Capítulo 1

Finanças Descentralizadas em Redes Blockchain: Perspectivas sobre Pesquisa e Inovação em Aplicações, Interoperabilidade e Segurança

Josué N. Campos, Ronan D. Mendonça, Alexandre Fontinele, Luís H. S. de Carvalho, Isdael R. Oliveira, Ítallo W. F. Cardoso, Rafael Coelho, Allan E. S. Freitas, Glauber D. Gonçalves, José A. M. Nacif, Alex B. Vieira

Abstract

The support for smart contracts in blockchain networks has led to the emergence of a decentralized and automated finance ecosystem, DeFi (Decentralized Finance). The DeFi ecosystem aims to enhance traditional financial services through token trading, to reduce intermediaries and barriers to credit, and to broaden access to financial services in the context of the decentralized web. In this chapter, we provide an in-depth discussion of three essential and driving aspects of the DeFi ecosystem: (1) innovative applications, (2) interoperability between blockchain networks, and (3) security of the decentralized protocols that rule blockchain. In this sense, we currently present the fundamentals and practical examples of the most popular DeFi applications. We also discuss state-of-the-art research focusing on the three aspects above that will support the growth of DeFi in the coming years.

Resumo

O suporte a contratos inteligentes em redes blockchain propiciou a emergência de um novo ecossistema de finanças descentralizado e automatizado, denominado DeFi, do inglês, decentralized finance. O ecossistema DeFi visa aprimorar os serviços financeiros tradicionais por meio da negociação de tokens e da redução de intermediários e de barreiras para o crédito, assim como do acesso mais amplo a serviços financeiros no contexto da web descentralizada. Este capítulo traz uma discussão profunda sobre três aspectos essenciais e impulsionadores do ecossistema DeFi: (1) aplicações inovadoras, (2) interoperabilidade entre redes blockchain, e (3) segurança para as vulnerabilidades de ambientes descentralizados e competitivos. Nesse sentido, apresentamos os fundamentos

e exemplos práticos das aplicações DeFi mais populares atualmente. Por conseguinte, discutimos as pesquisas no estado da arte sob os três aspectos acima mencionados que suportarão o crescimento de DeFi nos próximos anos.

1.1. Introdução

Blockchain é uma tecnologia disruptiva, em especial para o setor produtivo, i.e., organizações nas áreas da agricultura, indústria e serviços, pois fornece recursos para o registro público, seguro e descentralizado de dados [Xu et al. 2019]. Essa tecnologia permite processar e armazenar dados de forma distribuída sem delegar o controle a uma autoridade central, e mesmo na existência de alguma parte não-confiável – o que é uma abordagem prática para o Problema dos Generais Bizantinos, já conhecido em ciência da computação desde a década de 80 [Lamport et al. 2019].

A blockchain foi concebida originalmente para garantir segurança às transações da plataforma de pagamentos baseada na criptomoeda Bitcoin [Nakamoto 2008]. Porém, o potencial dessa tecnologia é amplo e existe um crescente interesse por novas aplicações [Casino et al. 2019]. Em particular, as aplicações descentralizadas ou DApps, que atualmente funcionam no topo das redes blockchain, já ocupam uma posição relevante nas atividades econômicas e financeiras internacionais, e despertam o interesse de governos e empresas [Schär 2021].

DApps são programas de computador autônomos baseados em contratos inteligentes [Szabo 1997] e desenvolvidos em linguagens de alto nível, suportados desde a segunda geração de redes blockchain como a Ethereum [Wood 2014]. Um DApp pode ser constituído por um ou vários contratos inteligentes. Esses contratos, uma vez iniciados, executam automaticamente e de acordo com seu código registrado na blockchain.

Existem uma variedade de DApps que oferecem um intricado ecossistema de serviços, desde a área de finanças ao entretenimento digital. Tal ecossistema vem propiciando a emergência de uma nova geração da Internet baseada na infraestrutura de blockchain, popularmente intitulada web descentralizada [Murray et al. 2023]. Essa nova geração busca unificar as vantagens das raízes descentralizadas da primeira geração da Internet, suportadas por conteúdos públicos e protocolos abertos (e.g., TCP/IP e HTTP), com as funcionalidades da geração vigente (dita Web2), baseadas em plataformas centralizadas, a exemplo, serviços em nuvem e redes sociais providas por Amazon, Google, Facebook etc. A promessa da web descentralizada é que esses serviços tenham versões alternativas em DApps, favorecendo não apenas "*Big Techs*" mas todo um ecossistema de vários pequenos provedores de serviços tecnológicos na web.

A essência dos negócios em DApps se baseia no conceito de *token*, que é um objeto digital registrado na blockchain. Um *token* é único e está associado a um usuário, que é o seu proprietário, e somente este pode transferir a propriedade do *token* a outro usuário. Isso garante a possibilidade de valor ao *token*, i.e., a sua escassez ou impossibilidade de posses duplicadas (gasto duplo), diferentemente de objetos digitais tradicionais da web facilmente. Blockchain atua como a tecnologia base que permite a propriedade de *tokens* em redes de acesso público e confiável com segurança garantida por criptografia.

DeFi, do inglês *Decentralized Finance*, é um ecossistema de DApps que visa repli-

car, aprimorar ou substituir os serviços financeiros tradicionais por meio de negociação e transferência de *tokens*. A ideia de DeFi é eliminar intermediários e barreiras para o crédito no sistema bancário vigente, fornecendo acesso mais amplo a serviços financeiros no contexto da web descentralizada. Nesse sentido, DeFi oferece operações financeiras fundamentais (câmbio, saque e empréstimo com ou sem garantias) na forma de protocolos codificados em DApps de redes blockchain populares como Ethereum, Binance, Avalanche, dentre outras. Os protocolos permitem que usuários interajam diretamente com a blockchain, e permitem também que desenvolvedores e usuários combinem diferentes protocolos para criar soluções financeiras personalizadas e inovadoras.

Inicialmente, concebemos os *tokens* como criptomoedas, ou associados e lastreados a elementos de commodities de mercado (*e.g.* um *token* cuja unidade esteja associado a uma tonelada de minério de ferro). Mas, como no mundo real, um *token* pode ser único per si, ou seja, estar associado a algum objeto único, personalíssimo. Assim, o conceito de *token* foi estendido para a versão não fungível ou NFT, do inglês *Non-Fungible Token*, algo único.

Um *token* não fungível pode ser tipicamente associado a objetos multimídia, cujo conteúdo de texto ou imagem o confere características únicas e o torna também um objeto colecionável. NFT oferece recursos adicionais como a definição de um autor (criador) e recebimento de royalties do autor em transferências do *token*. Devido a esses recursos, NFTs vêm sendo adotados por artistas para criação e distribuição de conteúdo digital, visando a proteção do direito autoral e do ganho com royalties na revenda de itens. Por exemplo, em 11 de março de 2021, o artista Beeple realizou a venda de sua obra de arte digital em formato de NFT na blockchain Ethereum pelo valor de US$ 69 milhões. É importante observar que essas obras podem ser acessadas gratuitamente na Internet por se tratarem de um objeto digital. Contudo, quanto mais popular é o NFT, mais benefícios ele pode trazer ao seu proprietário, que em tese possui direitos exclusivos sobre a sua comercialização e imagem [Okonkwo 2021].

DApps no ecossistema DeFi com seus respectivos *tokens* tradicionais ou NFT são abertos, transparentes e acessíveis a qualquer pessoa com uma conexão à Internet, permitindo que indivíduos cadastrados em uma blockchain via um par de chaves pública e privadas vendam, emprestem ou troquem seus *tokens* de maneira descentralizada. Esse ecossistema cresce gradativamente desde o seu apogeu em 2014 com o surgimento da blockchain Ethereum, e abre novas oportunidades de inovações tecnológicas em aplicações, mas também enfrenta desafios, como a interoperabilidade entre as redes blockchain e a segurança dos contratos inteligentes.

Esse capítulo versa sobre os principais tópicos tecnológicos e de pesquisa envolvendo DeFi. Nesse sentido, iniciamos descrevendo as principais aplicações DeFi e seus fundamentos técnicos na Seção 1.2. A seguir, apresentamos o tópico interoperabilidade entre redes blockchain como uma consequência da evolução do ecossistema DeFi na Seção 1.3. As questões de segurança, vitais para a existência e credibilidade de DeFi são discutidas na Seção 1.4. Finalmente, apresentamos nossas considerações finais com um resumo do capítulo na Seção 1.5. Importante mencionar que materiais adicionais, atualizações desse capítulo, códigos fonte, artigos e resultados de pesquisa desenvolvidos pelos autores podem ser obtidos no repositório `https://github.com/LABPAAD/blockchain_defi`.

1.2. Aplicações

Nesta seção, chamamos a atenção para as aplicações DeFi mais proeminentes recentemente. Contudo, antes de discuti-las aprofundamos nos fundamentos que abrangem todo o ecossistema DeFi.

A base das aplicações de finanças descentralizada é a tomada de decisão sem a necessidade de uma terceira parte confiável. Ou seja, aplicações que possibilitem diferentes tipos de soluções financeiras (desde empréstimos, a câmbio, a hipotecas, a investimentos etc.), sem existir uma entidade financeira como terceira parte envolvida para prover o cumprimento das cláusulas contratuais esperadas. A ausência desta entidade requer que as partes envolvidas estabeleçam um protocolo adequado para a tomada de decisão quanto à execução de uma transação, e um substrato adequado para o registro do resultado desta execução, ambos de forma descentralizada.

O mecanismo de tomada de decisão é o do consenso distribuído, um dos problemas fundamentais da computação distribuída e bloco de construção de diversas soluções descentralizadas. E uma vez atingido o resultado deste consenso, este é persistido de forma distribuída em uma estrutura baseada em uma cadeia de blocos com uso de técnicas criptográficas como chaves pública/privada e sumários criptográficos (*hash* de modo a prover mecanismos de audaitabilidade, autenticidade, não-repúdio e integridade dos dados [Greve et al. 2018]).

Com tais características, esta estrutura provê um livro-razão que garante a integridade das transações neste armazenado, mas mantido de forma distribuída, sem exigência da terceira parte confiável.

Um amplo conjunto de possibilidades de aplicações financeiras descentralizadas, ditas aplicações DeFi, podem ser construídas por meio deste livro-razão distribuído subjacente provido pela blockchain para o registro de transações replicadas em uma rede de pares (p2p). Tomamos a blockchain como uma base para DeFi e encaminhamos o leitor para outros trabalhos existentes (notavelmente [Greve et al. 2018]) para uma exposição mais completa sobre a tecnologia blockchain. Assim, assumimos no presente texto que blockchain é a base para aplicações DeFi, e por conseguinte, estas herdam todas as suas propriedades de segurança, consistência, integridade e disponibilidade de registros. Sem essas propriedades de segurança, aplicações DeFi se tornariam inerentemente inseguras.

1.2.1. Contratos Inteligentes e Transações

Os contratos inteligentes [Szabo 1997] desempenham um papel fundamental no ecossistema DeFi: permitem a automatização e execução de acordos financeiros entre as partes, e.g., vendedor e comprador ou credor e devedor, sem a necessidade de intermediários. Em nosso contexto, os contratos inteligentes são programas de computador autônomos, e são registrados em uma blockchain, estabelecendo regras e condições para orientar as interações entre as partes envolvidas. Logo, a execução de contratos é determinada por lógica programada, o que pode resultar em redução de custos e maior previsibilidade no cumprimento dos acordos financeiros.

A rede blockchain Ethereum foi a primeira a permitir contratos inteligentes [Wood 2014]. O Bitcoin, considerado a primeira blockchain [Nakamoto 2008], permite apenas

operações de transferência de valores entre usuários, sem a possibilidade de desenvolver programas que sejam registrados na blockchain. A maioria das redes blockchain que surgiram posteriormente ao Ethereum já incluem contratos inteligentes, dado a inovação que esse recurso possibilitou na construção de diferentes aplicações, incluindo as de DeFi baseadas em operações financeiras decentralizadas.

```
1  // SPDX-License-Identifier: GPL-3.0
2  pragma solidity >=0.7.0 <0.9.0;
3
4  contract C{
5      bytes32 nome;
6      function get(bytes32 _nome) public{
7       nome = _nome;
8      }
9      function set() public view returns (bytes32) {
10      return nome;
11     }
12 }
```

Algoritmo 1.1. Exemplo de código fonte de contrato inteligente tomando com referência a linguagem Solidity Ethereum, adaptado de [Palma et al. 2022].

Mais especificamente, um contrato inteligente é um programa de computador escrito em uma linguagem de alto nível, como ilustrado no Algoritmo 1.1, adaptado de [Palma et al. 2022]. Note que, após ser desenvolvido e testado, um contrato necessita ser implantado, ou seja, registrado de forma imutável, na blockchain.

Nesta seção, tomamos como referência o desenvolvimento e uso de contratos inteligentes em uma blockchain baseada em Ethereum. O registro do contrato é feito pela compilação deste em bytecode e implantado na blockchain por meio da operação denominada *contract create* Após a execução bem-sucedida dessa operação, o contrato recebe um endereço exclusivo na blockchain, permitindo que qualquer usuário interaja com suas funcionalidades.

O contrato implantado na blockchain com suas respectivas aplicações clientes, que transmitem as requisições dos usuários, constituem o que é conhecido como aplicação descentralizada ou DApp. Para interagir com o DApp, é necessário submeter uma transação à blockchain, tendo como destino o endereço do contrato e o nome da função desejada, como ilustra a Figura 1.1 proposta por [Palma et al. 2022] para fins didáticos. Contudo, essa transação deve ser realizada por meio de uma aplicação cliente conectada à rede blockchain e equipada com as interfaces de comunicação adequadas para o contrato em questão.

Outra forma de interagir com um contrato implantado na blockchain é por meio de outro contrato. Nesse caso, várias requisições podem ser previamente programadas seguindo uma estratégia automatizada para realizar um objetivo específico, o que tipicamente é conhecido como um *bot*. É importante observar ainda que um contrato na blockchain possui um estado definido pelas variáveis programadas no contrato. Consequentemente, a execução de funções podem alterar o estado deste contrato.

Para que um contrato inteligente seja executado, um usuário deve interagir com ele

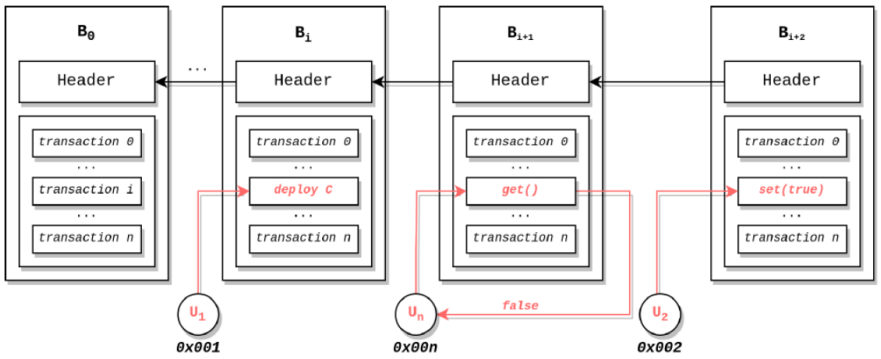

Figura 1.1. Exemplo de execução de uma transação que modifica um estado definido pelas variáveis programadas no contrato [Palma et al. 2022].

por meio de uma transação confirmada na blockchain. Após a confirmação da transação, o código do contrato é executado por um nó da rede e o estado é atualizado de acordo com as instruções do contrato. A tarifação das transações é determinada pelo consumo de "gás", uma unidade específica de custo computacional na blockchain. A Máquina Virtual Ethereum (EVM) usa um conjunto específico de instruções para execução de tarefas, cada uma consumindo uma quantidade específica de gás. O remetente de uma transação deve pagar o total de gás consumido por essa transação, calculado com base nas instruções executadas e no preço do gás naquele momento [Werner et al. 2022].

Execução de uma transação. Quando um participante da rede blockchain deseja fazer uma transação, os detalhes da transação não confirmada são primeiro transmitidos para uma rede de pares, validados e, em seguida, armazenados em uma área de espera (o *mempool* de um nó). Este grupo de transações é então propagado entre os nós da rede. Participantes do livro-razão subjacente responsável por garantir o consenso, os nós proponentes (que podem ser ditos mineradores ou forjadores, caso a rede seja, respectivamente, baseada em proof-of-work, PoW, ou proof-of stake, PoS) escolhem quais transações incluir em um determinado bloco, com base em parte na taxa de transação associada a cada transação. Transações em um bloco são executadas sequencialmente na ordem em que o minerador do respectivo bloco os incluía. Para um tratamento detalhado como esse processo funciona, encaminhamos o leitor para [Greve et al. 2018, Xu et al. 2019]. Nós proponentes têm a capacidade de controlar a sequência em que as transações são executadas. Consequentemente, os proponentes podem solicitar transações de formas que lhes renderão receitas e até mesmo inserir suas próprias transações para extrair mais receitas. Os mesmos podem ainda ser subornados para faça tal reordenação de transação, questões essas que serão discutidas na Seção 1.4.

1.2.2. Arquitetura DeFi

Aplicações DeFi, em blockchain, vem sendo foco de diversos estudos que abordam desde a taxonomia, à estruturação de conceitos e à arquitetura. A proposta mais proeminente e popularmente adotada até então é a arquitetura em camadas de [Schär 2021], que será a adotada neste capítulo. Essa arquitetura utiliza um modelo hierárquico em cinco camadas que têm propósitos distintos e complementares como mostra a Figura 1.2. Como usual nesse tipo de arquitetura, as camadas inferiores suportam as superiores em termos de

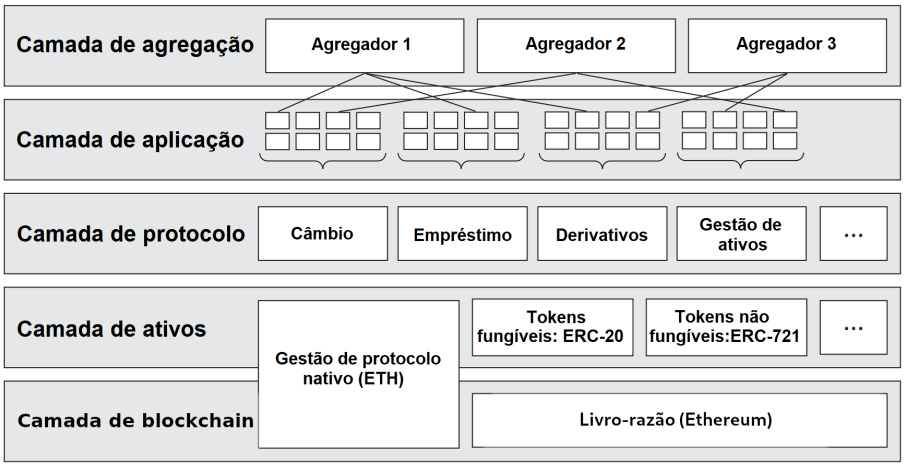

Figura 1.2. Arquitetura hierárquica de cinco camadas usualmente adotada para aplicações DeFi. Figura adaptada de [Schär 2021].

funcionalidades, o que permite às camadas mais altas ter objetivos específicos utilizando todos os recursos disponíveis na pilha de camadas ligeiramente inferior. É importante, contudo, observar que as camadas que aplicações DeFi são hierárquicas e seguras quanto às suas camadas inferiores. Ou seja, por exemplo, se a blockchain na camada mais inferior for comprometida, todas as camadas subsequentes não seriam seguras, e quaisquer esforços de descentralização nas camadas subsequentes seriam ineficazes.

1. Camada blockchain (Camada 1) consiste de um livro-razão para o registro de transações de modo imutável (*append only*) replicado entre vários computadores via redes de pares (e.g., Bitcoin ou Ethereum). Ela permite que a rede armazene informações com segurança e garante que quaisquer novos registros sejam anexados cumprindo um conjunto de regras acordadas entre os pares participantes da rede.

2. Camada de ativos (Camada 2) consiste de todos os ativos, i.e., *tokens*, que são emitidos sobre a camada de Blockchain. Isso inclui o ativo nativo da rede blockchain – e.g., Ether na rede Ethereum ou Matic na rede Polygon, bem como quaisquer outros ativos derivados destes que são emitidos sobre uma blockchain.

3. Camada de protocolos (Camada 3) provê padrões para casos de usos específicos, e.g., serviços para suportar trocas, empréstimos ou gerenciamento de ativos, que são registradas na blockchain. Esses padrões são implementados como um conjunto de contratos inteligentes que podem ser acessados por qualquer usuário (ou aplicativo DeFi).

4. Camada de aplicações (Camada 4) consiste em aplicativos direcionados aos usuários para se conectarem aos protocolos. Nessa camada, os contratos inteligentes geralmente são abstraídos por uma aplicação *front-end* baseado em navegador da Web ou até mesmo um software cliente de sistema operacional, tornando os protocolos fáceis de serem utilizados por usuários. É na camada de aplicação que se encontram as aplicações DeFi.

5. Camada de agregação (Camada 5) é uma extensão da camada de aplicação. Nessa camada os agregadores criam plataformas centradas no usuário que se conectam a vários aplicativos e protocolos. Os agregadores, geralmente, fornecem ferramentas para comparar e classificar protocolos permitindo a execução de tarefas complexas, conectando vários protocolos simultaneamente e combinando informações relevantes de maneira interativa. Geralmente, as aplicações na camada de agregação e aplicação funcionam da mesma maneira, através de uma interface Web ou aplicativo.

Visto o modelo conceitual ora apresentado para aplicações DeFi, vamos focar nas camadas de ativos e de protocolos. Logo, na seção seguinte serão discutidos os principais protocolos para gestão e operação com ativos na blockchain. Isso nos permite estabelecer a base necessária para os próximos conteúdos deste capítulo que tratam de interoperabilidade e segurança em DeFi.

1.2.3. *Tokens*

Negócios em DeFi se baseiam no conceito de *token*, que, de forma simplificada no contexto de redes blockchain, pode ser definido como um objeto digital registrado na blockchain. Um *token* está associado a um usuário, que é o seu proprietário, e somente este pode transferir a propriedade do *token* a outro usuário. Isso garante a possibilidade de valor ao *token*, i.e., a sua escassez ou impossibilidade de posses duplicadas (gasto duplo). Diferentemente de objetos digitais tradicionais da web, o *token* sempre tem um proprietário.

A ideia geral de *tokens* é tornar bens (i.e., ativos) tangíveis ou intangíveis mais acessíveis, assim como as transferências ou negociações desses ativos mais eficientes, assim um ativo so mundo real (dito *tokenizado*) pode ser representado na blockchain por um *token*, e uma transação que altere a propriedade deste *token* no livro-razão corresponderia a própria mudança de propriedade do ativo em si.

Dessa forma, ativos *tokenizados* podem ser transferidos facilmente em fração de segundos para qualquer pessoa ou organização no mundo. Por conseguinte, esses *tokens* podem ser usados por diferentes aplicações descentralizadas, desde que essas sejam constituídas pelos contratos ao qual os *tokens* estão vinculados. Importante mencionar que *tokens* estão associados a contratos inteligentes, por onde podem ser emitidos e transferidos aos respectivos proprietários.

Atualmente existem diferentes tipos de *tokens* onde cada um possui as suas características e a suas utilidades que serão discutidas a seguir.

1.2.3.1. *Token* Fungível

Token fungível foi o primeiro a ser desenvolvido na plataforma Ethereum, através de um padrão denominado ERC-20. Esse padrão tem como objetivo definir e controlar a emissão e distribuição de *tokens*. Assim, o padrão ERC-20 desempenha um papel crucial no ecossistema de *tokens* na Ethereum, simplificando a criação de *tokens* para desenvolvedores e possibilitando uma interação fluida entre usuários e uma diversidade de *tokens* em inúmeros aplicativos e serviços descentralizados.

O padrão ERC-20 estabelece um conjunto de funcionalidades características para um objeto digital ser considerado *token* fungível. São elas:

- **Nome e símbolo**: o *token* deve ter um nome descritivo e um símbolo associado a ele.
- **Decimais**: também deve ser especificado a quantidade de casas decimais que podem ser usadas ao exibir o saldo do *token*.
- ***Total supply***: registro do total de *tokens* em circulação emitidos e associados a um contrato padrão ERC-20.
- **Eventos**: existem dois eventos definidos por esse padrão. Um deles é acionado sempre que ocorre uma transferência de *tokens*, o que possibilita o registro e rastreamento das transferências (*Transfer*). O outro evento possibilita que o proprietário de *tokens* autorize outra conta a gastar uma quantidade específica de seus *tokens* em seu nome (*Approval*)
- ***Transfer***: permite que um proprietário possa enviar *tokens* para outros usuários da blockchain.
- **Emissão e queima**: permissão para emissão (*minting*) de novos *tokens* controlado pelos proprietários do contrato, ao passo que a queima (*burning*), i.e., destruição de *tokens* em circulação é controlada pelo proprietário do *token*.

Considerando esses conceitos e funcionalidades, *tokens* fungíveis são geralmente categorizados nos seguintes grupos:

- ***Tokens* de utilidade**: são criados com a finalidade de proporcionar acesso a serviços, produtos ou funcionalidades específicas dentro de um ecossistema blockchain. Eles permitem aos usuários pagarem por transações, votar em decisões de governança ou acessar serviços exclusivos oferecidos pela organização que emitiu os *tokens*.
- ***Tokens* de segurança**: representa ativos do mundo real, como ações, títulos, imóveis, obras de arte, fundos de investimento e outros ativos financeiros ou tangíveis. Jurisdições de alguns países como USA, Malásia, esses *tokens* podem ser considerados como título financeiro e estão sujeitos a regulamentações rigorosas, assim como títulos tradicionais emitidos por empresas. A principal característica de *token* de segurança é oferecer aos detentores direitos legais e econômicos sobre o ativo subjacente. Consequentemente, os seus possuidores adquirem privilégios tais como o recebimento de dividendos, participações em lucro, direito de voto em deliberações corporativas, entre outros, condicionados ao tipo de ativo que o *token* representa.
- ***Tokens* de governança**: representação digital que concede a seus detentores o privilégio de engajar-se nas deliberações de governança de uma organização autônoma descentralizada ou recursos e serviços especiais de uma plataforma blockchain. O propósito primordial desse *token* é capacitar a comunidade de usuários, assegurando que as determinações associadas ao desenvolvimento e operação de uma organização sejam efetuadas por meio de um processo descentralizado e democrático.

1.2.3.2. *Token* não Fungível

Os *tokens* não fungíveis, popularmente conhecidos como NFTs (*Non-Fungible Token*) é um tipo específico de objeto digital que representa a propriedade de um ativo único ou colecionável na blockchain. O que torna os NFTs únicos é o fato de serem não fungíveis, ou seja cada NFT é distinto e não pode ser trocado diretamente por outro de maneira idêntica em termos de valor, ao contrário das criptomoedas tradicionais, como o Bitcoin, que são fungíveis e podem ser trocadas umas pelas outras em proporções equivalentes.

O padrão de NFTs define um conjunto de regras e interfaces para a criação e interação com *tokens* digitais únicos e indivisíveis. O padrão mais comum para NFTs na blockchain Ethereum é o ERC-721 [Entriken et al. 2018], embora existam outros padrões e variações. As principais características definidas pelo padrão ERC-721 são:

- **Unicidade e Indivisibilidade**: Os *tokens* ERC-721 são únicos e indivisíveis, o que significa que cada *token* tem atributos e características exclusivas que o distinguem de outros *tokens*. Por exemplo, um NFT pode representar uma obra de arte digital específica, um item de jogo ou um bilhete de evento único. Em contraste, os *tokens* ERC-20 são fungíveis, o que significa que cada *token* é igual e pode ser trocado por outro *token* da mesma classe.
- **Transferências Individuais**: No padrão ERC-721, os *tokens* são transferidos individualmente, um de cada vez. Isso permite que cada NFT tenha sua própria história de propriedade e rastreabilidade na blockchain. Em contrapartida, os *tokens* ERC-20 podem ser transferidos em lotes, facilitando transações em massa de *tokens* fungíveis.
- **Métodos de Interface**: O padrão ERC-721 define uma série de métodos de interface padrão, incluindo *balanceOf* (para verificar o saldo de *tokens* de um proprietário), *ownerOf* (para verificar o proprietário de um *token*), *transferFrom* (para transferir a propriedade de um *token*) e outros. Esses métodos são adaptados para lidar com a singularidade e indivisibilidade dos *tokens* NFT.
- **Metadados e Informações Extras**: Os NFTs geralmente contêm metadados adicionais que descrevem o ativo representado pelo *token*. Esses metadados podem incluir informações sobre o autor da obra de arte, a data de criação, a descrição do item de jogo, entre outros detalhes relevantes. Os *tokens* ERC-721 permitem o armazenamento e a recuperação desses metadados de forma eficiente. Os *tokens* ERC-20 também podem armazenar metadados, mas geralmente se concentram em aspectos mais básicos, como nome, símbolo e número de casas decimais.

Cada NFT é único e possui suas próprias características. Um NFT pode ser, por exemplo, uma imagem, uma música ou até mesmo um item colecionável de um jogo (como uma roupa ou armamento). Um *token* não fungível também pode estar associado a um único objeto do mundo real, como um *token* de propriedade de uma obra de arte, ou mesmo um *token* correspondente a um ingresso em um dado lugar em uma apresentação única de uma peça de teatro. Os NFTs não podem ser divididos em partes menores pois são comercializados, por sua própria natureza, como unidades

únicas e indivisíveis. A titularidade de um NFT é registrada e verificada na blockchain, proporcionando autenticidade e singularidade ao ativo digital que ele simboliza. Esses *tokens* são comercializados a partir de plataformas de compra e venda ou até mesmo dentro do ecossistema o qual ele faz parte (dentro de um dApp como o Axie Infinity).

1.2.3.3. Stablecoin

Um ponto negativo nos *tokens* fungíveis é a sua alta volatilidade, o que cria obstáculos para usuários que buscam explorar os benefícios dos aplicativos DeFi, mas têm aversão ao risco associado a ativos voláteis, como o ETH. Como resposta a essa questão, surgiu uma categoria específica de criptomoedas conhecida como stablecoins. As stablecoins são projetadas para manter uma paridade de preço com um ativo específico, como o dólar americano ou o ouro.

Dentro de uma blockchain como a Ethereum, as stablecoins são definidas por padrões como os de *tokens* ERC-20. Esses *tokens* proporcionam a estabilidade necessária que os investidores procuram para participar de diversas aplicações DeFi, permitindo que uma solução nativa em *tokens* propicie posições em ativos digitais com menor volatilidade. Além disso, elas podem ser utilizadas para oferecer exposição *off-chain* aos retornos de ativos externos à blockchain subjacente, como, por exemplo, ouro, ações ou ETFs.

Os mecanismos pelos quais as stablecoins mantêm seu lastro podem variar dependendo da implementação [Goetze 2023] [Binance 2023]. No entanto, os quatro métodos principais são:

- **Stablecoins com garantia ou colateralizadas**: a ideia principal consiste em manter um fundo fiduciário na moeda de lastro. Dessa forma, se o valor da stablecoin sobe acima de uma unidade por moeda fiduciária então os arbitradores passaram a vender a stablecoin para trocar pela moeda fiduciária. Caso o valor da stablecoin caia abaixo de uma unidade então os arbitradores (usuários) passaram a comprar a stablecoin causando uma escassez dessa stablecoin, o que ocasiona uma alta no preço voltando novamente a relação de um para um. O mesmo vale para metais preciosos, petróleo e imóveis. Ainda que exista flutuações no preço essa ação faz com que a flutuação seja frações de centavos, mantendo o preço com uma flutuação de apenas algumas casas centesimais.
- **Stablecoins com garantia em criptomoeda ou cripto-colateralizadas**: pode ser considerado a segunda maior classe de stablecoins. Estas stablecoins são respaldadas por um fundo de outra criptomoeda. Seu valor pode ser ancorado de forma rígida ou flexível ao ativo de lastro, isso depende das diretrizes do protocolo. A stablecoin cripto-colateralizada mais popular é o DAI, criado pelo MakerDAO e é respaldado principalmente por ETH, com suporte colateral para alguns outros criptoativos. Ele está ancorado de forma flexível por meio de mecanismos econômicos que incentivam oferta e demanda para levar o preço a US$ 1. A capitalização de mercado do DAI é de pouco mais de US$ 5 bilhões.
- **Stablecoins sem garantia ou não colateralizadas**: este tipo de stablecoin não são apoiadas por um fundo de reserva. Este tipo de stablecoin utiliza um conjunto de

regras para controlar a oferta e demanda do *token* com o objetivo de manter seu valor próximo ao da moeda fiduciária desejada. Esse algoritmo tem a responsabilidade de ajustar automaticamente a oferta ou a demanda do *token* para trazê-lo de volta à paridade, caso o valor da stablecoin se desvia do valor desejado.

- **Stablecoins híbridas**: As stablecoins híbridas representam uma categoria de criptomoedas que incorpora características tanto das stablecoins colateralizadas quanto das não colateralizadas. Seu objetivo primordial é assegurar estabilidade de preço por meio de uma sinergia entre reservas de ativos tangíveis e algoritmos avançados. Esse modelo híbrido visa atingir um equilíbrio entre a estabilidade associada às stablecoins colateralizadas e a flexibilidade e descentralização características das não colateralizadas. À semelhança das stablecoins colateralizadas, as stablecoins híbridas podem manter reservas compostas por moedas fiduciárias, criptomoedas ou outros ativos digitais, que funcionam como garantia para sustentar o valor do *token*. Um exemplo de stablecoin híbrida é a Frax que mantém parte do seu funcionamento em garantia e parte do fornecimento algoritmo, onde a proporção em garantida e de algorítmicos depende do preço do atual do *token* em mercado.

De longe, a maior parte das stablecoins tem garantia fiduciária e normalmente estes são custodiados por uma entidade externa ou grupo de entidades que são submetidos a auditorias de rotina para verificar a existência das garantias. A maior stablecoin com garantia fiduciária é o Tether (USDT), com uma capitalização de mercado de mais de US$ 100 bilhões de dólares, tornando-se a terceira maior criptomoeda atrás do Bitcoin e do Ethereum. Tether também tem o maior volume de negociação que qualquer criptomoeda. O segundo maior é o USDC, apoiado pela Coinbase e pela Circle. USDT e USDC são muito populares para integração em protocolos DeFi, pois estão disponíveis para troca nas principais corretoras da rede Ethereum, como Uniswap e Jupiter, e a demanda por oportunidades de investimento em stablecoin é alta.

Um dos pontos negativos das stablecoins colateralizadas é a sua característica de que o fundo do ativo de lastro estar armazenado em uma entidade centralizada fora da blockchain. Neste ponto em questão, as stablecoins cripto-colateralizadas têm as vantagens da descentralização e garantia de colateral seguro. A desvantagem é que sua escalabilidade é limitada. Pois, para emitir mais unidades da stablecoin, um usuário deve necessariamente respaldar a emissão por meio de uma posição de dívida super colateralizada. Em alguns casos, como o DAI, há até mesmo um teto de dívida que limita ainda mais o crescimento do suprimento.

1.2.3.4. Central Bank Digital Currency

Uma moeda digital emitida por banco central, do inglês Central Bank Digital Currency (CBDC), é uma forma de moeda digital emitida pelo banco central de um país. É semelhante às criptomoedas, exceto que seu valor é fixado pelo banco central e, em regra, equivale à moeda fiduciária do país, tendo curso forçado, ou seja, é aceita pela economia por força de lei. A aceitação de stablecoins incentivou os bancos centrais a explorar os possíveis benefícios e custos da emissão de moedas digitais de banco central.

No campo das políticas públicas, há discussões intensas sobre os diferentes arranjos possíveis para as CBDCs [Allen et al. 2022]. Primeiramente, discute-se se o CBDC deve ser um instrumento de liquidação entre instituições financeiras (atacado) ou um sistema acessível a todos os consumidores (varejo), onde o CBDC seria um passivo do banco central. Outra possibilidade é, se adotada a CBDC no sistema de varejo, qual o papel do banco central: interagir diretamente com o público ou, delegar ao sistema financeiro a gestão de todas as atividades de atendimento ao cliente. Ainda, há intensos debates sobre as permissões na criação da CBDC, uma vez que diferentes características podem impactar a eficácia da política monetária e a estabilidade financeira.

Muitos países estão a desenvolver CBDCs, e alguns até já as implementaram. Por exemplo, a adoção do pagamento móvel e digital na China tem sido significativamente mais rápida do que na maioria dos outros países. Em 2019, os pagamentos via Alipay e WeChat ultrapassaram 500 milhões e 900 milhões de usuários ativos mensais, representando 36% e 65%, respectivamente, da população total da China [Frost et al. 2019]. Este ambiente de negócios foi favorável à criação de uma CBDC pelo banco central chinês, o Banco Popular da China (do inglês The People's Bank of China – POBC). No final de 2017, após aprovação do Conselho de Estado da China, o PBOC iniciou a colaboração com instituições comerciais para desenvolver e testar a moeda fiduciária digital, o e-CNY. O objetivo era estabelecer um sistema monetário respaldado pelo Estado, além de um simples sistema de pagamentos. Após alguns anos de trabalho, em abril de 2020, o PBOC anunciou testes em quatro cidades (Shenzhen, Suzhou, Xiong'an e Chengdu). Desde janeiro de 2022, a Tencent lançou serviços de e-CNY no WeChat, e diversas outras gigantes da Internet, como JD.com e Didi Taxi, também começaram a aceitar pagamentos em e-CNY nas cidades-piloto. Durante os Jogos Olímpicos de Inverno de 2022, a China testou com sucesso a aceitação do e-CNY, disponibilizando o aplicativo móvel e cartões de pagamento ou pulseiras do e-CNY para visitantes estrangeiros [Allen et al. 2022].

Um outro caso de estudo, ainda em desenvolvimento, é o DREX, CBDC brasileiro. O Brasil possui um dos sistemas bancários digitais mais avançados do mundo em seu ápice com o advento do PIX – mecanismo de pagamentos e transferência de valores que propiciou uma virtual universalização de movimentação de moeda de forma digital pela população brasileira. O Banco Central do Brasil tem sido transparente sobre motivações e processos no desenvolvimento do DREX, o que pode proporcionar uma oportunidade única de explorar o desenvolvimento e a implementação de um CBDC de uma perspectiva holística, considerando não apenas o design técnico, mas também as implicações políticas, econômicas e sociais [Sanches and Diniz 2024]. O seu projeto inclui preocupações quanto ao uso de tecnologia de contabilidade descentralizada, transações transfronteiriças eficientes e ênfase na inclusão financeira. A dinâmica da implementação de um CBDC em uma economia digital emergente como a do Brasil, deve ser observada com atenção, e pode potencializar o cenário de aplicações DeFi no contexto brasileiro.

1.2.4. Corretoras Descentralizadas

Entre janeiro e dezembro de 2023, somente na rede Ethereum, foram lançados mais de 370 mil *tokens* ERC-20. Com toda essa quantidade de *tokens* e um alto volume de capitalização de mercado torna-se indispensável o uso de plataformas onde os usuários interessados possam comprar, vender e trocar seus *tokens*. Essas plataformas permitem

que os proprietários desses ativos possam reequilibrar suas exposições de acordo com suas preferências e perfis de risco, ajustando as alocações de seus portfólios [Schär 2021].

Geralmente, as negociações de ativos descentralizados ocorrem através de corretoras centralizadas, pois em um primeiro momento é necessário trocar uma moeda fiduciária pelo ativo desejado e essa tarefa só é possível através de corretoras centralizadas ou de trocas diretas entre usuários. O grande problema das corretoras centralizadas é que os seus usuários estão expostos a: vazamento de dados pessoais, falência da corretora, ataques de terceiros e fraca regulação. É importante ressaltar que mesmo que os ativos sejam descentralizados e estejam de fato dentro da blockchain, na maioria dos casos, a posse desses ativos não está em uma conta que o usuário tem na blockchain e sim na posse da corretora. Ou seja, o único vínculo que o usuário tem com seus ativos é a sua conta padrão.

Com o objetivo de descentralizar e resolver os problemas associados às corretoras centralizadas, surgiram as corretoras descentralizadas, ou DEXes (*Decentralized Exchanges*). Os protocolos das DEXes são programados na blockchain por meio de contratos inteligentes, o que aumenta a confiabilidade entre as partes envolvidas na transação e mitiga os riscos de perda dos ativos. Dessa forma, os usuários mantêm controle total e exclusivo sobre seus ativos. Além disso, eles podem trocar seus ativos através de DApps fornecidos pela própria corretora ou utilizando contratos desenvolvidos por terceiros, já que as funções dos contratos da descentralizas podem ser acessadas por outros contratos dentro da rede.

As primeiras corretoras descentralizadas foram construídas de forma isolada e sem interação umas com as outras e entre seus protocolos. Inicialmente essas corretoras não possuíam uma liquidez compartilhada o que levava a um baixo volume de transações, grandes *spreads* entre compra e venda, altas taxas cobradas pela rede, processos complicados e lentos para mover fundos de uma uma corretoras para outra. Tudo isso tornava praticamente impossível as oportunidades de arbitragem entre os protocolos da rede.

Mais recentemente, houve uma movimentação em direção aos protocolos de corretoras abertas. Esses projetos visam simplificar a arquitetura das corretoras descentralizadas, estabelecendo padrões sobre como a troca de ativos pode ser conduzida. Eles permitem que qualquer corretoras, construída sobre esses protocolos, utilize *pools* de liquidez compartilhados e outros recursos integrados. O mais importante é que esses protocolos possibilitam que outros projetos DeFi utilizem esses *marketplaces* para trocar ou liquidar *tokens* conforme necessário. Nas próximas subseções apresentamos dois padrões populares de implementações para corretoras descentralizadas, e direcionamos os leitores interessados em outros padrões existentes para [Schär 2021].

1.2.4.1. Livro de Ordens Descentralizado

As corretoras descentralizadas com livro de ordens podem ser implementadas de várias maneiras. Todas utilizam contratos inteligentes para a liquidação das transações, mas diferem significativamente na forma como os livros de ordens são hospedados.

O livro de ordens é um registro digital que lista todas as ordens de compra e venda de um determinado ativo. Nas corretoras descentralizas que implementam esse método, o

livro de ordens é um registro onde os usuários podem cadastrar intenções de compra ou venda apresentando a quantidade de *tokens* e o valor da cotação que se deseja comprar ou vender. Quando duas ordens, de compra e venda, tem valores correspondentes, então a transação é realizada. Um comprador ou vendedor pode definir um valor de cotação, máximo ou mínimo, que deseja pagar, o que pode aumentar a chance de encontrar uma transação correspondente.

Nos protocolos das DEXes esse livro de ordens pode ser implementado dentro do protocolo, ou seja, dentro do contrato inteligente (*on-chain*), ou fora da blockchain (*off-chain*), através de uma aplicação de usuário ou de uma aplicação de terceiros.

Os livros de ordens *on-chain* têm a vantagem de serem totalmente descentralizados, pois as ordens são armazenadas dentro dos contratos inteligentes, o que elimina a necessidade de infraestrutura centralizada. No entanto, essa abordagem apresenta a desvantagem de que cada ação requer uma transação na blockchain, tornando o processo caro e lento. Mesmo a simples declaração de intenção de negociar resulta em taxas de rede. Em mercados voláteis, onde é comum o cancelamento frequente de ordens, essa desvantagem se torna ainda mais significativa.

Por essa razão, muitos protocolos de Descentralizado descentralizadas dependem de livros de ordens *off-chain*, utilizando a blockchain apenas como camada de liquidação. Os livros de ordens *off-chain* são hospedados e atualizados por terceiros centralizados. Eles fornecem aos compradores a informação necessária para selecionar uma ordem que desejam corresponder.

Um dos maiores protocolos que implementam essa abordagem é a dYdX [Juliano 2018]. A dYdX é uma DEX de negociação de derivativos que também oferece negociação de margem e empréstimos. Atualmente ela está entre as maiores corretoras descentralizadas da atualidade. Inicialmente a dYdX utilizava um sistema de livro de ordens *off-chain*, o qual era executado por eles. Porém, em 2023 a dYdX fez uma grande atualização descentralizando totalmente os seus serviços e passando a utilizar um livro de ordem *on-chain* na rede Cosmos [Product 2024].

1.2.4.2. Criadores de Mercado Automatizados

Criadores de Mercado Automatizados (AMM – *Automated Market Makers*) é um modelo de protocolo utilizado por DEXes que tem por objetivo definir a precificação de ativos digitais. O protocolo AMM se baseia em uma fórmula matemática para determinar os preços dos ativos. Alguns AMMs, inclusive, usam fórmulas simples, como é o caso do Uniswap. Provedores de liquidez são usuários que depositam ativos em *pools* de liquidez, que são espaços controlados por contratos inteligentes, e em troca recebem recompensas [Qin et al. 2021].

Qualquer ordem única de compra ou venda pode ser executada independentemente de outras negociações em AMM DEXes. Por exemplo, quando os comerciantes desejam trocar a criptomoeda A por B, eles podem invocar a função do contrato inteligente que transfere A da conta dos comerciantes para o *pool* de liquidez e envia B do *pool* de liquidez para a conta dos comerciantes. O processo de troca não envolve a participação de quaisquer

outros comerciantes. A taxa de câmbio entre A e B é determinada por funções pré definidas de forma transparente codificadas no contrato inteligente da AMM DEX [Wang et al. 2022].

Como as operações de mercado em AMM DEXes são invocadas por transações em blockchain, os usuários são obrigados a pagar uma taxa de transação aos mineradores. Especificamente, no Ethereum cada transação custa uma quantidade predeterminada de "gás", taxa de efetivação de uma transação na rede. O emissor da transação específica quanto está disposto a pagar por unidade de gás (ou seja, o preço do gás). A taxa de transação paga aos mineradores corresponde ao produto do consumo total de gás e do preço do gás [Wang et al. 2022].

A Figura 1.3 apresenta um modelo de produto constante que é aplicado em *pools* de liquidez. Este modelo pode ser expresso como $xy = k$, onde x e y correspondem as reservas dos *tokens* armazenados dentro da *pool*, e k uma constante. Porém, quando um usuário da rede realiza um troca de *tokens* utilizando esta *pool* os valores de x e y serão modificados, então obteremos $(x+\Delta x).(y+\Delta y) = k$. Dessa forma, podemos assumir que $\Delta y = (k/(x+\Delta x)) - y$. Ou seja, se um usuário realiza um *swap* de *tokens* x para y nesta *pool* serão depositados no contrato os *tokens* x e os *tokens* y equivalentes serão enviados para o usuário. Isso faz com que Δy assuma valores negativos pois alguns *tokens* y foram retirados do *pool*, e Δx assume valores positivos [Schär 2021].

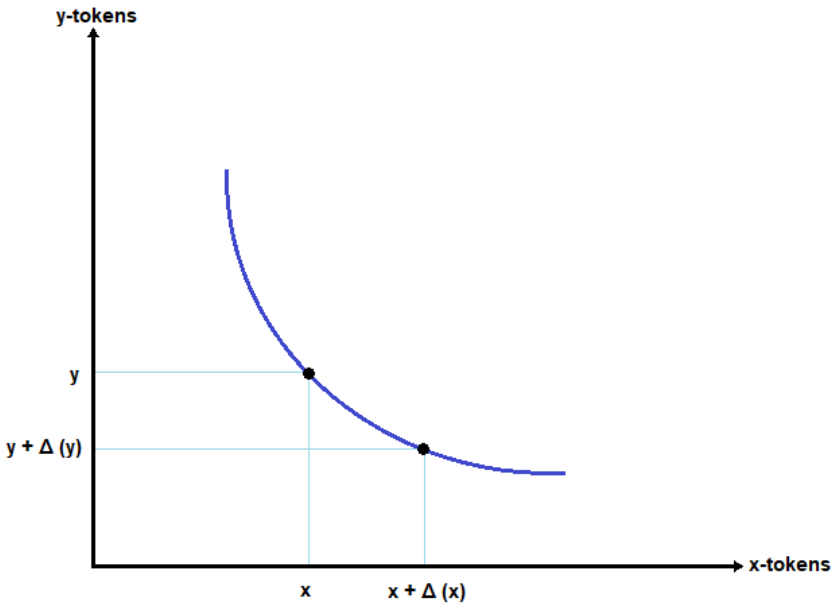

Figura 1.3. *Pool* de liquidez em um modelo de produto constante. Figura adaptada de [Schär 2021].

Toda troca realizada pelo *pool* resultará em um movimento em uma curva convexa, como mostra a Figura 1.3. Um *pool* de liquidez de produto constante modifica os valores dos *tokens* com base no movimento da curva fazendo com que o valor do *token* aumente infinitamente conforme a quantidade desse token se aproxima de zero, e, desta forma, que o *pool* jamais seja esgotado.

É importante ressaltar que os valores dos *tokens* são dados, somente, com base na função aplicada aquela *pool*. Isso significa que o valor de um *token* dentro de uma *pool* pode ser diferente do seu valor de mercado. Apesar dos valores dos *tokens* serem calculados de forma exponencial, dentro e fora do *pool* eles estarão sempre próximos. Quando o valor de mercado do *token* muda, os arbitradores poderão aproveitar a oportunidade para trocar os *tokens* com o contrato inteligente do *pool* até que o valor praticado dentro do *pool* de liquidez esteja igualado com o valor de mercado. Este processo garante que o valor do *token* dentro do *pool* esteja sempre próximo ou igualado com o valor de mercado.

1.2.5. Perspectivas em Aplicações

Nesta seção discutimos brevemente tópicos de pesquisa e inovação promissores em aplicações para finanças descentralizadas, reservando a discussão sobre interoperabilidade e segurança para seções seguintes.

Tokens não fungíveis (NFTs) tem sido alvo de pesquisas científicas, em especial, após o pico de comercialização superior a USD 2 bilhões desses tokens em 2021 e sua adoção pela indústria de mídia e artes digitais desde então. Um trabalho seminal conduzido por [Nadini et al. 2021] coletou dados de 4,7 milhões de NFTs e construiu a rede de interações entre usuários de NFTs visando identificar comunidades. Esse trabalho abriu caminho para pesquisas sobre aquisição de conhecimento aplicado ao mercado de NFTs via análises de grafos e aprendizagem de máquina. Exemplos proeminentes recentes são métricas de redes complexas para extrair comunidades de compradores e vendedores por categorias de NFT [White et al. 2022], e modelos de predição de preços baseado em processamento de imagem e texto com redes neurais profundas em [Costa et al. 2023].

Outro tópico de pesquisa que demanda contribuições da comunidade atualmente concerne a governança de DApps em especial DeFi. Em [Messias et al. 2023], a governança descentralizada é analisada via medições nos DApps Compound e Uniswap, que são duas DeFis populares em propor protocolos de votação para adoção de atualizações em seus códigos. Os autores revelam uma elevada concentração do poder de voto nesses DApps. Exemplos de esforços de pesquisa que fundamentam trabalhos nesse tópico incluindo contratos sociais e organizações autônomas descentralizadas (DAOs) são as análises sobre compromissos entre descentralização e desempenho proposto em [Chen et al. 2021], o estudo abrangente da teoria de governança e a reconceitualização do termo governança adaptada aos DAOs [Zwitter and Hazenberg 2020], e a taxonomia sobre o que constituem os DAOs e suas principais características proposta em [Hassan and De Filippi 2021]

Observando o cenário brasileiro, o desenvolvimento e implementação do DREX representam uma oportunidade significativa para o Brasil modernizar seu sistema financeiro, aumentar a inclusão financeira e digital, e fortalecer sua soberania econômica. No entanto, é fundamental abordar cuidadosamente os desafios técnicos para garantir que os benefícios do DREX sejam plenamente realizados e que os riscos sejam mitigados. A sua implementação pode impulsionar o cenário de finanças descentralizadas (DeFi) no Brasil, oferecendo novas oportunidades para inovação financeira, como empréstimos peer-to-peer, investimentos descentralizados e seguros automatizados. As competências desenvolvidas neste ecossistema podem ser transpostas ao cenário mundial, considerando a experiência digital avançada do sistema financeiro brasileiro, e adaptando-se às peculiaridades que

serão observadas em cada caso de implementação de novas CBDCs de acordo com as diferentes estratégias de cada Banco Central [Teixeira 2023].

1.3. Interoperabilidade

As redes blockchain permitem a execução de diferentes conjuntos de transações por meio de implementações distintas e normalmente isoladas entre si. Esta configuração acaba gerando sistemas heterogêneos, onde há grande dificuldade para troca de informações entre as redes, em especial aplicações DeFi e seus respectivos *tokens* que não podem funcionar em redes blockchain diferentes das quais foram inicialmente desenvolvidas e implantadas. Essas questões trouxeram em evidência a necessidade de interoperabilidade entre redes. Assim, vem se observando recentemente um esforço de várias organizações mantenedoras dessas redes em desenvolver soluções que ampliem as suas capacidades de cooperação, ainda que essas redes utilizem diferentes tecnologias, o que consiste na definição ampla do termo interoperabilidade [Wegner 1996].

Os tipos de interoperabilidade em blockchain mais comuns são: interoperabilidade entre redes blockchain (homogêneas), interoperabilidade entre dApps usando diferentes redes blockchain e interoperabilidade de redes blockchain com outras tecnologias de blockchain (heterogêneas) [Besançon et al. 2019]. Assim, a Figura 1.4 demonstra um diagrama de fragmentação desses tipos e apresenta como são definidas as transações entre os tipos. As transações de redes blockchain (homogêneas), são nomeadas como uma transação *cross-chain* (CC-Tx), onde "CC" significa *cross-chain* e "Tx" transação. Uma transação *cross-blockchain* (CB-Tx) é uma transação entre diferentes redes blockchain (heterogêneas) e, por fim, uma aplicação descentralizada *cross-chain* (CC-dApp) é um dApp que utiliza as transações *cross-blockchain* para implementar seus requisitos [Belchior et al. 2021].

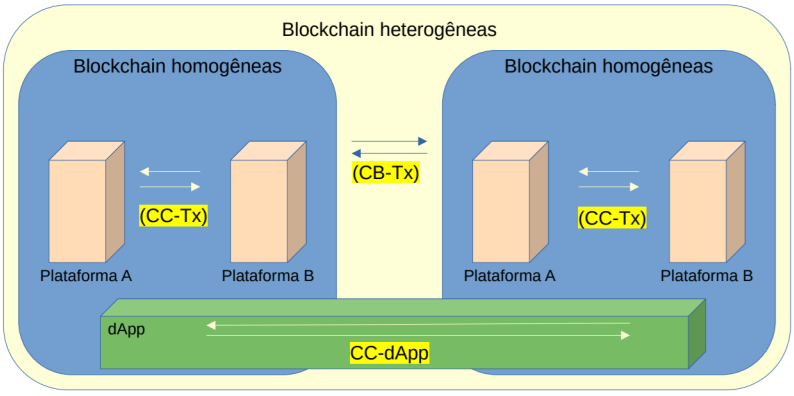

Figura 1.4. Tipos de interoperabilidade e transações. [Mendonça et al. 2024]

A transação *cross-chain* é o tipo de interoperabilidade mais utilizado atualmente. Ela envolve um par de redes blockchain em que um tipo de aplicação descentralizada facilita a transferência de ativos de uma blockchain para outra, provendo a interoperabilidade. Isso significa que é possível mover ativos, como criptomoedas e *tokens*, de uma blockchain para outra sem a necessidade de intermediários centralizados. Uma das maneiras de implementar *cross-chain* envolve a criação de pontos de conexão entre as redes blockchain,

conhecidos como “pontes". As pontes permitem que as transações sejam validadas em ambas as redes blockchain, garantindo a segurança e a integridade dos ativos transferidos. De maneira geral, as transferências de ativos entre cadeias seguem um procedimento atômico, baseado no protocolo *Cross-chain communication protocol (CCCP)* em que há o bloqueio de um ativo na origem, responsabilidade de transferência e a criação do ativo no destino [Belchior et al. 2021].

No contexto dos *tokens*, a interoperabilidade entre plataformas pode contribuir para a usabilidade, ao implicar a capacidade de transferir um ativo entre cadeias distintas, mantendo o estado e histórico consistentes. A interoperabilidade de cadeias deve atingir a eficiência de dois tipos, cada um dos quais trazendo considerações distintas porém contribuindo para a usabilidade. A troca de ativos digitais entre cadeias é um dos tipos de interoperabilidade. Ele deveria conter a capacidade de transferir e trocar ativos originários de diferentes cadeias sem intermediários confiáveis, como trocas centralizadas. Um exemplo disso seria tornar um *token*, originário de uma cadeia, válido em qualquer outra cadeia disponível. Outro tipo de interoperabilidade desejada se diz respeito à troca de informações que mantém a capacidade de fazer algo em uma cadeia que reflete em outra cadeia. Esta troca deve permitir o rastreamento não só de ativos ou itens negociáveis, mas também as operações executadas. Como exemplo, o compartilhamento do histórico de transações de um determinado item contendo negociações e proprietários.

1.3.1. Mecanismos de Interoperabilidade

Com tantas oportunidades de aplicativos de negócios com requisitos em interoperar redes blockchain, prover soluções com mecanismos genéricos de *cross-chain* para conectar redes blockchain homogêneas e heterogêneas, amplia o espaço de desenvolvimento desta tecnologia [Buterin 2016]. Algumas soluções foram propostas para interoperar acesso a dados e transações entre redes blockchain, como as *sidechains*, o mecanismo notarial (*Notary mechanism*) e bloqueio de hash (*Hash-locking* ou *Hash time lock*) [Belchior et al. 2021]. Estes mecanismos propõem soluções que podem abranger um número maior de variedades de aplicações e distintas soluções de blockchain.

1.3.1.1. Sidechain

Uma *sidechain* é uma blockchain independente que opera em paralelo à blockchain principal (ou *mainchain*) e seu conceito foi baseado para permitir a transferência segura de ativos e informações entre ambas. Esse mecanismo de funcionamento é essencial para a interoperabilidade de diferentes redes blockchain, pois amplia a funcionalidade da *mainchain* sem sobrecarregar as limitações de escalabilidade da rede principal, transferindo certas operações e transações para a cadeia secundária.

O conceito de *sidechain* [Back et al. 2014] foi introduzido como uma solução para diversos problemas de escalabilidade, flexibilidade e experimentação de uma *sidechain*, é possível implementar novas funcionalidades e realizar atualizações sem a necessidade de divisões na blockchain principal, o que contribui para sua estabilidade e segurança.

A figura 1.5 ilustra o conceito de *sidechain* com um exemplo de uma blockchain hipotética. A *mainchain* é a cadeia de blocos original onde as transações são registradas

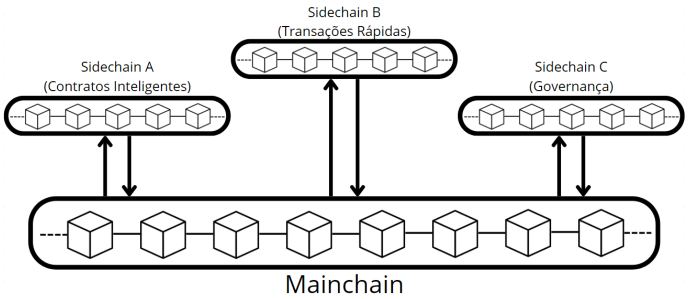

Figura 1.5. Exemplo de sidechains

e a integridade e segurança dos dados são asseguradas. As setas indicam que há uma relação entre a blockchain principal e *sidechains*, demonstrando que os ativos podem ser transferidos entre elas. *Sidechains* são cadeias de blocos adicionais que estão conectadas à blockchain principal. Cada *sidechain* pode ter características e funcionalidades específicas que não estão presentes na cadeia principal. No exemplo ilustrado, a *sidechain* A é projetada para suportar contratos inteligentes. A *sidechain* B é otimizada para transações rápidas, ou seja, podendo lidar com um volume maior de transações por segundo do que a cadeia principal. Por fim, a *sidechain* C foca na governança, ou seja, em como são tomadas decisões sobre a evolução e as regras do sistema blockchain.

Uma das principais características das *sidechains* é a capacidade de transferir ativos de forma segura entre a *mainchain* e a *sidechain*. Isso geralmente é feito através de um mecanismo conhecido como *two-way peg* (ponte bidirecional), que permite a transferência de *tokens* entre as duas cadeias [Singh et al. 2020]. Inicialmente, um usuário envia uma quantidade de *tokens* para um endereço específico (chamado de *lock-box*) na *mainchain*, bloqueando os *tokens* nessa cadeia. Em seguida, a *sidechain* recebe a transação de bloqueio que foi confirmada na *mainchain* e emite réplicas dos *tokens* originais. O usuário pode então utilizar esses *tokens* representados para pagar por serviços ou realizar transferências na *sidechain*. Eventualmente, o usuário pode retirar seus *tokens* da *sidechain* para a *mainchain*, exigindo que os *tokens* representados sejam bloqueados ou queimados na *sidechain*, dependendo da implementação.

Atualmente, existem três principais mecanismos para implementar um *two-way peg*: *two-way peg* centralizado, *two-way peg* federado e verificação simplificada de pagamento (SPV) [Ren et al. 2023]. No *two-way peg* centralizado [Singh et al. 2020], a implementação é realizada por uma terceira parte confiável que fica responsável por garantir o bloqueio e desbloqueio de *tokens* tanto na *mainchain* quanto na *sidechain*. Embora seja fácil de implementar, este esquema centralizado contraria a característica de descentralização de redes blockchain públicas e pode tornar um ponto único de falha.

O *two-way peg* federado [Singh et al. 2020] melhora o método anterior ao adicionar um grupo de partes ditas notários (do inglês *notaries*) para processar as operações de bloqueio e desbloqueio de *tokens*, utilizando esquemas de multi assinaturas. Desse modo, a operação de bloquear ou desbloquear *tokens* da *mainchain* só ocorre se a maioria das partes do grupo concordar com a transação. Embora essa abordagem possa diminuir a centralização, ainda pode apresentar riscos se a maioria dos participantes tiver intenções

maliciosas ou o grupo de assinantes for pequeno.

A verificação simplificada de pagamento (SPV) é outro mecanismo utilizado para implementar a interoperabilidade entre *sidechains* e a *mainchain*. Inicialmente descrito por Satoshi Nakamoto em [Nakamoto and Bitcoin 2008], esse mecanismo permite que nós verifiquem transações sem precisar baixar toda a blockchain. Em um contexto de *sidechains*, a SPV permite que a *sidechain* verifique transações na *mainchain* de forma eficiente, sem a necessidade de sincronizar todos os dados da *mainchain*.

Sidechains operam com seus próprios mecanismos de consenso, regras e sistemas de governança, ou seja, são responsáveis pela sua segurança e não herdam tais propriedades da *mainchain*. Isso possibilita o desenvolvimento e a experimentação de novos protocolos e aplicações descentralizadas sem interferir na operação da blockchain principal. Uma *sidechain* pode ser configurada para suportar contratos inteligentes mais complexos, transações mais rápidas ou diferentes algoritmos de consenso. Por exemplo, a RSK [Lerner et al. 2022] provê uma *sidechain* com suporte a contratos inteligentes para interoperar com a *mainchain* da Bitcoin.

Em resumo, as *sidechains* proporcionam uma maneira de escalar redes, experimentar novas funcionalidades e melhorar a interoperabilidade entre diferentes redes blockchain. Ao permitir a coexistência de múltiplas cadeias especializadas, as *sidechains* ampliam o potencial das aplicações em blockchain, tornando-as mais versáteis e adaptáveis às necessidades específicas dos usuários e desenvolvedores.

1.3.1.2. Mecanismo de Bloqueio de *Hash*

O mecanismo de bloqueio de *hash* ou *hash time-lock contract* (HTLC) representa um marco significativo na evolução dos mecanismos de troca entre redes blockchain, proporcionando uma solução inovadora para realizar transações entre redes sem depender de intermediários confiáveis [Ou et al. 2022]. Ao implementar contratos HTLC nas redes blockchain envolvidas na negociação, o processo de troca de ativos é seguro e confiável. Esse contrato atua como uma garantia, bloqueando os ativos envolvidos até que as condições acordadas sejam atendidas. A utilização do conceito de *hash* adiciona uma camada adicional de segurança, criando uma trava com uma palavra secreta que deve ser correspondente em ambas as extremidades da transação. Além disso, a imposição de um limite de tempo para a conclusão da troca aumenta a eficiência e a segurança do processo. Em caso de não cumprimento dentro do prazo estipulado, o contrato automaticamente cancela a transação, revertendo os *tokens* para suas respectivas carteiras de origem. Esse mecanismo desempenha um papel fundamental na facilitação de trocas descentralizadas, promovendo a confiança e a segurança nas transações entre redes blockchain [Belchior et al. 2021].

A arquitetura do mecanismo de bloqueio de *hash* exige a implementação de contratos inteligentes. Neste caso, o contrato é responsável pela troca segura dos ativos, ou seja, ele é implantado nas duas redes e possui a tarefa de conectá-las. Contudo, não há um terceiro confiável. O contrato inteligente atua sincronizando as redes no que diz respeito à verificação das transações, da palavra secreta e a devolução dos valores, caso necessário. Sendo assim, o contrato HTLC possui as funcionalidades de bloquear os fundos que serão transferidos, registrar o horário da transação e exigir uma palavra secreta no momento

da retirada dos fundos para o destinatário da segunda rede. Caso o tempo tenha ultrapassado o período de bloqueio, o contrato é capaz de reverter todo o processo, conforme a definição do mecanismo. Apesar disso, questões referentes à segurança do mecanismo também devem ser consideradas. O comprometimento do contrato inteligente que realiza a sincronização pode levar ao comportamento desordenado das chamadas de retirada e bloqueio dos fundos, causando possíveis perdas ou resultados inesperados.

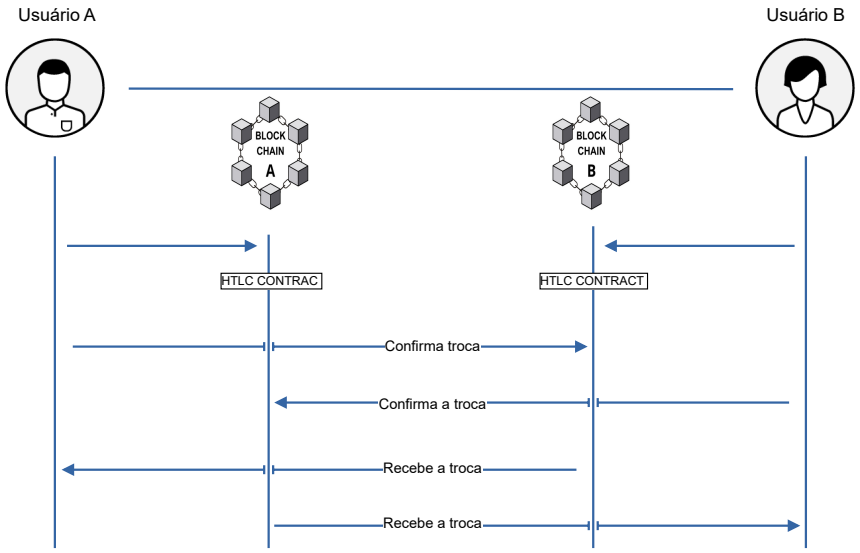

Figura 1.6. Arquitetura implementada do Mecanismo *Hash-Time Lock*.

No contexto do mecanismo de bloqueio de *hash*, conforme a Figura 1.6, o *Usuário A* deseja fazer a transferir seu *token* para uma conta do *Usuário B* em outra rede. Para fazer essa transferência, ele escolhe uma "palavra secreta", e utiliza o *Hash* juntamente com o endereço de B para criar o contrato HTLC. Por meio do contrato criado na *Blockchain A*, ele bloqueia o *Token A* para a transferência ser realizada. De maneira similar, o *Usuário B* implanta o HTLC na *Blockchain B* com o endereço de A e a palavra secreta em *Hash*. Sendo assim, o *Usuário A* faz a retirada (*withdraw*) do *Token B* na *Blockchain B* com sua palavra secreta. Ao fazer isso, a palavra secreta pode ser usada pelo *Usuário B* para retirar o *Token A* da *Blockchain A*.

1.3.1.3. Mecanismo Notarial

O mecanismo notarial é uma forma de implementar a interoperabilidade entre cadeias. Ele consiste em verificar e encaminhar mensagens entre cadeias por meio de uma entidade confiável intermediária chamada de notário. Quando há troca e transferência de ativos entre diferentes redes blockchain, uma ou mais organizações são designadas como notários para monitorar eventos entre as cadeias, e alcançar um consenso sobre a ocorrência do evento por meio de um algoritmo de consenso específico, e, por fim, responder de forma tempestiva [Belchior et al. 2021]. O mecanismo notarial se divide em mecanismo notarial de assinatura única e de múltiplas assinaturas [Ou et al. 2022].

O mecanismo notarial de assinatura única, também denominado mecanismo notarial centralizado, consiste em designar um único nó ou instituição independente para atuar como notário, e o notário assume as tarefas de coleta de dados, verificação e confirmação de transações no processo de interação entre cadeias. O notário é composto por, pelo menos, uma conta nas cadeias de origem e de destino. Este mecanismo consegue ter um processamento rápido de transações e é bastante adaptável, apesar do escopo restrito, limitando-se a troca de ativos.

No mecanismo notarial de múltiplas assinaturas o notário é geralmente composto por vários nós, onde cada nó possui uma chave e somente quando uma determinada porcentagem destes nós assinam em conjunto é que há um consenso e as transações entre cadeias podem ser confirmadas. Durante a verificação da transação, uma parte dos notários é selecionada aleatoriamente do grupo notarial, diminuindo o grau de dependência da confiabilidade dos notários.

Na arquitetura deste mecanismo, os usuários envolvidos na transferência de *tokens* devem interagir com o notário. Essa interação pode ocorrer por meio de *dApps* (aplicativos descentralizados executados em blockchain) ou contratos inteligentes, com o domínio de um terceiro confiável. O notário desempenha o papel de receptor do *token* do usuário A (remetente) na *blockchain* A, transferindo-o para o usuário B (destinatário) na *blockchain* B e registrando informações sobre as transações realizadas. O notário deve garantir a entrega segura dos recursos ao destinatário designado.

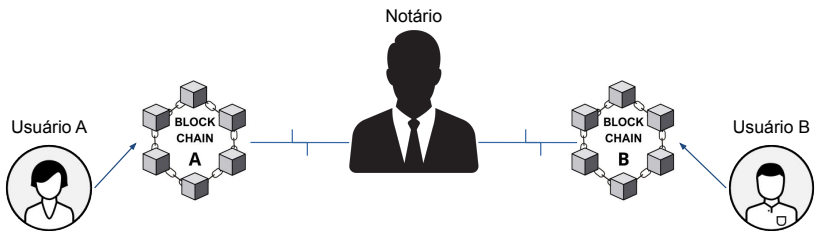

Figura 1.7. Arquitetura implementada para o Mecanismo Notarial.

Vale ressaltar que, embora o mecanismo seja eficaz, ele ainda possui limitações. Em particular a segurança, que é de suma importância em qualquer aplicação blockchain. No caso desta arquitetura em específico, o mecanismo notarial atua de maneira centralizada, sendo assim a segurança da transação depende da integridade do notário, visto que ele é o responsável por receber os fundos na blockchain de origem e transferi-los para a blockchain de destino. Se o notário for comprometido de alguma forma, isso pode acarretar em perdas financeiras para os usuários das redes. Porém, uma vez que o notário seja exaustivamente testado e reconhecido como confiável o mecanismo se torna extremamente eficiente.

Atualmente, algumas organizações utilizam mecanismos notariais, como o proto-

colo CCIP da Chainlink, para garantir a interoperabilidade entre redes blockchain. No entanto, diversos projetos que inicialmente adotaram essa abordagem agora estão explorando alternativas. Por exemplo, o Interledger, criado pela Ripple, foi projetado para facilitar pagamentos entre diferentes redes blockchain utilizando o Protocolo Interledger (ILP), um conjunto aberto de protocolos que permite transações de forma atômica e universal [Thomas and Schwartz 2015].

Originalmente, o ILPv1 visava proporcionar uma maneira flexível de realizar pagamentos entre diversas redes blockchain. Ele implementava transações atômicas através de dois modos: *hash-locks* e quórum de notários, conhecido como Protocolo de Transporte Atômico. Os *hash-locks*, baseados em contratos HTLC, são bloqueios criptográficos que podem ser desbloqueados ao revelar um segredo *s* cujo resultado da função de *hash H(s)* corresponde ao valor configurado no bloqueio [Siris et al. 2019]. No modo atômico, além dos *hash-locks*, as transações são coordenadas por meio de um grupo AD-HOC de notários selecionados pelos participantes para verificá-las e validá-las.

Por outro lado, o modo universal do ILP permitia transações entre conectores não confiáveis, dispensando notários. Enquanto o modo atômico utiliza notários para garantir a execução adequada de um pagamento, o modo universal depende dos incentivos de participantes racionais para eliminar a necessidade de coordenação externa. Este modo fornece segurança para todos os participantes não defeituosos conectados, sob a suposição de sincronia limitada com um limite conhecido. Em vez de notários, utilizava o XRP, a moeda nativa do Ripple, para facilitar essas transações [Thomas and Schwartz 2015].

Com a evolução para o ILPv4, houve uma mudança significativa na abordagem do protocolo devido à preocupações de segurança associadas aos longos tempos de espera nas transações de garantia. O protocolo ILPv1 sofria com deficiências, como o problema da opção gratuita, onde remetentes e destinatários poderiam manipular as taxas de câmbio, e ataques de negação de serviço que vinculavam fundos dos conectores intermediários. Para mitigar esses riscos, o ILPv4 foi redesenhado para usar transferências rápidas de pacotes de baixo valor [Siris et al. 2019]. Esse novo modelo elimina a necessidade de notários e compromissos de alto valor, permitindo que os conectores estabeleçam relações de confiança bilaterais, de forma que as as *sidechains* podem ser usadas como um sistema de liquidação entre duas entidades em interação.

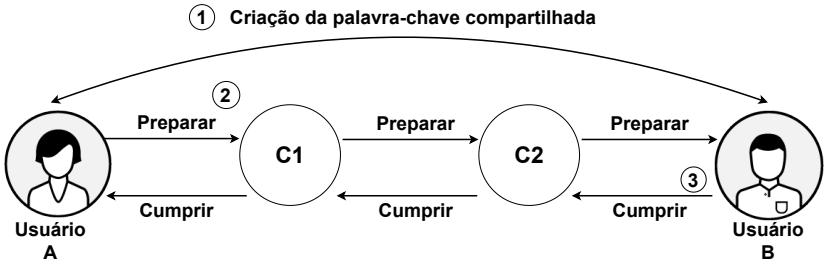

Figura 1.8. Representação de pagamento utilizando o ILPv4.

Um pagamento ILPv4 entre o usuário A (remetente) e o usuário B (destinatário) usando dois conectores prossegue conforme visualizado na Figura 1.8, essa transferência de valor ocorre em duas fases: preparar e cumprir. Inicialmente, o remetente gera um pacote de preparação contendo o valor e o *hash* de uma palavra-chave conhecida apenas pelo remetente e pelo destinatário. Este pacote é encaminhado ao receptor através de conectores. Cada conector, ao encaminhar o pacote, compromete-se a pagar ao próximo conector somente se este fornecer prova de pagamento ao conector subsequente. Quando o receptor recebe o pacote de preparação, ele gera um pacote de cumprimento revelando a palavra-chave e o envia de volta pelo mesmo caminho, começando pelo último conector. Cada conector valida a palavra-chave, efetua o pagamento e encaminha o pacote de cumprimento ao conector anterior. Esse processo permite que os pacotes de preparação e cumprimento se propaguem rapidamente, evitando os atrasos significativos associados aos métodos de pagamento baseados em garantia do ILPv1 [Siris et al. 2019].

1.3.2. Soluções para Interoperabilidade

Diversas soluções têm sido implementadas para superar as barreiras da fragmentação, promovendo um ecossistema mais integrado e funcional no que diz respeito à interoperabilidade de redes blockchain. Estas soluções variam desde protocolos de comunicação padronizados até plataformas de troca e *bridges* que facilitam a interoperabilidade entre diferentes redes blockchain. Nesta subseção, examinaremos duas das principais abordagens implementadas para alcançar a interoperabilidade em blockchain: Cosmos e Chainlink.

1.3.2.1. Cosmos

O Cosmos pode ser definido como a "internet das redes blockchain". Essa estrutura descentralizada concentra-se principalmente na interoperabilidade e escalabilidade, oferecendo maior flexibilidade tanto para desenvolvedores quanto para usuários. O Cosmos Hub é a principal blockchain da rede, ele atua como ponto central de coordenação e segurança. No Cosmos Hub, todas as atividades e transações são registradas, e a criptomoeda nativa, ATOM, é hospedada. O *Inter-Blockchain Communication Protocol* (IBC) permite que várias redes blockchain se conectem ao Cosmos Hub, possibilitando a troca segura de informações entre estas redes blockchain. Estas redes blockchain conectadas são chamadas de Zonas, cada uma delas sendo uma blockchain individual com suas próprias funcionalidades. Com a conexão estabelecida, as Zonas podem interoperar entre si. Isso significa que dados podem ser trocados entre redes blockchain com diferentes mecanismos de consenso e verificação. A representação na Figura 1.9 ilustra essa interconexão.

A rede Cosmos, faz uso do mecanismo de interoperabilidade *sidechain*, proposto para dimensionar redes blockchain alternativas que são "atreladas bidirecionalmente". A indexação bidirecional é equivalente a uma ponte. Assim, o Cosmos Hub atua como clientes leves um do outro, utilizando provas SPV para determinar quando as moedas devem ser transferidas para a *sidechain* e vice-versa [Kwon and Buchman 2019]. Além de sua função como rede, o Cosmos é um projeto de código aberto. Ele oferece aos desenvolvedores ferramentas e estruturas para construir suas próprias redes blockchain específicas, as chamadas Zonas. Essas Zonas podem ser profundamente customizadas para atender às necessidades de diferentes usuários. Nesse sentido, para o funcionamento do

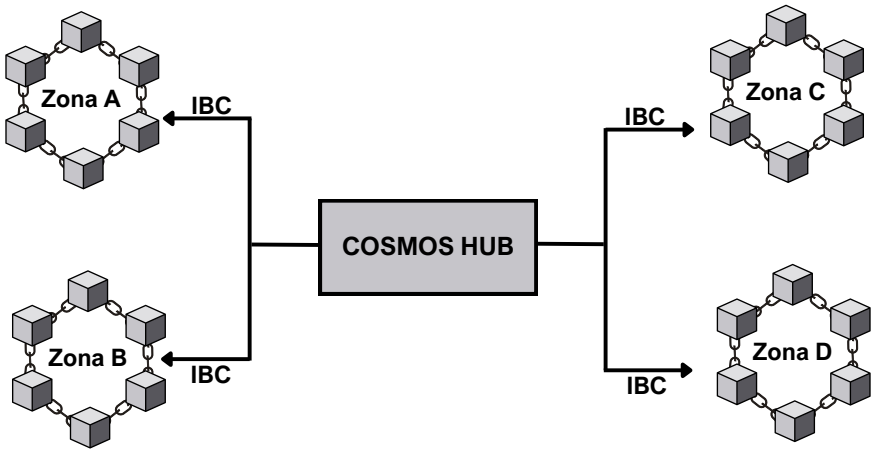

Figura 1.9. Representação do cosmos.

Cosmos há três componentes fundamentais: *Tendermint*, Cosmos SDK e o *Inter-Blockchain Communication Protocol* (IBC).

O *Tendermint* é um conjunto de ferramentas de desenvolvimento que tem como objetivo simplificar o desenvolvimento de camadas de rede e consenso através do *Tendermint Core* (motor de consenso) e a *Application Blockchain Interface* (ABCI) (interface de aplicação genérica).

- **Tendermint Core:** Garante que as mesmas transações sejam registradas em todas as máquinas na mesma ordem e tolera até um terço de suas máquinas falharem arbitrariamente. Isto inclui comportamento explicitamente malicioso.
- **Application BlockChain Interface (ABCI):** Permite que as transações sejam processadas em qualquer linguagem de programação.

O Cosmos SDK é um kit de desenvolvimento de software de código-aberto disponibilizado que permite a criação de redes blockchain e com compatibilidade na rede cosmos. O protocolo de consenso padrão é o *Tendermint Core*, mas há grande variedade de módulos integrados disponíveis. Ele simplifica consideravelmente o processo e oferece todos os padrões para criação de uma blockchain. Atualmente, há alguns projetos conhecidos que foram desenvolvidos com o Cosmos SDK, entre eles a Binance Smart Chain (BSC), KAVA, Celestia, Band Protocol e Terra.

O *Inter-Blockchain Communication Protocol* (IBC) é o protocolo responsável por estabelecer a comunicação *interblockchain*. Ele é responsável por lidar com a autenticação e permite que diferentes redes blockchain transfiram dados e valores entre si diretamente, sem depender de intermediários. O IBC, por meio de um nó de processamento de transações, executa a hospedagem de fundos, envia dados de bloqueio de ativos para outra cadeia, inicia a proposta e desbloqueia o ativo especificado para a cadeia de destino. Além disso, o modo de funcionamento do IBC é semelhante à comunicação TCP/IP utilizada para troca de dados entre a Internet e outras redes. É composto por duas camadas: a camada de transporte (TAO) e a camada de aplicação (APP) [Goes 2020].

A camada de transporte fornece a infraestrutura chave para conectar-se com segurança a outras cadeias e é a base para a construção de outras aplicações. Ela é responsável pelas etapas de autenticação, transporte e ordenação de pacotes de dados. A camada de transporte consiste em canais, clientes leves, conexões e retransmissores. Por outro lado, a camada de aplicação é construída na camada de transporte e determina como os pacotes de dados são empacotados, interpretados e utilizados pelas cadeias de envio e recebimento. Através da interface da camada de aplicação, os usuários finais podem interagir com a *Interchain*, interagindo com NFTs, *tokens* ou outras aplicações que utilizam a camada de transporte.

1.3.2.2. Chainlink

O *Cross-Chain Interoperability Protocol* (CCIP) da Chainlink surge como um protocolo inovador que revoluciona a interoperabilidade entre redes blockchain. Através de uma rede descentralizada de oráculos e contratos inteligentes, o CCIP facilita a transferência de *tokens* e dados entre diferentes redes blockchain, permitindo um ecossistema blockchain interconectado [Breidenbach et al. 2022]. Diferente de *bridges* tradicionais centralizadas, o CCIP interopera de dados e *tokens* descentralizados usando o mecanismo notarial. O protocolo da Chainlink utiliza de três redes descentralizadas e roteadores para fazer a ponte entre redes blockchain, como mostra a Figura 1.10. Para que a transação via CCIP seja efetivada, é necessário confiar nas redes descentralizadas e roteadores, visto que elas serão o notário da transação.

Para que uma transação de envio de dados ou *tokens* seja transportada de uma blockchain para a outra, é necessário que o contrato inteligente da blockchain de origem envie uma transação para o roteador. O contrato que atua como roteador da rede origem recebe como transação uma mensagem codificada com as instruções sobre a blockchain de destino, qual é o destinatário, alguns *tokens* fundíveis como taxa pelo serviço (*feeToken*) e os dados ou *tokens* que serão transferidos. Dessa maneira, o roteador de origem envia a transação para a rede de *commiting* que, por conseguinte, envia para a blockchain de destino, na qual a transação fica armazenada em uma *commit store*. Assim, todo o mecanismo aguarda pela aprovação da *risk management network*, que verifica nas redes de origem e destino se as informações entre elas são compatíveis e não houve alteração. Após a verificação, uma terceira rede, a *executing* efetiva a transação para que o roteador de destino receba a mensagem e execute a instrução no contrato destino.

Apesar de ser um protocolo descentralizado, o CCIP possui a necessidade de confiar em redes blockchain, que estão passíveis de falhas. Além disso, a compatibilidade de redes é um fator importante para o funcionamento do protocolo, ou seja, é necessário que exista um contrato roteador na rede de origem e na rede de destino. Além disso, os *tokens* interoperáveis entre as duas redes podem ser incompatíveis com o protocolo, necessitando a utilização de outras *bridges*.

1.3.3. Perspectivas em Interoperabilidade

A interoperabilidade de redes blockchain, embora tenha avançado significativamente, ainda enfrenta diversos desafios devido à infraestrutura heterogênea dessas redes. Um dos

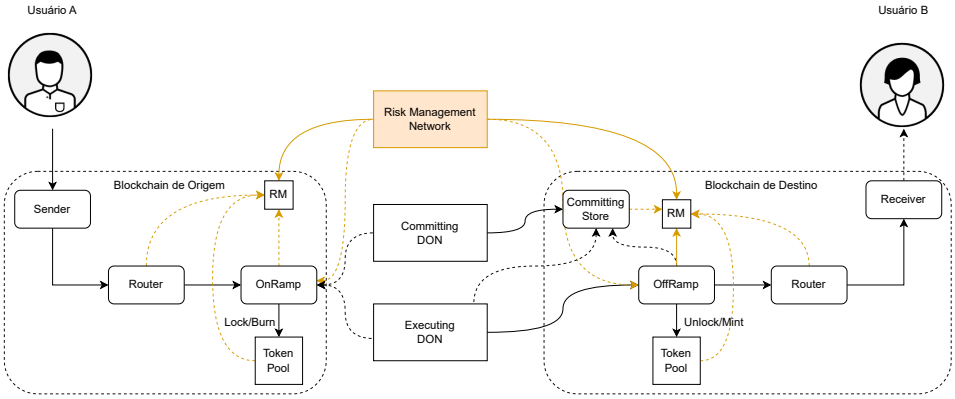

Figura 1.10. Representação do protocolo CCIP.

principais problemas é a integração entre uma blockchain permissionada (privada) e uma não permissionada (pública) [Helliar et al. 2020]. Uma blockchain permissionada exige autenticação e permissões de acesso, dificultando a confiança e a verificação de transações por clientes de uma blockchain não permissionada.

Além disso, uma blockchain permissionada geralmente utiliza protocolos de consensos diferentes, como o RAFT [Ongaro and Ousterhout 2014] (um protocolo CFT – *Crash Fault Tolerance*), ou o BFT-SMART [Bessani et al. 2014] (um protocolo BFT – *Byzantine Fault Tolerance* baseado em quórums), ao invés de PoW e PoS (protocolos BFT com abordagens probabilísticas), comuns nas redes blockchain públicas. Por exemplo, o uso de CFT assume que todos os nós são confiáveis e apenas falham por colapso. Abordagens baseadas em BFT são mais custosas por tolerar nós que podem agir de maneira maliciosa ou arbitrária, e abordagens baseadas em quórum tendem a favorecer consistência e disponibilidade, um par diverso de abordagens probabilísticas quando consideramos a tríade do teorema CAP [Gilbert and Lynch 2002]. Esta diferença de consensos envolve um *trade-off* entre eficiência e robustez.

Outro desafio é a interoperabilidade com redes blockchain que utilizam linguagens de *script* limitadas, como o Bitcoin, que não suporta contratos inteligentes Turing-completos. Isso contrasta com redes blockchain como a Ethereum [Tikhomirov 2018], que permite cálculos complexos necessários para protocolos de cadeia cruzada, dificultando a conexão entre essas redes. Além disso, esquemas de assinatura digital e algoritmos de *hashing* entre diferentes redes blockchain tornam a verificação e implementação de transações cruzadas complexas, já que cada blockchain pode usar métodos distintos para essas operações fundamentais.

1.4. Segurança

Nesta seção serão apresentados os aspectos principais relacionados à segurança no contexto de contratos inteligentes e consequentemente aplicações DeFi. De fato, nos últimos anos, diferentes estudos tratam das pesquisas relacionadas à segurança nos ambientes voltados aos contratos inteligentes [Rouhani and Deters 2019, Harz and Knottenbelt 2018], discutindo a importância e a constante evolução desses. O conceito de segurança em

uma rede blockchain está diretamente relacionado aos métodos de ataque e também aos mecanismos de defesa a esses ataques [Sayeed et al. 2020].

Nesse sentido, tratamos nas seções seguintes, primeiramente, do tópico auditoria de contratos inteligentes, que aborda mecanismos de defesa no aspecto estático e dinâmico do código fonte desses programas. A seguir, focamos no comportamento dos usuários do sistema e as ameaças ocasionadas a partir desse, em especial, ataques de *front-running*, manipulações de oráculos, *flash-loans* e ataque de liquidez, que são algumas das ameaças mais recentes em redes blockchain públicas. Por conseguinte, discutimos na seção final as perspectivas em segurança explorando as pesquisas estado-da-arte que abordam ambos os aspectos estáticos e dinâmicos em segurança de contratos inteligentes e do ecossistema DeFi.

1.4.1. Auditoria de Contratos Inteligentes

Conforme discutido na Seção 1.2, contratos inteligentes são programas escritos em uma linguagem de programação, que utilizam redes blockchain como camada de execução [Zou et al. 2019]. Atualmente, os contratos inteligentes são amplamente utilizados em aplicações de finanças descentralizadas (DeFi) para a emissão de *tokens*, delegação de posse, transferência de ativos digitais e para pagamentos P2P automáticos. Eles ganharam popularidade nos últimos anos por incorporar propriedades de redes blockchain. O código intermediário de um contrato inteligente (*bytecode*) e as transações dele derivadas são gravados de maneira imutável, consistente e descentralizada em vários nós da rede blockchain. Logo, contratos inteligentes possibilitam a implantação e a operação de aplicações descentralizadas (DApps) com segurança, o que é um atrativo para diferentes áreas de negócios, desde finanças ao entretenimento digital [Harvey et al. 2021].

Semelhantemente, as redes blockchain também ganharam visibilidade nos últimos anos e, com isso, diversas redes tais como Ethereum [Buterin et al. 2014] e HyperLedger Fabric [Androulaki et al. 2018] surgiram. Enquanto algumas redes são públicas, isto é, o acesso é aberto e para executar uma transação basta apenas pagar as taxas da rede (mineração e complexidade das operações no contrato), outras redes são permissionadas, restringindo seu acesso a membros ou grupos confiáveis. Pelo fato das redes públicas exigirem taxas ao executar uma operação do contrato inteligente, seu código deve ser simples e otimizado. Por outro lado, as redes permissionadas permitem às organizações e/ou indivíduos executarem transações em contratos com códigos contendo operações mais complexas que nas redes públicas.

Independentemente da complexidade do contrato inteligente implementado, faz-se necessário verificar a sua corretude, ou seja, se a implementação não possui vulnerabilidades de segurança e segue boas práticas de codificação. Uma vulnerabilidade de segurança pode ser definida como uma falha em um sistema que pode ser explorada por um agente malicioso [Almakhour et al. 2020]. Se tratando de contratos inteligentes e blockchain, uma vulnerabilidade pode ser uma falha devido à má codificação, como o uso inadequado de funcionalidades da linguagem de programação do contrato, ou uma falha explorada a partir da arquitetura blockchain utilizada. Em ambos os casos, uma vulnerabilidade pode acarretar em comportamentos inesperados ou a perda do controle parcial ou total do contrato pelo agente malicioso [Ivanov et al. 2023].

Adicionalmente, para corretude e boas práticas de programação de contratos inteligentes é importante verificar as regras de negócio embutidas no contrato, isto é, se o mesmo atende às regras esperadas e as desempenham de forma segura. Esta verificação independe da rede Blockchain utilizada, seja permissionada ou pública, diferindo apenas na forma e nos métodos de verificação. Uma vez que o contrato inteligente esteja instalado e em funcionamento na rede blockchain, ainda assim ele pode estar suscetível a vulnerabilidades de segurança [Yamashita et al. 2019]. Nesse caso, as transações do contrato também devem ser monitoradas visando identificar uma vulnerabilidade antes que essa seja explorada.

Nesse contexto, diversas pesquisas estão sendo desenvolvidas para garantir a segurança de contratos inteligentes em redes Blockchain [Kushwaha et al. 2022]. Por exemplo, [Kalra et al. 2018] propõe a ferramenta Zeus, criada com o objetivo de analisar vulnerabilidades em contratos inteligentes voltados para as redes Ethereum e HyperLedger Fabric, utilizando a técnica de verificação formal *Model Checking*. Já em [Ding et al. 2021], é apresentada a ferramenta HFContractFuzzer, que mapeia as vulnerabilidades de contratos inteligentes de redes permissionadas codificados na linguagem de alto-nível Go. Com o auxílio da técnica de Fuzzing, os autores conseguiram identificar falhas de segurança em contratos de diversas redes Blockchain. Além das vulnerabilidades de segurança associadas aos contratos inteligentes, as regras de negócio embutidas também devem ser consideradas como pontos importantes de revisão e, dessa forma, em [Liao et al. 2022] é discutido a plataforma ModCon, utilizada principalmente para a geração e execução de testes funcionais voltados para contratos inteligentes escritos na linguagem Solidity.

Trabalho	Análise	Entrada	Técnica	Rede
[Xu et al. 2021]	Estática	Código	Aprendizado de máquina	Ethereum
[Yan et al. 2022]	Estática	Código	Aprendizado de máquina	Ethereum
[Li et al. 2022]	Estática/Dinâmica	Código	Execução simbólica	Hyperledger
[Ghaleb et al. 2023]	Estática	Bytecode	Execução simbólica	Ethereum
[Beillahi et al. 2022]	Estática	Opcodes	Grafo de fluxo de controle	Ethereum
[Zhang et al. 2023a]	Estática	Código	Grafo de fluxo de controle	Ethereum
[Xu et al. 2023]	Estática	Código	Árvore Sintática Abstrata	Hyperledger
[Yadav and Naval 2023]	Estática	Bytecode	Execução simbólica	Ethereum
[Ye et al. 2020]	Estática	Código	Grafo de fluxo de controle	Ethereum
[Wang et al. 2020]	Dinâmica	Opcodes	Aprendizado de máquina	Ethereum
[Rodler et al. 2023]	Dinâmica	Bytecode	Fuzzing	Ethereum
[Liu et al. 2023]	Dinâmica	Código	Fuzzing	Ethereum
[Liao et al. 2022]	Estática	Bytecode	Rede Neural	Ethereum

Tabela 1.1. Trabalhos que propõem ferramentas para auditoria de contratos inteligentes.

A auditoria de contratos inteligentes é uma das etapas mais importantes no ciclo de desenvolvimento de uma aplicação descentralizada envolvendo contratos inteligentes [David et al. 2023]. Geralmente, a auditoria ocorre após a etapa de desenvolvimento do código dos contratos, sendo realizada por empresas especializadas em segurança de software. O objetivo da auditoria é detectar falhas e comportamentos inesperados, *i.e.*, vulnerabilidades, antes da instalação e utilização do contrato na rede blockchain principal. O processo de auditoria consiste principalmente em inspeção manual do código pelo auditor, baseado em históricos de vulnerabilidades catalogados por comunidades de especialistas em segurança.

Nesse caso, o catálogo *Common Weakness Enumeration* é um dos mais conhecidos, sendo mantido por comunidades de desenvolvedores da indústria, academia e governos[1]. Importante também mencionar consórcios de corporações com negócios baseados em redes blockchain que formam alianças para definir especificações de segurança em contratos inteligentes nessas redes. Por exemplo, a *Enterprise Ethereum Alliance* (EEA) propuseram e vem mantendo um conjunto de especificações para a rede Ethereum denominado *EEA EthTrust Security Levels Specification*[2].

Adicionalmente à inspeção manual, existem métodos de análise automáticas de códigos fonte que auxiliam o trabalho de auditores, e os mais conhecidos são os métodos de análise estática e dinâmica. Uma forma simples de conceituar esses métodos concerne o ambiente onde os contratos são executados, *i.e.*, a rede blockchain. A análise estática lida com a identificação de vulnerabilidades no código do contrato sem propriamente implantá-lo no ambiente de execução [Ivanov et al. 2023]. Por sua vez, a análise dinâmica atua identificando vulnerabilidades no que diz respeito ao ambiente de execução em que o código foi implantado, i.e., vulnerabilidades relacionadas ao comportamento das funcionalidades do contrato e das transações gravadas na rede blockchain [Almakhour et al. 2020].

Há um grande esforço no desenvolvimento de ferramentas automáticas que auxiliam o processo de auditoria de contratos inteligentes. Contudo, existem questões práticas de adoção dessas ferramentas que precisam ser cuidadosamente consideradas [Chaliasos et al. 2024]. Primeiro, uma parte das ferramentas de código-aberto já difundidas na literatura, que atuam como *benchmarks* em diversos trabalhos, podem ter sido descontinuadas, *e.g.* Oyente [Luu et al. 2016] e Vandal [Brent et al. 2018]. Segundo, é comum que ferramentas de auditoria deixem de ser projetos de código-aberto e passam a ser incorporadas à uma organização privada, *e.g.* Mythril [ConsenSys 2024] e Smartcheck [Tikhomirov et al. 2018].

A Tabela 1.1 apresenta 13 projetos de ferramentas automáticas de análise de contratos inteligentes que abordam os desafios e mecanismos de verificação automática de código para o contexto do presente trabalho, indicando se a ferramenta utiliza análise análise estática ou dinâmica e se a entrada para a análise é o código fonte, *opcodes* ou *bytecode* do código fonte. Ainda, descrevemos a técnica base para a análise e a rede blockchain utilizada, sendo Ethereum e Hyperledger Fabric as redes que essas ferramentas suportam até o momento.

Em cada ferramenta são utilizadas uma ou mais técnicas de verificação de código já existentes na literatura para verificação de software no geral. Normalmente, as técnicas exigem a conversão do código do contrato inteligente para uma Representação Intermediária (IR), semelhante aos compiladores de linguagens de programação convencionais. Na literatura, as ferramentas variam ou combinam técnicas de acordo com as vulnerabilidades que desejam identificar. Por exemplo, a ferramenta Manticore [Mossberg et al. 2019] utiliza a técnica Execução Simbólica para explorar o espaço de estados do programa. Conforme a Figura 1.11, a partir de uma entrada simbólica, o código de um contrato inteligente pode ser analisado pelos estados gerados de acordo com a entrada, com o objetivo de identificar

[1] `https://cwe.mitre.org/`

[2] `https://entethalliance.org/specs/ethtrust-sl/`

se estados e/ou propriedades que não podem ser atingidos (*i.e.* vulnerabilidades) realmente não são.

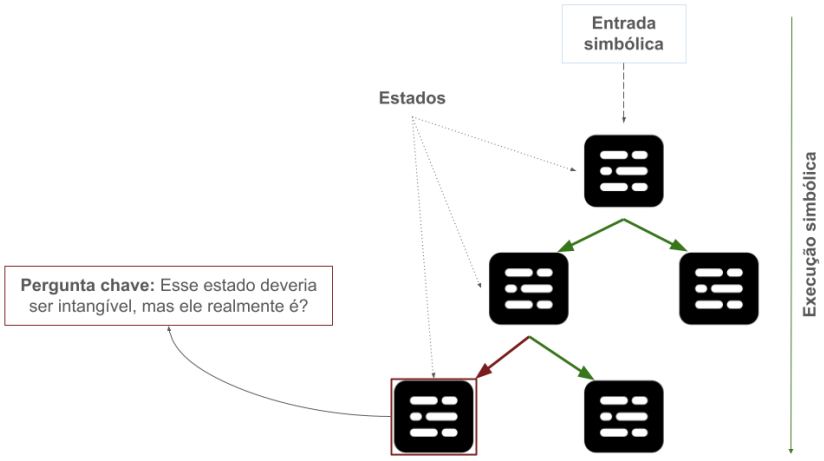

Figura 1.11. Fluxo da execução simbólica sobre os estados de um contrato inteligente.

Geralmente, a técnica de execução simbólica pode ser combinada com as técnicas de verificação formal, como o *Model Checking*. Em [Song et al. 2022], é apresentada a ferramenta ESBMC-Solidity, que utiliza a verificação formal em conjunto com a execução simbólica para checar as propriedades de vulnerabilidades em contratos inteligentes escritos na linguagem de programação Solidity. Dessa maneira, ameaças intrínsecas ao código e da linguagem do contrato podem ser identificadas. De contrapartida, brechas de segurança referentes ao fluxo de execução do contrato inteligente exigem técnicas capazes de analisar o comportamento das funcionalidades do código que são invocadas durante sua utilização. Sendo assim, técnicas de análise dinâmica como *Fuzzing* mostram-se como uma solução promissora na literatura.

Inicialmente, a técnica de *Fuzzing* foi concebida como um mecanismo de testagem de software [Zeller et al. 2019]. Os recentes trabalhos expandem este conceito para o contexto de blockchain e contratos inteligentes, em que a técnica pode ser utilizada como um mecanismo de verificação do comportamento de utilização do código de um contrato, como mostra a Figura 1.12. A partir de entradas geradas aleatoriamente e em grande quantidade, o contrato inteligente pode ser monitorado de acordo com os resultados das execuções de suas funcionalidades, com o objetivo de serem identificadas vulnerabilidades que dizem respeito não somente ao código do contrato, mas também da arquitetura da rede blockchain utilizada e as regras de negócio embutidas.

Nesse sentido, recentes trabalhos integram o *fuzzing* de diversas maneiras com o ecossistema de contratos e blockchain. Em [Wüstholz and Christakis 2020] é proposto a ferramenta Harvey, que estende o conceito de geração de entradas da técnica convencional para gerar resultados satisfatórios no contexto de contratos inteligentes. Além disso, a ferramenta executa os testes com as entradas priorizando a exploração de estados possíveis do contrato de maneira inteligente, levando em consideração as transações que ocorrem na rede devido às funcionalidades do contrato. Semelhantemente em [Jiang et al. 2018], a ferramenta propõe a utilização da técnica *fuzzing* combinada com a criação de oráculos de

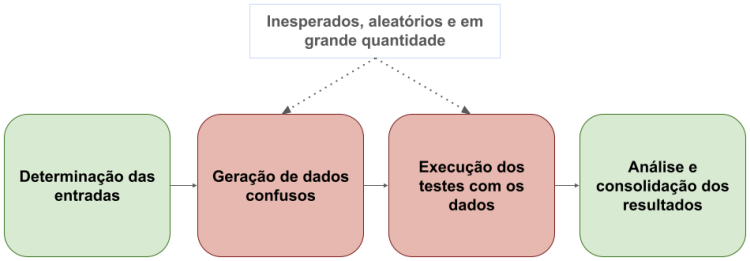

Figura 1.12. Fluxo da técnica de *fuzzing* para identificação de falhas.

teste, em que cada oráculo é responsável por identificar uma ameaça específica.

Em suma, o processo de auditoria dos contratos inteligentes auxilia na mitigação de perdas de ativos financeiros no ecossistema DeFi. A junção das técnicas de verificação de software convencionais com o contexto de contratos inteligentes podem acarretar em análises assertivas, desde que haja especificações detalhadas de possíveis falsos positivos e constante suporte às ferramentas difundidas na literatura.

1.4.2. *Front-running* em Redes Blockchain

Qualquer ação realizada pelo usuário que modifique o estado da blockchain é registrada como uma transação. Uma vez efetivada, a transação não pode ser revertida e é esse recurso que torna a blockchain um livro razão imutável. Nesse sentido, os contratos inteligentes, assim como os usuários externos, são capazes de armazenar ativos em contas endereçáveis e realizar transações que alteram o estado da rede [Varun et al. 2022].

Contratos inteligentes são programas criados usando uma linguagem de programação de alto nível como Solidity. Eles são projetados para serem executados quando um conjunto predeterminado de condições forem atendidas. Geralmente, eles automatizam a execução de um ativo ou acordo que garante que todos os participantes conheçam instantaneamente o resultado, sem o envolvimento de qualquer parte intermediária. Os contratos inteligentes também podem automatizar processos, ou seja, desencadear outra ação assim que o contrato atual for executado. Uma vez que são implantados na blockchain, quando as condições exigidas são atendidas, eles são executados por uma rede de nós que chegam a um consenso antes que o estado executado seja armazenado na blockchain.

Uma blockchain é um registro de transações anexado. Por sua vez, as transações são armazenadas em uma *pool* e tratadas igualmente, independentemente do horário específico em que elas foram adicionadas, para posteriormente serem combinadas em um bloco por nós mineradores. Dessa maneira, mais de 95% dos mineradores optam por ordenar as transações em relação ao preço do gás, que é a taxa de transação que os mineradores recebem por anexar uma transação em um bloco [Zhou et al. 2021].

Uma das principais ameaças ao ecossistema DeFi na blockchain Ethereum atualmente é a manipulação na ordem de transações que serão efetivadas na rede por usuários estratégicos que monitoram constantemente a fila pública de transações pendentes (i.e., *mempool*), ataque esse conhecido como *front-running*.

Como mostra a Figura 1.13, nesse tipo de ataque o valor da tarifa da transação é

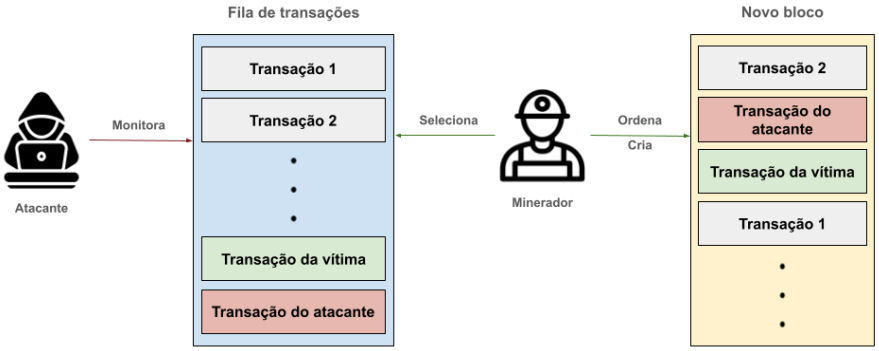

Figura 1.13. Demonstração do cenário de ataque *front-running*.

utilizado pelo atacante para manipular a ordem das transações que aguardam na *mempool* para constituírem o novo bloco. Por essência, o consenso distribuído em blockchain elimina uma autoridade central para gerenciar a *mempool* e evitar ataques *front-running*. Assim, vem ocorrendo um aumento no número de atacantes, bem como várias formas de ataques *front-unning* [Varun et al. 2022, Torres et al. 2021, Zhang et al. 2023c, Zhang et al. 2023b].

Existem três tipos principais de ataques *front-running*: deslocamento, inserção e supressão. O ataque de inserção é conhecido como ataque sanduíche. No ataque de deslocamento, a transação de ataque desloca a transação da vítima por ter um preço de gás mais alto. Como a transação de ataque oferece melhor incentivo, ela é extraída antes da transação da vítima. Este ataque é usado em jogos de quebra-cabeça onde é preciso enviar a chave para vencer o jogo. Assim que o atacante vê a solução da vítima, ele cria a mesma transação com um preço de gás mais alto, inutilizando a transação da vítima. No ataque de supressão, um atacante faz múltiplas transações com um preço de gás muito mais alto do que a transação da vítima para evitar que a transação da vítima seja explorada no mesmo bloco. Esse ataque também é conhecido como *cluster* de supressão. É frequentemente usado em jogos de loteria que têm como regra que a última pessoa a entrar na loteria ganha.

O ataque sanduíche, que é um tipo *front-running*, é uma estratégia de negociação já conhecida nos sistemas financeiros tradicionais, onde um usuário com visão privilegiada do sistema identifica um negócio promissor de outro usuário e o executa antecipadamente obtendo os benefícios desse negócio. Esse tipo de ataque vem chamando a atenção de pesquisadores recentemente no contexto da blockchain Ethereum e DeFi [Torres et al. 2021, Zhang et al. 2023b]. Nesse caso, um atacante inicia monitorando a *mempool* em busca de transações pendentes que estão prestes a negociar grandes somas de um determinado ativo. Uma grande transação resultará em uma flutuação no preço do ativo. Na sequência, o atacante cria o chamado "sanduíche", cercando esta grande transação com duas de suas próprias transações. Na primeira transação, o atacante executa uma grande transação para comprar ou vender alguma quantidade de ativo antes que o preço do ativo flutue. Na segunda transação, o atacante retrocede a grande transação para recomprar o ativo original por um preço mais baixo ou vender o ativo recém-adquirido por um preço mais alto. O atacante obtém lucro devido à diferença de preço e a vítima pode sofrer prejuízo [Weintraub et al. 2022].

Os atacantes podem ser mineradores ou não mineradores. Os mineradores não são obrigados a pagar um preço mais alto de gás para manipular a ordem das transações, pois têm controle total sobre as transações que são incluídas em um bloco. Os não mineradores, por outro lado, são obrigados a pagar um preço mais elevado de gás para antecipar as transações de outros não mineradores. Em [Torres et al. 2021] é assumido que o atacante é um não minerador financeiramente racional com a capacidade de monitorar a *mempool* de transações. O atacante precisa processar as transações na *mempool*, encontrar uma vítima e criar transações de ataque antes que a transação da vítima seja minerada.

O atacante não seria capaz de reagir rápido o suficiente para realizar todas as tarefas necessárias para efetuar o ataque manualmente. Portanto, seguindo [Torres et al. 2021], o atacante possui pelo menos um programa de computador (*Bot*) que executa automaticamente as tarefas necessárias para o ataque. O *Bot* precisa de pelo menos uma ou mais contas de propriedade externa (EOA - *Externally Owned Accounts*) para atuar como remetente de qualquer transação de ataque. O uso de várias contas de propriedade externa auxilia os atacantes a ocultar suas atividades, semelhante aos esquemas de lavagem de dinheiro. Assumimos que o atacante possui um saldo suficientemente grande em todas as suas contas, a partir do qual pode enviar transações de ataque com gás suficientemente maior que o gás da transação da vítima. No entanto, o atacante também pode empregar contratos inteligentes para manter parte da lógica do ataque. Esses contratos inteligentes são referidos como contratos de *Bot*, que são invocados pelas contas do atacante.

A Figura 1.14 apresenta um cenário com ataque sanduíche. No ataque sanduíche, o atacante envia duas transações, uma com um preço de gás mais alto do que a transação da vítima e outra com um preço de gás mais baixo para intercalar a transação da vítima. É usado em plataformas descentralizadas de câmbio (DEXes - *Descentralized Exchanges*) para fazer sanduíches de transações prestes a negociar grandes somas de um determinado ativo, também conhecido na literatura como transações baleia [Varun et al. 2022].

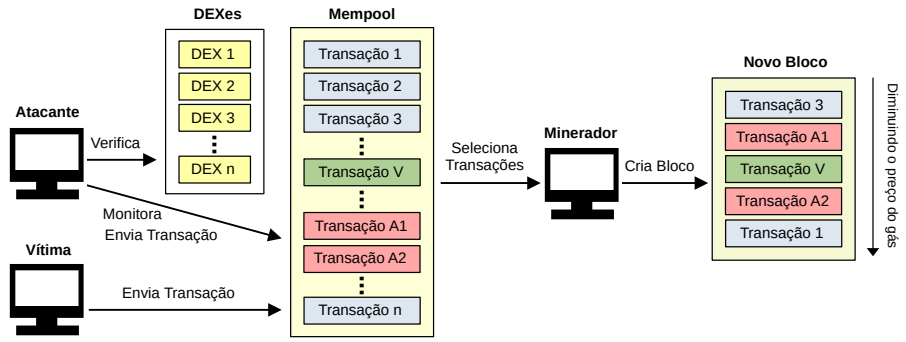

Figura 1.14. Demonstração de um cenário com ataque sanduíche.

Observa-se na Figura 1.14 que o atacante monitora a *mempool* na espera de uma possível transação vítima (Transação V). O atacante verifica nas DEXes se o ativo da transação da vítima pode gerar lucro. Em caso positivo, o atacante gera duas transações (Transação A1 e Transação A2). Elas, por sua vez, possuem valores de gás que fazem com que realizem um "sanduíche" com a transação da vítima quando o minerador selecionar transações para criar um bloco seguindo a ordenação convencional pelo preço do gás.

Os ataques sanduíche exploram as AMM DEXes. Uma ordem de compra aumentará o preço de um ativo, enquanto uma ordem de venda diminuirá o preço do ativo. Portanto, os especuladores podem monitorar continuamente a rede (*i.e. mempool* e AMMs) para encontrar transações pendentes para AMM DEXes (*i.e.* transação da vítima) que implicarão diferenças de preços. Dessas DEXes, os especuladores podem comprar o ativo por um preço baixo antes que a transação da vítima seja executada (transação T_{A1}) e venderem o ativo após a transação da vítima aumentar de preço (transação T_{A2}), gerando lucro para si. Ao final do ataque, a cotação da transação vítima é pior do que seria sem a transação T_{A1}, resultando em perda financeira para a vítima [Wang et al. 2022].

Para identificar uma arbitragem de ataque sanduíche, podem ser verificadas todas as transações de transferência em um bloco. Uma transação de transferência é definida como $T = (s, r, a, c, h, i)$, em que s é remetente dos *tokens*, r é o receptor dos *tokens*, a é o número de *tokens* transferidos, c é o endereço do contrato do *token*, h é o *hash* da transação, e i é o índice da transação. Ao verificar todas as transações em um bloco, busca-se encontrar três transações de transferência: T_{A1}, T_V e T_{A2}. As transações T_{A1} e T_{A2} estão relacionadas com o atacante, e a transação T_V está relacionada com a vítima.

Um possível algoritmo para detecção de arbitragem de ataque sanduíche segue as suposições a seguir baseadas em [Torres et al. 2021]. As transações T_{A1}, T_V e T_{A2} devem estar nessa ordem, ou seja, o índice de T_{A1} deve ser menor do que o índice de T_V e o índice de T_V deve ser menor que o índice de T_{A2} ($i_{A1} < i_V < i_{A2}$). O atacante e a vítima compram *tokens* em T_{A1} e em T_V, respectivamente. Em seguida, o atacante vende os *tokens* em T_{A2} que comprou anteriormente em T_{A1}. O número de *tokens* comprados por T_{A1} deve ser semelhante ao número de *tokens* vendidos por T_{A2} (ou seja, $a_{A1} \approx a_{A2}$). O atacante e a vítima realizam transações sobre os mesmos *tokens*, ou seja, os endereços de contrato de *token* de T_{A1}, T_V e T_{A2} devem ser idênticos ($c_{A1} = c_V = c_{A2}$). O remetente de T_{A1} deve ser idêntico ao remetente de T_V, bem como o receptor de T_{A2}, e o receptor de T_{A1} deve ser idêntico ao remetente de T_{A2} (ou seja, $s_{A1} = s_V = r_{A2} \wedge r_{A1} = s_{A2}$). Os *hashes* de transação de T_{A1}, T_V e T_{A2} devem ser diferentes (ou seja, $h_{A1} \neq h_V \neq h_{A2}$). O preço do gás de T_{A1} deve ser maior que o preço de gás de T_V. E o preço do gás de T_{A2} deve ser menor ou igual ao preço do gás de T_V (ou seja, $g_{A1} > g_V \geq g_{A2}$).

O algoritmo para detecção de arbitragem de ataque sanduíche assume que os ataques sanduíche sempre ocorrem dentro do mesmo bloco. Essa suposição permite verificar os blocos em paralelo, uma vez que só é preciso comparar as transações dentro de um bloco. No entanto, esta suposição nem sempre se aplica à realidade, uma vez que as transações podem ser dispersas por diferentes blocos durante o processo de mineração. Sendo assim, podem existir ataques sanduíches realizados em vários blocos e que o algoritmo não é capaz de detectar. Portanto, essa abordagem representa um limite inferior para análises de ataques sanduíches na blockchain Ethereum. Contudo, existem abordagens que expandem a análise para mais de um bloco, como no trabalho de [Zhang et al. 2023b] que analisa os ataques em uma janela de 3 blocos consecutivos.

1.4.3. Manipulação de Oráculos

Um oráculo conecta o mundo blockchain on-chain com o mundo off-chain, por meio de interface com provedores de dados externos.

Oráculos provêm informações fundamentais para ativar gatilhos e tomadas de decisão em contratos inteligentes. Por serem entidades de confiança, têm o "privilégio"de fornecer dados aceitos incondicionalmente. Este detalhe é crucial, pois todo o ecossistema da blockchain gira em torno do conceito de imutabilidade e interação sem confiança por meio da descentralização. Conectar a blockchain a um ponto de falha centralizado, como um oráculo, resulta fundamentalmente em uma perda de descentralização. Esse dilema é conhecido como "o problema do oráculo"e afeta todas as aplicações blockchain do mundo real. Dependendo do setor e do tipo de oráculo, diferentes consequências podem surgir [Caldarelli and Ellul 2021].

Existem várias circunstâncias em que um oráculo pode deixar de fornecer dados confiáveis, assumindo um modelo de falhas bizantino: Os oráculos podem ser mal programados, apresentar *bugs*, sofrer sabotagem ou mau funcionamento. O problema do oráculo pode envolver tanto questões técnicas, mas também aspectos sociais.

Como um ponto único de falha, a chance de um oráculo ser comprometido ou dos administradores conspirarem para alterar os dados fornecidos pode estar correlacionada ao valor dos contratos inteligentes: quanto maior o valor dos ativos manipulados pelo contrato inteligente, maior a probabilidade dele ser alvo de ataques. Um oráculo pode assim ser comprometido por meio de conluio ou suborno dos administradores da entidade gestora do oráculo que pode alterar a transferência de dados para fins egoístas.

Um oráculo que provê informações sobre quantidade e preço de ativos financeiros negociados provê um conjunto de informações pública que pode ser facilmente verificável. O uso de um mecanismo de quórum com diferentes provedores desta informação pode prover um oráculo decentralizado resiliente a provedores bizantinos. O provimento de canais seguros para evitar a adulteração de dados providos pelo oráculo também pode ser utilizado. Por fim, aspectos temporais podem ser observados, em face a situações que um atacante tente atrasar a atualização de informações providas por um oráculo (*e.g.*, a alteração de preço de um dado ativo) para tomar vantagem da posição desatualizada.

1.4.4. Ataques de *Flash Loan*

Um ataque de *flash loan* é um mecanismo de exploração sofisticado no ecossistema DeFi por meio do uso de empréstimos rápidos (ditos *flash loans*). Estes empréstimos rápidos são uma proposta sem precedentes do ecossistema de Finanças Descentralizadas (DeFi), em 2018 o projeto *Marble 1* idealizou o mecanismo para permitir que qualquer pessoa emprestasse ativos sem garantia para aproveitar oportunidades de arbitragem, desde que os fundos fossem devolvidos no âmbito da mesma transação [Cao et al. 2021].

O desafio é que o atacante utiliza desta possibilidade para manipular valores de um dado ativo e obter vantagens indevidas. Em um passo a passo exemplificamos como um ataque de *flash loan* pode acontecer:

1. Tomada do Empréstimo: O atacante adquire um empréstimo rápido de uma grande quantidade de um dado *token* sem fornecer garantias. Isso é possível porque o reembolso do empréstimo é garantido pelo próprio protocolo de empréstimo dentro da mesma transação;

2. Manipulação do Mercado: Usando os fundos do empréstimo, o atacante pode

manipular o preço de um ativo em uma ou várias corretoras descentralizadas. Por exemplo, ele pode comprar uma grande quantidade de um dado *token* para inflar seu preço ou vender em massa para diminuir o preço;

3. Realização do Lucro: Após manipular o mercado e explorar as vulnerabilidades, o atacante troca os ativos de volta para a criptomoeda original, mas agora com lucro; e

4. Reembolso do Empréstimo: Finalmente, dentro da mesma transação, o atacante reembolsa o empréstimo rápido junto com a taxa de empréstimo. Como toda a operação ocorre dentro de um único bloco, se em algum ponto o atacante não puder reembolsar o empréstimo, a transação é revertida, e nada acontece.

Este mecanismo é sofisticado e requer uma análise do padrão comportamental das transações para inferir possíveis intenções dos remetentes [Wang et al. 2021].

1.4.5. Ataques de Liquidez

Os mecanismos descentralizados inerentes a DeFi provêm uma assimetria de informações que pode ser utilizada por atacantes. Por exemplo, ataques de arbitragem de liquidez exploram as diferenças de preço entre diferentes *pools* de liquidez ou corretoras. Um atacante pode usar *bots* de arbitragem para detectar e explorar essas discrepâncias de preços, comprando *tokens* em uma plataforma onde o preço é baixo e vendendo-os em outra onde o preço é mais alto. Embora a arbitragem em si não seja maliciosa, quando feita de forma agressiva ou com informações privilegiadas, pode levar à drenagem de liquidez de certos *pools*, prejudicando os participantes honestos.

Outra possibilidade é manipular o preço de um ativo dentro de um pool de liquidez injetando uma grande quantidade de um *token* em um pool, o que inflaciona artificialmente o preço do *token*. Uma vez que o preço é manipulado, o atacante pode executar negociações subsequentes em outras plataformas ou explorar contratos inteligentes que dependem de oráculos de preço que agora refletem o preço manipulado.

Os ataques de manipulação de preços originam-se de vulnerabilidades lógicas de aplicativos DeFi, o que torna a detecção não trivial. Ou seja, a detecção de tais ataques exige que analisemos as transações entre vários contratos inteligentes e compreendamos a semântica de alto nível dos aplicativos DeFi [Wu et al. 2021].

1.4.6. Perspectivas em Segurança de DApps

A segurança e correção de contratos inteligentes representam um desafio importante para o ecossistema de aplicações DeFi e blockchain. Esses contratos, que gerenciam grandes quantidades de criptomoedas e ativos digitais são alvos frequentes de ataques [Ivanov et al. 2023]. Devido à imutabilidade das redes blockchain, contratos inteligentes precisam ser projetados e testados rigorosamente antes da implantação. Apesar dos avanços na segurança dos contratos inteligentes, ataques recentes a plataformas, destacam vulnerabilidades persistentes. *Bugs* podem surgir tanto das limitações das linguagens de programação de contratos inteligentes quanto de erros na lógica de negócios, sendo o último mais difícil de detectar e corrigir, exigindo uma modelagem precisa e análise cuidadosa por especialistas.

Nesse sentido, a combinação de técnicas de verificação de contratos inteligentes

pode ser um caminho promissor. Alguns trabalhos propõem a utilização de análise estática e dinâmica para identificação de brechas em contratos [Linoy et al. 2021], enriquecendo a etapa de análise de brechas e riscos para uma aplicação DeFi no geral. Embora a combinação de técnicas desempenhe um papel inovador para o campo de verificação de contratos, estratégias que melhoram o desempenho das soluções atuais também são importantes. Os contratos inteligentes implantados em redes blockchain, muitas das vezes, possuem diversas chamadas de contratos externos, contratos estes que podem possuir brechas significativas para um prejuízo financeiro [Liao et al. 2022]. Logo, a verificação *cross-contract* mostra-se como um desafio para as soluções atuais difundidas na literatura [Ye et al. 2020]. Por fim, a utilização de técnicas *out-of-the-box* traçam um caminho inovador para a verificação de contratos, isto é, a utilização de mecanismos e técnicas que não são do campo de verificação de software no geral, como o aprendizado de máquina. Esta prática, como mostra [Xu et al. 2021], apesar de ser um meio de verificação que deve ser extensivamente analisado em busca de falsos positivos e negativos, pode gerar *insights* e oportunidades de pesquisa capazes de propor estratégias de mitigação de certas brechas que são comuns para redes blockchain.

No que diz respeito à interoperabilidade entre redes blockchain, a segurança torna-se um desafio crítico, com ataques a pontes de cadeia cruzada causando perdas financeiras significativas. Um dos principais problemas de segurança é o ataque de duplo gasto [Chohan 2021] entre redes blockchain. Esse ataque ocorre quando uma criptomoeda é usada mais de uma vez, explorando fraquezas em redes blockchain PoW, especialmente quando um grupo de mineradores controla mais de 50% do poder de mineração [Conti et al. 2018]. No contexto de cadeias cruzadas, esses ataques podem acontecer antes ou depois das transações, impactando negativamente ambas as redes blockchain envolvidas. Para mitigar esses riscos, protocolos de cadeia cruzada geralmente implementam tempos de espera elevados para confirmar transações, pois isso permite uma verificação mais robusta e garante que a transação seja irreversível e aceita por todos os nós da rede.

Adicionalmente, a privacidade é um desafio significativo na interoperabilidade devido ao potencial de violação do anonimato e rastreabilidade. Em abordagens tradicionais de interoperabilidade, partes confiáveis atuam como intermediárias, comprometendo o anonimato dos usuários ao verificar transações e bloquear ativos. Para preservar a privacidade, é necessário desenvolver uma infraestrutura de interoperabilidade que elimine a necessidade de intermediários e manter o mesmo nível de anonimato que os sistemas de blockchain originais. Uma solução para isso que vem sendo avaliada por pesquisadores é a utilização do mecanismo de *Zero Knowledge Proof* (ZKP). Neste mecanismo, uma informação sensível como dados de transações de interoperabilidade podem ser propagadas para a rede sem necessariamente revelá-las [Sun et al. 2021]. Nesse sentido, um dos principais problemas nesse contexto é a rastreabilidade das transações entre redes blockchain. Por exemplo, trocas HTLC (*Hashed Time-Lock Contract*) são utilizadas em transações de criptomoedas para garantir que duas partes só possam concluir uma troca se certas condições forem atendidas. No entanto, se o mesmo valor *hash* for usado em várias transações, pode haver um risco de segurança, pois o valor secreto revelado em uma transação pode ser usado para reivindicar outras transações que dependem do mesmo *hash*. Para resolver isso, são necessários mecanismos adicionais de preservação da privacidade, como assinaturas adaptadoras [Deshpande and Herlihy 2020], semelhante ao ZKP, em que

permitem que uma transação seja assinada de uma maneira que não revele informações sobre a transação em si. Ainda, ambientes de execução confiáveis [Sabt et al. 2015] [Li et al. 2021] podem ser avaliados, garantindo que os dados e códigos sejam mantidos em segredo, mesmo quando são processados para impedir que essas conexões sejam detectadas.

Além disso, a privacidade dos dados também é um desafio crítico no que diz respeito às aplicações descentralizadas, especialmente em setores sensíveis a esse tipo de problema. O compartilhamento e transferência de dados sensíveis que circulam nas redes podem gerar riscos às plataformas financeiras. Por exemplo, alguns trabalhos discutem possibilidades estratégicas de mitigação de ataques criando filas de transações pendentes privadas [Eskandari et al. 2020], para que aplicações não exponham transações importantes livremente na rede. Apesar de ser uma solução promissora, o seu uso impacta diretamente no pilar de descentralização de redes blockchain, sendo portanto um desafio de constantes pesquisas e inovações. Já em [Ma et al. 2019], são discutidos os mecanismos de proteção de privacidade implementados no *Hyperledger Fabric* e sua aplicação no contexto da cadeia de suprimentos financeira. De maneira semelhante, essa perspectiva beneficia diretamente quando se trata de interoperabilidade em redes blockchain garantindo a privacidade de todas as partes envolvidas. A interoperabilidade deve garantir que a privacidade seja mantida, mesmo quando interconectando diferentes redes blockchain protegendo informações sensíveis de acesso não autorizado.

1.5. Considerações Finais

Este capítulo abordou os conceitos, aplicações e perspectivas sobre o ecossistema de finanças descentralizadas. Com o advento das redes blockchain e o uso de contratos inteligentes com a rede Ethereum, as aplicações de finanças descentralizadas (DeFi - *Decentralized Finance*) emergiram como uma solução flexível para movimentação de ativos financeiros. Este novo ecossistema visa aprimorar os serviços financeiros tradicionais por meio da negociação de *tokens*, da redução de intermediários e de barreiras para o crédito, bem como do acesso mais amplo a serviços financeiros no contexto da *web* descentralizada. Este novo ecossistema propiciou um conjunto de aplicações inovadoras, desafios a interoperabilidade entre diferentes redes blockchain, bem como novas demandas de segurança para tratar as possíveis vulnerabilidades e ameaças neste ambiente descentralizado. Neste sentido, apresentamos os fundamentos e exploramos, com exemplos práticos, algumas das mais utilizadas aplicações DeFi. Observando o ecossistema DeFi sob a ótica destes três aspectos, aplicações, interoperabilidade e segurança, discutimos as perspectivas, os desafios e o estado-da-arte que suportará o crescimento e uso de DeFi nos próximos anos.

Em aplicações, discutimos a arquitetura de DeFi no contexto de redes blockchain públicas. A partir da introdução dos *tokens* em conjunto com os contratos inteligentes, as finanças descentralizadas representam majoritariamente as transações que ocorrem em redes blockchain como a Ethereum. É possível notar a introdução de instituições financeiras e governamentais neste ecossistema por meio das *stablecoins* e CDBCs. Além disso, as corretoras descentralizadas representam o meio mais utilizado atualmente para os usuários usufruírem dos recursos que o ecossistema DeFi tem a oferecer com a utilização de redes blockchain.

Em interoperabilidade apontamos os mecanismos de transferência de ativos entre

redes mais utilizados atualmente. A interoperabilidade, se tratando de redes blockchain e DeFi, representa a comunicação entre duas redes distintas, ou seja, o usuário possui a capacidade de trafegar entre redes diferentes, movimentar ativos e utilizar plataformas de câmbio de maneira segura e escalável. Além disso, discutimos as soluções implementadas em redes como Ethereum, levando em consideração aspectos de segurança e centralização de ativos. Por exemplo, o mecanismo de *sidechains* permite que as redes blockchain principais (*mainchains*) possuam diferentes implementações com objetivos distintos, *e.g sidechains* com foco em governança, latência de transações por segundo, etc. Por outro lado, enquanto mecanismos como o HTLC abordam transações atômicas sem a necessidade de um terceiro confiável, mas geram riscos de privacidade, mecanismos como o notarial introduzem um terceiro confiável, com o objetivo de proporcionar maior privacidade, porém centraliza toda a troca de ativos em um notário.

Em segurança abordamos os desafios práticos de implantação de uma aplicação descentralizada por meio de contratos inteligentes. O código destes contratos podem estar suscetíveis à vulnerabilidades de segurança e, portanto, passam por um rigoroso processo de auditoria. Por meio de técnicas convencionais e *out-of-the-box* ("fora da caixa") de verificação de softwares, pesquisadores e iniciativas privadas fomentam um campo de pesquisa de criação de técnicas e *frameworks* capazes de identificar potenciais ameaças em ambientes redes blockchain. Nesse sentido, apresentamos algumas das principais vulnerabilidades que ocorrem em redes blockchain públicas como a Ethereum e ecossistemas DeFi, *i.e. front-running*, manipulações de oráculos, *flash-loans* e ataques de liquidez. Estas práticas representam um perigo para redes públicas, devido à influência negativa que estas transações podem ocasionar em preços de *tokens* ao tratarmos de finanças descentralizadas.

Finalmente, condensamos as perspectivas de segurança ao abordamos os aspectos de auditoria, interoperabilidade e aplicações. Evidenciamos oportunidades de pesquisa nas três subáreas que constituem o ecossistema DeFi, de forma que este capítulo proporcione uma base de conhecimento e perspectivas do estado-da-arte para interessados na área de finanças descentralizadas.

Agradecimentos

Os autores agradecem o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) processo no. 88887.918434/2023-00, Fundação de Amparo à Pesquisa do Piauí (FAPEPI) processo no. 00110.000235/2022-78 e o Comitê Técnico Blockchain da Rede Nacional de Pesquisas CT-Blockchain RNP. Agradecem também o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

Referências

[Allen et al. 2022] Allen, F., Gu, X., and Jagtiani, J. (2022). Fintech, cryptocurrencies, and cbdc: Financial structural transformation in china. *Journal of International Money and Finance*, 124:102625.

[Almakhour et al. 2020] Almakhour, M., Sliman, L., Samhat, A. E., and Mellouk, A. (2020). Verification of smart contracts: A survey. *Pervasive and Mobile Computing*,

67:101227.

[Androulaki et al. 2018] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15.

[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*, 72:201–224.

[Beillahi et al. 2022] Beillahi, S. M., Keilty, E., Nelaturu, K., Veneris, A., and Long, F. (2022). Automated auditing of price gouging tod vulnerabilities in smart contracts. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–6. IEEE.

[Belchior et al. 2021] Belchior, R., Vasconcelos, A., Guerreiro, S., and Correia, M. (2021). A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)*, 54(8):1–41.

[Besançon et al. 2019] Besançon, L., Silva, C. F. D., and Ghodous, P. (2019). Towards blockchain interoperability: Improving video games data exchange. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 81–85.

[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.

[Binance 2023] Binance (2023). O que é uma stablecoin? `https://academy.binance.com/pt/articles/what-is-a-stablecoin`. (Accessed on 05/23/2024).

[Breidenbach et al. 2022] Breidenbach, L., Cachin, C., Chan, B., Coventry, A., Ellis, S., Juels, A., Koushanfar, F., Miller, A., Magauran, B., Moroz, D., et al. (2022). Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. 2021. *Available also from: https://research. chain. link/whitepaper-v2. pdf*.

[Brent et al. 2018] Brent, L., Jurisevic, A., Kong, M., Liu, E., Gauthier, F., Gramoli, V., Holz, R., and Scholz, B. (2018). Vandal: A scalable security analysis framework for smart contracts. *arXiv preprint arXiv:1809.03981*.

[Buterin 2016] Buterin, V. (2016). Chain interoperability. *R3 research paper*, 9:1–25.

[Buterin et al. 2014] Buterin, V. et al. (2014). A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1.

[Caldarelli and Ellul 2021] Caldarelli, G. and Ellul, J. (2021). The blockchain oracle problem in decentralized finance—a multivocal approach. *Applied Sciences*, 11(16):7572.

[Cao et al. 2021] Cao, Y., Zou, C., and Cheng, X. (2021). Flashot: a snapshot of flash loan attack on defi ecosystem. *arXiv preprint arXiv:2102.00626*.

[Casino et al. 2019] Casino, F., Dasaklis, T. K., and Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and informatics*, 36:55–81.

[Chaliasos et al. 2024] Chaliasos, S., Charalambous, M. A., Zhou, L., Galanopoulou, R., Gervais, A., Mitropoulos, D., and Livshits, B. (2024). Smart contract and defi security tools: Do they meet the needs of practitioners? In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, pages 1–13.

[Chen et al. 2021] Chen, Y., Richter, J. I., and Patel, P. C. (2021). Decentralized governance of digital platforms. *Journal of Management*, 47(5):1305–1337.

[Chohan 2021] Chohan, U. W. (2021). The double spending problem and cryptocurrencies. *Available at SSRN 3090174*.

[ConsenSys 2024] ConsenSys (2024). Mythril: A security analysis tool for evm bytecode.

[Conti et al. 2018] Conti, M., Kumar, E. S., Lal, C., and Ruj, S. (2018). A survey on security and privacy issues of bitcoin. *IEEE communications surveys & tutorials*, 20(4):3416–3452.

[Costa et al. 2023] Costa, D., La Cava, L., and Tagarelli, A. (2023). Show me your nft and i tell you how it will perform: Multimodal representation learning for nft selling price prediction. In *Proceedings of the ACM Web Conference 2023*, pages 1875–1885.

[David et al. 2023] David, I., Zhou, L., Qin, K., Song, D., Cavallaro, L., and Gervais, A. (2023). Do you still need a manual smart contract audit? *arXiv preprint arXiv:2306.12338*.

[Deshpande and Herlihy 2020] Deshpande, A. and Herlihy, M. (2020). Privacy-preserving cross-chain atomic swaps. In *International Conference on Financial Cryptography and Data Security*, pages 540–549. Springer.

[Ding et al. 2021] Ding, M., Li, P., Li, S., and Zhang, H. (2021). Hfcontractfuzzer: Fuzzing hyperledger fabric smart contracts for vulnerability detection. In *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering*, pages 321–328.

[Entriken et al. 2018] Entriken, W., Shirley, D., Evans, J., and Sachs, N. (2018). Ethereum improvement proposal 721.

[Eskandari et al. 2020] Eskandari, S., Moosavi, S., and Clark, J. (2020). Sok: Transparent dishonesty: front-running attacks on blockchain. In *Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*, pages 170–189. Springer.

[Frost et al. 2019] Frost, J., Gambacorta, L., Huang, Y., Shin, H. S., and Zbinden, P. (2019). Bigtech and the changing structure of financial intermediation. *Economic policy*, 34(100):761–799.

[Ghaleb et al. 2023] Ghaleb, A., Rubin, J., and Pattabiraman, K. (2023). Achecker: Statically detecting smart contract access control vulnerabilities. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 945–956. IEEE.

[Gilbert and Lynch 2002] Gilbert, S. and Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News*, 33(2):51–59.

[Goes 2020] Goes, C. (2020). The interblockchain communication protocol: An overview. *arXiv preprint arXiv:2006.15918*.

[Goetze 2023] Goetze, C. (2023). Stablecoins: o que são e quais os tipos? `https://hubdoinvestidor.com.br/stablecoins-o-que-sao-e-quais-os-tipos/`. (Accessed on 05/23/2024).

[Greve et al. 2018] Greve, F., Sampaio, L., Abijaude, J., Coutinho, A. A., Brito, I., and Queiroz, S. (2018). Blockchain e a Revolução do Consenso sob Demanda. In *Proc. of SBRC Minicursos*.

[Harvey et al. 2021] Harvey, C. R., Ramachandran, A., and Santoro, J. (2021). *DeFi and the Future of Finance*. John Wiley & Sons.

[Harz and Knottenbelt 2018] Harz, D. and Knottenbelt, W. (2018). Towards safer smart contracts: A survey of languages and verification methods. *arXiv preprint arXiv:1809.09805*.

[Hassan and De Filippi 2021] Hassan, S. and De Filippi, P. (2021). Decentralized autonomous organization. *Internet Policy Review*, 10(2):1–10.

[Helliar et al. 2020] Helliar, C. V., Crawford, L., Rocca, L., Teodori, C., and Veneziani, M. (2020). Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, 54:102136.

[Ivanov et al. 2023] Ivanov, N., Li, C., Yan, Q., Sun, Z., Cao, Z., and Luo, X. (2023). Security threat mitigation for smart contracts: A comprehensive survey. *ACM Computing Surveys*, 55(14s):1–37.

[Jiang et al. 2018] Jiang, B., Liu, Y., and Chan, W. K. (2018). Contractfuzzer: Fuzzing smart contracts for vulnerability detection. In *Proceedings of the 33rd ACM/IEEE international conference on automated software engineering*, pages 259–269.

[Juliano 2018] Juliano, A. (2018). dydx: A standard for decentralized margin trading and derivatives. *URl: https://whitepaper. dydx. exchange*.

[Kalra et al. 2018] Kalra, S., Goel, S., Dhawan, M., and Sharma, S. (2018). Zeus: analyzing safety of smart contracts. In *Ndss*, pages 1–12.

[Kushwaha et al. 2022] Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., and Lee, H.-N. (2022). Ethereum smart contract analysis tools: A systematic review. *Ieee Access*, 10:57037–57062.

[Kwon and Buchman 2019] Kwon, J. and Buchman, E. (2019). Cosmos whitepaper. *A Netw. Distrib. Ledgers*, 27:1–32.

[Lamport et al. 2019] Lamport, L., Shostak, R., and Pease, M. (2019). *The Byzantine Generals Problem*, page 203–226. Association for Computing Machinery, New York, NY, USA.

[Lerner et al. 2022] Lerner, S. D., Cid-Fuentes, J. Á., Len, J., Fernàndez-València, R., Gallardo, P., Vescovo, N., Laprida, R., Mishra, S., Jinich, F., and Masini, D. (2022). Rsk: A bitcoin sidechain with stateful smart-contracts. *Cryptology ePrint Archive*.

[Li et al. 2021] Li, M., Weng, J., Li, Y., Wu, Y., Weng, J., Li, D., Xu, G., and Deng, R. (2021). Ivycross: A privacy-preserving and concurrency control framework for blockchain interoperability. *Cryptology ePrint Archive*.

[Li et al. 2022] Li, P., Li, S., Ding, M., Yu, J., Zhang, H., Zhou, X., and Li, J. (2022). A vulnerability detection framework for hyperledger fabric smart contracts based on dynamic and static analysis. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*, pages 366–374.

[Liao et al. 2022] Liao, Z., Zheng, Z., Chen, X., and Nan, Y. (2022). Smartdagger: a bytecode-based static analysis approach for detecting cross-contract vulnerability. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 752–764.

[Linoy et al. 2021] Linoy, S., Ray, S., and Stakhanova, N. (2021). Etherprov: Provenance-aware detection, analysis, and mitigation of ethereum smart contract security issues. In *2021 IEEE International Conference on Blockchain (Blockchain)*, pages 1–10. IEEE.

[Liu et al. 2023] Liu, Z., Qian, P., Yang, J., Liu, L., Xu, X., He, Q., and Zhang, X. (2023). Rethinking smart contract fuzzing: Fuzzing with invocation ordering and important branch revisiting. *IEEE Transactions on Information Forensics and Security*, 18:1237–1251.

[Luu et al. 2016] Luu, L., Chu, D.-H., Olickel, H., Saxena, P., and Hobor, A. (2016). Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 254–269.

[Ma et al. 2019] Ma, C., Kong, X., Lan, Q., and Zhou, Z. (2019). The privacy protection mechanism of hyperledger fabric and its application in supply chain finance. *Cybersecurity*, 2(1):1–9.

[Mendonça et al. 2024] Mendonça, R. D., Cardoso, I. W. F., Coelho, R., Campos, J. N., Gonçalves, G. D., Vieira, A. B., and Nacif, J. A. (2024). Mecanismos de interoperabilidade em blockchains: Um comparativo de custo de transações cross-chain para tokens

erc-20. In *Anais do VII Workshop em Blockchain: Teoria, Tecnologias e Aplicações*. SBC.

[Messias et al. 2023] Messias, J., Pahari, V., Chandrasekaran, B., Gummadi, K. P., and Loiseau, P. (2023). Understanding blockchain governance: Analyzing decentralized voting to amend defi smart contracts.

[Mossberg et al. 2019] Mossberg, M., Manzano, F., Hennenfent, E., Groce, A., Grieco, G., Feist, J., Brunson, T., and Dinaburg, A. (2019). Manticore: A user-friendly symbolic execution framework for binaries and smart contracts. In *2019 34th IEEE/ACM (ASE)*, pages 1186–1189. IEEE.

[Murray et al. 2023] Murray, A., Kim, D., and Combs, J. (2023). The promise of a decentralized internet: What is web3 and how can firms prepare? *Business Horizons*, 66(2):191–202.

[Nadini et al. 2021] Nadini, M., Alessandretti, L., Di Giacinto, F., Martino, M., Aiello, L. M., and Baronchelli, A. (2021). Mapping the nft revolution: market trends, trade networks, and visual features. *Scientific reports*, 11(1):1–11.

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

[Nakamoto and Bitcoin 2008] Nakamoto, S. and Bitcoin, A. (2008). A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4.

[Okonkwo 2021] Okonkwo, I. E. (2021). Nft, copyright; and intellectual property commercialisation. *SSRN*. https://ssrn.com/abstract=3856154.

[Ongaro and Ousterhout 2014] Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference (USENIX ATC 14)*, pages 305–319.

[Ou et al. 2022] Ou, W., Huang, S., Zheng, J., Zhang, Q., Zeng, G., and Han, W. (2022). An overview on cross-chain: Mechanism, platforms, challenges and advances. *Computer Networks*.

[Palma et al. 2022] Palma, L., Martina, J., and Vigil, M. (2022). On and off: Extracting the transaction history of permissioned blockchain networks. Universidade Federal de Santa Catarina. Computing Science PhD Candidate Dissertation.

[Product 2024] Product (2024). dYdX v4 - full decentralization. `https://dydx.exchange/blog/v4-full-decentralization`. (Accessed on 05/20/2024).

[Qin et al. 2021] Qin, K., Zhou, L., and Gervais, A. (2021). Quantifying blockchain extractable value: How dark is the forest? *CoRR*, abs/2101.05511.

[Ren et al. 2023] Ren, K., Ho, N.-M., Loghin, D., Nguyen, T.-T., Ooi, B. C., Ta, Q.-T., and Zhu, F. (2023). Interoperability in blockchain: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

[Rodler et al. 2023] Rodler, M., Paaßen, D., Li, W., Bernhard, L., Holz, T., Karame, G., and Davi, L. (2023). Ef cf: High performance smart contract fuzzing for exploit generation. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 449–471. IEEE.

[Rouhani and Deters 2019] Rouhani, S. and Deters, R. (2019). Security, performance, and applications of smart contracts: A systematic survey. *IEEE Access*, 7:50759–50779.

[Sabt et al. 2015] Sabt, M., Achemlal, M., and Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/Ispa*, volume 1, pages 57–64. IEEE.

[Sanches and Diniz 2024] Sanches, B. H. and Diniz, E. (2024). From crypto-libertarian utopia to central bank digital currencies: The transfigurative convergence of bitcoin's prefiguration. *Available at SSRN 4818428*.

[Sayeed et al. 2020] Sayeed, S., Marco-Gisbert, H., and Caira, T. (2020). Smart contract: Attacks and protections. *IEEE Access*, 8:24416–24427.

[Schär 2021] Schär, F. (2021). Decentralized finance: On blockchain-and smart contract-based financial markets. *FRB of St. Louis Review*.

[Singh et al. 2020] Singh, A., Click, K., Parizi, R. M., Zhang, Q., Dehghantanha, A., and Choo, K.-K. R. (2020). Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications*, 149:102471.

[Siris et al. 2019] Siris, V. A., Nikander, P., Voulgaris, S., Fotiou, N., Lagutin, D., and Polyzos, G. C. (2019). Interledger approaches. *Ieee Access*, 7:89948–89966.

[Song et al. 2022] Song, K., Matulevicius, N., de Lima Filho, E. B., and Cordeiro, L. C. (2022). Esbmc-solidity: An smt-based model checker for solidity smart contracts. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*, pages 65–69.

[Sun et al. 2021] Sun, X., Yu, F. R., Zhang, P., Sun, Z., Xie, W., and Peng, X. (2021). A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205.

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

[Teixeira 2023] Teixeira, D. (2023). O caminho para o real digital. *Revista LIFT papers*, 5(5).

[Thomas and Schwartz 2015] Thomas, S. and Schwartz, E. (2015). A protocol for interledger payments. *URL https://interledger. org/interledger. pdf*.

[Tikhomirov 2018] Tikhomirov, S. (2018). Ethereum: state of knowledge and research perspectives. In *Foundations and Practice of Security: 10th International Symposium, FPS 2017, Nancy, France, October 23-25, 2017, Revised Selected Papers 10*, pages 206–221. Springer.

[Tikhomirov et al. 2018] Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E., and Alexandrov, Y. (2018). Smartcheck: Static analysis of ethereum smart contracts. In *Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain*, pages 9–16.

[Torres et al. 2021] Torres, C. F., Camino, R., and State, R. (2021). Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1343–1359. USENIX Association.

[Varun et al. 2022] Varun, M., Palanisamy, B., and Sural, S. (2022). Mitigating frontrunning attacks in ethereum. In *Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure*, BSCI '22, page 115–124. Association for Computing Machinery.

[Wang et al. 2021] Wang, D., Wu, S., Lin, Z., Wu, L., Yuan, X., Zhou, Y., Wang, H., and Ren, K. (2021). Towards a first step to understand flash loan and its applications in defi ecosystem. In *Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing*, pages 23–28.

[Wang et al. 2020] Wang, W., Song, J., Xu, G., Li, Y., Wang, H., and Su, C. (2020). Contractward: Automated vulnerability detection models for ethereum smart contracts. *IEEE Transactions on Network Science and Engineering*.

[Wang et al. 2022] Wang, Y., Zuest, P., Yao, Y., Lu, Z., and Wattenhofer, R. (2022). Impact and user perception of sandwich attacks in the defi ecosystem. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. Association for Computing Machinery.

[Wegner 1996] Wegner, P. (1996). Interoperability. *ACM Computing Surveys (CSUR)*, 28(1):285–287.

[Weintraub et al. 2022] Weintraub, B., Torres, C. F., Nita-Rotaru, C., and State, R. (2022). A flash(bot) in the pan: measuring maximal extractable value in private pools. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 458–471. Association for Computing Machinery.

[Werner et al. 2022] Werner, S., Perez, D., Gudgeon, L., Klages-Mundt, A., Harz, D., and Knottenbelt, W. (2022). Sok: Decentralized finance (defi). In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, pages 30–46.

[White et al. 2022] White, B., Mahanti, A., and Passi, K. (2022). Characterizing the opensea nft marketplace. In *Companion Proceedings of the Web Conference 2022*, pages 488–496.

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

[Wu et al. 2021] Wu, S., Wang, D., He, J., Zhou, Y., Wu, L., Yuan, X., He, Q., and Ren, K. (2021). Defiranger: Detecting price manipulation attacks on defi applications. *arXiv preprint arXiv:2104.15068*.

[Wüstholz and Christakis 2020] Wüstholz, V. and Christakis, M. (2020). Harvey: A greybox fuzzer for smart contracts. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1398–1409.

[Xu et al. 2023] Xu, X., Hu, T., Li, B., and Liao, L. (2023). Ccdetector: Detect chaincode vulnerabilities based on knowledge graph. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 699–704. IEEE.

[Xu et al. 2019] Xu, X., Weber, I., and Staples, M. (2019). *Architecture for blockchain applications*. Springer.

[Xu et al. 2021] Xu, Y., Hu, G., You, L., and Cao, C. (2021). A novel machine learning-based analysis model for smart contract vulnerability. *Security and Communication Networks*, 2021:1–12.

[Yadav and Naval 2023] Yadav, K. and Naval, S. (2023). Cfg analysis for detecting vulnerabilities in smart contracts. In *International Conference on Smart Trends for Information Technology and Computer Communications*, pages 753–763. Springer.

[Yamashita et al. 2019] Yamashita, K., Nomura, Y., Zhou, E., Pi, B., and Jun, S. (2019). Potential risks of hyperledger fabric smart contracts. In *2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 1–10. IEEE.

[Yan et al. 2022] Yan, X., Wang, S., and Gai, K. (2022). A semantic analysis-based method for smart contract vulnerability. In *2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity)*, pages 23–28. IEEE.

[Ye et al. 2020] Ye, J., Ma, M., Lin, Y., Sui, Y., and Xue, Y. (2020). Clairvoyance: Cross-contract static analysis for detecting practical reentrancy vulnerabilities in smart contracts. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, pages 274–275.

[Zeller et al. 2019] Zeller, A., Gopinath, R., Böhme, M., Fraser, G., and Holler, C. (2019). The fuzzing book.

[Zhang et al. 2023a] Zhang, P., Yu, Q., Xiao, Y., Dong, H., Luo, X., Wang, X., and Zhang, M. (2023a). Bian: Smart contract source code obfuscation. *IEEE Transactions on Software Engineering*.

[Zhang et al. 2023b] Zhang, W., Wei, L., Cheung, S.-C., Liu, Y., Li, S., Liu, L., and Lyu, M. R. (2023b). Combatting front-running in smart contracts: Attack mining, benchmark construction and vulnerability detector evaluation. *IEEE Transactions on Software Engineering*, 49(6):3630–3646.

[Zhang et al. 2023c] Zhang, Y., Liu, P., Wang, G., Li, P., Gu, W., Chen, H., Liu, X., and Zhu, J. (2023c). Frad: Front-running attacks detection on ethereum using ternary classification model. *arXiv preprint arXiv:2311.14514*.

[Zhou et al. 2021] Zhou, L., Qin, K., Torres, C. F., Le, D. V., and Gervais, A. (2021). High-frequency trading on decentralized on-chain exchanges. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 428–445. IEEE.

[Zou et al. 2019] Zou, W., Lo, D., Kochhar, P. S., Le, X.-B. D., Xia, X., Feng, Y., Chen, Z., and Xu, B. (2019). Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering*.

[Zwitter and Hazenberg 2020] Zwitter, A. and Hazenberg, J. (2020). Decentralized network governance: blockchain technology and the future of regulation. *Frontiers in Blockchain*, 3:12.