. 2021] encontra-se um relatório abrangente sobre diferentes instâncias de estudos e aplicativos blockchain propostos pela comunidade de pesquisa e seus respectivos impactos na blockchain e seu uso em outros aplicativos ou cenários. O artigo [Leng et al. 2020] sob a perspectiva dos sistemas de informação aponta a nível de processo, de dados e de infraestrutura questões de segurança na blockchain, sugerindo direções futuras de pesquisa.

4.7.5. Problemas Relativos à Combinação com IoT

A integração das tecnologias necessárias à realização de aplicações envolvendo blockchain e IoT não é trivial. Os desafios mencionados nas subseções anteriores (atividades ilegais, *throughout* de transações, problemas de armazenamento e privacidade e segu-

rança) também podem se aplicar à integração com IoT, e alguns desafios ainda se tornam mais proeminentes neste contexto. Os principais problemas de segurança em relação à arquitetura em camadas da IoT, além dos protocolos usados para rede, comunicação e gerenciamento são catalogados e mostrados como a blockchain pode ser um habilitador-chave para resolvê-los [Khan and Salah 2018].

4.7.6. Problemas Relativos à Combinação com Outras Tecnologias

A blockchain possui potencial para ser aplicado em diversos campos, podendo assim ser combinada a várias outras tecnologias. Como exemplo, podemos citar a computação quântica, computação em nuvem e névoa, *Big Data*, inteligência artificial e realidade virtual. Para cada uma destas tecnologias há desafios de pesquisa em aberto.

Em relação à computação quântica, os protocolos criptográficos usados pela blockchain são suscetíveis a ataques pelo desenvolvimento de um computador quântico. Por exemplo, a segurança do Bitcoin será quebrada pelo enorme poder de computação dos computadores quânticos dentro de 10 anos. Tecnicamente, o esquema de assinatura da curva elíptica usado pelo Bitcoin poderia ser completamente quebrado por um computador quântico já em 2027 [Wu et al. 2019]. Em [Chen 2020] discute-se direções promissoras de criptografia, viabilidade da distribuição de chaves quânticas, gargalos encontrados pela rede de distribuição de chave quântica, conflitos simultâneos de links de retransmissão quântica em grande escala, atraso de retransmissão e acesso inconveniente a aplicativos que ainda não foram resolvidos completamente.

O uso de blockchain com Inteligência Artificial têm recebido grande parte da atenção das pesquisas nos últimos anos. Em [Ekramifard et al. 2020] são listados 23 artigos que demonstram desafios e aplicações inpiradoras que aumentam a segurança, a eficiência e a produtividade dos aplicativos.

Blockchain com computação em nuvem/névoa tem como uma grande dificuldade o estabelecimento de um ambiente seguro. Os dados na nuvem precisam ser transferidos pela Internet, então os desafios e comparação de vários problemas no ambiente de nuvem e problemas de segurança usando blockchain são relatados em [Pavithra et al. 2019].

O uso de blockchain com Realidade Virtual também possui desafios de pesquisa em aberto. Uma visão geral das oportunidades investigadas pelas soluções atuais combinando realidade virtual/realidade aumentada e blockchain é discutida, evidenciando oportunidades que poderiam promover a convergência dessas tecnologias e impulsioná-las em [Cannavo and Lamberti 2020].

Combinar *Big Data* com blockchain representa um grande potencial para melhorar os serviços e aplicativos de *big data*. Oportunidades de pesquisa e desafios em aberto para aplicativos de *big data* em diferentes domínios verticais, como cidade inteligente, saúde inteligente, transporte inteligente e rede inteligente são apresentados e discutidos para conduzir pesquisas adicionais nesta área promissora [Deepa et al. 2020].

4.7.7. Padrões de Blockchain

Algumas organizações ou institutos internacionais têm se dedicado ativamente ao estabelecimento de padrões de blockchain para regular e facilitar seu desenvolvimento. A

Tabela 4.4 mostra os padrões em desenvolvimento da ISO/TC (um comitê técnico internacional estabelecido em 2016, com o objetivo de padronizar blockchains e DLT's, do inglês *Distributed Ledger Technology*).

Tabela 4.4. Padrões em desenvolvimento da ISO/TC 307

Padrão	Objetivo	Estágio
ISO/AWI 22739	Terminologia	20.00
ISO/NP TR 23244	Visão geral da privacidade e proteção das informações de identificação pessoal	10.99
ISO/NP TR 23245	Risco de segurança e vulnerabilidades	10.99
ISO/NP TR 23246	Visão geral do gerenciamento de identidade usando tecnologias de blockchain e ledger distribuída	10.99
ISO/AWI 23257	Arquitetura	20.00
ISO/AWI TS 23258	Taxonomia e Ontologia	20.00
ISO/AWI TS 23259	Contratos inteligentes legalmente vinculativos	20.00
ISO/NP TR 23455	Visão geral e interações entre contratos inteligentes em blockchain e sistemas DLT	10.99
ISO/NP TR 23576	Segurança de ativos digitais	10.99
ISO/NP TR 23578	Problemas de descoberta relacionados à interoperabilidade	10.99

Além destes padrões, existem também:

- *Blockchain Reference Architecture*: Lançado pela China Blockchain Technology (CBT). Esta arquitetura divide a blockchain nas camadas de usuário, serviços, core e básica [Tecnology 2017b].
- *Blockchain–Data format specification*: Também lançado pela CBT. Este padrão organiza o formato dos dados, incluindo a estrutura e relacionamento, classificando os dados em seis categorias (dados da conta, do bloco, da transação, de entidades, dos contratos inteligentes e de configuração) [Tecnology 2017a]
- IEEE P2418: Padrão em desenvolvimento pela CTB para um framework de blockchain e IoT [Tecnology 2017c].
- *The Web Ledger Protocol* 1.0: Publicado pelo grupo de blockchain da W3C com o objetivo principal de proporcionar a flexibilidade e conectividade entre algoritmos [Sporny and D. 2017].

4.8. Conclusão

Este capítulo conduziu um levantamento das principais pesquisas na área de Internet das Coisas, blockchain e contratos inteligentes, desenvolvendo conceitos iniciais para nivelamento, descrevendo a arquitetura básica, os protocolos de consenso, as aplicações de blockchain e IoT, a montagem de um curso para estas tecnologias, os principais desafios e uma prática.

A Seção 4.2 trouxe conceitos de blockchain, contratos inteligentes, Internet das coisas e blockchain para IoT como objetivo de nivelar os conhecimentos.

A Sessão 4.3 apresentou as arquiteturas de blockchain de aplicações blockchain-IoT, e apresentou uma tabela comparativa entre as aplicações que usam blockchain e IoT, contemplando a camada de middleware, plataforma de blockchain, camada de rede, camada física e a classe de aplicação.

A Seção 4.4 apresentou os protocolos de consenso, agrupando-os em PoX, BFT e DAG. Os protocolos PoX são divididos em dois grandes grupos (PoW e PoS) e suas variantes. Logo após, os protocolos da família BFT e DAG foram descritos. Esta sessão também apresentou uma tabela comparativa entre estes protocolos, salientando as suas respectivas vantagens, desvantagens e aplicações.

A Seção 4.5 apresentou um histórico da evolução da blockchain e seus principais marcos, seguidas de uma análise das aplicações possíveis para a blockchain, agrupadas em : Mercado, Governo, saúde, Aplicações Gerais e IoT. Para encerrar esta seção, os autores apresentaram a prática e a página do minicurso com informações complementares, além de um tutorial para a execução.

Direcionamentos de como implementar um curso que contemple o conteúdo deste minicurso foram abordados na Seção 4.6. Por fim, a Seção 4.7 apresentou os desafios de pesquisa atuais nesta área do conhecimento.

Referências

[Abijaude et al. 2021] Abijaude, J., Serra, H., Sobreira, P., and Greve, F. (2021). Minicurso blockchain e contratos inteligentes para aplicações em iot, uma abordagem prática. `https://github.com/lifuesc/jai2021`. Acessado em 14/05/2021.

[Ahmad et al. 2021] Ahmad, R. W., Salah, K., Jayaraman, R., Yaqoob, I., Ellahham, S., and Omar, M. (2021). The role of blockchain technology in telehealth and telemedicine. *International Journal of Medical Informatics*, 148:104399.

[Alladi et al. 2019] Alladi, T., Chamola, V., Rodrigues, J. J., and Kozlov, S. A. (2019). Blockchain in smart grids: A review on different use cases. *Sensors*, 19(22):4862.

[Almatarneh 2020] Almatarneh, A. (2020). Blockchain technology and corporate governance: The issue of smart contracts—current perspectives and evolving concerns.

[Alphand et al. 2018] Alphand, O., Amoretti, M., Claeys, T., Dall'Asta, S., Duda, A., Ferrari, G., Rousseau, F., Tourancheau, B., Veltri, L., and Zanichelli, F. (2018). Iotchain: A blockchain security architecture for the internet of things. In *2018 IEEE wireless communications and networking conference (WCNC)*, pages 1–6. IEEE.

[Alves 2021] Alves, D. (2021). Proof-of-concept (poc) of restaurant's food requests in the lisk blockchain/sidechain. In *Journal of Physics: Conference Series*, volume 1828, page 012110. IOP Publishing.

[Angraal et al. 2017] Angraal, S., Krumholz, H. M., and Schulz, W. L. (2017). Blockchain technology: applications in health care. *Circulation: Cardiovascular quality and outcomes*, 10(9):e003800.

[Antonopoulos 2017] Antonopoulos, A. (2017). *Mastering Bitcoin: Programming the Open Blockchain*. O'reilly, 2nd edition.

[Araujo et al. 2019] Araujo, P., Viana, W., Veras, N., Farias, E. J., and de Castro Filho, J. A. (2019). Exploring students perceptions and performance in flipped classroom designed with adaptive learning techniques: A study in distributed systems courses. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 219.

[Atonomi 2019] Atonomi (2019). Atonomi—bringing trust and security to iot. `https://atonomi.io/`. Acessado em 04/05/2021.

[Atzori 2016] Atzori, M. (2016). Blockchain-based architectures for the internet of things: A survey (2016). *Available at SSRN 2846810 eLibrary*.

[Azaria et al. 2016] Azaria, A., Ekblaw, A., Vieira, T., and Lippman, A. (2016). Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30. IEEE.

[Bach et al. 2018] Bach, L. M., Mihaljevic, B., and Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1545–1550. IEEE.

[Back et al. 2014] Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timón, J., and Wuille, P. (2014). Enabling blockchain innovations with pegged sidechains. *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, 72.

[Bahga and Madisetti 2016] Bahga, A. and Madisetti, V. K. (2016). Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, 9(10):533–546.

[Balasubramaniam et al. 2020] Balasubramaniam, A., Gul, M. J. J., Menon, V. G., and Paul, A. (2020). Blockchain for intelligent transport system. *IETE Technical Review*, pages 1–12.

[Bartoletti and Pompianu 2017] Bartoletti, M. and Pompianu, L. (2017). An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International conference on financial cryptography and data security*, pages 494–509. Springer.

[Berdik et al. 2021] Berdik, D., Otoum, S., Schmidt, N., Porter, D., and Jararweh, Y. (2021). A survey on blockchain for information systems management and security. *Information Processing & Management*, 58(1):102397.

[Bessani et al. 2014] Bessani, A., Sousa, J., and Alchieri, E. E. (2014). State machine replication for the masses with bft-smart. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 355–362. IEEE.

[Biswas and Muthukkumarasamy 2016] Biswas, K. and Muthukkumarasamy, V. (2016). Securing smart cities using blockchain technology. In *2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS)*, pages 1392–1393. IEEE.

[Bogner et al. 2016] Bogner, A., Chanson, M., and Meeuw, A. (2016). A decentralised sharing app running a smart contract on the ethereum blockchain. In *Proceedings of the 6th International Conference on the Internet of Things*, pages 177–178. ACM.

[Buterin et al. 2014] Buterin, V. et al. (2014). A next-generation smart contract and decentralized application platform. *white paper*.

[Cannavo and Lamberti 2020] Cannavo, A. and Lamberti, F. (2020). How blockchain, virtual reality and augmented reality are converging, and why. *IEEE Consumer Electronics Magazine*.

[Castro et al. 1999] Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186.

[Chaer et al. 2019] Chaer, A., Salah, K., Lima, C., Ray, P. P., and Sheltami, T. (2019). Blockchain for 5g: opportunities and challenges. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE.

[Chen 2020] Chen, H. (2020). Quantum relay blockchain and its applications in key service. In *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, pages 95–99.

[Chowdhury et al. 2018] Chowdhury, M. J. M., Colman, A., Kabir, M. A., Han, J., and Sarda, P. (2018). Blockchain as a notarization service for data sharing with personal data store. In *2018 17th ieee international conference on trust, security and privacy in computing and communications/12th ieee international conference on big data science and engineering (TrustCom/BigDataSE)*, pages 1330–1335. IEEE.

[Christidis and Devetsikiotis 2016] Christidis, K. and Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *Ieee Access*, 4:2292–2303.

[Christin 2013] Christin, N. (2013). Traveling the silk road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd international conference on World Wide Web*, pages 213–224.

[Cocco et al. 2017] Cocco, L., Pinna, A., and Marchesi, M. (2017). Banking on blockchain: Costs savings thanks to the blockchain technology. *Future internet*, 9(3):25.

[Croman et al. 2016] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., et al. (2016). On scaling decentralized blockchains. In *International conference on financial cryptography and data security*, pages 106–125. Springer.

[Dashkevich et al. 2020] Dashkevich, N., Counsell, S., and Destefanis, G. (2020). Blockchain application for central banks: A systematic mapping study. *IEEE Access*, 8:139918–139952.

[De Angelis et al. 2018] De Angelis, S., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., and Sassone, V. (2018). Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain.

[Deepa et al. 2020] Deepa, N., Pham, Q.-V., Nguyen, D. C., Bhattacharya, S., Prabadevi, B., Gadekallu, T. R., Maddikunta, P. K. R., Fang, F., and Pathirana, P. N. (2020). A survey on blockchain for big data: Approaches, opportunities, and future directions. *arXiv preprint arXiv:2009.00858*.

[Dennis and Owen 2015] Dennis, R. and Owen, G. (2015). Rep on the block: A next generation reputation system based on the blockchain. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 131–138. IEEE.

[Dettling 2018] Dettling, W. (2018). How to teach blockchain in a business school. In *Business Information Systems and Technology 4.0*, pages 213–225. Springer.

[Díaz et al. 2016] Díaz, M., Martín, C., and Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing. *Journal of Network and Computer applications*, 67:99–117.

[Dorri et al. 2016] Dorri, A., Kanhere, S. S., and Jurdak, R. (2016). Blockchain in internet of things: Challenges and solutions. *CoRR*, abs/1608.05187.

[Dorri et al. 2017] Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017). Blockchain for iot security and privacy: The case study of a smart home. In *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pages 618–623. IEEE.

[Dorri et al. 2019] Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2019). Lsb: A lightweight scalable blockchain for iot security and anonymity. *Journal of Parallel and Distributed Computing*, 134:180–197.

[Dziembowski et al. 2015] Dziembowski, S., Faust, S., Kolmogorov, V., and Pietrzak, K. (2015). Proofs of space. In *Annual Cryptology Conference*, pages 585–605. Springer.

[Ekramifard et al. 2020] Ekramifard, A., Amintoosi, H., Seno, A. H., Dehghantanha, A., and Parizi, R. M. (2020). A systematic literature review of integration of blockchain and artificial intelligence. *Blockchain cybersecurity, trust and privacy*, pages 147–160.

[Etash 2020] Etash (2020). Etash protocol. https://eth.wiki/en/concepts/ethash/ethash. Acessado em 04/05/2021.

[Ethereum 2014] Ethereum, W. (2014). A secure decentralised generalised transaction ledger [j]. *Ethereum project yellow paper*, 151:1–32.

[EVM 2020] EVM (2020). Ethereum virtual machine. https://eth.wiki/en/concepts/evm/ethereum-virtual-machine-(evm)-awesome-list. Acessado em 04/05/2021.

[Fakhri and Mutijarsa 2018] Fakhri, D. and Mutijarsa, K. (2018). Secure iot communication using blockchain technology. In *2018 International Symposium on Electronics and Smart Devices (ISESD)*, pages 1–6. IEEE.

[Frankenfield 2018] Frankenfield, J. (2018). Proof of burn. https://www.investopedia.com/terms/p/proof-burn-cryptocurrency. Acessado em 04/05/2021.

[Greve et al. 2018] Greve, F., Sampaio, L., Abijaude, J., Coutinho, A., Valcy, Í., and Queiroz, S. (2018). Blockchain e a revolução do consenso sob demanda. *Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (Minicursos_-SBRC)*, 36.

[Greve 2005] Greve, F. G. P. (2005). Protocolos fundamentais para o desenvolvimento de aplicaç oes robustas. In *Minicursos SBRC 2005: Brazilian Symposium on Computer Networks*, pages 330–398.

[Guo and Liang 2016] Guo, Y. and Liang, C. (2016). Blockchain application and outlook in the banking industry. *Financial Innovation*, 2(1):1–12.

[Hou 2017] Hou, H. (2017). The application of blockchain technology in e-government in china. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–4.

[Huh et al. 2017] Huh, S., Cho, S., and Kim, S. (2017). Managing iot devices using blockchain platform. In *2017 19th international conference on advanced communication technology (ICACT)*, pages 464–467. IEEE.

[Ikeda 2018] Ikeda, K. (2018). Security and privacy of blockchain and quantum computation. In *Advances in Computers*, volume 111, pages 199–228. Elsevier.

[Ji et al. 2018] Ji, Y., Zhang, J., Ma, J., Yang, C., and Yao, X. (2018). Bmpls: blockchain-based multi-level privacy-preserving location sharing scheme for telecare medical information systems. *Journal of medical systems*, 42(8):1–13.

[Jiang et al. 2016] Jiang, J., Wen, S., Yu, S., Xiang, Y., and Zhou, W. (2016). Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys & Tutorials*, 19(1):465–481.

[Khan and Salah 2018] Khan, M. A. and Salah, K. (2018). Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411.

[Kiayias et al. 2017] Kiayias, A., Russell, A., David, B., and Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer.

[Kuperberg et al. 2019] Kuperberg, M., Kemper, S., and Durak, C. (2019). Blockchain usage for government-issued electronic ids: A survey. In *International Conference on Advanced Information Systems Engineering*, pages 155–167. Springer.

[Kwon 2014] Kwon, J. (2014). Tendermint: Consensus without mining. *Draft v. 0.6, fall*, 1(11).

[Lamport et al. 1982] Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401.

[Lao et al. 2020] Lao, L., Li, Z., Hou, S., Xiao, B., Guo, S., and Yang, Y. (2020). A survey of iot applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR)*, 53(1):1–32.

[Larimer 2014] Larimer, D. (2014). Delegated proof-of-stake (dpos). *Bitshare whitepaper*, 81:85.

[Lee and Lee 2015] Lee, I. and Lee, K. (2015). The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440.

[Leng et al. 2020] Leng, J., Zhou, M., Zhao, L. J., Huang, Y., and Bian, Y. (2020). Blockchain security: A survey of techniques and research directions. *IEEE Transactions on Services Computing*.

[Lerner 2015] Lerner, S. D. (2015). Dagcoin: a cryptocurrency without blocks. *White paper*.

[Lin and Liao 2017] Lin, I.-C. and Liao, T.-C. (2017). A survey of blockchain security issues and challenges. *IJ Network Security*, 19(5):653–659.

[Liu et al. 2020] Liu, B., Xiao, L., Long, J., Tang, M., and Hosam, O. (2020). Secure digital certificate-based data access control scheme in blockchain. *IEEE Access*, 8:91751–91760.

[Liu et al. 2017] Liu, B., Yu, X. L., Chen, S., Xu, X., and Zhu, L. (2017). Blockchain based data integrity service framework for iot data. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 468–475. IEEE.

[Liu 2016] Liu, P. T. S. (2016). Medical record system using blockchain, big data and tokenization. In *International conference on information and communications security*, pages 254–261. Springer.

[Matthews 2015] Matthews, C. M. (2015). Silk road creator found guilty of cybercrimes,. https://www.wsj.com/articles/silk-road-creator-found-guilty-of-cybercrimes-1423083107. Acessado em 04/05/2021.

[Mazieres 2015] Mazieres, D. (2015). The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation*, 32.

[Mehmood et al. 2017] Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., and Guizani, S. (2017). Internet-of-things-based smart cities: Recent advances and challenges. *IEEE Communications Magazine*, 55(9):16–24.

[Meiklejohn et al. 2013] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., and Savage, S. (2013). A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140.

[Mendling et al. 2018] Mendling, J., Weber, I., Aalst, W. V. D., Brocke, J. V., Cabanillas, C., Daniel, F., Debois, S., Ciccio, C. D., Dumas, M., Dustdar, S., et al. (2018). Blockchains for business process management-challenges and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, 9(1):1–16.

[Merkle-Patricia-Tree 2020] Merkle-Patricia-Tree (2020). Modified merkle patricia trie specification. https://eth.wiki/en/fundamentals/patricia-tree. Acessado em 04/05/2021.

[Milutinovic et al. 2016] Milutinovic, M., He, W., Wu, H., and Kanwal, M. (2016). Proof of luck: An efficient blockchain consensus protocol. In *proceedings of the 1st Workshop on System Software for Trusted Execution*, pages 1–6.

[Mistry et al. 2020] Mistry, I., Tanwar, S., Tyagi, S., and Kumar, N. (2020). Blockchain for 5g-enabled iot for industrial automation: A systematic review, solutions, and challenges. *Mechanical Systems and Signal Processing*, 135:106382.

[Mqtt 2017] Mqtt (2017). Message queuing telemetry transport. http://mqtt.org. Acessado em 04/05/2021.

[Nakamoto 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Technical report, Bitcoin Org.

[Narayanan et al. 2016] Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies*. Princeton University Press.

[Park et al. 2018] Park, J.-S., Youn, T.-Y., Kim, H.-B., Rhee, K.-H., and Shin, S.-U. (2018). Smart contract-based review system for an iot data marketplace. *Sensors*, 18(10).

[Pavithra et al. 2019] Pavithra, S., Ramya, S., and Prathibha, S. (2019). A survey on cloud security issues and blockchain. In *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*, pages 136–140. IEEE.

[PoET 2018] PoET (2018). Poet 1.0 specification. https://sawtooth.hyperledger.org/docs/core/releases/1.0/architecture/poet.html. Acessado em 04/05/2021.

[Popov 2018] Popov, S. (2018). The tangle. *White paper*, 1:3.

[Popov et al. 2020] Popov, S., Moog, H., and et al. (2020). The coordicide. *white paper Iota Foundation*.

[Rahimi et al. 2020] Rahimi, P., Khan, N. D., Chrysostomou, C., Vassiliou, V., and Nazir, B. (2020). A secure communication for maritime iot applications using blockchain technology. In *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 244–251. IEEE.

[Rao and Dave 2019] Rao, A. R. and Dave, R. (2019). Developing hands-on laboratory exercises for teaching stem students the internet-of-things, cloud computing and blockchain applications. In *2019 IEEE Integrated STEM Education Conference (ISEC)*, pages 191–198. IEEE.

[Rathee et al. 2021] Rathee, G., Balasaraswathi, M., Chandran, K. P., Gupta, S. D., and Boopathi, C. (2021). A secure iot sensors communication in industry 4.0 using blockchain technology. *Journal of Ambient Intelligence and Humanized Computing*, 12(1):533–545.

[Ruta et al. 2017] Ruta, M., Scioscia, F., Ieva, S., Capurso, G., and Di Sciascio, E. (2017). Semantic blockchain to improve scalability in the internet of things. *Open Journal of Internet Of Things (OJIOT)*, 3(1):46–61.

[Sagirlar et al. 2018] Sagirlar, G., Carminati, B., Ferrari, E., Sheehan, J. D., and Ragnoli, E. (2018). Hybrid-iot: Hybrid blockchain architecture for internet of things-pow sub-blockchains. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1007–1016. IEEE.

[Saint-Andre et al. 2004] Saint-Andre, P. et al. (2004). Extensible messaging and presence protocol (xmpp): Core.

[Samaniego et al. 2016] Samaniego, M., Jamsrandorj, U., and Deters, R. (2016). Blockchain as a service for iot. In *2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*, pages 433–436. IEEE.

[Seitz et al. 2017] Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and Tschofenig, H. (2017). Authentication and authorization for constrained environments (ace). *Internet Engineering Task Force, Internet-Draft draft-ietf-aceoauth-authz-07*.

[Shae and Tsai 2017] Shae, Z. and Tsai, J. J. (2017). On the design of a blockchain platform for clinical trial and precision medicine. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, pages 1972–1980. IEEE.

[Sharma et al. 2017] Sharma, P. K., Singh, S., Jeong, Y.-S., and Park, J. H. (2017). Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks. *IEEE Communications Magazine*, 55(9):78–85.

[Shelby et al. 2014] Shelby, Z., Hartke, K., and Bormann, C. (2014). The constrained application protocol (coap).

[Silvano and Marcelino 2020] Silvano, W. F. and Marcelino, R. (2020). Iota tangle: A cryptocurrency to communicate internet-of-things data. *Future Generation Computer Systems*, 112:307–319.

[Sornin et al. 2015] Sornin, N., Luis, M., Eirich, T., Kramp, T., and Hersent, O. (2015). Lorawan specification. *LoRa alliance*.

[Sousa et al. 2018] Sousa, J., Bessani, A., and Vukolic, M. (2018). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 51–58. IEEE.

[Sporny and D. 2017] Sporny, M. and D., L. (2017). The web ledger protocol 1.0. `http://standards.ieee.org/develop/project/2418.html`. Acessado em 04/05/2021.

[Stojkoska and Trivodaliev 2017] Stojkoska, B. L. R. and Trivodaliev, K. V. (2017). A review of internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140:1454–1464.

[Szabo 1997] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, 2(9).

[Sztajnberg et al. 2018] Sztajnberg, A., da Silva Macedo, R., and Stutzel, M. (2018). Protocolos de aplicação para a internet das coisas: conceitos e aspectos práticos. *Sociedade Brasileira de Computação*.

[Ta-Shma et al. 2017] Ta-Shma, P., Akbar, A., Gerson-Golan, G., Hadash, G., Carrez, F., and Moessner, K. (2017). An ingestion and analytics architecture for iot applied to smart city use cases. *IEEE Internet of Things Journal*, 5(2):765–774.

[Tecnology 2017a] Tecnology, C. B. (2017a). Blockchain data format specification. `http://www.cbdforum.cn/bcweb/index/bz/1-0.html`. Acessado em 04/05/2021.

[Tecnology 2017b] Tecnology, C. B. (2017b). Blockchain reference architecture. `http://www.cbdforum.cn/bcweb/index/article/bzwrr-1.html`. Acessado em 04/05/2021.

[Tecnology 2017c] Tecnology, C. B. (2017c). Standard for the framework of blockchain use in internet of things (iot). `http://standards.ieee.org/develop/project/2418.htm`. Acessado em 04/05/2021.

[Todd 2015] Todd, P. (2015). Ripple protocol consensus algorithm review. *Ripple Labs Inc White Paper (May, 2015) https://raw. githubusercontent. com/petertodd/rippleconsensus-analysis-paper/master/paper. pdf*.

[Underwood 2016] Underwood, S. (2016). Blockchain beyond bitcoin. *Communications of the ACM*, 59(11):15–17.

[Vinoski 2006] Vinoski, S. (2006). Advanced message queuing protocol. *IEEE Internet Computing*, 10(6):87–89.

[Vučinić et al. 2015] Vučinić, M., Tourancheau, B., Rousseau, F., Duda, A., Damon, L., and Guizzetti, R. (2015). Oscar: Object security architecture for the internet of things. *Ad Hoc Networks*, 32:3–16.

[Wu et al. 2019] Wu, M., Wang, K., Cai, X., Guo, S., Guo, M., and Rong, C. (2019). A comprehensive survey of blockchain: From theory to iot applications and beyond. *IEEE Internet of Things Journal*, 6(5):8114–8154.

[Xie et al. 2020] Xie, R., Wang, Y., Tan, M., Zhu, W., Yang, Z., Wu, J., and Jeon, G. (2020). Ethereum-blockchain-based technology of decentralized smart contract certificate system. *IEEE Internet of Things Magazine*, 3(2):44–50.

[Xu et al. 2017] Xu, C., Wang, K., and Guo, M. (2017). Intelligent resource management in blockchain-based cloud datacenters. *IEEE Cloud Computing*, 4(6):50–59.

[Xu et al. 2019] Xu, X., Zhang, X., Gao, H., Xue, Y., Qi, L., and Dou, W. (2019). Become: blockchain-enabled computation offloading for iot in mobile edge computing. *IEEE Transactions on Industrial Informatics*, 16(6):4187–4195.

[Yermack 2017] Yermack, D. (2017). Corporate governance and blockchains. *Review of Finance*, 21(1):7–31.

[Yoo and Ko 2020] Yoo, H. and Ko, N. (2020). Blockchain based data marketplace system. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1255–1257. IEEE.

[Yue et al. 2016] Yue, X., Wang, H., Jin, D., Li, M., and Jiang, W. (2016). Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):1–8.

[Zhang and Chen 2020] Zhang, C. and Chen, Y. (2020). A review of research relevant to the emerging industry trends: Industry 4.0, iot, blockchain, and business analytics. *Journal of Industrial Integration and Management*, 5(01):165–180.

[Zhang et al. 2020] Zhang, W., Luo, Y., Fu, S., and Xie, T. (2020). Privacy-preserving reputation management for blockchain-based mobile crowdsensing. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE.

[Zheng et al. 2017] Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*, pages 557–564. IEEE.